API Яндекс.Директа

Руководство разработчика

5.05.2011

Яндекс

API Яндекс.Директа. Руководство разработчика. Версия 3.0
Дата сборки документа: 5.05.2011.
Этот документ является составной частью технической документации Яндекса.
Сайт справки к сервисам Яндекса: http://help.yandex.ru
© 2008—2011 ООО «ЯНДЕКС». Все права защищены.

Предупреждение об исключительных правах

Яндексу (а также указанному им правообладателю) принадлежат исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации, используемые при разработке, поддержке и эксплуатации сервиса API Яндекс.Директа. К таким результатам могут относиться, но не ограничиваясь указанными, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования. Эти права охраняются в соответствии с Гражданским кодексом РФ и международным правом.

Вы можете использовать сервис API Яндекс.Директа или его составные части только в рамках полномочий, предоставленных вам Пользовательским соглашением сервиса API Яндекс.Директа или специального соглашения.

Нарушение требований по защите исключительных прав правообладателя влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с российским законодательством.

Контактная информация

ООО «ЯНДЕКС»
http://www.yandex.ru
Тел.: +7 495 739 7000
Email: pr@yandex-team.ru
Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

Содержание

Введение ... 5

Первоначальные сведения ... 6
- Версии API ... 6
- Формат JSON ... 8
- Протокол SOAP ... 10
- Авторизационные токены ... 14
- Цифровые сертификаты SSL ... 14

Практика использования ... 16
- Ограничения на использование API ... 16
- Работа с рекламной кампанией ... 17
- Получение отчета по статистике ... 18
- Прогноз бюджета ... 18
- Отладка приложения ... 19
- Получение цифровых сертификатов ... 19

Методы API ... 23

Статистика кампании ... 23
- CreateNewReport ... 23
- GetReportList ... 26
- DeleteReport ... 27
- GetStatGoals ... 27

Подбор ключевых слов ... 29
- CreateNewWordstatReport ... 29
- GetWordstatReportList ... 30
- GetWordstatReport ... 31
- GetKeywordsSuggestion ... 32

Прогноз бюджета ... 34
- CreateNewForecast ... 34
- GetForecastList ... 35
- GetForecast ... 36
- DeleteForecastReport ... 39

Информация о кампаниях ... 40
- GetCampaignParams ... 40
- GetCampaignsList ... 43
- GetCampaignsListFilter ... 45
- GetBalance ... 47
- GetBanners ... 49
- GetBannerPhrases ... 55
- GetBannerPhrasesFilter ... 58

Редактирование кампаний и объявлений ... 63
- CreateOrUpdateCampaign ... 63
- CreateOrUpdateBanners ... 67
- SetAutoPrice ... 71
- UpdatePrices ... 73

Состояние кампаний и объявлений ... 76
- StopCampaign ... 76
- ResumeCampaign ... 76
- ArchiveCampaign ... 77
- UnArchiveCampaign ... 78
- DeleteCampaign ... 78
- ModerateBanners ... 79
- StopBanners ... 80
- ResumeBanners ... 81
- ArchiveBanners ... 82
- UnArchiveBanners ... 82

DeleteBanners ... 83

Работа с клиентами ... 85
GetClientInfo ... 85
GetClientsList ... 86
GetClientsUnits ... 87
GetSubClients ... 88
CreateNewSubclient ... 89

Яндекс.Каталог и регионы ... 92
GetRegions ... 92
GetRubrics ... 92

Прочие методы ... 94
GetVersion ... 94
GetAvailableVersions ... 94
GetTimeZones ... 95
PingAPI ... 97

Коды ошибок ... 98

Примеры ... 101

Предметный указатель ... 169

Введение

Документ содержит сведения о программном интерфейсе сервиса Яндекс.Директ (далее по тексту — «API Яндекс.Директа» или «API»).

API обеспечивает автоматизированный обмен данными с сервисом контекстной рекламы Яндекс.Директ. Через API внешние приложения могут получать сведения о рекламных кампаниях, результаты торгов и статистику кампаний, редактировать и добавлять кампании, объявления, ключевые слова.

Основные разделы документа:

- Первоначальные сведения — взаимодействие с API Яндекс.Директа по протоколу SOAP и в формате JSON, авторизация пользователей;
- Практика использования — выполнение различных задач с помощью API Яндекс.Директа и ограничения на использование API.
- Методы API — описание методов, входных и результирующих данных;
- Примеры — примеры вызова методов на Perl, PHP, Python с применением различных библиотек. Здесь же содержатся примеры приложений и примеры работы с отчетами.

Документ адресован разработчикам приложений по управлению контекстной рекламой Яндекса. Предполагается, что читатели знакомы с возможностями и терминологией сервиса Яндекс.Директ.

Первоначальные сведения

API Яндекс.Директа поддерживает два способа взаимодействия:

- запросы в формате JSON;
- протокол SOAP;

Запросы JSON меньше нагружают сервер и быстрее обрабатываются.

Примечание:
Существует большое количество библиотек JSON и SOAP для разных языков программирования. Использование некоторых библиотек демонстрируется в разделе Примеры.

Доступ к API предоставляется через криптографический протокол SSL, который защищает данные от несанкционированного доступа при передаче по открытым каналам связи. Установка SSL-соединения обязательна для вызова методов API.

API доступен только авторизованным пользователям Яндекс.Директа. Пользователи идентифицируют себя одним из двух способов:

- с помощью авторизационных токенов;
- с помощью цифровых сертификатов SSL.

Оба способа доступны для SOAP и JSON. Авторизация по токенам — более прогрессивный способ.

Версии API

Текущая версия API имеет номер 3.

Новые версии анонсируются в рассылке API Яндекс.Директа. Предыдущие версии поддерживаются в течение одного-двух месяцев после выпуска новой версии. При попытке использовать неподдерживаемую версию выдается сообщение об ошибке 508.

Примечание:
Настоятельно рекомендуется переходить на новые версии API при их появлении. Полная совместимость между версиями не гарантируется.

При обращении к API номер версии указывается в адресной строке. Следующий адрес используется для запросов к API в формате JSON:

```
https://soap.direct.yandex.ru/json-api/v3/
```

Ниже показан адрес WSDL-описания текущей версии API:

```
http://soap.direct.yandex.ru/wsdl/v3/
```

SOAP-запросы отправляют на следующий адрес:

```
https://soap.direct.yandex.ru/api/v3/
```

Сведения о доступных версиях API можно получить с помощью метода GetAvalibleVersions. Среди прочего, он возвращает даты планируемого отключения предыдущих версий.

Версии API:
Версия 3

Версия 2
Версия 1

Версия 3

Текущая версия API Яндекс.Директа.

Третья версия API выпущена 1 декабря 2010 г.

Добавлено в версии

1. Методы CreateNewWordstatReport, GetWordstatReportList, GetWordstatReport для подбора ключевых слов.
2. Метод GetKeywordsSuggestion для получения подсказок к ключевым словам.
3. Метод CreateNewSubclient, создающий клиента для рекламного агентства.
4. Метод GetStatGoals, возвращающий информацию о целях счетчиков Яндекс.Метрики, которые установлены на сайтах рекламодателя.
5. Поддержка авторизации по токенам, которые выдает OAuth-сервер Яндекса.
6. Взаимодействие с API в формате JSON.
7. Параметр `locale` позволяет выбирать язык ответных сообщений при вызове методов (см. locale в JSON и locale в SOAP).

Изменено в версии

1. Метод CreateOrUpdateCampaign — добавлены входные параметры `MinusKeywords`, `AddRelevantPhrases` и `RelevantPhrasesBudgetLimit`. Получить значения настроек кампании можно с помощью метода GetCampaignParams.
2. Метод CreateOrUpdateBanners — добавлен входной параметр `Sitelinks`.
3. Структура ContactInfo — добавлены параметры `OGRN` и `PointOnMap` (возможные значения можно получить с помощью метода GetBanners).
4. Метод GetBannerPhrasesFilter — добавлен входной параметр `RequestPrices`.
5. Метод CreateNewReport — добавлена возможность группировать данные отчета по целям (параметр `clStatGoals`) и по позициям в показах (параметр `clPositionType`). Эти же параметры добавлены в структуру NewReportFilterInfo.
6. Метод GetBanners — добавлены входные параметры `Sitelinks`, `StatusSitelinksModerate`, `AdWarnings`, `ModerateRejectionReasons`, `FixedOnModeration`.
7. Метод GetClientInfo — добавлен входной параметр `Discount`.

Версия 2

Поддержка версии будет прекращена в апреле 2011 г.

Вторая версия API выпущена 23 августа 2010 г. Использовать эту версию не рекомендуется, поскольку ее поддержка скоро будет прекращена.

Документация: Руководство разработчика в.2 (PDF, 1.14 Мб).

Добавлено в версии

1. Метод GetSubClients. Возвращает список подчиненных пользователей первого уровня для указанного логина
2. Метод GetCampaignParams. Возвращает список параметров рекламной кампании и их значения.

Изменено в версии

1. Метод CreateOrUpdateCampaign:
   - Добавлены поля `ContextLimit`, `DisabledDomains`, `DisabledIps`, `ConsiderTimeTarget`, `Status`, `StatusContextStop`, `StatusMetricaControl`, `StatusOpenStat`.
   - Добавлена структура EmailNotificationInfo для управления уведомлениями по электронной почте. Структура включает поля `Email`, `SendWarn`, `MoneyWarningValue`, `WarnPlaceInterval`, `SendAccNews`.
   - Добавлена структура SmsNotificationInfo для управления SMS-уведомлениями. Структура включает поля `SmsPhone`, `SmsTimeFrom`, `SmsTimeTo`, `ModerateResultSms`, `MetricaSms`, `MoneyInSms`, `MoneyOutSms`.
   - Удалены поля `Email`, `MoneyWarningValue`, `SendWarn`. Параметры для управления уведомлениями по электронной почте объединены в структуру EmailNotificationInfo.
   - Удалены поля `SmsSendWarnMoneyOut`, `SmsSendWarnMoneyIn`, `SmsSendModerate`, `SmsTimeHourFrom`, `SmsTimeMinFrom`, `SmsTimeHourTo`, `SmsTimeMinTo`. Параметры для управления SMS-уведомлениями объединены в структуру SmsNotificationInfo.
2. Методы UpdatePrices и CreateOrUpdateBanners — добавлены входные параметры `AutoBudgetPriority` и `ContextPrice`.
3. Метод GetBannerPhrasesFilter — добавлен входной параметр `ConsiderTimeTarget`.
4. Методы GetBannerPhrases и GetBannerPhrasesFilter — к результирующей структуре BannerPhraseInfo добавлены параметры `ContextPrice`, `ContextLowCTR`, `ContextCoverage`.
5. Методы GetCampaignsList и GetCampaignsListFilter — вместо массива `CampaignInfo` возвращается массив ShortCampaignInfo, который:
   - содержит параметры `CampaignID`, `Login`, `Name`, `StartDate`, `Sum`, `Rest`, `Shows`, `Clicks`, `Status`, `StatusShow`, `StatusModerate`, `StatusActivating`, `StatusArchive`, `IsActive`;
   - не содержит параметры `TimeTarget`, `Autobudget`, `AutobudgetSum`, `AutobudgetMaxPrice` (их значения можно получить с помощью метода GetCampaignParams).
6. Метод GetClientInfo позволяет получать информацию о нескольких клиентах в виде структур ClientInfo. В структуру `ClientInfo` добавлены поля `Role`, `SendWarn`, `SendNews`, `SendAccNews`, `NonResident`.
7. Методы ArchiveCampaign, UnArchiveCampaign, ResumeCampaign, StopCampaign, DeleteCampaign принимают на вход структуру CampaignIDInfo с новым параметром `CampaignID`. Совместимость с предыдущей версией API сохранена.

Версия 1

Поддержка версии прекращена.

Первая версия API выпущена 18 июня 2010 г. Поддержка прекращена 24 февраля 2011 г.

Документация: Руководство разработчика в.1 (PDF, 1.14 Мб).

Формат JSON

Взаимодействие с API Яндекс.Директа в формате JSON.

API Яндекс.Директа поддерживает универсальный формат обмена данными JSON. Преимущество JSON в большей компактности по сравнению с SOAP/XML, а также в скорости анализа запросов на стороне Яндекс.Директа.

Во многих языках программирования существуют встроенные модули и подключаемые библиотеки для работы с JSON. Они формируют массивы JSON из структур, свойственных языку, например из словарей и ассоциативных массивов. Ниже перечислены рекомендуемые модули и библиотеки.

- **Perl**: модуль JSON::Syck.
- **PHP**: встроенная поддержка JSON началась в версии 5.2.0. В предыдущих версиях можно использовать библиотеку Services_JSON.
- **Python**: модуль json.

Примеры работы с JSON содержатся в разделах Пример работы с JSON на Perl и Пример работы с JSON на PHP.

Запросы в формате JSON

Данные в формате JSON передаются методом HTTP POST. Запросы отправляют на адрес API Яндекс.Директа для JSON-запросов:

```
https://soap.direct.yandex.ru/json-api/v3/
```

Передаваемый массив JSON состоит по меньшей мере из двух ключей — `method` и `param` (см. пример ниже). В первом ключе указывают имя метода, во втором — входные параметры. Формат ключа `param` зависит от вызываемого метода.

Пример запроса JSON

Пример демонстрирует вызов метода `GetClientInfo`. В ключе `param` передается массив из одной строки, которая является логином пользователя. Ключ `locale` устанавливает русский язык для ответных сообщений.

```
POST /json-api/v3/ HTTP/1.1
Host: soap.direct.yandex.ru
Content-Length: 76
Content-Type: application/json; charset=utf-8

{
    "method": "GetClientInfo",
    "param": ["teux"],
    "locale": "ru"
}
```

Выбор языка ответных сообщений

Массив JSON может содержать ключ `locale`, указывающий язык ответных сообщений. На выбранном языке выводятся статусы кампаний, статусы баннеров и сообщения об ошибках. Возможны следующие значения `locale`:

- ru — русский;
- ua — украинский;
- en — английский.

Если ключ `locale` отсутствует, подразумевается английский язык. Исключением являются методы `GetRegions`, `ReguRubrics`, `GetTimeZones`. Для них в текущей версии API (версия 3) предустановленным является русский язык. В следующей версии API он будет заменен на английский.

Авторизация по цифровому сертификату

Авторизация по цифровому сертификату выполняется при установке SSL-соединения и не влияет на формирование запросов JSON. Подробнее см. раздел Цифровые сертификаты SSL.

Авторизация по токену

Авторизация по токену позволяет обращаться к API Яндекс.Директа от имени любого пользователя, который разрешил приложению доступ к данным о своих кампаниях (см. раздел Авторизационные токены). В запросе указывают следующие идентификаторы:

- `login` — логин пользователя, к данным которого приложение собирается получить доступ;
- `application_id` — идентификатор приложения, присвоенный OAuth-сервером Яндекса;
- `token` — авторизационный токен, выданный OAuth-сервером Яндекса с согласия пользователя.

Пример запроса JSON с авторизацией по токену

Запрос от имени пользователя «teux» с авторизацией по токену, выданному OAuth-сервером Яндекса.

```
POST /json-api/v3/ HTTP/1.1
Host: soap.direct.yandex.ru
Content-Length: 204
Content-Type: application/json; charset=utf-8

{
    "method": "GetClientInfo",
    "param": ["teux"],
    "locale": "ru",
    "login": "teux",
    "application_id": "eef1c9fce4874819b8399bb0bac9b9d9",
    "token": "e4d3b4d2a7444fb387a18dda54d1c8d9"
}
```

Сообщения об ошибках

При возникновении ошибок обработка запроса прекращается и возвращается сообщение об ошибке в формате JSON. Пример сообщения показан ниже.

```
{
    "error_detail":"Метод HTTP запроса должен быть POST",
    "error_str":"Неверный метод запроса",
    "error_code":512
}
```

Параметр `error_code` содержит код ошибки, `error_str` — краткое описание ошибки, `error_detail` — дополнительные пояснения, если имеются. Краткое описание и дополнительное пояснение могут выводиться на разных языках, что определяется параметром locale, указанным при вызове метода.

См. также
Описание формата JSON
Пример работы с JSON на Perl

Протокол SOAP

Взаимодействие с API Яндекс.Директа по протоколу SOAP.

API Яндекс.Директа поддерживает протокол SOAP (Simple Object Access Protocol) версии 1.1. Обмен данными по этому протоколу ведется посредством XML-сообщений.

Существует множество библиотек, реализующих протокол SOAP. Некоторые из них перечислены ниже:

- **Perl:** SOAP::Lite и Crypt::SSLeay;
- **PHP:** nuSOAP и SOAPClient;
- **Python:** SOAPpy и suds.

WSDL

WSDL (Web Services Description Language) — это язык описания веб-сервисов, основанный на XML. С помощью WSDL создается модель сервиса, в которой перечислены его методы, указаны входные и результирующие структуры данных, типы данных, сетевые адреса для обращения к сервису и другое.

Описание API Яндекс.Директа на языке WSDL доступно по адресу:

```
http://soap.direct.yandex.ru/wsdl/v3/
```

Первоначально WSDL был разработан компаниями IBM, Microsoft и Ariba для использования вместе с протоколом SOAP. Наиболее функциональные библиотеки SOAP используют WSDL-описания для формирования запросов к сервисам и проверки передаваемых данных. Библиотеки скачивают WSDL-описание, анализируют его и формируют структуры данных, необходимые для вызова методов. Приложению остается заполнить эти структуры данными и выполнить запрос. Информация из ответного XML-сообщения так же выдается в виде структуры, свойственной языку программирования. Все это избавляет приложение от необходимости обрабатывать XML.

Примечание:
Приложение может самостоятельно формировать SOAP-запросы в формате XML, однако,
это трудоемкая задача и при наличии полнофункциональной SOAP-библиотеки выполнять ее вручную нецелесообразно.

Запросы в формате SOAP

Данные в формате SOAP передаются методом HTTP POST. Запросы отправляют на адрес API Яндекс.Директа:

```
https://soap.direct.yandex.ru/api/v3/
```

Способ отправки запросов зависит от SOAP-библиотеки. Если библиотека поддерживает WSDL, достаточно указать адрес WSDL-файла. Из него библиотека получает адрес API Яндекс.Директа и выполняет необходимые действия для отправки запроса. Если библиотека не поддерживает WSDL, необходимо явно указывать адрес API.

Сообщение SOAP состоит из заголовка — элемент `SOAP-ENV:Header` (см. пример ниже) и основной части — элемент `SOAP-ENV:Body`. Заголовок может содержать метаданные, относящиеся к сообщению в целом. В теле сообщения передается элемент `params` с параметрами вызываемого метода. Формат элемента `params` отличается для разных методов.

Пример запроса в формате SOAP

Вызов метода `GetClientInfo` для получения данных о пользователе «teux». Заголовок содержит элемент `locale`, устанавливающий русский язык для ответных сообщений.

```
POST /api/v3/ HTTP/1.1
Content-Length: 524
Content-Type: text/xml; charset=utf-8
SOAPAction: "API#GetClientInfo"
Host: soap.direct.yandex.ru

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:ns0="API"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <locale>ru</locale>
  </SOAP-ENV:Header>
```

```
  <SOAP-ENV:Body>
    <ns0:GetClientInfo SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
      <params soapenc:arrayType="xsd:string[]">
        <xsd:string>teux</xsd:string>
        <Login>teux</Login>
      </params>
    </ns0:GetClientInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Пример ответа в формате SOAP

Ответ имеет формат XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:namesp2="http://namespaces.soaplite.com/perl"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <namesp3:GetClientInfoResponse xmlns:namesp3="API">
      <SOAP-ENC:Array xsi:type="namesp2:ArrayOfClientInfo" SOAP-
ENC:arrayType="namesp2:ClientInfo[1]">
        <item xsi:type="namesp2:ClientInfo">
          <SendAccNews xsi:type="xsd:string">Yes</SendAccNews>
          <Discount xsi:nil="true" xsi:type="xsd:float"/>
          <Login xsi:type="xsd:string">teux</Login>
          <Email xsi:type="xsd:string">teux@yandex.ru</Email>
          <Role xsi:type="xsd:string">Client</Role>
          <FIO xsi:type="xsd:string">Andrew Smith</FIO>
          <DateCreate xsi:type="xsd:date">2011-01-06</DateCreate>
          <StatusArch xsi:type="xsd:string">No</StatusArch>
          <SendNews xsi:type="xsd:string">Yes</SendNews>
          <Phone xsi:nil="true" xsi:type="xsd:string"/>
          <NonResident xsi:type="xsd:string">No</NonResident>
          <SendWarn xsi:type="xsd:string">Yes</SendWarn>
        </item>
      </SOAP-ENC:Array>
    </namesp3:GetClientInfoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Выбор языка ответных сообщений

Заголовок сообщения SOAP может содержать элемент `locale`, указывающий язык ответных сообщений. На выбранном языке выводятся статусы кампаний, статусы баннеров и сообщения об ошибках. Возможны следующие значения `locale`:

- ru — русский;
- ua — украинский;
- en — английский.

Если элемент `locale` отсутствует, подразумевается английский язык. Исключением являются методы `GetRegions`, `ReguRubrics`, `GetTimeZones`. Для них в текущей версии API (версия 3) предустановленным является русский язык. В следующей версии API он будет заменен на английский.

Авторизация по цифровому сертификату

Авторизация по цифровому сертификату выполняется при установке SSL-соединения и не влияет на формирование запросов SOAP. Подробнее см. раздел Цифровые сертификаты SSL.

Авторизация по токену

Авторизация по токену позволяет обращаться к API Яндекс.Директа от имени любого пользователя, который разрешил приложению доступ к данным о своих кампаниях (см. раздел Авторизационные токены). В заголовке SOAP-запроса указывают следующие идентификаторы:

- `login` — логин пользователя, к данным которого приложение собирается получить доступ;
- `application_id` — идентификатор приложения, присвоенный OAuth-сервером Яндекса;
- `token` — авторизационный токен, выданный OAuth-сервером Яндекса с согласия пользователя.

Пример запроса SOAP с авторизацией по токену

Вызов метода `GetClientInfo` для получения данных о пользователе «teux».

```
POST /api/v3/ HTTP/1.1
Content-Length: 686
Content-Type: text/xml; charset=utf-8
SOAPAction: "API#GetClientInfo"
Host: soap.direct.yandex.ru

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:ns0="API"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <locale>ru</locale>
    <login>teux</login>
    <token>e4d3b4d2a7444fb387a18dda54d1c8d9</token>
    <application_id>eef1c9fce4874819b8399bb0bac9b9d9</application_id>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns0:GetClientInfo SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
      <params soapenc:arrayType="xsd:string[]">
        <xsd:string>teux</xsd:string>
        <Login>teux</Login>
      </params>
    </ns0:GetClientInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Сообщения об ошибках

При возникновении ошибки обработка запроса прекращается и возвращается сообщение об ошибке в формате SOAP. Пример сообщения показан ниже.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:namesp8="http://xml.apache.org/xml-soap"
                   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:namesp4="http://namespaces.soaplite.com/perl"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode xsi:type="xsd:string">SOAP-ENV:53</faultcode>
      <faultstring xsi:type="xsd:string">Authorization error</faultstring>
      <detail xsi:type="xsd:string"/>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Элемент `faultcode` содержит код ошибки, `faultstring` — краткое описание ошибки, `detail` — дополнительные пояснения, если имеются. Краткое описание и дополнительное пояснение могут выводиться на разных языках, что определяется элементом locale.

См. также
Simple Object Access Protocol (SOAP) 1.1
Web Services Description Language (WSDL) 1.1

Авторизационные токены

Идентификация пользователей с помощью токенов, выдаваемых OAuth-сервером Яндекса.

Использование авторизационных токенов удобно в приложениях с динамично меняющимся кругом пользователей. Приложение получает доступ к данным пользователя по специальному коду — токену. Он выдается OAuth-сервером Яндекса и выражает согласие пользователя на доступ приложения к его данным в Яндекс.Директе.

Инициатором получения токена является приложение. Оно перенаправляет пользователя на OAuth-сервер Яндекса, где пользователь проходит авторизацию и подтверждает согласие на доступ приложения к данным в Яндекс.Директе. OAuth-сервер выдает приложению токен, который приложение указывает во всех запросах к API от имени пользователя.

Примечание:
OAuth-сервер находится внутри защищенной сети Яндекса, поэтому пользователю психологически комфортно доверить ему пароль. В любой момент пользователь может прекратить действие токена через интерфейс OAuth-сервера на странице Доступ внешних приложений.

Инструкции по получению токенов содержатся в документе OAuth-авторизация. Руководство разработчика. Для тестирования приложения можно получить отладочный токен.Предварительно необходимо зарегистрировать приложение на OAuth-сервере, в результате чего приложению присваивается уникальный идентификатор. На OAuth-сервере он именуется `client_id`, в API Яндекс.Директа известен как `application_id`.

В запросах к API Яндекс.Директа необходимо указывать логин пользователя, авторизационный токен и идентификатор приложения (см. Пример запроса JSON с авторизацией по токену и Пример запроса SOAP с авторизацией по токену). Если указанные данные не верны или срок действия токена истек, возвращается сообщение об ошибке с кодом 54.

Цифровые сертификаты SSL

Идентификация пользователей с помощью цифровых сертификатов, выдаваемых Яндекс.Директом.

Цифровые сертификаты являются частью протокола SSL, поэтому их применение для идентификации пользователей органично сочетается с установкой SSL-соединения. Если при установке соединения на сервер отправлен действующий сертификат пользователя, последующие запросы через данное соединение считаются принадлежащими этому пользователю.

Примечание:
Если в запросе указан авторизационный токен, пользователь идентифицируется по токену, и только в случае неудачи — по цифровому сертификату.

Сертификаты получают через веб-интерфейс Яндекс.Директа. Это ограничивает применение сертификатов, поскольку вынуждает пользователей предпринимать дополнительные действия по получению сертификатов и передавать их приложению. В связи с этим авторизация по сертификатам больше подходит приложениям с небольшим и редко меняющимся кругом пользователей.

Простейший способ получить персональный сертификат — создать его в интерфейсе Яндекс.Директа. При этом Яндекс.Директ возвращает следующие файлы:

- `cert.crt` — персональный сертификат пользователя;
- `private.key` — секретный ключ шифрования (используется для декодирования данных, получаемых от API Яндекс.Директа);
- `cacert.pem` — собственный сертификат API Яндекс.Директа (так называемый CA-сертификат);
- `req.csr` — запрос на создание персонального сертификата.

Для установки SSL-соединения необходимы файлы `cert.crt` и `private.key`. Некоторые SSL-библиотеки дополнительно требуют CA-сертификат (файл `cacert.pem`). Он служит для проверки того, то SSL-соединение установлено именно с сервером Яндекс.Директа, а не с каким-либо другим сервером. Часть SSL-библиотек не выполняет такую проверку и не требует CA-сертификат.

Внимание!
Файл `private.key` следует защищать от несанкционированного доступа. Он должен быть доступен приложению, но не доступен внешним пользователям.

Файл `req.csr` не нужен для установки SSL-соединения. Он создается во время генерации персонального сертификата и передается пользователю для полноты состава возвращаемых данных

Некоторые SSL-библиотеки ожидают получать персональный сертификат и секретный ключ в одном файле. В Unix-подобных системах можно объединить сертификат с ключом следующей командой:

```
cat cert.crt private.key > solid-cert.crt
```

Секретный ключ шифрования можно сгенерировать локально с помощью библиотеки OpenSSL. Основная причина для этого — ключ не передается по открытым каналам связи. В этом случае процедура получения сертификата изменяется, как описано в разделе Получение сертификата с помощью csr-запроса.

Примеры приложений в настоящем документе демонстрируют установку SSL-соединений на разных языках программирования с использованием различных библиотек:

Perl: раздел Создание рекламной кампании на Perl (SOAP::Lite);

PHP: разделы Создание рекламной кампании на PHP (NuSOAP) и Создание рекламной кампании на PHP (SOAPClient) ;

Python: Создание рекламной кампании на Python (SOAPpy).

Примечание:
Для установки SSL-соединения не обязательно применять ключи и сертификаты, полученные от Яндекс.Директа. Некоторые библиотеки SSL автоматически создают ключи и сертификаты прозрачно для приложения. Однако по таким сертификатам невозможно идентифицировать пользователя, поэтому необходимо применять авторизацию по токену.

Практика использования

В разделе содержатся сведения о выполнении различных задач с помощью API Яндекс.Директа и об ограничениях на использование API.

Ограничения на использование API

Ограничения доступа к API

По умолчанию доступ к API запрещен пользователям «Легкого» интерфейса и клиентам агентств. Если вы являетесь пользователем «Легкого» интерфейса, для получения доступа к API Директа необходимо перейти на «Профессиональный» интерфейс.

Доступ к API может быть ограничен по IP-адресу пользователя для обеспечения дополнительной безопасности. Чтобы запретить работу со своими рекламными кампаниями пользователям с любыми IP-адресами, кроме своего, напишите письмо в службу поддержки по адресу: api@direct.yandex.ru.

Пользователям, которые не имеют доступа к API, сервис возвращает ошибку с кодом 510 (Access denied).

Технические ограничения

Накладываются на количество вызовов метода в течение суток:

- UpdatePrices — 3000 вызовов на кампанию, за один вызов разрешено передавать не более 1000 фраз.
- CreateNewReport — 300 вызовов на кампанию.
- CreateOrUpdateCampaign — 100 кампаний на одну учетную запись.
- CreateOrUpdateBanners — 1000 объявлений на одну кампанию.
- GetKeywordsSuggestion — 3000 вызовов на одну учетную запись.
- Все остальные методы — 50 000 вызовов на каждый метод в сутки.

При превышении разрешенного количества вызовов любого из методов вступает в действие следующее ограничение: до начала новых суток каждый метод может вызываться не чаще одного раза в 10 минут.

Также ограничения накладываются на количество кампаний, объявлений или фраз в методах:

- UpdatePrices — 1000 фраз.
- CreateNewForecast — 100 фраз.
- CreateNewWordstatReport — 10 фраз в одном запросе и 1000 фраз в течение суток.
- GetBanners — 2000 объявлений или 10 кампаний в одном запросе.
- GetBannerPhrases, GetBannerPhrasesFilter — 1000 объявлений в одном запросе.

Ограничения накладываются на количество клиентов в методах:

- CreateNewSubclient — 100 субклиентов для одного агентства в сутки.

Ограничения накладываются на количество одновременно выполняющихся запросов. Разрешается отправлять не более пяти одновременных запросов на одну учетную запись.

Балльные ограничения

API Яндекс.Директа предоставляется бесплатно, но количество и характер доступных операций, которые пользователь может произвести в течение суток, зависят от количества баллов на аккаунте. Баллы API пересчитываются и начисляются один раз в сутки.

На количество доступных для использования баллов влияют следующие факторы:

- количество отклоненных на модерации объявлений;
- количество фраз, отключенных за низкий CTR;
- средний CTR рекламной кампании;
- средний бюджет рекламной кампании.

Чтобы увеличить количество доступных баллов, пользователю необходимо оптимизировать свои рекламные кампании: добавлять и уточнять ключевые слова, работать с текстом объявления с целью увеличения CTR, увеличивать минимальные ставки.

Примечание:
Для проверки количества доступных баллов на аккаунте пользователя предназначен метод GetClientsUnits.

Стоимость операций, имеющих балльные ограничения:

- добавление объявления — 12 баллов;
- редактирование объявления — 4 балла;
- добавление новой фразы — 2 балла;
- редактирование фразы — 1 балл;
- получение подсказок к ключевым словам — 3 балла за запрос;
- получение статистики запросов — 10 баллов за фразу.

Баллы списываются со счета при вызове методов:

- CreateOrUpdateBanners;
- GetKeywordsSuggestion;
- CreateNewWordstatReport.

После совершения операции ее стоимость и остаток баллов можно узнать из ответа сервера.

Работа с рекламной кампанией

Чтобы создать новую рекламную кампанию:

1. Инициализируйте набор параметров для рекламной кампании.
2. Создайте запрос в необходимом формате (см. CreateOrUpdateCampaign).
3. Вызовите метод CreateOrUpdateCampaign с выбранным набором параметров.

С помощью этого же метода можно изменить параметры существующей рекламной кампании. Для этого в параметрах необходимо указать ее `CampaignID`. Затем, с помощью метода CreateOrUpdateBanners, можно создавать и изменять объявления внутри кампании.

См. также
Справочник методов API
Примеры

Получение отчета по статистике

Чтобы получить XML-отчет по статистике рекламной кампании:

1. Подготовьте и передайте на вход метода CreateNewReport структуру данных, в которой указано, какие данные необходимо видеть в отчете, и каков способ их сортировки.

   В ответ получите `ReportID` будущего отчета.

2. По полученному `ReportID` с интервалом 10-30 секунд проверяйте статус отчета с помощью метода GetReportList до тех пор, пока он не будет создан.
3. Скачайте готовый отчет с помощью **HTTPS**-запроса по адресу, указанному в параметре `URL`.

Внимание!
При загрузке отчета необходимо передавать сертификат API.

Отчет будет доступен по указанному адресу в течение пяти часов. Если по каким-либо причинам забрать отчет не получилось, повторите всю процедуру заново. Формат отчета описан в примере.

См. также
CreateNewReport
GetReportList
Пример файла отчета по статистике рекламной кампании

Прогноз бюджета

Примечание:
Реальный бюджет кампании может отличаться от прогнозируемого, так как он подсчитан на основе анализа ставок конкурентов и **CTR** их кампаний, а эти параметры могут изменяться в процессе работы рекламной кампании. Кроме этого, в прогнозе бюджета не учитываются настройки временного таргетинга и показы объявлений на сайтах-участниках Рекламной сети Яндекса.

Чтобы рассчитать бюджет рекламной кампании:

1. Подготовьте необходимые фразы, рубрики каталога и регионы для показа.
2. Создайте заявку на расчет бюджета (с помощью метода CreateNewForecast).

   Метод вернет номер заявки (`ForecastID`) на составление отчета с прогнозом бюджета.

3. Периодически проверяйте готовность отчета, вызывая метод GetForecastList.

   Метод возвращает список заявок на прогноз бюджета и их статусы. Как только метод сообщит, что заявка с интересующим номером имеет статус «Done», можно загружать отчет.

4. Загрузите отчет по номеру заявки с помощью метода GetForecast.

Отладка приложения

При отладке клиентского приложения полезно иметь возможность просматривать XML-данные, которыми обмениваются веб-сервис и клиент. Для этого при создании SOAP-запросов используйте методы отладки, предоставляемые выбранной SOAP-библиотекой.

Примечание:
В примерах вся информация, включая текст **SOAP-запроса** и ответ сервера (заголовок+XML), будет выведена в **STDOUT** (стандартный вывод).

Perl (SOAP::Lite)

```
$soap -> on_debug( sub { print @_ })
```

PHP (nuSOAP)

```
print $client->request;        # вывод XML-запроса клиентского приложения
print $client->response;       # вывод XML-ответа сервера
```

PHP (SOAPClient)

```
$client->__getLastRequest();   # вывод XML-запроса клиентского приложения*
$client->__getLastResponse();  # вывод XML-ответа сервера*
```

Python (SOAPpy)

```
config.debug = 1
```

См. также
Протокол SOAP
Примеры

Получение цифровых сертификатов

Инструкции по получению сертификатов и криптографических ключей через веб-интерфейс Яндекс.Директа.

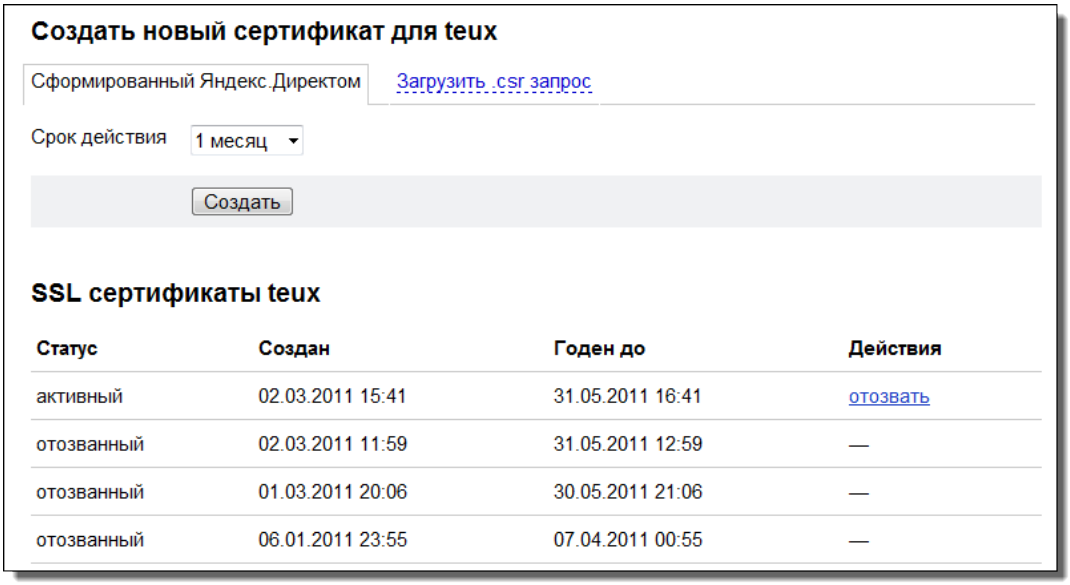
Получение сертификата в интерфейсе Яндекс.Директа

Персональный сертификат для установки SSL-соединения и идентификации пользователя можно получить в интерфейсе Яндекс.Директа, как описано ниже.

1. На странице создания сертификата перейдите на вкладку **Сформированный Яндекс.Директом**.

На вкладке отображается список персональных сертификатов.

Вкладка «Сформированный Яндекс.Директом»



Примечание:
Активными могут быть не более двух сертификатов. Если это так, для создания сертификата необходимо отозвать один из активных сертификатов, нажав на ссылку **Отозвать**.

2. Выберите срок действия сертификата — 1 месяц, 2 месяца или 3 месяца.
3. Нажмите на кнопку **Создать**.

   Яндекс.Директ отправит zip-архив со следующими файлами:

   - `cert.crt` — персональный сертификат пользователя;
   - `private.key` — секретный ключ шифрования (используется для декодирования данных, получаемых от API Яндекс.Директа);
   - `cacert.pem` — собственный сертификат API Яндекс.Директа (так называемый CA-сертификат);
   - `req.csr` — запрос на создание персонального сертификата.

Назначение файлов описано в разделе Цифровые сертификаты SSL.

Получение сертификата с помощью csr-запроса

Применение csr-запросов разделяет процедуру получения сертификата на два этапа. Первый этап выполняется на локальной машине с помощью библиотеки OpenSSL и заключается в генерации секретного ключа и запроса на сертификат (файл с расширением `csr`). Секретный ключ можно дополнительно защитить паролем, что не делается при генерации ключа на стороне Яндекс.Директа.

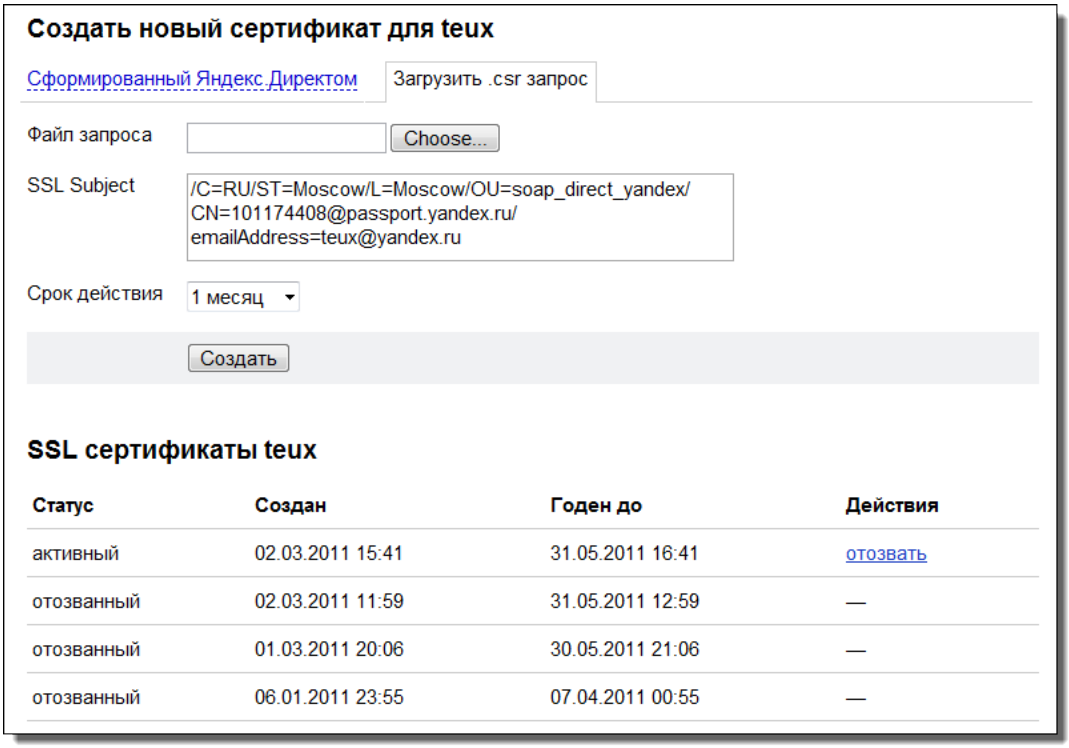
На втором этапе csr-запрос отправляют в Яндекс.Директ, а в ответ получают персональный сертификат, который может использоваться вместе с ранее созданным секретным ключом для установки SSL-соединений и идентификации пользователя.

Ниже описана процедура получения сертификата с помощью csr-запроса.

1. На странице создания сертификата перейдите на вкладку **Загрузить .csr запрос**.

На вкладке отображаются элементы для получения нового сертификата и список выданных сертификатов.

Вкладка «Загрузить .csr запрос»



Примечание:
Активными могут быть не более двух сертификатов. Если это так, для создания сертификата необходимо отозвать один из активных сертификатов, нажав на ссылку **Отозвать**.

---

2. Скопируйте текст из поля **SSL Subject**. Он понадобится для генерации csr-запроса на локальной машине.
3. На локальной машине выполните команду:

```
openssl req -new -keyout <имя_секретного_ключа>.pem -out <логин>.csr -subj
<SSL-subject>
```

В команде задают следующие аргументы:

- `keyout` — имя файла с расширением `pem`, в который помещается секретный ключ;
- `out` — имя файла с расширением `csr`, в который помещается запрос на сертификат (рекомендуется, чтобы имя файла совпадало с логином пользователя);
- `subj` — строка **SSL Subject**, скопированная из интерфейса Яндекс.Директа.

Ниже показан пример команды.

```
openssl req -new -keyout direct_private_key.pem -out teux.csr -subj /C=RU/
ST=Moscow/L=Moscow/OU=soap_direct_yandex/CN=101174108@passport.yandex.ru/
emailAddress=teux@yandex.ru
```

4. В ответ на приглашение введите и подтвердите пароль доступа к секретному ключу.

Команда создаст секретный ключ (файл с расширением `pem`) и запрос на сертификат (файл с расширением `csr`).

5. Вернитесь в интерфейс Яндекс.Директа на вкладку **Загрузить .csr запрос**.
6. Нажмите на кнопку **Choose** и выберите только что созданный csr-файл.
7. Нажмите на кнопку **Создать**.

   Яндекс.Директ возвратит zip-архив с персональным сертификатом `cert.crt`. Его и ранее созданный секретный ключ используют для установки SSL-соединения. Также из архива может понадобиться файл `cacert.pem`, что зависит от используемой SSL-библиотеки.

По окончанию срока действия сертификата csr-файл можно использовать для получения нового сертификата. Узнать срок действия сертификата можно с помощью команды:

```
openssl x509 -in certificate.crt -text -noout
```

В Windows срок действия сертификата можно узнать, дважды щелкнув по файлу сертификата в проводнике.

Методы API

Ниже приведены ссылки на описание методов API. Методы сгруппированы по целевому назначению.

Статистика и аналитика	Общие данные
Статистика кампании	**Работа с клиентами**
CreateNewReport	GetClientInfo
GetReportList	GetClientsList
DeleteReport	GetClientsUnits
GetStatGoals	GetSubClients
Подбор ключевых слов	CreateNewSubclient
CreateNewWordstatReport	**Яндекс.Каталог и регионы**
GetWordstatReportList	GetRegions
GetWordstatReport	GetRubrics
GetKeywordsSuggestion	**Прочие методы**
Прогноз бюджета	GetVersion
CreateNewForecast	GetAvailableVersions
GetForecastList	GetTimeZones
GetForecast	PingAPI
DeleteForecastReport	
Кампании и объявления	**Управление состояниями**
Информация о кампаниях	**Состояние компаний**
GetCampaignParams	StopCampaign
GetCampaignsList	ResumeCampaign
GetCampaignsListFilter	ArchiveCampaign
GetBalance	UnArchiveCampaign
GetBanners	DeleteCampaign
GetBannerPhrases	**Состояние объявлений**
GetBannerPhrasesFilter	ModerateBanners
Редактирование кампаний и объявлений	StopBanners
CreateOrUpdateCampaign	ResumeBanners
CreateOrUpdateBanners	ArchiveBanners
SetAutoPrice	UnArchiveBanners
UpdatePrices	DeleteBanners

Примечание:
В разделе Примеры демонстрируется вызов методов на популярных языках программирования с применением различных сетевых библиотек.

Статистика кампании

CreateNewReport

Описание

```
int CreateNewReport(NewReportInfo)
```

Запрос на создание отчета по статистике рекламной кампании за указанный период.

Возвращает ID будущего отчета. С помощью параметров `StatusReport` и `ReportID` метода GetReportList можно проверить готовность отчета и узнать идентификатор для скачивания.

Одновременно можно запросить не более пяти отчетов. При заказе шестого отчета возвращается сообщение об ошибке с кодом 31.

Внимание!
Количество вызовов метода для одной рекламной кампании в течение суток не может превышать 300. Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура `NewReportInfo`:

NewReportInfo

Поле	Тип	Значение
`CampaignID`	`int`	Идентификатор рекламной кампании (обязательно).
`StartDate`	`string`	Дата начала отчетного периода в формате `YYYY-MM-DD`. Например, `2007-05-01` (обязательно).
`CompressReport`	`int`	Возможные значения: • `1` — сжать готовый файл с помощью gzip; • 0 — не сжимать.
`EndDate`	`string`	Дата окончания отчетного периода (обязательно).
`Filter`	NewReportFilterInfo	Отфильтровать отчет по указанным в структуре NewReportFilterInfo параметрам.
`GroupByColumns`	`string[COLUMNS]`	Сгруппировать строки отчета по перечисленным в массиве колонкам. Колонки (`COLUMNS`): • `clBanner` — `BannerID`; • `clDate` — дата сбора статистики; • `clPage` — ID площадок показов; • `clGeo` — ID регионов показов; • `clPhrase` — ID фраз; • `clStatGoals`[v3] — ID целей; • `clPositionType`[v3] — позиция в показах на поиске Яндекса.
`GroupByDate`	`string`	Сгруппировать отчет по дням, неделям или месяцам: • `day` — по дням (по умолчанию); • `week` — по неделям; • `month` — по месяцам. При группировке отчета по дням недели или месяцам, в качестве `StartDate` будет указан первый день недели (месяца), а значения параметров `sum`, `shows`, и `clicks` будут просуммированы, соответственно, за неделю или за месяц.
`Limit`	`int`	Максимальное количество строк в отчете.
`Offset`	`int`	Отступить указанное количество строк от начала отчета. Указывается, если задан `Limit`.
`OrderBy`	`string[COLUMNS]`	Упорядочить строки отчета по указанным колонкам, см. `COLUMNS`.
`TypeResultReport`	`string`	Тип файла с отчетом. Единственно возможное значение — `xml`.

Структура `NewReportFilterInfo`.

NewReportFilterInfo

Поле	Тип	Значение
`Banner`	`int[]`	Массив идентификаторов объявлений (`BannerID`).
`Geo`	`int[]`	Массив идентификаторов регионов (`GeoID`).
`PageType`	`string`	Тип площадок. Возможные значения: • `search` — показывать статистику только для поисковых площадок; • `context` — показывать статистику только для тематических площадок; • `all` — показывать статистику для всех площадок (по умолчанию).
`PageName`	`string[]`	Массив названий площадок, на которых были показы. Название площадки Яндекса — «Яндекс». Для других площадок используются названия доменов, зарегистрированных в Яндекс.Директе, например `mail.ru`, `nigma.ru`.
`Phrase`	`string[]`	Массив искомых фраз. В отчет выводятся строки, содержащие любую из указанных фраз в качестве подстроки.
`StatGoals`[v3]	`int[]`	Идентификатор цели рекламной кампании. Указывается как элемент массива, однако массив может содержать только один идентификатор. Если идентификатор не задан, в отчет выводятся данные по всем целям. Список целей рекламной кампании можно получить с помощью метода GetStatGoals.
`PositionType`[v3]	`string`	Позиция объявления в показах на поиске Яндекса. Возможные значения: • `premium` — спецразмещение; • `other` — другие позиции.

Возвращает

Параметр	Тип	Значение
`ReportID`	`int`	Идентификатор запрошенного отчета.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Получение отчета по статистике
Пример файла отчета по статистике рекламной кампании
GetReportList
DeleteReport
Коды ошибок при работе со статистикой

GetReportList

Описание

```
array GetReportList()
```

Возвращает структуру данных с информацией о всех доступных отчетах.

Входные параметры

Для того чтобы скачать список отчетов, необходимо передать сертификат API.

Возвращает

Массив структур данных `ReportInfo`:

```
[
    {ReportInfo},
      ...
    {ReportInfo}
]
```

ReportInfo

Поле	Тип	Описание
`ReportID`	`int`	Идентификатор отчета.
`Url`	`string`	URL сгенерированного отчета. Если отчет еще не готов, то `Url` содержит пустую строку.
`StatusReport`	`string`	Отражает текущий статус подготовки отчета: • `Done` — отчет готов; • `Pending` — отчет еще готовится. Если отчет готов, то он становится доступен по адресу, указанному в параметре `Url`. После получения отчета его можно удалить с помощью метода DeleteReport.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Получение отчета по статистике
CreateNewReport
DeleteReport
Пример файла отчета по статистике рекламной кампании
Коды ошибок при работе со статистикой

DeleteReport

Описание

```
int DeleteReport(int)
```

Удаляет отчет с указанным `reportID`.

Входные параметры

Поле	Тип	Значение
Обязательные		
`ReportID`	`int`	Идентификатор отчета.

Возвращает

Параметр	Тип	Значение
`result`	`int`	`1` — если отчет был успешно удален.

Примечание:
Отчеты автоматически удаляются через 5 часов после создания. При создании 6-ой заявки первая будет удалена независимо от статуса готовности.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Получение отчета по статистике
CreateNewReport
GetReportList
Пример файла отчета по статистике рекламной кампании
Коды ошибок при работе со статистикой

GetStatGoals

Описание

```
array GetStatGoals(CampaignIDInfo)
```

Возвращает информацию о целях счетчиков Яндекс.Метрики, установленных на сайтах рекламодателя. Список целей может потребоваться для анализа статистики по рекламной кампании (метод CreateNewReport).

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

Массив структур данных `StatGoalInfo`:

```
[
    {StatGoalInfo},
      ...
    {StatGoalInfo}
]
```

StatGoalInfo

Поле	Тип	Значение
`GoalID`	`int`	Идентификатор цели.
`Name`	`string`	Название цели.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateNewReport

Подбор ключевых слов

CreateNewWordstatReport

Описание

```
int CreateNewWordstatReport(NewWordstatReportInfo)
```

Делает запрос на создание отчета по статистике запросов пользователей к поиску Яндекса. Возвращает ID будущего отчета. Через некоторое время по этому номеру можно получить готовый отчет с помощью метода GetWordstatReport.

Готовность отчета можно проверить с помощью метода GetWordstatReportList.

Примечание:
Одновременно вы можете работать не более чем с 5 отчетами. При заказе 6-ого отчета первый отчет будет удален, а заказанный окажется в конце очереди.

Каждый запрос может содержать не более 10 фраз. В течение суток может быть запрошена статистика не более чем для 1000 фраз.

Внимание!
При вызове метода расходуются баллы с аккаунта владельца кампании. Если для выполнения операции недостаточно баллов, веб-сервис Яндекс.Директа возвращает ошибку с кодом 152.

Подробнее о балльных и технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура `NewWordstatReportInfo`:

NewWordstatReportInfo

Поле	Тип	Значение
Обязательные		
`Phrases`	`string[]`	Список ключевых слов, для которых необходимо получить статистику запросов.
Необязательные		
`GeoID`	`int[]`	Идентификаторы регионов показа.

Возвращает

Параметр	Тип	Значение
`ReportID`	`int`	Идентификатор запрошенного отчета.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetWordstatReportList
GetWordstatReport
GetRegions
GetKeywordsSuggestion
Общие коды ошибок

GetWordstatReportList

Описание

```
array GetWordstatReportList()
```

Возвращает структуру данных с информацией о всех доступных отчетах по статистике запросов, которые были созданы с помощью метода CreateNewWordstatReport.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `WordstatReportStatusInfo`:

```
[
    {WordstatReportStatusInfo},
      ...
    {WordstatReportStatusInfo}
]
```

WordstatReportStatusInfo

Поле	Тип	Описание
`ReportID`	`int`	Идентификатор отчета.
`StatusReport`	`string`	Отражает текущий статус подготовки отчета: • `Done` — отчет готов; • `Pending` — отчет еще готовится. Если отчет готов, его можно получить с помощью метода GetWordstatReport.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateNewWordstatReport
GetWordstatReport
Общие коды ошибок

GetWordstatReport

Описание

```
array GetWordstatReport(int)
```

Возвращает отчет по статистике запросов пользователей к поиску Яндекса.

Входные параметры

Параметр	Тип	Значение
ReportID	int	Идентификатор отчета.

Возвращает

Массив структур данных `WordstatReportInfo`:

```
[
    {WordstatReportInfo},
      ...
    {WordstatReportInfo}
]
```

WordstatReportInfo

Поле	Тип	Описание
Phrase	string	Фраза, для которой собрана статистика запросов.
GeoID	int[]	Идентификаторы регионов показа, если статистика запросов собрана для пользователей из заданных регионов.
SearchedWith	WordstatItem[]	Массив структур WordstatItem. Содержит статистику запросов на Яндексе, включающих заданную фразу.

Поле	Тип	Описание
SearchedAlso	WordstatItem[]	Массив структур WordstatItem. Содержит статистику других запросов, которые делали люди, искавшие заданную фразу.

WordstatItem

Поле	Тип	Описание
Phrase	string	Фраза запроса.
Shows	int	Предварительный прогноз числа показов в месяц.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateNewWordstatReport
GetWordstatReportList
Общие коды ошибок

GetKeywordsSuggestion

Описание

```
string[] GetKeywordsSuggestion(KeywordsSuggestionInfo)
```

Возвращает список «подсказок» к заданным ключевым словам.

Примечание:
Количество вызовов метода в сутки ограничено, а при вызове расходуются баллы с аккаунта кампании (см. Ограничения на использование API). Если баллов недостаточно, возвращается сообщение об ошибке 152. Чтобы обновить только ставки за клик, используется метод UpdatePrices, при вызове которого баллы не расходуются.

Входные параметры

KeywordsSuggestionInfo

Поле	Тип	Значение
Keywords	string[]	Список ключевых слов, для которых необходимо получить подсказку.

Возвращает

Поле	Тип	Значение
`Keywords`	`string[]`	Подсказки к ключевым словам.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Общие коды ошибок
CreateNewWordstatReport

Прогноз бюджета

CreateNewForecast

Описание

```
int CreateNewForecast(ForecastInfo)
```

Запрашивает прогноз бюджета рекламной кампании и возвращает `ForecastID` (номер заявки на составление прогноза). Через некоторое время по этому номеру можно получить готовый прогноз с помощью метода GetForecast.

Примечание:
Возможно одновременное размещение не более 5 (пяти) заявок на расчет бюджета. При создании 6-ой заявки первая будет удалена, независимо от статуса готовности. Созданный отчет доступен в течение 5 часов.

Внимание!
Количество фраз в запросе не может превышать 100.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура данных `ForecastInfo`:

```
{
    GeoID  => int[],
    Phrases => string[],
    Categories => int[]
}
```

ForecastInfo

Поле	Тип	Значение
Необязательные		
`GeoID`	`int[]`	Массив `GeoID` регионов показа объявления. Полный список регионов можно получить с помощью метода GetRegions. В этом поле не допускается использование буквенных символов. Если требуется прогноз по всем регионам, `GeoID` должен иметь значение 0 (задан по умолчанию).
`Phrases`	`string[]`	Массив фраз для расчета бюджета. Все фразы должны быть в кодировке UTF-8 и удовлетворять требованиям к ключевым словам.
`Categories`	`int[]`	Массив `RubricID` рубрик Яндекс.Каталога. Полный список можно получить с помощью метода GetRubrics.

Прогноз бюджета можно рассчитывать как для показов на поиске по фразам, так и для показов на страницах Яндекс.Каталога. Для этого необходимо в параметре `Categories` запроса указать список рубрик, полученных с помощью метода GetRubrics.

Пример структуры на входе

```
{
    GeoID  => [1,2,192],
    Phrases => ['слон','мамонт','большой слон -мамонт'],
    Categories => [11082,477,478]
}
```

Возвращает

Параметр	Тип	Значение
`ForecastID`	`int`	Номер заявки на составление прогноза.

В случае ошибки метод возвращает сообщение об ошибке посредством `SOAP::Fault`.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Прогноз бюджета
GetForecast
GetForecastList
Коды ошибок при работе с прогнозом бюджета

GetForecastList

Описание

```
array GetForecastList()
```

Возвращает список всех заказанных прогнозов и их текущих статусов.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `ForecastStatusInfo`:

```
[
    {ForecastStatusInfo},
      ...
    {ForecastStatusInfo}
]
```

ForecastStatusInfo

Поле	Тип	Значение
`ForecastID`	`int`	Номер заявки на составление прогноза, полученный с помощью метода CreateNewForecast.
`StatusForecast`	`string`	Текущий статус прогноза: • `Done` — прогноз готов; • `Pending` — прогноз находится в процессе подготовки.

По готовности (`StatusForecast => Done`) отчет можно получить с помощью метода GetForecast.

Примечание:
Если отсутствуют заказанные прогнозы, то вызов `GetForecastList` возвращает пустую структуру.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Прогноз бюджета
CreateNewForecast
GetForecast
Коды ошибок при работе с прогнозом бюджета

GetForecast

Описание

```
ForecastData GetForecast(int)
```

Возвращает прогноз стоимости кампании, предварительно заказанный с помощью метода CreateNewForecast.

В случае, если прогноз еще не готов (или удален), метод возвращает сообщение об ошибке `` `Forecast is not ready` ``.

Входные параметры

Обязательные		
ForecastID	int	Номер заявки на составление прогноза.

Возвращает

Структуру данных следующего формата:

```
{
    Phrases    => [
                   {BannerPhraseInfo},
                     ...
                  ],
    Categories => [
                   {BannerPhraseInfo},
                     ...
                  ],
    Common     => {ForecastCommonInfo}
}
```

BannerPhraseInfo

Поле	Тип	Описание
Phrase	string	Фраза или идентификатор Яндекс.Каталога.
IsRubric	string	Значение «Yes» — в параметре Phrase указан идентификатор Яндекс.Каталога, значение «No» — указана фраза.
Clicks	int	Прогнозируемое количество кликов по объявлению.
FirstPlaceClicks	int	Прогнозируемое количество кликов по объявлению при показе на первом месте.
PremiumClicks	int	Прогнозируемое количество кликов по объявлению при показе в спецразмещении.
Shows	int	Прогнозируемое количество показов объявления при поиске по указанной фразе или в указанной рубрике Яндекс.Каталога.
Min	double	Стоимость входа в гарантированные показы.
Max	double	Стоимость показа на первом месте.
PremiumMin	double	Стоимость входа в спецразмещение.
PremiumMax	double	Стоимость показа на первом месте в спецразмещении.
Market	double	Стоимость клика в Яндекс.Маркете.
CTR	double	Прогнозируемый CTR (Click-Through-Rate) .
FirstPlaceCTR	double	Прогнозируемый CTR при показе на первом месте.
PremiumCTR	double	Прогнозируемый CTR при показе в спецразмещении.

ForecastCommonInfo

Общая информация о прогнозируемой стоимости кампании.

Поле	Тип	Описание
Geo	int	Регионы показа объявления.
Shows	int	Прогнозируемое суммарное количество показов по данным фразам.
Clicks	int	Прогнозируемое суммарное количество кликов по данным фразам.
FirstPlaceClicks	int	Прогнозируемое суммарное количество кликов по данным фразам на первом месте.

Поле	Тип	Описание
PremiumClicks	int	Прогнозируемое суммарное количество кликов по данным фразам при ставке, достаточной для входа в спецразмещение.
Min	double	Прогнозируемый бюджет кампании при условии показов всех фраз по ставке входа в гарантированные показы.
Max	double	Прогнозируемый бюджет кампании при условии показов всех фраз на первом месте.
PremiumMin	double	Прогнозируемый бюджет кампании при условии показов всех фраз по ставке, достаточной для входа в спецразмещение.

Пример

Пример возвращаемой структуры данных:


```
{
  "data":
  {
    "Categories":[
      {
        "LowCTRWarning":"Yes",
        "CTR":0.167074342622521,
        "FirstPlaceClicks":168,
        "Max":0.01,
        "StatusPhraseModerate":"New",
        "PremiumClicks":916,
        "FirstPlaceCTR":0.183781776884773,
        "PremiumCTR":1.00026207740019,
        "Min":0.01,
        "IsRubric":"Yes",
        "PremiumMax":16.27,
        "PremiumMin":16.27,
        "LowCTR":"No",
        "Clicks":153,
        "Shows":91576,
        "Phrase":1,
        "Market":null,
        "ContextLowCTR":"No"
      }
    ],
    "Phrases":[
      {
        "LowCTRWarning":"No",
        "CTR":1.38364779874214,
        "FirstPlaceClicks":12,
        "Max":0.01,
        "StatusPhraseModerate":"New",
        "PremiumClicks":52,
        "FirstPlaceCTR":1.52201257861635,
        "PremiumCTR":6.54088050314465,
        "Min":0.01,
        "IsRubric":"No",
        "PremiumMax":1.59,
        "PremiumMin":1.59,
        "LowCTR":"No",
        "Clicks":11,
        "Shows":795,
        "Phrase":"butter",
        "Market":null,
        "ContextLowCTR":"No"
      }
    ],
    "Common":{
      "Max":1.8,
```

```
      "PremiumClicks":968,
      "Geo":2,
      "Clicks":164,
      "Min":1.64,
      "Shows":92371,
      "FirstPlaceClicks":180,
      "PremiumMin":14986
    }
  }
}
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Прогноз бюджета
CreateNewForecast
GetForecastList
Коды ошибок при работе с прогнозом бюджета

DeleteForecastReport

Удаляет прогноз бюджета по указанному идентификатору.

Удалять прогнозы бюджета не обязательно, поскольку они автоматически удаляются спустя пять часов после формирования.

Входные данные

Ниже показаны входные данные в формате JSON.

```
{
   method: "DeleteForecastReport",
   param: (int)
}
```

В параметре `param` указывают идентификатор удаляемого прогноза. Идентификатор можно получить методом GetForecastList.

Результирующие данные

При успешном удалении прогноза возвращается значение 1, как показано ниже.

```
{
   "data": 1
}
```

См. также
Примеры DeleteForecastReport

Информация о кампаниях

GetCampaignParams

Описание

```
CampaignInfo GetCampaignParams(CampaignIDInfo)
```

Возвращает полный список параметров кампании со значениями.

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

Структуру данных `CampaignInfo`:

CampaignInfo

Поле	Тип	Значение
`FIO`	`string`	ФИО владельца кампании.
`CampaignID`	`int`	ID рекламной кампании.
`Login`	`string`	Логин владельца кампании.
`Name`	`string`	Название рекламной кампании.
`StartDate`	`string`	Дата начала кампании в формате `YYYY-MM-DD`.
`AgencyName`	`string`	Название агентства. Поле содержит значение, если рекламная кампания создана через агентство.
`ManagerName`	`string`	Имя менеджера Яндекса. Поле содержит значение, если кампания обслуживается по тарифу «Беззаботный».
Данные статистики		
`Clicks`	`int`	Количество кликов за весь период существования кампании.
`Shows`	`int`	Количество показов за весь период существования кампании.
`Rest`	`float`	Остаток средств на счете кампании.
`Sum`	`float`	Общая сумма, внесенная на счет рекламной кампании за весь период ее существования.
Настройки		
`AddRelevantPhrases`[v3]	`string`	Использование показов по дополнительным релевантным фразам: • `Yes` — показы по дополнительным релевантным фразам включены; • `No` — показы по дополнительным фразам выключены.

Поле	Тип	Значение
Autobudget	string	Статус автобюджета: • Yes — включен; • No — выключен.
AutobudgetMaxPrice	double	Максимальная ставка для автобюджета, в условных единицах.
AutobudgetSum	double	Установленный для рекламной кампании бюджет на неделю, в условных единицах.
AutoOptimization	string	Использование автофокуса для кампании: • Yes — используется автофокус; • No — не используется. Автофокус рекомендуется использовать всегда. Преимущества описаны в справочной информации по сервису Яндекс.Директ.
ContextPricePercent	int	Максимальная цена за клик для показов на тематических сайтах РСЯ. Указывается в процентах от максимальной цены на поиске. Возможные значения: 10, 20, ..., 100.
ConsiderTimeTarget	string	Расчет цен по позициям: • Yes — не учитываются объявления конкурентов, остановленные временным таргетингом; • No — учитываются объявления конкурентов, остановленные временным таргетингом в момент выполнения текущего запроса.
ContextLimit	string	Способ ограничения показов на тематических сайтах РСЯ: • Default — расход бюджета на показы в РСЯ не задан; • Limited — расходы на показы в РСЯ удерживать в пределах, заданных параметром ContextLimitSum.
ContextLimitSum	int	Максимально допустимая доля расхода на показы на тематических площадках (в процентах) от общего расхода кампании. Возможные значения: 0, 10, 20, ..., 100.
DisabledDomains	string	Список сайтов-участников РСЯ, на которых запрещены показы объявлений из данной кампании.
DisabledIps	string	Список IP-адресов пользователей, для которых запрещены показы объявлений из данной кампании.
EmailNotification	EmailNotificationInfo	Уведомления по электронной почте о состоянии кампании.
MinusKeywords[v3]	string[]	Список единых минус-слов для всех объявлений кампании. Единые минус-слова дополняют существующие минус-слова во фразах, но не заменяют их.
RelevantPhrasesBudgetLimit[v3]	int	Максимально допустимая доля расхода на показы по дополнительным релевантным фразам (в процентах) от расхода на поиске Яндекса. Возможные значения: 10, 20, ..., 100, -1. Значение 100 используется по умолчанию. Значение -1 показывает, что бюджет на показы по релевантным фразам неограничен.
SmsNotification	SmsNotificationInfo	SMS-уведомления о состоянии кампании. Номер телефона должен быть указан в персональных данных пользователя для сервисов Яндекса.

Поле	Тип	Значение
StatusContextStop	string	Остановка показов объявлений на тематических сайтах РСЯ при отключении фраз за низкий CTR на поиске Яндекса: • Yes — показы в рекламной сети останавливаются; • No — показы не останавливаются.
StatusMetricaControl	string	Управление показами при неработающем сайте: • Yes — показы объявлений останавливаются, если сайт не работает; • No — показы не останавливаются. Для управления показами на сайте требуется Метрика.
StatusOpenStat	string	Интеграция с внешними системами статистики: • Yes — в ссылку рекламного объявления добавляется метка для учета внешними системами статистики; • No — внешние системы статистики не используются.
TimeTarget	TimeTargetInfo	Временной таргетинг кампании. Если поле имеет пустое значение, объявления кампании показываются без ограничений по времени.
Статусы		
IsActive	string	Статус показа объявлений в кампании на Яндекс.Директе: • Yes — объявления показываются в соответствии с настройками временного таргетинга, если на счете кампании достаточно средств; • No — объявления в кампании не показываются.
Status	string	Статус кампании (текстовое описание). Например: активная, на модерации, остановлена, архивирована и т. д.
StatusActivating	string	Активизирована ли кампания: • Yes — кампания активизирована; • Pending — идет активизация кампании.
StatusArchive	string	Помещена ли кампания в архив: • Yes — кампания помещена в архив; • No — кампания не архивировалась.
StatusBehavior	string	Использование поведенческого таргетинга при показе кампании: • Yes — учитываются предпочтения пользователей; • No — поведенческий таргетинг отключен.
StatusModerate	string	Принята ли кампания на модерации: • Yes — кампания принята на модерации; • No — кампания отклонена на модерации; • New — кампании присвоен статус «Черновик»; • Pending — кампания ожидает модерации.
StatusShow	string	Статус кампании, заданный пользователем: • Yes — кампания включена; • No — кампания остановлена. Значение параметра изменяется при вызове методов StopCampaign, ResumeCampaign.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetCampaignsListFilter
Общие коды ошибок

GetCampaignsList

Описание

```
array GetCampaignsList(string[])
```

Позволяет получить список кампаний для указанных логинов и их текущий статус.

Входные параметры

Поле	Тип	Значение
Необязательные		
Login	string[]	Массив логинов клиентов Яндекс.Директа. Если не задан, то считается, что запрошен список кампаний для текущего клиента. Параметр является необязательным только для прямых клиентов Яндекс.Директа. Для агентств должен указываться всегда.

Возвращает

Массив структур данных `ShortCampaignInfo`:

```
[
    {ShortCampaignInfo},
      ...
    {ShortCampaignInfo}
]
```

ShortCampaignInfo

Поле	Тип	Значение
CampaignID	int	ID рекламной кампании.
Login	string	Логин владельца кампании.
Name	string	Название кампании.
StartDate	date	Время начала кампании.
AgencyName	string	Название агентства. Поле содержит значение, если рекламная кампания создана через агентство.
ManagerName	string	Имя менеджера Яндекса. Поле содержит значение, если кампания обслуживается по тарифу «Беззаботный».

Поле	Тип	Значение
Sum	float	Общая сумма, внесенная на счет рекламной кампании за весь период ее существования. Сумма, потраченная на рекламную кампанию за период существования, определяется как разность значений: (Sum – Rest).
Rest	float	Остаток средств на счете кампании.
Shows	int	Количество показов за весь период существования кампании.
Clicks	int	Количество кликов за весь период существования кампании.
Status	string	Статус кампании (текстовое описание). Например: активная, на модерации, остановлена, архивирована и т. д.
StatusShow	string	Статус кампании, заданный пользователем: • Yes — кампания включена; • No — кампания остановлена. Значение параметра изменяется при вызове методов StopCampaign, ResumeCampaign.
StatusModerate	string	Принята ли кампания на модерации: • Yes — кампания принята на модерации; • No — кампания отклонена на модерации; • Draft — кампании присвоен статус «Черновик»; • Pending — кампания ожидает модерации.
StatusActivating	string	Активизирована ли кампания: • Yes — кампания активизирована; • Pending — идет активизация кампании.
StatusArchive	string	Помещена ли кампания в архив: • Yes — кампания помещена в архив; • No — кампания не архивировалась.
IsActive	string	Статус показа объявлений в кампании на Яндекс.Директе: • Yes — объявления показываются в соответствии с настройками временного таргетинга, если на счете кампании достаточно средств; • No — объявления в кампании не показываются.

Пример возвращаемой структуры

```
[
    {
        CampaignID => 635646,
        Login => MyYandexLogin,
        Name => "Новая",
        StartDate => 2007-02-15,
        Sum => 50,
        Rest => 50,
        Shows => 26,
        Clicks => 0,
        Status => Идут показы,
        StatusShow => Yes,
        StatusArchive => No,
        StatusActivating => Yes,
        StatusModerate => Yes,
        IsActive => Yes,
    },
    ...
]
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsListFilter
GetBanners
GetBannerPhrases
UpdatePrices
GetBalance
Общие коды ошибок

GetCampaignsListFilter

Описание

```
array GetCampaignsListFilter(GetCampaignsInfo)
```

Возвращает список кампаний для указанных логинов клиентов Яндекс.Директа и их текущий статус. Метод работает так же, как и GetCampaignsList, но дополнительно позволяет ограничить список возвращаемых полей, см. параметр Filter.

Входные параметры

Структура данных `GetCampaignsInfo`:

GetCampaignsInfo

Поле	Тип	Значение
Необязательные		

Поле	Тип	Значение
Logins	string[]	Массив логинов клиентов Яндекс.Директа. Если не задан, то считается, что запрошен список кампаний для текущего клиента. Параметр является необязательным только для прямых клиентов Яндекс.Директа. Для агентств должен указываться всегда.
Filter	CampaignsFilterInfo	Структура данных, содержащая возможные значения для статусов рекламной кампании. Если параметр не задан, то считается, что запрошен список всех кампаний для указанных клиентов.

CampaignsFilterInfo

Поле	Тип	Значение
StatusModerate	string[]	Возможные значения StatusModerate.
StatusActivating	string[]	Возможные значения StatusActivating.
StatusShow	string[]	Возможные значения StatusShow.
IsActive	string[]	Возможные значения IsActive.
StatusArchive	string[]	Возможные значения StatusArchive.

Возвращает

Массив структур данных `ShortCampaignInfo`:

```
[
    {ShortCampaignInfo},
      ...
    {ShortCampaignInfo}
]
```

ShortCampaignInfo

Поле	Тип	Значение
CampaignID	int	ID рекламной кампании.
Login	string	Логин владельца кампании.
Name	string	Название кампании.
StartDate	date	Время начала кампании.
AgencyName	string	Название агентства. Поле содержит значение, если рекламная кампания создана через агентство.
ManagerName	string	Имя менеджера Яндекса. Поле содержит значение, если кампания обслуживается по тарифу «Беззаботный».
Sum	float	Общая сумма, внесенная на счет рекламной кампании за весь период ее существования. Сумма, потраченная на рекламную кампанию за период существования, определяется как разность значений: (`Sum — Rest`).
Rest	float	Остаток средств на счете кампании.
Shows	int	Количество показов за весь период существования кампании.
Clicks	int	Количество кликов за весь период существования кампании.
Status	string	Статус кампании (текстовое описание). Например: активная, на модерации, остановлена, архивирована и т. д.

Поле	Тип	Значение
StatusShow	string	Статус кампании, заданный пользователем: • Yes — кампания включена; • No — кампания остановлена. Значение параметра изменяется при вызове методов StopCampaign, ResumeCampaign.
StatusModerate	string	Принята ли кампания на модерации: • Yes — кампания принята на модерации; • No — кампания отклонена на модерации; • Draft — кампании присвоен статус «Черновик»; • Pending — кампания ожидает модерации.
StatusActivating	string	Активизирована ли кампания: • Yes — кампания активизирована; • Pending — идет активизация кампании.
StatusArchive	string	Помещена ли кампания в архив: • Yes — кампания помещена в архив; • No — кампания не архивировалась.
IsActive	string	Статус показа объявлений в кампании на Яндекс.Директе: • Yes — объявления показываются в соответствии с настройками временного таргетинга, если на счете кампании достаточно средств; • No — объявления в кампании не показываются.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetBanners
GetBannerPhrases
UpdatePrices
GetBalance
Общие коды ошибок

GetBalance

Описание

```
array GetBalance(array)
```

Возвращает информацию о текущих суммах на счетах указанных кампаний.

Входные параметры

Поле	Тип	Значение
Обязательные		
CampaignID	int[]	Массив идентификаторов кампаний.

Список кампаний можно получить с помощью метода GetCampaignsList.

Возвращает

Массив структур данных CampaignBalanceInfo:

```
{
    CampaignID => {CampaignBalanceInfo},
     ...
    CampaignID => {CampaignBalanceInfo}
}
```

CampaignBalanceInfo

Поле	Тип	Значение
CampaignID	int	Идентификатор кампании.
Rest	float	Остаток средств на счете кампании.
Sum	float	Общая сумма, внесенная на счет рекламной кампании за весь период ее существования. Сумма, потраченная на рекламную кампанию за период существования, определяется как разность значений: (Sum — Rest).

Пример

Пример возвращаемой структуры данных (Perl):

```
$VAR1 = [
         {
             CampaignID => 123456,
             Sum => 110,
             Rest => 59
         },
   ...
         {
             CampaignID => 567890,
             Sum => 220,
             Rest => 58
         }
        ];
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetBannerPhrases
GetBanners
UpdatePrices
Коды ошибок при работе со статистикой

GetBanners

Описание

```
array GetBanners(GetBannersInfo)
```

Возвращает информацию об указанных объявлениях.

Внимание!
Для метода ограничен размер запроса: не более 2000 объявлений или не более 10 кампаний.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

В качестве входных параметров в структуре `GetBannersInfo` необходимо передавать либо идентификаторы тех объявлений, информацию о которых необходимо получить, либо идентификаторы кампаний, содержащих необходимые объявления.

GetBannersInfo

Поле	Тип	Значение
`BannerIDS`	`int[]`	Массив идентификаторов объявлений. Не указывается, если задан `CampaignIDS`. Может содержать не более 2000 идентификаторов.
`CampaignIDS`	`int[]`	Массив идентификаторов кампаний. Не указывается, если задан `BannerIDS`. Может содержать не более 10 идентификаторов.
`Filter`	BannersFilterInfo	Структура данных, содержащая возможные значения статусов рекламных кампаний и объявлений.

Примечание:
Если одновременно заданы идентификаторы объявлений `BannerIDS` и идентификаторы кампаний `CampaignIDS`, при выполнении запроса учитываются только идентификаторы объявлений.

BannersFilterInfo

Поле	Тип	Значение
`IsActive`	`string[]`	Возможные значения IsActive.
`StatusActivating`	`string[]`	Возможные значения StatusActivating.
`StatusArchive`	`string[]`	Возможные значения StatusArchive.
`StatusBannerModerate`	`string[]`	Возможные значения StatusBannerModerate.
`StatusPhoneModerate`	`string[]`	Возможные значения StatusPhoneModerate.
`StatusPhrasesModerate`	`string[]`	Возможные значения StatusPhrasesModerate.
`StatusShow`	`string[]`	Возможные значения StatusShow.

Возвращает

Массив структур данных `BannerInfo`:

```
{
    {BannerInfo},
      ...
    {BannerInfo}
}
```

BannerInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID кампании.
`BannerID`	`int`	ID объявления.
`Title`	`string`	Заголовок объявления.
`Text`	`string`	Текст объявления.
`Href`	`string`	Ссылка на страницу, на которую ведет рекламное объявление.
`Sitelinks`[v3]	`Sitelink[]`	Массив структур Sitelink, содержащий информацию о дополнительных ссылках. Используются для показов на первой позиции в спецразмещении. При создании дополнительных ссылок необходимо учитывать требования к текстам и адресам страниц.

Поле	Тип	Значение
Domain	string	URL объявления, показываемый пользователю. Совпадает со значением поля Href, кроме следующих случаев: • значение Domain установлено вручную менеджером Яндекс.Директа; • URL, указанный в поле Href, является редиректом (в поле Domain будет записан конечный адрес объявления).
Geo	int[]	Массив ID регионов показа объявления. Список всех регионов показа объявлений сервиса Яндекс.Директ можно получить с помощью метода GetRegions.
ContactInfo	ContactInfo	Виртуальная визитка рекламодателя.
AdWarnings[v3]	string[]	Отметки модератора для объявлений товаров и услуг, реклама которых ограничивается действующим законодательством. Объявления медицинской, алкогольной или табачной тематики сопровождаются предупреждениями стандартного вида (см. п. 9 в Правилах показа). Возможные значения: • medicine — для рекламы лекарственных средств, медицинских услуг, в том числе методов лечения, медицинской техники; • abortion — для рекламы медицинских услуг по искусственному прерыванию беременности; • tobacco — для рекламы табака и табачных изделий; • alcohol — для рекламы алкогольной продукции, пива и напитков, изготавливаемых на его основе.
IsActive	string	Показывается ли объявление: • Yes — объявление показывается в соответствии с настройками временного таргетинга, если на счете кампании достаточно средств; • No — объявление не показывается.
ModerateRejectionReasons[v3]	RejectReason[]	Массив структур RejectReason , который содержит описание причин отклонения по результатам модерации для текста объявления, фраз, контактной информации или дополнительных ссылок.
StatusActivating	string	Активизирована ли кампания: • Yes — кампания активизирована; • Pending — идет активизация кампании.
StatusArchive	string	Помещено ли объявление в архив: • Yes — объявление помещено в архив; • No — объявление не архивировалось.
StatusBannerModerate	string	Статус модерации для объявления: • Yes — объявление принято на модерации; • No — объявление отклонено на модерации; • New — объявлению присвоен статус «Черновик»; • Pending — объявление ожидает модерации; • PreliminaryAccept — объявление предварительно принято к показам, окончательный результат модерации будет позже.

Поле	Тип	Значение
StatusPhoneModerate	string	Статус модерации для визитки: • Yes — визитка принята на модерации; • No — визитка отклонена на модерации; • New — визитке присвоен статус «Черновик»; • Pending — визитка ожидает модерации.
StatusPhrasesModerate	string	Приняты ли фразы на модерации: • Yes — фразы приняты на модерации, • No — фразы отклонены на модерации, • New — фразам присвоен статус «Черновик»; • Pending — фразы ожидают модерации; • PreliminaryAccept — фразы предварительно приняты к показам, окончательный результат модерации будет позже.
StatusSitelinksModerate[v3]	string	Статус модерации для дополнительных ссылок: • Yes — дополнительные ссылки приняты на модерации; • No — дополнительные ссылки отклонены на модерации; • New — дополнительным ссылкам присвоен статус «Черновик»; • Pending — дополнительные ссылки ожидают модерации.
StatusShow	string	Включено ли объявление пользователем: • Yes — объявление включено; • No — показ объявления остановлен. Значение параметра изменяется при вызове методов StopBanners, ResumeBanners.
FixedOnModeration[v3]	string	Исправлен ли текст объявления во время модерации (исправляются только опечатки): • Yes — текст объявления исправлен модератором; • No — текст объявления остался без изменений.

ContactInfo

Поле	Тип	Значение
Country	string	Страна местонахождения владельца объявления.
City	string	Город.
CountryCode	string	Код страны для контактного телефона. Значение должно начинаться со знака «+». Например, для России указывается значение «+7».
CityCode	string	Код города для контактного телефона.
Phone	string	Контактный телефон.
PhoneExt	string	Добавочный номер телефона.
CompanyName	string	Название компании или ФИО предпринимателя.
OGRN[v3]	string	ОГРН, если рекламодатель — юридическое лицо.
Street	string	Улица.
House	string	Номер дома.
Build	string	Строение/корпус.
Apart	string	Квартира.

Поле	Тип	Значение
ContactPerson	string	Контактное лицо.
IMClient	string	Тип программы для мгновенного обмена сообщениями (IM): 'icq', 'jabber', 'skype', 'mail_agent'.
IMLogin	string	ID пользователя IM.
ExtraMessage	string	Дополнительная информация о рекламируемом товаре или услуге.
ContactEmail	string	Адрес электронной почты владельца объявления.
WorkTime	string	Представление интервалов показа объявления в виде списка значений, разделенных запятой (CSV), см. ниже.
PointOnMap[v3]	MapPoint	Координаты местоположения рекламодателя для метки на карте. Для создания метки на карте адрес рекламодателя должен быть указан с точностью до улицы. Если координаты не заданы, метка на карте ставится в соответствии с адресом.

Формат WorkTime:

W1;W2;H1;M1;H2;M2[[;W1;W2;H1;M1;H2;M2;]...], где

W1, W2 — номера дней недели начала и конца интервала показа;
H1, H2 — часы начала и конца интервала показа;
M1, M2 — минуты начала и конца интервала показа.

MapPoint

Поле	Тип	Значение
x	float	Долгота точки.
y	float	Широта точки.
x1	float	Долгота левого нижнего угла области на карте. Может принимать значение от -180 до 180.
y1	float	Широта левого нижнего угла области на карте. Может принимать значение от -90 до 90.
x2	float	Долгота правого верхнего угла области на карте. Может принимать значение от -180 до 180.
y2	float	Широта правого верхнего угла области на карте. Может принимать значение от -90 до 90.

RejectReason

Поле	Тип	Значение
Type	string	Тип объекта, отклоненного на модерации. Может принимать значения: • Banner; • Phrases; • ContactInfo; • Sitelinks.
Text	string	Описание причины отклонения.

Пример возвращаемой структуры

```
[
    {
        BannerID => 824952,
        StatusModerate => Yes,
        CampaignID => 635646,
        StatusArchive => No,
        Domain => ,
        Geo => ,
        StatusActivating => Yes,
        StatusShow => Yes,
        IsActive => Yes,
        StatusBannerModerate => Yes,
        Href => help.yandex.ru/direct/,
        Title =>  Эффективная реклама на Я.Директ!,
        Text => Как повысить эффективность рекламных кампаний. Узнай больше!,
        ContactInfo => {
                        WorkTime => ,
                        Build => 21,
                        ContactPerson => ,
                        ExtraMessage => ,
                        Street => Самокатная,
                        IMLogin => ,
                        House => 1,
                        City => Москва,
                        IMClient => ,
                        CountryCode => ,
                        CityCode => ,
                        Phone => ,
                        PhoneExt => ,
                        Country => Россия,
                        Apart => ,
                        CompanyName => ООО "Яндекс",
                        ContactEmail =>
                       }
    }
];
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetBannerPhrases
GetBalance
UpdatePrices
Коды ошибок при работе со статистикой
Коды ошибок при работе с кампаниями и объявлениями

GetBannerPhrases

Описание

```
array GetBannerPhrases(int[])
```

Возвращает список фраз для указанных объявлений.

Внимание!
Количество объявлений в запросе не может превышать 1000.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Поле	Тип	Значение
Обязательные		
BannerIDS	int[]	Массив BannerIDS. Массив может содержать не более 1000 идентификаторов.

Возвращает

Массив структур данных BannerPhraseInfo:

```
[
    {BannerPhraseInfo},
      ...
    {BannerPhraseInfo}
]
```

BannerPhraseInfo

Поле	Тип	Значение
CampaignID	int	Идентификатор рекламной кампании.
PhraseID	int	Идентификатор фразы.
BannerID	int	Идентификатор объявления.
Phrase	string	Текст фразы или идентификатор рубрики каталога (если IsRubric = Yes).
IsRubric	string	Является ли фраза рубрикой Яндекс.Каталога: • Yes — является рубрикой; • No — не является рубрикой. Если IsRubric = Yes, поле Phrase содержит идентификатор рубрики каталога.
Price	double	Текущая цена по фразе.
ContextPrice	double	Цена для фразы на сайтах-участниках Рекламной сети Яндекса.
AutoBudgetPriority	string	Приоритет фразы при использовании автобюджета. Определяет, как часто будет показываться объявление по заданной фразе. Возможные значения: • Low — низкий; • Medium — средний; • High — высокий.

Поле	Тип	Значение
CurrentOnSearch	string	Текущая цена на поиске по заданной фразе.
Max	double	Цена клика, обеспечивающая показ объявления на первом месте справа от результатов поиска.
Min	double	Цена клика, обеспечивающая вход в гарантированные показы.
PremiumMax	double	Стоимость первого спецразмещения, если отличается от PremiumMin. Иначе — цена входа в спецразмещение.
PremiumMin	double	Стоимость входа в спецразмещение.
Shows	int	Количество показов объявления по заданной фразе за последние 28 дней, в течение которых были показы объявлений по фразе.
Clicks	int	Количество кликов по фразе за последние 28 дней, в течение которых были показы объявлений по фразе.
Market	double	Максимальная ставка Яндекс.Маркета по данной фразе.
StatusPhraseModerate	string	Принята ли фраза на модерации: • Yes — фраза принята на модерации; • No — фраза отклонена на модерации; • New — фраза находится в статусе «Черновик».
LowCTR	string	Отключена ли фраза за низкий CTR: • Yes — фраза отключена; • No — показы объявлений по фразе продолжаются.
ContextLowCTR	string	Отключена ли фраза за низкий CTR на сайтах-участниках РСЯ: • Yes — фраза отключена; • No — показы объявлений на сайтах РСЯ по фразе продолжаются.
LowCTRWarning	string	Близка ли фраза к отключению за низкий CTR: • Yes — фраза может быть отключена за низкий CTR; • No — фраза продолжает работать без предупреждений.
MinPrice	double	Минимальная цена, действующая на поиске — это цена за клик, при которой возможны показы объявления на страницах результатов поиска Яндекса и поисковых сервисов Яндекса (http://direct.yandex.ru/help/?id=991166).
Prices	float[]	Массив ставок по указанной фразе, отсортированный по позициям размещения. Содержит список ставок, которые могут быть списаны за клик по объявлению, для пользователей из разных регионов и для различных позиций в рекламном блоке. Данные в массиве передаются без детализации. Для определения гарантированной цены клика рекомендуется использовать параметры Max, Min, PremiumMax, PremiumMin.
Coverage	CoverageInfo[]	Массив структур CoverageInfo, позволяющих по ставке определить охват аудитории на выдаче.
ContextCoverage	CoverageInfo[]	Массив структур CoverageInfo, позволяющих по ставке определить охват аудитории в Рекламной сети Яндекса.

Поле	Тип	Значение
AutoBroker	string	Статус Автоброкера для фразы: • Yes — включен; • No — выключен.

CoverageInfo

Поле	Тип	Значение
Price	float	Цена ставки.
Probability	float	Предполагаемый охват аудитории. Представляет собой прогноз вероятности показа объявления на страницах: • результатов поиска Яндекса при точном совпадении запроса пользователя с заданной ключевой фразой; • Яндекс.Каталога в выбранной рубрике; • Рекламной сети Яндекса. Охват аудитории зависит от общего количества объявлений по данной ключевой фразе, кликабельности и заданной максимальной цены клика. Для параметра Coverage значение возвращается в долях в диапазоне от 0 до 1. Например: 0,2. Для параметра ContextCoverage возвращается целочисленное значение в процентах в диапазоне от 0 до 100. Например: 20.

Пример возвращаемой структуры

```
[
    {
        Prices => [10.41, 4.02, 4.01, 3.91, 3.46, 3.39, 2.2, 2.03, 1.76, 1.38,
0.17, 0.11, 0.07, 0.05, 0.02, 0.01],
        Coverage => [
                      {
                       Probability => 0.474879
                       Price => 0.01
                      },
                      {
                       Probability => 0.62007
                       Price => 0.4
                      },
                      {
                       Probability => 0.661976
                       Price => 0.79
                      },
                      {
                       Probability => 0.698253
                       Price => 1.18
                      },
                      {
                       Probability => 0.729784
                       Price => 1.57
                      },
                      {
                       Probability => 0.757297
                       Price => 1.96
                      },
                      {
                       Probability => 0.780812
                       Price => 2.34
                      },
                      {
                       Probability => 0.802055
```

```
                      Price => 2.73
                     },
                     {
                      Probability => 0.820783
                      Price => 3.12
                     },
                     {
                      Probability => 0.837345
                      Price => 3.51
                     },
                     {
                      Probability => 0.852036
                      Price => 3.9
                     }
                    ],

        AutoBroker => Yes,
        Max => 4.02,
        PhraseID => 42478475,
        CurrentOnSearch => 0.02,
        CampaignID => 1329426,
        Min => 3.91,
        IsRubric => No,
        PremiumMax => 10.41,
        PremiumMin => 10.41,
        BannerID => 4063871,
        Price => 0.03,
        Clicks => 0,
        Shows => 0,
        Phrase => 'Доставка суши',
        Market => 0,
        MinPrice => 0.01,
    }
]
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetBannerPhrasesFilter
GetCampaignsList
GetBanners
UpdatePrices
GetBalance
Коды ошибок при работе со статистикой

GetBannerPhrasesFilter

Описание

```
array GetBannerPhrasesFilter(BannerPhrasesFilterRequestInfo)
```

Возвращает список фраз для указанных объявлений. Метод работает так же, как и GetBannerPhrases, но дополнительно позволяет ограничить список возвращаемых полей, см. параметр FieldsNames.

Примечание:
Количество объявлений в запросе не может превышать 1000.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура данных `BannerPhrasesFilterRequestInfo`:

BannerPhrasesFilterRequestInfo

Поле	Тип	Значение
`BannerIDS`	`int[]`	Массив BannerIDS. Массив может содержать не более 1000 идентификаторов.
`FieldsNames`	`string[]`	Массив, задающий имена возвращаемых полей. Возможные значения: • Phrase • IsRubric • Price • AutoBudgetPriority • CurrentOnSearch • Max • Min • PremiumMax • PremiumMin • Shows • Clicks • Market • StatusPhraseModerate • LowCTR • LowCTRWarning • MinPrice • Prices • Coverage • AutoBroker Полный список полей приведен в описании метода GetBannerPhrases.
`ConsiderTimeTarget`	`string`	Расчет цен по позициям: • `Yes` — не учитывать объявления конкурентов, остановленные временным таргетингом; • `No` — учитывать объявления конкурентов, остановленные временным таргетингом в момент выполнения текущего запроса.
`RequestPrices`[v3]	`string`	Возвращать в результате запроса данные статистики (количество показов и кликов) и данные торгов (рассчитанные цены по позициям, прогноз охвата аудитории): • `Yes` — результат запроса содержит данные статистики и данные торгов (используется по умолчанию); • `No` — результат запроса содержит только список фраз и их параметры. Значение `No` рекомендуется использовать для снижения времени выполнения запроса.

Возвращает

Массив структур данных `BannerPhraseInfo`:

```
[
    {BannerPhraseInfo},
      ...
    {BannerPhraseInfo}
]
```

BannerPhraseInfo

Поле	Тип	Значение
CampaignID	int	Идентификатор рекламной кампании.
PhraseID	int	Идентификатор фразы.
BannerID	int	Идентификатор объявления.
Phrase	string	Текст фразы или идентификатор рубрики каталога (если `IsRubric = Yes`).
IsRubric	string	Является ли фраза рубрикой Яндекс.Каталога: • `Yes` — является рубрикой; • `No` — не является рубрикой. Если `IsRubric = Yes`, поле `Phrase` содержит идентификатор рубрики каталога.
Price	double	Текущая цена по фразе.
ContextPrice	double	Цена для фразы на сайтах-участниках Рекламной сети Яндекса.
AutoBudgetPriority	string	Приоритет фразы при использовании автобюджета. Определяет, как часто будет показываться объявление по заданной фразе. Возможные значения: • `Low` — низкий; • `Medium` — средний; • `High` — высокий.
CurrentOnSearch	string	Текущая цена на поиске по заданной фразе.
Max	double	Цена клика, обеспечивающая показ объявления на первом месте справа от результатов поиска.
Min	double	Цена клика, обеспечивающая вход в гарантированные показы.
PremiumMax	double	Стоимость первого спецразмещения, если отличается от `PremiumMin`. Иначе — цена входа в спецразмещение.
PremiumMin	double	Стоимость входа в спецразмещение.
Shows	int	Количество показов объявления по заданной фразе за последние 28 дней, в течение которых были показы объявлений по фразе.
Clicks	int	Количество кликов по фразе за последние 28 дней, в течение которых были показы объявлений по фразе.
Market	double	Максимальная ставка Яндекс.Маркета по данной фразе.
StatusPhraseModerate	string	Принята ли фраза на модерации: • `Yes` — фраза принята на модерации; • `No` — фраза отклонена на модерации; • `New` — фраза находится в статусе «Черновик».

Поле	Тип	Значение
LowCTR	string	Отключена ли фраза за низкий CTR: • Yes — фраза отключена; • No — показы объявлений по фразе продолжаются.
ContextLowCTR	string	Отключена ли фраза за низкий CTR на сайтах-участниках РСЯ: • Yes — фраза отключена; • No — показы объявлений на сайтах РСЯ по фразе продолжаются.
LowCTRWarning	string	Близка ли фраза к отключению за низкий CTR: • Yes — фраза может быть отключена за низкий CTR; • No — фраза продолжает работать без предупреждений.
MinPrice	double	Минимальная цена, действующая на поиске — это цена за клик, при которой возможны показы объявления на страницах результатов поиска Яндекса и поисковых сервисов Яндекса (http://direct.yandex.ru/help/?id=991166).
Prices	float[]	Массив ставок по указанной фразе, отсортированный по позициям размещения. Содержит список ставок, которые могут быть списаны за клик по объявлению, для пользователей из разных регионов и для различных позиций в рекламном блоке. Данные в массиве передаются без детализации. Для определения гарантированной цены клика рекомендуется использовать параметры Max, Min, PremiumMax, PremiumMin.
Coverage	CoverageInfo[]	Массив структур CoverageInfo, позволяющих по ставке определить охват аудитории на выдаче.
ContextCoverage	CoverageInfo[]	Массив структур CoverageInfo, позволяющих по ставке определить охват аудитории в Рекламной сети Яндекса.
AutoBroker	string	Статус Автоброкера для фразы: • Yes — включен; • No — выключен.

CoverageInfo

Поле	Тип	Значение
Price	float	Цена ставки.
Probability	float	Предполагаемый охват аудитории. Представляет собой прогноз вероятности показа объявления на страницах: • результатов поиска Яндекса при точном совпадении запроса пользователя с заданной ключевой фразой; • Яндекс.Каталога в выбранной рубрике; • Рекламной сети Яндекса. Охват аудитории зависит от общего количества объявлений по данной ключевой фразе, кликабельности и заданной максимальной цены клика. Для параметра Coverage значение возвращается в долях в диапазоне от 0 до 1. Например: 0,2. Для параметра ContextCoverage возвращается целочисленное значение в процентах в диапазоне от 0 до 100. Например: 20.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetBanners
UpdatePrices
GetBalance
Коды ошибок при работе со статистикой

Редактирование кампаний и объявлений

CreateOrUpdateCampaign

Описание

```
int CreateOrUpdateCampaign(CampaignInfo)
```

Создает или редактирует рекламную кампанию.

Внимание!
Количество вызовов метода в течение суток ограничено.

Подробнее о балльных и технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура `CampaignInfo`:

CampaignInfo

Поле	Тип	Значение
Обязательные		
`Login`	`string`	Логин владельца кампании.
`CampaignID`	`int`	ID рекламной кампании, 0 — если требуется создать новую кампанию.
`Name`	`string`	Название рекламной кампании.
`FIO`	`string`	ФИО владельца кампании.
`EmailNotification`	EmailNotificationInfo	Уведомления по электронной почте о состоянии кампании.
Необязательные		
`AddRelevantPhrases`[v3]	`string`	Использование показов по дополнительным релевантным фразам: • `Yes` — показы по дополнительным релевантным фразам включены; • `No` — показы по дополнительным фразам выключены.
`Autobudget`	`string`	Статус автобюджета: • `Yes` — включен; • `No` — выключен.
`AutobudgetMaxPrice`	`double`	Максимальная ставка для автобюджета, в условных единицах.
`AutobudgetSum`	`double`	Установленный для рекламной кампании бюджет на неделю, в условных единицах.
`AutoOptimization`	`string`	Использование автофокуса для кампании: • `Yes` — использовать автофокус; • `No` — не использовать.

Поле	Тип	Значение
		Автофокус рекомендуется использовать всегда. Преимущества описаны в справочной информации по сервису Яндекс.Директ.
ConsiderTimeTarget	string	Расчет цен по позициям: • Yes — не учитывать объявления конкурентов, остановленные временным таргетингом; • No — учитывать объявления конкурентов, остановленные временным таргетингом в момент выполнения текущего запроса.
ContextLimit	string	Способ ограничения показов на тематических сайтах РСЯ: • Default — расход бюджета на показы в РСЯ не задан; • Limited — расходы на показы в РСЯ удерживать в пределах, заданных параметром ContextLimitSum.
ContextLimitSum	int	Максимально допустимая доля расхода на показы на тематических площадках (в процентах) от общего расхода кампании. Возможные значения: 0, 10, 20, ..., 100.
ContextPricePercent	int	Максимальная цена за клик для показов на тематических сайтах РСЯ. Указывается в процентах от максимальной цены на поиске. Возможные значения: 10, 20, ..., 100.
DisabledDomains	string	Список сайтов-участников РСЯ, на которых необходимо запретить показы объявлений из данной кампании.
DisabledIps	string	Список IP-адресов пользователей, для которых необходимо запретить показы объявлений из данной кампании.
MinusKeywords[v3]	string[]	Список единых минус-слов для всех объявлений кампании. Единые минус-слова дополняют существующие минус-слова во фразах. Единые минус-слова не изменяют существующие фразы объявлений. Для редактирования фраз используйте метод CreateOrUpdateBanners. При пересечении одного из единых минус-слов с каким-либо словом в исходной ключевой фразе, минус-слово игнорируется.
RelevantPhrasesBudgetLimit[v3]	int	Максимально допустимая доля расхода на показы по дополнительным релевантным фразам (в процентах) от расхода на поиске Яндекса. Возможные значения: 10, 20, ..., 100, -1. Значение 100 используется по умолчанию. Значение -1 показывает, что бюджет на показы по релевантным фразам неограничен.
SmsNotification	SmsNotificationInfo	SMS-уведомления о состоянии кампании. Перед изменением параметров SMS-уведомлений убедитесь, что номер телефона указан в персональных данных пользователя для сервисов Яндекса.
Status	string	Статус кампании (текстовое описание). Например: активная, на модерации, остановлена, архивирована и т. д.
StatusBehavior	string	Использование поведенческого таргетинга при показе кампании: • Yes — учитывать предпочтения пользователей; • No — не учитывать.
StatusContextStop	string	Остановка показов объявлений на тематических сайтах РСЯ при отключении фраз за низкий CTR на поиске Яндекса: • Yes — останавливать показы в рекламной сети, • No — продолжать показывать.
StartDate	string	Дата начала кампании в формате YYYY-MM-DD.

Поле	Тип	Значение
StatusMetricaControl	string	Управление показами при неработающем сайте: • Yes — останавливать показы, если сайт не работает; • No — не останавливать показы. Для управления показами на сайте требуется Метрика.
StatusOpenStat	string	Интеграция с внешними системами статистики: • Yes — добавлять в ссылку рекламного объявления метку для учета внешними системами статистики; • No — внешние системы статистики не используются.
TimeTarget	TimeTargetInfo	Временной таргетинг кампании.

EmailNotificationInfo

Поле	Тип	Описание
Email	string	Адрес электронной почты для уведомлений.
SendWarn	string	Yes — получать уведомления на указанный адрес электронной почты, No — не получать.
MoneyWarningValue	int	Нижний порог остатка денег на счету (в процентах от суммы последней оплаты), по достижению которого начнут рассылаться уведомления.
WarnPlaceInterval	int	Интервал в минутах, задающий частоту рассылки уведомлений.
SendAccNews	string	Yes — получать уведомления о событиях по данной кампании, No — не получать. Значение задается для клиентов с тарифом «Беззаботный».

SmsNotificationInfo

Поле	Тип	Описание
SmsPhone	string	Номер телефона для рассылки SMS-уведомлений. Для получения уведомлений необходимо на них подписаться и подтвердить подписку. Номер телефона хранится в персональных данных пользователя для всех сервисов Яндекса.
SmsTimeFrom	string	Начало интервала времени для рассылки SMS-уведомлений. Значение задается в формате HH:MM. Например, для рассылки уведомлений может быть выбрана первая половина дня.
SmsTimeTo	string	Окончание интервала времени для рассылки SMS-уведомлений. Значение задается в формате HH:MM.
ModerateResultSms	string	Yes — рассылать SMS-уведомления о результатах модерации, No — не рассылать.
MetricaSms	string	Yes — рассылать SMS-уведомления о результатах мониторинга сайтов, No — не рассылать.
MoneyInSms	string	Yes — рассылать SMS-уведомления о поступлении средств на счет кампании, No — не рассылать.
MoneyOutSms	string	Yes — рассылать SMS-уведомления об окончании средств на счете кампании, No — не рассылать.

TimeTargetInfo

Поле	Тип	Описание
Необязательные		

Поле	Тип	Описание
ShowOnHolidays	string	Показ объявлений в праздничные дни: • Yes — показывать; • No — запретить показы в праздничные дни.
HolidayShowFrom	int	С какого часа показывать объявления в праздничные дни.
HolidayShowTo	int	До какого часа показывать объявления в праздничные дни.
TimeZone	string	Временная зона пользователя (владельца рекламной кампании). Список временных зон можно получить с помощью метода GetTimeZones.
Обязательные		
DaysHours	TimeTargetItem[]	Массив структур TimeTargetItem, содержащих данные о днях и часах для показа объявлений.

TimeTargetItem

Поле	Тип	Описание
Days	int[]	Перечисление рабочих дней кампании. Дни нумеруются числами от 1 до 7 включительно. Например: • 1 — понедельник; • 7 — воскресенье.
Hours	int[]	Перечисление рабочих часов кампании. Часы нумеруются числами от 0 до 23 включительно. Например: • 0 — показы объявлений с 00:00 до 00:59:59; • 23 — показы объявлений с 23:00 до 23:59:59.

Возвращает

Параметр	Тип	Значение
CampaignID	int	Идентификатор созданной или отредактированной кампании.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateOrUpdateBanners
Общие коды ошибок

CreateOrUpdateBanners

Описание

```
int[] CreateOrUpdateBanners(array)
```

Добавляет или редактирует объявление. Кампания может иметь не более 1000 объявлений.

Внимание!
Количество вызовов метода в сутки ограничено, а при вызове расходуются баллы с аккаунта кампании (см. Ограничения на использование API). Если баллов недостаточно, возвращается сообщение об ошибке 152. Чтобы обновить только ставки за клик, используется метод UpdatePrices, при вызове которого баллы не расходуются.

Редактирование отличается от добавления только тем, что в запросе указывают идентификатор редактируемого объявления. Чтобы добавить объявление, идентификатор не указывают.

При редактировании необходимо указывать все параметры объявления, даже если они остаются неизменными. Отсутствие какого-либо параметра приводит к удалению его значения. Это относится к тексту, заголовку, фразам, контактной информации и другим параметрам.

Рекомендуется начинать редактирование с получения параметров объявления методами GetBanners и GetBannerPhrases. Из полученных параметров формируют структуру для вызова метода `CreateOrUpdateBanners` (описана ниже), вносят необходимые изменения, например редактируют или добавляют параметры, и вызывают метод.

Примечание:
Добавленные и измененные объявления автоматически отправляются на модерацию при условии, что кампания активна. Для кампаний в статусе «Черновик» отправить объявление на модерацию можно с помощью метода ModerateBanners. При этом статус «Черновик» меняется на активный статус, после чего добавляемые и редактируемые объявления автоматически отправляются на модерацию.

Одобренные объявления начинают показываться согласно настройкам кампании. Остановить показ можно с помощью метода StopBanners.

Входные параметры

Массив структур данных `BannerInfo`:

```
[
    {BannerInfo},
      ...
    {BannerInfo}
]
```

BannerInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID кампании.
`BannerID`	`long`	Идентификатор объявления, если необходимо изменить его параметры. Чтобы создать новое объявление, не указывайте значение параметра или укажите значение 0.

Поле	Тип	Значение
Title	string	Заголовок объявления.
Text	string	Текст объявления.
Geo	string	Список идентификаторов для регионов показа объявления. Значения должны быть разделены запятыми. Список всех регионов показа объявлений сервиса Яндекс.Директ можно получить с помощью метода GetRegions.
Phrases	BannerPhraseInfo[]	Массив структур BannerPhraseInfo, содержащих информацию о фразах объявлений. Необходимо включить в массив по меньшей мере одну фразу.
Обязателен хотя бы один из следующих параметров		
Href	string	Ссылка на сайт. Параметр обязательный, если не указана контактная информация в параметре ContactInfo.
ContactInfo	ContactInfo	Информация о клиенте Яндекс.Директа. Параметр обязательный, если не указана ссылка на сайт в параметре Href.
Необязательные		
Sitelinks[v3]	Sitelink[]	Массив структур Sitelink. Содержит информацию о дополнительных ссылках, которые используются для показов на первой позиции в спецразмещении. При создании дополнительных ссылок необходимо учитывать требования к текстам и адресам страниц.

Примечание:
Необходимо включить по меньшей мере одну фразу в массив Phrases и указать ссылку на сайт в параметре Href либо контактную информацию в структуре ContactInfo (либо то и то). В противном случае возвращается сообщение об ошибке 71.

BannerPhraseInfo

Поле	Тип	Описание
Phrase	string	Текст фразы или идентификатор рубрики каталога (если IsRubric = Yes).
PhraseID	int	ID фразы.
IsRubric	string	Является ли фраза рубрикой Яндекс.Каталога: • Yes — является рубрикой; • No — не является рубрикой. Если IsRubric = Yes, поле Phrase должно содержать идентификатор рубрики каталога.
Price	double	Текущая цена для фразы.
ContextPrice	float	Цена для фразы на сайтах-участниках Рекламной сети Яндекса.
AutoBroker	string	Включен ли автоброкер для фразы: • Yes — включен; • No — выключен. Автоброкер рекомендуется включать всегда. Преимущества описаны в справочной информации по сервису Яндекс.Директ.

Поле	Тип	Описание
AutoBudgetPriority	string	Приоритет фразы при использовании автобюджета. Определяет, как часто будет показываться объявление по заданной фразе. Возможные значения: • Low — низкий; • Medium — средний; • High — высокий.

Примечание:
При попытке изменить значение поля ContextPrice для фразы, которая НЕ была отключена за низкий CTR на поисковых площадках, метод вернет ошибку с кодом 191.

Для активных фраз управляйте ценой клика на тематических сайтах РСЯ с помощью параметра ContextPricePercent (см. описание метода CreateOrUpdateCampaign).

Примечание:
Для фраз, отключенных на поиске за низкий CTR, в торгах учитывается только значение параметра ContextPrice. Значение параметра Price можно изменять, но оно учитываться не будет.

Sitelink

Поле	Тип	Описание
Title	string	Текст ссылки. При создании дополнительных ссылок необходимо учитывать требования к текстам и адресам страниц.
Href	string	Ссылка на страницу сайта.

Пример структуры на входе

```
CreateOrUpdateBanners => [
    {
        CampaignID => '327634',
        BannerID => '2000135',
        Title => 'API title',
        Text => 'API banner body',
        Href => 'http://market-click2.ya.ru/?q=qu',
        Geo => '2,183,3,5',
        Phrases => [
            {
                IsRubric => 'No',
                Phrase => 'new phrase',
                PhraseID => '23021604',
                AutoBroker => 'No',
                Price => '0.77',
                ContextPrice => '0.62',
                AutobudgetPriority => 'High'
            },
            {
                IsRubric => 'No',
                Phrase => 'this is phrase -2',
                PhraseID => '23021605',
                AutoBroker => 'Yes',
                Price => '0.33',
                ContextPrice => '0.25',
                AutobudgetPriority => 'Medium'
```

```
            },
            {
                IsRubric => 'Yes',
                Phrase => '231',
                AutoBroker => 'Yes',
                Price => '0.17',
                ContextPrice => '0.14',
                AutobudgetPriority => 'Low'
            },
        ],

        ContactInfo => {
            WorkTime => '0;4;10;00;18;00;5;6;13;00;16;00',
            City => 'Москва',
            Street => 'Самокатная',
            House => 21,
            Apart => 6,
            Build => 2,
            ContactPerson => 'контактное лицо',
            ExtraMessage => 'подробнее о товаре/услуге',
            IMClient => 'jabber',
            IMLogin => 'user@ya.ru',
            CountryCode => '+7',
            PhoneExt => '495'
            Phone => '5552233',
            Country => 'Россия',
            CompanyName => 'ООО "Яндекс"',
            ContactEmail => 'user@yandex.ru'
        },

        Sitelinks => [
            {
                'Title' => 'Дополнительная ссылка 1',
                'Href' => 'http://market-click2.ya.ru/?q=qu&action=1'
            },
            {
                'Title' => 'Дополнительная ссылка 2',
                'Href' => 'http://market-click2.ya.ru/?q=qu&action=2'
            },
            {
                'Title' => 'Дополнительная ссылка 3',
                'Href' => 'http://market-click2.ya.ru/?q=qu&action=3'
            }
        ]
    },
    {еще объявление},
     ...
    {еще объявление}
]
```

Возвращает

Параметр	Тип	Значение
BannerID	int[]	Массив идентификаторов измененных или созданных объявлений.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateOrUpdateCampaign
Коды ошибок при работе с обновлением ставок рекламной кампании

SetAutoPrice

Описание

```
array SetAutoPrice(AutoPriceInfo)
```

Устанавливает цены за клик в кампаниях, используя сложную форму. Цена за клик может быть либо фиксированная (режим SinglePrice), либо гибкая (режим Wizard).

Цены за клик не изменяются постоянно и автоматически, а устанавливаются однократно при вызове метода. Чтобы назначить другие цены, необходимо вызвать метод с новыми значениями параметров.

Примечание:
В настройках рекламной кампании цены изменяются в момент вызова метода. Перед тем как изменения вступят в силу и будут учитываться при показах в результатах поиска, кампания проходит активизацию. Активизация может занять до 40 минут, в периоды особенно высокой нагрузки — не более 3 часов.

Входные параметры

Структура данных `AutoPriceInfo`:

AutoPriceInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	Идентификатор рекламной кампании.
`Mode`	`string`	Режим назначения цены за клик: • `SinglePrice` — установлена фиксированая цена за клик. В режиме `SinglePrice` не указывается цена для режима `Wizard`. • `Wizard` — цена может изменяться по установленным правилам.
Параметры режима SinglePrice		
`SinglePrice`	`double`	Цена за клик в режиме `SinglePrice`.
Параметры режима Wizard		

Поле	Тип	Значение
PriceBase	string	Базовая цена. Возможные значения: • pmin — цена входа в спецразмещение; • pmax — цена показа на первом месте в спецразмещении; • max — цена первого места; • min — цена входа в гарантированное размещение.

ProcBase	string	Надбавка к базовой цене. Возможные значения:
		• value — определенный процент (Proc) от базовой цены (PriceBase);
		• diff — определенный процент (Proc) от разницы между текущей ценой клика и:
		  • ценой первого места (если PriceBase = min);
		  • ценой входа в спецразмещение (если PriceBase = max);
		  • ценой первого места в спецразмещении (если PriceBase = pmin).

MaxPrice	double	Максимально допустимая цена за клик. Не может быть выше 30 у.е.
Proc	int	Максимально допустимый процент отклонения от базовой цены (при ProcBase = value) или разницы цен (при ProcBase = diff). Например, значение 10 при ProcBase = value означает +10% от PriceBase.
UpdatePhrases	string	Изменять ли также цены фраз: • Yes — изменять цены фраз; • No — не изменять.
UpdateCategories	string	Изменять ли также цены рубрик: • Yes — изменять цены рубрик; • No — не изменять.

Пример структуры на входе

Режим SinglePrice:

```
params = {
          'CampaignID': 1477023,
          'Mode': 'SinglePrice',
          'SinglePrice': 0.8
         }
```

Режим Wizard:

```
params = {
          'CampaignID': 1477023,
          'Mode': 'Wizard',
          'PriceBase': 'pmin',
          'ProcBase': 'value',
          'MaxPrice': 30,
          'Proc': 60,
          'UpdatePhrases': 'Yes',
          'UpdateCategories': 'Yes'
}
```

Возвращает

Массив структур данных PhrasePriceInfo, который содержит измененные цены для фраз:

```
[
    {PhrasePriceInfo},
      ...
    {PhrasePriceInfo}
]
```

PhrasePriceInfo

Поле	Тип	Значение
CampaignID	int	Идентификатор рекламной кампании.
BannerID	int	Идентификатор объявления.
PhraseID	int	Идентификатор фразы.
Price	double	Установленная цена.

Пример возвращаемой структуры

```
[
    {
        CampaignID => 1477023,
        BannerID => 2245646,
        PhraseID => 1398789,
        Price => 0.12
    },
    {
        CampaignID => 1477023,
        BannerID => 2245646,
        PhraseID => 1298789,
        Price => 0.01
    }
]
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Коды ошибок при работе со статистикой

UpdatePrices

Описание

```
int UpdatePrices(array)
```

Изменяет цену для указанных фраз в указанной кампании. Список кампаний можно получить с помощью метода GetCampaignsList, список объявлений и их `BannerID` — с помощью GetBanners, список фраз и их `PhraseID` можно получить с помощью метода GetBannerPhrases.

Внимание!
Количество вызовов метода для одной рекламной кампании в течение суток не может превышать 3000. За один вызов разрешено передавать не более 1000 фраз.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Примечание:
В настройках рекламной кампании цены изменяются в момент вызова метода. Перед тем как изменения вступят в силу и будут учитываться при показах в результатах поиска, кампания проходит активизацию. Активизация может занять до 40 минут, в периоды особенно высокой нагрузки — не более 3 часов.

Входные параметры

Массив структур `BannerPriceInfo`:

```
[
    {BannerPriceInfo},
      ...
    {BannerPriceInfo}
]
```

BannerPriceInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID кампании.
`BannerID`	`int`	ID объявления.
`PhraseID`	`int`	ID искомой фразы.
Обязателен хотя бы один из следующих параметров		
`Price`	`double`	Новая цена.
`ContextPrice`	`float`	Цена для фразы на тематических сайтах РСЯ.
`AutoBroker`	`string`	Статус автоброкера для фразы: • `Yes` — включен; • `No` — выключен. Автоброкер рекомендуется включать всегда. Преимущества описаны в справочной информации по сервису Яндекс.Директ.
`AutoBudgetPriority`	`string`	Приоритет фразы при использовании автобюджета. Определяет, как часто будет показываться объявление по заданной фразе. Возможные значения: • `Low` — низкий; • `Medium` — средний; • `High` — высокий.

Примечание:
Требуется задать значение для, как минимум, одного из полей: `Price`, `ContextPrice`, `AutoBroker`, `AutoBudgetPriority`. Если значение ни одного из указанных полей не задано, метод вернет ошибку с кодом 242 (Bad price). Если не заданы значения для одного, двух или трех из указанных полей, то для фразы с данным `PhraseID` они останутся без изменений.

Примечание:
При попытке изменить значение поля `ContextPrice` для фразы, которая НЕ была отключена за низкий CTR на поисковых площадках, метод вернет ошибку с кодом 191.

Для активных фраз управляйте ценой клика на тематических сайтах РСЯ с помощью параметра ContextPricePercent (см. описание метода CreateOrUpdateCampaign).

Пример структуры на входе

```
{
    CampaignID => 635646,
    BannerID => 824952,
    PhraseID => 11141888,
    Price => 0.14,
    ContextPrice => 0.11,
    AutoBroker => Yes,
    AutoBudgetPriority => 'Low'
}
```

Возвращает

Поле	Тип	Значение
`Result`	`int`	Результат операции: • `1` — цена успешно изменена; • `0` — ошибка.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetCampaignsList
GetBalance
GetBanners
GetBannerPhrases
Коды ошибок при работе со статистикой

Состояние кампаний и объявлений

StopCampaign

Описание

```
int StopCampaign(CampaignIDInfo)
```

Останавливает указанную рекламную кампанию.

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

В случае успеха возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
ResumeCampaign
Общие коды ошибок

ResumeCampaign

Описание

```
int ResumeCampaign(CampaignIDInfo)
```

Запускает указанную рекламную кампанию.

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

В случае успеха возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
StopCampaign
Общие коды ошибок

ArchiveCampaign

Описание

```
int ArchiveCampaign(CampaignIDInfo)
```

Помещает рекламную кампанию в архив. Архивировать можно только остановленные кампании с нулевым балансом счетов.

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

В случае успеха возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
UnArchiveCampaign
Общие коды ошибок

UnArchiveCampaign

Описание

```
int UnArchiveCampaign(CampaignIDInfo)
```

Извлекает указанную рекламную кампанию из архива.

Входные параметры

Структура данных `CampaignIDInfo`:

CampaignIDInfo

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

В случае успеха возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
ArchiveCampaign
Общие коды ошибок

DeleteCampaign

Описание

```
int DeleteCampaign(CampaignIDInfo)
```

Удаляет указанную рекламную кампанию.

Примечание:
Кампанию можно удалить, если она находится в статусе «Черновик». Кампании, которые уже были оплачены и по которым была собрана статистика, удалить нельзя. Такие кампании можно только поместить в архив.

Входные параметры

Структура данных `CampaignIDInfo`:

`CampaignIDInfo`

Поле	Тип	Значение
`CampaignID`	`int`	ID рекламной кампании.

Возвращает

В случае успеха возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
StopCampaign
Общие коды ошибок

ModerateBanners

Описание

```
int ModerateBanners(CampaignBidsInfo)
```

Отправляет на модерацию указанные объявления, если они находятся на вкладке «Черновики». Вызов этого метода равносилен принятию оферты в интерфейсе службы Яндекс.Директ.

Чтобы получить идентификаторы объявлений в статусе «Черновик», вызовите метод GetBanners со следующими параметрами:

- `CampaignIDS` — укажите идентификатор кампании, объявления которой необходимо отправить на модерацию;
- `Filter:StatusBannerModerate` — укажите значение `New`.

Входные параметры

Структура данных `CampaignBidsInfo`:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID рекламной кампании.
`BannerIDS`	`int[]`	Массив `BannerID`.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
StopCampaign
Общие коды ошибок

StopBanners

Описание

```
int StopBanners(CampaignBidsInfo)
```

Приостанавливает показ указанных объявлений.

Входные параметры

Структура данных `CampaignBidsInfo`:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID рекламной кампании.
`BannerIDS`	`int[]`	Массив `BannerID`.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
StopCampaign
Общие коды ошибок

ResumeBanners

Описание

```
int ResumeBanners(CampaignBidsInfo)
```

Запускает показ указанных объявлений.

Входные параметры

Структура данных `CampaignBidsInfo`:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID рекламной кампании.
`BannerIDS`	`int[]`	Массив `BannerID`.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
StopBanners
Общие коды ошибок

ArchiveBanners

Описание

```
int ArchiveBanners(CampaignBidsInfo)
```

Помещает указанные объявления в архив.

Входные параметры

Структура данных `CampaignBidsInfo`:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID рекламной кампании.
`BannerIDS`	`int[]`	Массив `BannerID`.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
UnArchiveBanners
Общие коды ошибок

UnArchiveBanners

Описание

```
int UnArchiveBanners(CampaignBidsInfo)
```

Извлекает указанные объявления из архива.

Входные параметры

Структура данных `CampaignBidsInfo`:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
`CampaignID`	`int`	ID рекламной кампании.

Поле	Тип	Значение
BannerIDS	int[]	Массив BannerID.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
ArchiveBanners

DeleteBanners

Описание

```
int DeleteBanners(CampaignBidsInfo)
```

Удаляет указанные объявления.

Примечание:
Объявления, которые уже были в показе и по которым была собрана статистика, удалить нельзя. Такие объявления можно только поместить в архив.

Входные параметры

Структура данных CampaignBidsInfo:

CampaignBidsInfo

Поле	Тип	Значение
Обязательные		
CampaignID	int	ID рекламной кампании.
BannerIDS	int[]	Массив BannerID.

Возвращает

В случае успешной отработки возвращает 1.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
DeleteCampaign
ArchiveBanners
Общие коды ошибок

Работа с клиентами

GetClientInfo

Описание

```
ClientInfo GetClientInfo(string[])
```

Возвращает регистрационную информацию об указанных клиентах Яндекс.Директа.

Входные параметры

Поле	Тип	Значение
Необязательные		
Login	string[]	Массив логинов клиентов Яндекс.Директа. Если не задан, то считается, что запрошен список кампаний для текущего клиента, на которого выписан используемый сертификат службы Яндекс.Директ.

Возвращает

Массив структур данных ClientInfo:

```
[
    {ClientInfo},
      ...
    {ClientInfo}
]
```

ClientInfo

Поле	Тип	Значение
Login	string	Логин клиента Яндекс.Директа.
FIO	string	ФИО клиента.
Email	string	Адрес электронной почты клиента.
DateCreate	date	Дата создания логина в формате YYYY-MM-DD.
Phone	string	Телефон клиента, указанный при регистрации.
Role	string	Роль клиента. Возможные значения: • Client; • Agency.
Discount[v3]	float	Скидка клиента. Рассчитывается в соответствии со скидочной политикой Яндекса.
StatusArch	string	Помещен ли данный клиент в архив: • Yes — помещен в архив; • No — не помещен в архив. Возможность архивирования предоставляется только для субклиентов Яндекс.Директа.

Поле	Тип	Значение
SendWarn	string	Присылать предупреждения о позициях и отключениях объявлений: • Yes — присылать; • No — не присылать.
SendNews	string	Присылать новости сервиса Яндекс.Директ: • Yes — присылать; • No — не присылать.
SendAccNews	string	Присылать уведомления о состоянии кампаний с тарифом «Беззаботный»: • Yes — присылать; • No — не присылать.
NonResident	string	Является ли клиент нерезидентом РФ: • Yes — нерезидент; • No — резидент.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetClientsList

GetClientsList

Описание

```
array GetClientsList()
```

Возвращает список клиентов агентства.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `ClientInfo`:

```
[
    {ClientInfo},
      ...
    {ClientInfo}
]
```

См. описание структуры `ClientInfo` в методе GetClientInfo.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetClientInfo

GetClientsUnits

Описание

```
array GetClientsUnits(string[])
```

Возвращает количество баллов («юнитов»), оставшихся на счете указанных пользователей.

Баллы расходуются во время запросов, на выполнение которых существуют специальные ограничения. Подробнее читайте в разделе Ограничения на использование API.

Входные параметры

Поле	Тип	Значение
Необязательные		
Login	string[]	Массив логинов клиентов Яндекс.Директа.

Возвращает

Массив структур данных `ClientsUnitInfo`:

```
[
    {ClientsUnitInfo},
      ...
    {ClientsUnitInfo}
]
```

ClientsUnitInfo

Поле	Тип	Значение
Login	string	Логин клиента Яндекс.Директа.
UnitsRest	int	Текущее количество баллов на счете клиента.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
CreateOrUpdateCampaign
CreateOrUpdateBanners
Ограничения на использование API

GetSubClients

Описание

```
array GetSubClients(ShortClientInfo)
```

Возвращает список пользователей, которые находятся на первом уровне в иерархии подчиненных пользователей относительно указанного логина. Пользователь, выполняющий запрос, должен обладать правами на управление логином, который указан во входных параметрах.

Например, для клиента (главного представителя) метод возвращает список всех представителей, которые ему подчиняются.

Для агентства (главного представителя) метод возвращает список следующих пользователей:

- неограниченных представителей агентства;
- ограниченных представителей агентства;
- субклиентов (главных представителей) с правом редактирования агентских кампаний;
- субклиентов (главных представителей) без права редактирования агентских кампаний.

Входные параметры

Структура данных `ShortClientInfo`:

ShortClientInfo

Поле	Тип	Значение
Необязательные		
`Login`	`string`	Логин клиента Яндекс.Директа. Если не задан, запрашивается список подчиненных клиентов для текущего пользователя.
`FIO`	`string`	ФИО клиента. Во входных параметрах не используется.
`Role`	`string`	Роль клиента. Во входных параметрах не используется.

Пример структуры на входе

```
{
    Login => "my-client-1"
}
```

Возвращает

Массив структур данных следующего формата:

```
[
    {ShortClientInfo},
      ...
    {ShortClientInfo}
]
```

Если текущий пользователь не обладает правами на управление логином, который указан во входных параметрах, метод возвращает ошибку в стандартном формате `SOAP::Fault`.

Возможные значения поля `Role`:

Role

Значение	Описание
Для клиентов	
`ChiefRepClient`	Главный представитель клиента, имеет кампании.
`UnlimitedRepClient`	Представитель клиента: имеет кампании, подчиняется главному представителю (`ChiefRepClient`).
Для агентств	
`ChiefRepAgency`	Главный представитель агентства.
`UnlimitedRepAgency`	Неограниченный представитель агентства: имеет доступ ко всем клиентам, подчиняется главному представителю агентства (`ChiefRepAgency`).
`LimitedRepAgency`	Ограниченный представитель агентства: имеет доступ к части клиентов, подчиняется главному представителю агентства (`ChiefRepAgency`).
Для субклиентов	
`ChiefRepSubClient`	Клиент агентства без права редактировать свои агентские кампании, имеет кампании.
`UnlimitedRepSubClient`	Представитель субклиента: имеет кампании, подчиняется субклиенту без права редактировать свои агентские кампании (`ChiefRepSubClient`).

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetClientInfo
CreateNewSubclient

CreateNewSubclient

Описание

```
CreateNewSubclientResponse CreateNewSubclient(CreateNewSubclientRequest)
```

Создает для агентства нового субклиента.

Примечание:
Метод доступен только после специального разрешения менеджера. Чтобы получить разрешение, напишите запрос по адресу api@direct.yandex.ru.

Внимание!
Разрешается создавать не более 100 субклиентов для одного агентства в сутки.

Подробнее о технических ограничениях читайте в разделе Ограничения на использование API.

Входные параметры

Структура данных `CreateNewSubclientRequest`:

`CreateNewSubclientRequest`

Поле	Тип	Значение
Обязательные		
`Login`	`string`	Логин субклиента.
`Name`	`string`	Имя субклиента.
`Surname`	`string`	Фамилия субклиента.
Необязательные		
`AgencyLogin`	`string`	Логин главного представителя агентства. Значение указывается в том случае, если запрос на создание субклиента выполняется от имени менеджера. Если запрос выполняется от имени представителя агентства, по умолчанию используется логин текущего пользователя.

Примечание:
Невозможно создать субклиента с логином, который уже используется на Яндексе другим пользователем.

Если логин уже зарегистрирован, метод возвращает ошибку с кодом 252 (Login is occupied).

Примечание:
Для параметров `Name` и `Surname` выполняются следующие проверки:

- Длина значения не должна превышать 20 символов.
- Значение не должно содержать недопустимые символы: `&`, =, <, >.
- В качестве значения не может быть передана пустая строка.

Если имя и фамилия субклиента не проходят перечисленные проверки, метод возвращает ошибку с кодом 71 (Bad parameters).

Возвращает

Структуру данных `CreateNewSubclientResponse`:

`CreateNewSubclientResponse`

Поле	Тип	Значение
`Login`	`string`	Логин субклиента.
`Password`	`string`	Пароль субклиента для доступа к учетной записи в Яндекс.Паспорте. Генерируется случайным образом.
`FIO`	`string`	Имя субклиента.
`Email`	`string`	Адрес электронной почты субклиента.
`ClientID`	`int`	Идентификатор субклиента.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetSubClients

Яндекс.Каталог и регионы

GetRegions

Описание

```
array GetRegions()
```

Возвращает структуру данных с информацией о всех регионах показов сервиса Яндекс.Директ. Список регионов может потребоваться для анализа статистики по рекламной кампании, при использовании сервиса прогноза бюджета (метод CreateNewForecast), при создании и редактировании объявлений.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `RegionInfo`:

```
[
    {RegionInfo},
      ...
    {RegionInfo}
]
```

RegionInfo

Поле	Тип	Значение
`RegionID`	`int`	Идентификатор региона показов. Например, `'1'`.
`RegionName`	`string`	Название региона. Например, `'Москва и область'`.
`ParentID`	`int`	Идентификатор родительского региона. Например, `'3'`.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetRubrics

GetRubrics

Описание

```
array GetRubrics()
```

Возвращает структуру данных с информацией о всех рубриках сервиса **Яндекс.Каталог**. Список рубрик может потребоваться для анализа статистики по рекламной кампании, при использовании сервиса прогноза бюджета (метод CreateNewForecast), при создании и редактировании кампаний и объявлений.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `RubricInfo`:

```
[
    {RubricInfo},
      ...
    {RubricInfo}
]
```

RubricInfo

Поле	Тип	Значение
RubricID	int	Идентификатор рубрики.
RubricName	string	Название рубрики.
RubricFullName	string	Полное название рубрики.
Url	string	URL рубрики.
ParentID	int	Идентификатор родительского раздела каталога.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetRegions

Прочие методы

GetVersion

Описание

```
int GetVersion()
```

Возвращает текущую версию API Яндекс.Директа.

Входные параметры

отсутствуют

Возвращает

Числовое значение текущей версии API.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetAvailableVersions
PingAPI
Общие коды ошибок

GetAvailableVersions

Описание

```
array GetAvailableVersions()
```

Возвращает список доступных версий API Яндекс.Директа.

Предлагается использовать этот метод для получения уведомлений о появлении свежих версий API, а также об истечении срока жизни предыдущих доступных версий.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `VersionDesc`:

```
[
    {VersionDesc},
      ...
    {VersionDesc}
]
```

VersionDesc

Поле	Тип	Значение
VersionNumber	float	Номер версии API Яндекс.Директа.
BorderDate	date	Время жизни указанной версии. Не определяется для самой свежей версии API.
LinkWSDL	string	Ссылка на WSDL-файл для указанной версии API.

Пример результата

```
'result' => [
                {
                    'BorderDate' => '2010-06-10',
                    'VersionNumber' => '1',
                    'LinkWSDL' => 'https://soap.direct.yandex.ru/wsdl/v1/'
                },
                {
                    'BorderDate' => undef,
                    'VersionNumber' => '2'
                    'LinkWSDL' => 'https://soap.direct.yandex.ru/wsdl/v2/'
                }
            ]
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
PingAPI
Общие коды ошибок

GetTimeZones

Описание

```
array GetTimeZones()
```

Возвращает список временных зон для установки и отображения временного таргетинга в настройках рекламной кампании.

Входные параметры

отсутствуют

Возвращает

Массив структур данных `TimeZoneInfo`:

```
[
    {TimeZoneInfo},
      ...
    {TimeZoneInfo}
]
```

TimeZoneInfo

Поле	Тип	Значение
`TimeZone`	`string`	Обозначение временной зоны.
`GMTOffset`	`int`	Разница с GMT в секундах.
`Name`	`string`	Название временной зоны.

Пример результата

```
'result' => [
             {
                 'GMTOffset' => '10800',
                 'TimeZone' => 'Europe/Kaliningrad',
                 'Name' => 'Калининград (MSK -01:00)'
             },
             {
                 'GMTOffset' => '14400',
                 'TimeZone' => 'Europe/Moscow'
                 'Name' => 'Москва'
             },
             {
                 'GMTOffset' => '21600',
                 'TimeZone' => 'Asia/Yekaterinburg'
                 'Name' => 'Екатеринбург (MSK +02:00)'
             },
             ...
             {
                 'GMTOffset' => '10800',
                 'TimeZone' => 'Europe/Tallinn'
                 'Name' => 'Эстония (GMT +03:00)'
             },
             {
                 'GMTOffset' => '32400',
                 'TimeZone' => 'Asia/Seoul'
                 'Name' => 'Южная Корея (GMT +09:00)'
             },
             {
                 'GMTOffset' => '-18000',
                 'TimeZone' => 'America/Jamaica'
                 'Name' => 'Ямайка (GMT -05:00)'
             },
             {
                 'GMTOffset' => '32400',
                 'TimeZone' => 'Asia/Tokyo'
                 'Name' => 'Япония (GMT +09:00)'
             }
            ]
```

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
Общие коды ошибок

PingAPI

Описание

```
int PingAPI()
```

Проверяет успешность авторизации.

Входные параметры

отсутствуют

Возвращает

1 — если авторизация прошла успешно.

Примеры

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

См. также
GetAvailableVersions
Проверка подключения. Простой скрипт на PHP (NuSOAP)
Общие коды ошибок

Коды ошибок

В разделе приведены коды и описание ошибок в ответных сообщениях API Яндекс.Директа.

Примечание:
Краткое описание ошибок приведено на русском языке. В таком виде оно возвращается, если в запросе указан параметр locale со значением «ru».

Код	Краткое описание	Пояснение
Коды ошибок при работе со статистикой		
1	Неверный CampaignID	Неверный идентификатор кампании CampaignID.
2	Нет статистики для данной кампании	Статистика по указанной кампании отсутствует.
3	Неверная дата	Неверный формат даты начала или даты окончания периода формирования статистики — параметр StartDate или EndDate соответственно.
5	Неверный временной интервал	Неверный период формирования статистики: дата начала находится после даты окончания.
8	Указанные лимиты неверны	Неверно задано максимальное количество строк в отчете — параметр Limit. Значение параметра должно быть числом больше нуля.
9	Поле должно быть массивом	Неверный тип одного из параметров. Ожидается массив, но задан скаляр или структура данных.
10	Неверное условие фильтрации баннера	Неверный фильтр по объявлениям — параметр Banner.
11	Неверное условие фильтрации геотаргетинга	Неверный фильтр по регионам — параметр Geo.
12	Неверное условие фильтрации площадки	Неверный фильтр по площадкам — параметр PageName.
13	Неверное условие фильтрации типа площадки	Неверный тип сайта в фильтре по площадкам — параметр Page.
14	Неверное условие фильтрации параметров фразы	Неверный фильтр по фразам кампании — параметр Phrase.
15	Неверный тип площадки	Неверный фильтр по типу площадок — параметр PageType. Например, значение определено нестрого.
16	Неверно указаны параметры группировки по дате	Неверный фильтр по датам — параметр Date.
17	Неверное название поля	Неверно задано значение параметра GroupByColumns.
18	Неверно указан отступ	Неверный отступ — параметр Offset.
20	Неверный тип отчета	Неверный тип файла с отчетом — параметр TypeResultReport.
21	Неверно указан способ компрессии отчета	Неверный тип сжатия отчета — параметр CompressReport.
22	Неверный ReportID	Неверный идентификатор отчета `ReportID` при вызове методов GetReportList и DeleteReport.
23	Неверный BannerID	Неверный идентификатор объявления `BannerID` при вызове методов CreateNewReport, GetBannerPhrase, GetBanners, UpdatePrices.
24	Указанный отчет не существует	Указанный отчет не создан.
31	Достигнут лимит отчетов в очереди	Очередь отчетов заполнена.
32	Неверный PositionType	Неверный фильтр по позициям — параметр PositionType.

Код	Краткое описание	Пояснение
33	Неверное условие фильтрации по целям	Неверный фильтр по целям — параметр StatGoals.
Общие коды ошибок		
25	Поле может содержать значения: Yes и No	Входной параметр может принимать только значения «Yes» и «No».
30	Массив не должен быть пустым	Входной параметр не может являться пустым массивом и должен содержать значения.
53	Ошибка авторизации	Указано несуществующее имя пользователя. Проверьте, не истек ли срок действия сертификата или авторизационного токена.
54	Недостаточно прав	Недостаточно прав для выполнения запрошенной операции.
55	Метод не существует	Вызывается несуществующий метод.
56	Превышен лимит запросов	Достигнуто техническое ограничение на вызов метода.
241	Превышен допустимый размер массива	Превышена допустимая длина массива элементов.
500	Внутренняя ошибка сервера	Внутренняя ошибка сервера. Попробуйте вызвать метод через некоторое время. При повторении ошибки обратитесь в службу технической поддержки Яндекс.Директа.
501	Неверный запрос	Неверный запрос к API Яндекс.Директа.
503	Сервис временно недоступен	Сервер API Яндекс.Директа временно недоступен.
506	Превышен лимит одновременных запросов	Превышено ограничение на количество одновременных соединений с сервером.
508	Запрашиваемая версия API недоступна	Версия API устарела и более не поддерживается. Используйте последнюю версию API Яндекс.Директа.
509	Метод недоступен в данной версии API	Вызываемый метод не доступен в используемой версии API Яндекс.Директа.
510	Доступ заблокирован	Доступ к API запрещен.
511	Неизвестный язык	Указана неизвестная локаль.
512	Неверный метод запроса	Запрос выполнен методом, отличным от HTTP POST. Запросы к Яндекс.Директу должны выполняться методом HTTP POST.
Коды ошибок при работе с прогнозом бюджета и ключевыми словами		
71	Параметры запроса указаны неверно	Неверно задан один из параметров.
72	Неверный ForecastID	Неверный номер заявки на составление прогноза — параметр ForecastID.
73	Указанный отчет с прогнозом не существует	Прогноза бюджета с указанным идентификатором не создан.
74	Отчет с прогнозом в процессе подготовки	Формирование прогноза бюджета не закончено.
75	Неверный CategoryID	Неверно заданы рубрики Яндекс.Каталога — параметр Categories.
76	Неверно указана фраза	Неверно задана фраза для расчета прогноза бюджета — параметр Phrases.
77	Неверный GeoID	Неверно заданы регионы показа объявлений — параметр GeoID.
91	Указанный отчет wordstat не существует	Неверный идентификатор отчета — параметр ReportID. Проверьте, что отчет с таким идентификатором присутствует в списке заказанных отчетов.
92	Отчет wordstat в процессе подготовки	Формирование отчета по ключевым словам не закончено.
93	Неверный WordstatReportID	Неверный формат параметра ReportID, например значение не задано или содержит символы.
Коды ошибок при работе с кампаниями и объявлениями		
26	Значение должно быть от 0 до 100 и кратным 10	Неверно задано значение процента. Возможные значения: 0, 10, 20, ..., 100.
27	Неверный BannerID	Неверно задан параметр `BannerID`.
28	Неверный PhraseID	Неверно задан параметр `PhraseID`.

Код	Краткое описание	Пояснение
29	Неверно указан тип кампании	Запрещено редактировать указанную кампанию.
111	Неверно указаны параметры кампании	Неправильно описана структура CampaignInfo в запросе.
113	Неверное условие фильтрации кампании	Неверные значения фильтров в параметре Filter.
151	Неверно указаны параметры баннера	Неправильно описана структура BannerInfo в запросе.
152	Недостаточно баллов	Недостаточно баллов для выполнения операции.
153	Превышен лимит баннеров в запросе	Количество объявлений в запросе превышает допустимое значение.
154	Структура PointOnMap заполнена неверно	Неверные координаты местоположения рекламодателя при вызове метода CreateOrUpdateBanners. Местоположение задается в параметре `ContactInfo -> PointOnMap`, который содержит структуру MapPoint .
192	Неверно указаны минус-слова	Неверно указаны минус-слова в параметре MinusKeywords. Минус-слова должны состоять из допустимых символов, не превышать максимальную длину и не являться словосочетаниями.
Коды ошибок при работе с обновлением ставок рекламной кампании		
191	Запрещено менять ContextPrice для активных на поиске фраз	Запрещено менять ставки за клик в Рекламной сети Яндекса для фраз, которые *не* были отключены за низкий CTR на поисковых площадках.
242	Неверно указана цена	Неверно задана цена для фразы (UpdatePrices).
243	Неверно указано значение поля Autobroker	Неверно задан атрибут `AutoBroker` (CreateOrUpdateBanners, GetBannerPhrases, UpdatePrices).
Коды ошибок при работе с клиентами		
251	Неверный логин пользователя	Неверно задан логин собственника кампании (GetCampaignsList).
252	Логин занят	Логин используется на Яндексе другим пользователем.
253	Ошибка при создании логина	Не удалось создать учетную запись клиента методом CreateNewSubclient. Указанный логин занят.
254	Ошибка при создании клиента	Не удалось создать учетную запись клиента методом CreateNewSubclient. Причина ошибки не уточняется.
255	Невозможно связать клиента с агентством	Не удалось создать учетную запись клиента методом CreateNewSubclient.

Примеры

Ниже приведены ссылки на примеры вызова методов API.

Статистика и аналитика	Общие данные
Статистика кампании	**Работа с клиентами**
CreateNewReport	GetClientInfo
GetReportList	GetClientsList
DeleteReport	GetClientsUnits
GetStatGoals	GetSubClients
Подбор ключевых слов	CreateNewSubclient
CreateNewWordstatReport	**Яндекс.Каталог и регионы**
GetWordstatReportList	GetRegions
GetWordstatReport	GetRubrics
GetKeywordsSuggestion	**Прочие методы**
Прогноз бюджета	GetVersion
CreateNewForecast	GetAvailableVersions
GetForecastList	GetTimeZones
GetForecast	PingAPI
DeleteForecastReport	

Кампании и объявления	Управление состояниями
Информация о кампаниях	**Состояние кампаний**
GetCampaignParams	StopCampaign
GetCampaignsList	ResumeCampaign
GetCampaignsListFilter	ArchiveCampaign
GetBalance	UnArchiveCampaign
GetBanners	DeleteCampaign
GetBannerPhrases	**Состояние объявлений**
GetBannerPhrasesFilter	ModerateBanners
Редактирование кампаний и объявлений	StopBanners
CreateOrUpdateCampaign	ResumeBanners
CreateOrUpdateBanners	ArchiveBanners
SetAutoPrice	UnArchiveBanners
UpdatePrices	DeleteBanners

Примеры приложений

Создание рекламной кампании на Perl (SOAP::Lite)
Создание рекламной кампании на PHP (NuSOAP)
Создание рекламной кампании на PHP (SOAPClient)
Создание рекламной кампании на Python (SOAPpy)
Проверка подключения. Простой скрипт на PHP (NuSOAP)
Пример работы с JSON на Perl
Пример работы с JSON на PHP

Работа с отчетами

Получение файла отчета (PHP, cURL)
Пример файла отчета по статистике рекламной кампании

Вызов методов

ArchiveBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 123121,
          'BannerIDS': [54654, 46546, 878545]
         }
result = proxy.ArchiveBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->call('ArchiveBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->ArchiveBanners($params);
```

Пример на Perl

```
$params = {
   CampaignID => 123121,
   BannerIDS => [54654, 46546, 878545]
};
$result = $client->ArchiveBanners($params);
```

См. также
ArchiveBanners

ArchiveCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 54654564,
}
result = proxy.ArchiveCampaign(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->call('ArchiveCampaign', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->ArchiveCampaign($params);
```

Пример на Perl

```
$params = {
   CampaignID => 54654564,
};
$result = $client->ArchiveCampaign($params);
```

См. также
ArchiveCampaign

CreateNewForecast

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'Phrases': [u'nokia', u'phone', u'cars'],
          'GeoID': [0],
          'Categories': []
         }
result = proxy.CreateNewForecast(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
                'Phrases' => array(
                                    'nokia',
                                    'doors',
                                    'windows'
                                ),
                'GeoID' => array(0),
                'Categories' => array(10882, 2898)
);

$result = $client->call('CreateNewForecast', array('params' =>
$params));
```

Пример на PHP SoapClient

```
PHP SoapClient

$params = array(
                'Phrases' => array(
                                    'nokia',
                                    'doors',
                                    'windows'
                                ),
                'GeoID' => array(0),
                'Categories' => array(10882, 2898)
);

$result = $client->CreateNewForecast($params);
```

Пример на Perl

```
$params = {
           'Phrases' => ['nokia', 'doors', 'windows'],
           'GeoID' => [0],
           'Categories' => [10882, 2898],
};

$result = $client->CreateNewForecast($params);
```

См. также
CreateNewForecast

CreateNewReport

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 170198,
          'StartDate': date(2005, 01, 01),
          'EndDate': date(2007, 11, 01),
          'GroupByDate': 'day',
          'GroupByColumns': ['clPhrase']
         }

result = proxy.CreateNewReport(params = params)
```

Пример на PHP NuSOAP

```
$createNewReportData = array(
                             'CampaignID'     => "1342250",
                             'StartDate'      => "2005-01-01",
                             'EndDate'        => "2007-11-01",
                             'GroupByDate'    => "day",
                             'GroupByColumns' =>
array("clPhrase"),
                            'Filter'         => array("PageType"
=> 'search')
                       );

$result = $client->call('CreateNewReport', array('params' =>
$createNewReportData));
$result = $client->call('GetReportList', array('params' => 594));
```

Пример на PHP SoapClient

```
$createNewReportData = array(
                             'CampaignID'     => "1342250",
                             'StartDate'      => "2005-01-01",
                             'EndDate'        => "2007-11-01",
                             'GroupByDate'    => "day",
                             'GroupByColumns' =>
array("clPhrase"),
                            'Filter'         => array("PageType"
=> 'search')
                       );

$result = $client->CreateNewReport($createNewReportData);
```

Пример на Perl

```
$createNewReportData = {
                        CampaignID     => "1342250",
                        StartDate      => "2005-01-01",
                        EndDate        => "2007-11-01",
                        GroupByDate    => "day",
                        GroupByColumns => ["clPhrase"],
                        Filter         => {
                                            PageType => 'search'
                                           },
                       };

$result = $client->CreateNewReport($createNewReportData);
```

См. также
CreateNewReport

CreateNewSubclient

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'Login': 'john_smith',
    'Name': 'John',
    'Surname': 'Smith'
}
result = proxy.CreateNewSubclient(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    'Login' => 'john_smith',
    'Name' => 'John',
    'Surname' => 'Smith'
);
$result = $client->call('CreateNewSubclient', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    'Login' => 'john_smith',
    'Name' => 'John',
    'Surname' => 'Smith'
);
$result = $client->CreateNewSubclient($params);
```

Пример на Perl

```
$params = {
    'Login' => 'john_smith',
    'Name' => 'John',
    'Surname' => 'Smith'
};
my $request = $client->CreateNewSubclient($params);
```

См. также
CreateNewSubclient

CreateNewWordstatReport

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'Phrases': ['фраза1', 'фраза2'],
          'GeoID': [1, 2, 3],
         }

result = proxy.CreateNewWordstatReport(params = params)
```

Пример на PHP NuSOAP

```
$createNewReportData = array(
                              'Phrases' => array("фраза1",
"фраза2"),
                              'GeoID'   => array(1, 2, 3)
                        );

$result = $client->call('CreateNewWordstatReport',
array('params' => $createNewReportData));
```

Пример на PHP SoapClient

```
$createNewReportData = array(
                              'Phrases' => array("фраза1",
"фраза2"),
                              'GeoID'   => array(1, 2, 3)
                        );

$result = $client->CreateNewWordstatReport($createNewReportData);
```

Пример на Perl

```
$createNewReportData = {
                        Phrases => ['фраза1', 'фраза2'],
                        GeoID   => [1, 2, 3]
                        };

$result = $client->CreateNewWordstatReport($createNewReportData);
```

См. также
CreateNewWordstatReport

CreateOrUpdateBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = [
    {
        'CampaignID': 1327837
      , 'BannerID': 2571700
      , 'Title': 'Python banner title'
      , 'Text': u'Python banner body'
      , 'Href': 'http://market-click2.ya.ru/?q=qu'
      , 'Geo': '2,183,3,5'
      , 'Phrases': [
            {
                'IsRubric': No
              , 'Phrase': u'phrase 1'
              , 'PhraseID': 0
              , 'AutoBroker': No
              , 'Price': 0.11
              , 'ContextPrice': 0.9
              , 'AutobudgetPriority': 'Medium'
            },
            {
                'IsRubric': No
              , 'Phrase': u'phrase 2'
              , 'PhraseID': 0
              , 'AutoBroker': Yes
              , 'Price': 0.11
              , 'ContextPrice': 0.9
              , 'AutobudgetPriority': 'Medium'
            },
        ]
      , 'ContactInfo': {
            'Apart': 2
          , 'Build': 2
          , 'City': u'Иваново'
          , 'CityCode': 493
          , 'ContactEmail': 'ivanl@ivanov.ru'
          , 'ContactPerson': u'Иван Иванов'
          , 'Country': u'Россия'
          , 'CountryCode': '+7'
          , 'ExtraMessage': u'подробнее о товаре/услуге'
          , 'House': '2'
          , 'IMClient': 'jabber'
          , 'IMLogin': 'ivan@ivanov.ru'
          , 'CompanyName': u'ivanov-agency'
          , 'Phone': '2128506'
          , 'PhoneExt': '223'
          , 'Street': u'Ивановская'
          , 'WorkTime':'0;4;10;00;18;00;5;6;13;00;16;00'
        }

      , 'Sitelinks': {
            {
                'Title': 'Дополнительная ссылка 1'
              , 'Href': 'http://market-click2.ya.ru/?
q=qu&action=1'
            }
          , {
                'Title' => 'Дополнительная ссылка 2'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
            }
```

```
          , {
                'Title' => 'Дополнительная ссылка 3'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
            }
        }
    }
]

result = proxy.CreateOrUpdateBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    array(
        'CampaignID' => 5454
      , 'BannerID' => 0
      , 'Title' => 'Заголовок объявления'
      , 'Text' => 'Текст объявления'
      , 'Href' => 'http://market-click2.ya.ru/?q=qu'
      , 'Geo' => '2,183,3,5'
      , 'Phrases' => array(
            array(
                'IsRubric' => No
              , 'Phrase' => 'Текст фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => No
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            ),
            array(
                'IsRubric' => No
              , 'Phrase' => 'Текст еще одной фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => Yes
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            ),
        )

      , 'ContactInfo' => array(
            'Apart' => 2
          , 'Build' => 2
          , 'City' => 'Москва'
          , 'CityCode' => 495
          , 'ContactEmail' => 'email@ya.ru'
          , 'ContactPerson' => 'контактное лицо'
          , 'Country' => 'Россия'
          , 'CountryCode' => '+7'
          , 'ExtraMessage' => 'подробнее о товаре/услуге'
          , 'House' => '2'
          , 'IMClient' => 'jabber'
          , 'IMLogin' => 'email@ya.ru'
          , 'CompanyName' => 'ФИО2'
          , 'Phone' => '2128506'
          , 'PhoneExt' => '223'
          , 'Street' => 'улица'
          , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
        )

      , 'Sitelinks' => array(
            array(
                    'Title' => 'Дополнительная ссылка 1'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
                 )
```

```
          , array(
                    'Title' => 'Дополнительная ссылка 2'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
                 )
          , array(
                    'Title' => 'Дополнительная ссылка 3'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
                 )
        )
    )
);

$result = $client->call('CreateOrUpdateBanners', array('params'
=> $params));
```

Пример на PHP SoapClient

```
$params = array(
    array(
        'CampaignID' => 5454
      , 'BannerID' => 0
      , 'Title' => 'Заголовок объявления'
      , 'Text' => 'Текст объявления'
      , 'Href' => 'http://market-click2.ya.ru/?q=qu'
      , 'Geo' => '2,183,3,5'
      , 'Phrases' => array(
            array(
                'IsRubric' => No
              , 'Phrase' => 'Текст фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => No
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            ),
            array(
                'IsRubric' => No
              , 'Phrase' => 'Текст еще одной фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => Yes
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            ),
        )

      , 'ContactInfo' => array(
            'Apart' => 2
          , 'Build' => 2
          , 'City' => 'Москва'
          , 'CityCode' => 495
          , 'ContactEmail' => 'email@ya.ru'
          , 'ContactPerson' => 'контактное лицо'
          , 'Country' => 'Россия'
          , 'CountryCode' => '+7'
          , 'ExtraMessage' => 'подробнее о товаре/услуге'
          , 'House' => '2'
          , 'IMClient' => 'jabber'
          , 'IMLogin' => 'email@ya.ru'
          , 'CompanyName' => 'ФИО2'
          , 'Phone' => '23455'
          , 'PhoneExt' => '223'
          , 'Street' => 'улица'
          , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
        )
```

```
      , 'Sitelinks' => array(
            array(
                    'Title' => 'Дополнительная ссылка 1'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
                )
          , array(
                    'Title' => 'Дополнительная ссылка 2'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
                )
          , array(
                    'Title' => 'Дополнительная ссылка 3'
                  , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
                )
        )
    )
);

$result = $client->CreateOrUpdateBanners($params);
```

Пример на Perl

```
my $params = [
    {
        'CampaignID' => 5454
      , 'BannerID' => 0
      , 'Title' => 'Заголовок объявления'
      , 'Text' => 'Текст объявления'
      , 'Href' => 'http://market-click2.ya.ru/?q=qu'
      , 'Geo' => '2,183,3,5'
      , 'Phrases' => [
            {
                'IsRubric' => No
              , 'Phrase' => 'Текст фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => No
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            },
            {
                'IsRubric' => No
              , 'Phrase' => 'Текст еще одной фразы'
              , 'PhraseID' => 0
              , 'AutoBroker' => Yes
              , 'Price' => 0.11
              , 'ContextPrice' => 0.9
              , 'AutobudgetPriority' => 'Medium'
            },
        ]

      , 'ContactInfo' => {
            'Apart' => 2
          , 'Build' => 2
          , 'City' => 'Москва'
          , 'CityCode' => 495
          , 'ContactEmail' => 'email@ya.ru'
          , 'ContactPerson' => 'контактное лицо'
          , 'Country' => 'Россия'
          , 'CountryCode' => '+7'
          , 'ExtraMessage' => 'подробнее о товаре/услуге'
          , 'House' => '2'
          , 'IMClient' => 'jabber'
          , 'IMLogin' => 'email@ya.ru'
          , 'CompanyName' => 'ФИО2'
          , 'Phone' => '23455'
```

```
          , 'PhoneExt' => '223'
          , 'Street' => 'улица'
          , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
        }

      , 'Sitelinks' => [
            {
                'Title' => 'Дополнительная ссылка 1'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
            }
          , {
                'Title' => 'Дополнительная ссылка 2'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
            }
          , {
                'Title' => 'Дополнительная ссылка 3'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
            }
        ]
    }
];

my $request = $client->CreateOrUpdateBanners($params);
```

См. также
CreateOrUpdateBanners

CreateOrUpdateCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 0,
    'Login' : u'user_login',
    'StatusContextStop': 'No',
    'Name': u'Новая кампания',
    'FIO': u'Иванов Иван Иванович',
    'StartDate': '2009-01-15',

    'AutoOptimization': 'No',

    'DisabledDomains': 'domain1.ru,domain2.ru',
    'DisabledIps': '64.234.23.21',

    'StatusBehavior': 'Yes',

    'TimeTarget':
    {
        'TimeZone': 'Europe/Moscow',
        'DaysHours': [
            {
                'Hours': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12],
```

```
                'Days': [1, 2, 3, 4]
            }
        ],
    },
    'StatusMetricaControl': 'Yes',
    'ConsiderTimeTarget': 'No',
    'StatusOpenStat': 'No',
    'ContextPricePercent': 10,

    'ContextLimit': 'Limited',
    'ContextLimitSum': 70,

    'Autobudget': 'Yes',
    'AutobudgetSum': 76,
    'AutobudgetMaxPrice': 17,

    'EmailNotification': {
        'Email': 'user@ya.ru',
        'MoneyWarningValue': 1,
        'WarnPlaceInterval': 15,
        'SendWarn': 'Yes',
        'SendAccNews': 'Yes'
    },

    'SmsNotification': {
        'MetricaSms': 'Yes',
        'ModerateResultSms': 'Yes',
        'MoneyInSms': 'Yes',
        'MoneyOutSms': 'Yes',
        'SmsTimeFrom': '09:15',
        'SmsTimeTo': '22:15'
    },

    'MinusKeywords': {'keyword1', 'keyword2', 'keyword3'},

    'AddRelevantPhrases': 'Yes',
    'RelevantPhrasesBudgetLimit': 50
}

result = proxy.CreateOrUpdateCampaign(params = params)
```

Пример на PHP NuSOAP

```
params = array(
    'CampaignID' => 0,
    'Login' => 'user_login',
    'StatusContextStop' => 'No',
    'Name' => 'Новая кампания',
    'FIO' => 'Иванов Иван Иванович',
    'StartDate' => '2009-01-15',

    'AutoOptimization' => 'No',

    'DisabledDomains' => 'domain1.ru,domain2.ru',
    'DisabledIps' => '64.234.23.21',

    'StatusBehavior' => 'Yes',

    'TimeTarget' => array(
        'TimeZone' => 'Europe/Moscow',
        'DaysHours' => array(
            array(
                'Hours' => array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12),
                'Days' => array(1, 2, 3, 4)
            )
        ),
    ),
```

```
    'StatusMetricaControl' => 'Yes',
    'ConsiderTimeTarget' => 'No',
    'StatusOpenStat' => 'No',
    'ContextPricePercent' => 10,

    'ContextLimit' => 'Limited',
    'ContextLimitSum' => 70,

    'Autobudget' => 'Yes',
    'AutobudgetSum' => 76,
    'AutobudgetMaxPrice' => 17,

    'EmailNotification' => array(
        'Email' => 'user@ya.ru',
        'MoneyWarningValue' => 1,
        'WarnPlaceInterval' => 15,
        'SendWarn' => 'Yes',
        'SendAccNews' => 'Yes'
    ),

    'SmsNotification' => array(
        'MetricaSms' => 'Yes',
        'ModerateResultSms' => 'Yes',
        'MoneyInSms' => 'Yes',
        'MoneyOutSms' => 'Yes',
        'SmsTimeFrom' => '09:15',
        'SmsTimeTo' => '22:15'
    ),

    'MinusKeywords' => array('keyword1', 'keyword2', 'keyword3'),

    'AddRelevantPhrases' => 'Yes',
    'RelevantPhrasesBudgetLimit' => 50
);

$result = $client->call('CreateOrUpdateCampaign', array('params'
=> $params));
```

Пример на PHP SoapClient

```
params = array(
    'CampaignID' => 0,
    'Login' => 'user_login',
    'StatusContextStop' => 'No',
    'Name' => 'Новая кампания',
    'FIO' => 'Иванов Иван Иванович',
    'StartDate' => '2009-01-15',

    'AutoOptimization' => 'No',

    'DisabledDomains' => 'domain1.ru,domain2.ru',
    'DisabledIps' => '64.234.23.21',

    'StatusBehavior' => 'Yes',

    'TimeTarget' => array(
        'TimeZone' => 'Europe/Moscow',
        'DaysHours' => array(
            array(
                'Hours' => array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12),
                'Days' => array(1, 2, 3, 4)
            )
        ),
    ),
    'StatusMetricaControl' => 'Yes',
    'ConsiderTimeTarget' => 'No',
    'StatusOpenStat' => 'No',
```

```
    'ContextPricePercent' => 10,

    'ContextLimit' => 'Limited',
    'ContextLimitSum' => 70,

    'Autobudget' => 'Yes',
    'AutobudgetSum' => 76,
    'AutobudgetMaxPrice' => 17,

    'EmailNotification' => array(
        'Email' => 'user@ya.ru',
        'MoneyWarningValue' => 1,
        'WarnPlaceInterval' => 15,
        'SendWarn' => 'Yes',
        'SendAccNews' => 'Yes'
    ),

    'SmsNotification' => array(
        'MetricaSms' => 'Yes',
        'ModerateResultSms' => 'Yes',
        'MoneyInSms' => 'Yes',
        'MoneyOutSms' => 'Yes',
        'SmsTimeFrom' => '09:15',
        'SmsTimeTo' => '22:15'
    ),

    'MinusKeywords' => array('keyword1', 'keyword2', 'keyword3'),

    'AddRelevantPhrases' => 'Yes',
    'RelevantPhrasesBudgetLimit' => 50
);

$result = $client->CreateOrUpdateCampaign($params);
```

Пример на Perl

```
my $params = {
    'CampaignID' => 0,
    'Login' => 'user_login',
    'StatusContextStop' => 'No',
    'Name' => 'Новая кампания',
    'FIO' => 'Иванов Иван Иванович',
    'StartDate' => '2009-01-15',

    'AutoOptimization' => 'No',

    'DisabledDomains' => 'domain1.ru,domain2.ru',
    'DisabledIps' => '64.234.23.21',

    'StatusBehavior' => 'Yes',

    'TimeTarget' => {
        'TimeZone' => 'Europe/Moscow',
        'DaysHours' => [
            {
                'Hours' => [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12],
                'Days' => [1, 2, 3, 4]
            }
        ],
    },
    'StatusMetricaControl' => 'Yes',
    'ConsiderTimeTarget' => 'No',
    'StatusOpenStat' => 'No',
    'ContextPricePercent' => 10,

    'ContextLimit' => 'Limited',
    'ContextLimitSum' => 70,
```

```
    'Autobudget' => 'Yes',
    'AutobudgetSum' => 76,
    'AutobudgetMaxPrice' => 17,

    'EmailNotification' => {
        'Email' => 'user@ya.ru',
        'MoneyWarningValue' => 1,
        'WarnPlaceInterval' => 15,
        'SendWarn' => 'Yes',
        'SendAccNews' => 'Yes'
    },

    'SmsNotification' => {
        'MetricaSms' => 'Yes',
        'ModerateResultSms' => 'Yes',
        'MoneyInSms' => 'Yes',
        'MoneyOutSms' => 'Yes',
        'SmsTimeFrom' => '09:15',
        'SmsTimeTo' => '22:15'
    },

    'MinusKeywords' => ['keyword1', 'keyword2', 'keyword3'],

    'AddRelevantPhrases' => 'Yes',
    'RelevantPhrasesBudgetLimit' => 50
}

my $request = $client->CreateOrUpdateCampaign($params);
```

См. также
CreateOrUpdateCampaign

DeleteBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          CampaignID: 456465,
          BannerIDS: [445566, 727854, 8795465]
         }
result = proxy.DeleteBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array("CampaignID" => 456465,
                "BannerIDS" => array(445566, 727854, 8795465));
$result = $client->call('DeleteBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array("CampaignID" => 456465,
                "BannerIDS" => array(445566, 727854, 8795465));
$result = $client->DeleteBanners($params);
```

Пример на Perl

```
$params = {
           CampaignID => 456465,
           BannerIDS => [445566, 727854, 8795465]
          };
$result = $client->DeleteBanners($params);
```

См. также
DeleteBanners

DeleteForecastReport

Пример на Python

```
result = proxy.DeleteForecastReport(params = 36)
```

Пример на PHP NuSOAP

```
$result = $client->call('DeleteForecastReport', array('params'
=> 36));
```

Пример на PHP SoapClient

```
$params = '36';
$result = $client->DeleteForecastReport($params);
```

Пример на Perl

```
$params = '36';
$result = $client->DeleteForecastReport($params);
```

DeleteCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 54654564,
}
result = proxy.DeleteCampaign(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->call('DeleteCampaign', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->DeleteCampaign($params);
```

Пример на Perl

```
$params = {
   CampaignID => 54654564,
};
$result = $client->DeleteCampaign($params);
```

См. также
DeleteCampaign

DeleteReport

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.DeleteReport(params = 36)
```

Пример на PHP NuSOAP

```
$result = $client->call('DeleteReport', array('params' => 17));
```

Пример на PHP SoapClient

```
$params = '12';
$result = $client->DeleteReport($params);
```

Пример на Perl

```
$params = '12';
$result = $client->DeleteReport($params);
```

См. также
DeleteReport

GetAvailableVersions

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetAvailableVersions()
```

Пример на PHP NuSOAP

```
$result = $client->call("GetAvailableVersions", array());
```

Пример на PHP SoapClient

```
$result = $client->GetAvailableVersions();
```

Пример на Perl

```
my $result = $client->GetAvailableVersions();
```

См. также
GetAvailableVersions

GetBalance

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetBalance(params = [987469])
```

Пример на PHP NuSOAP

```
$params = array(716240);
$result = $client->call('GetBalance', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(716240);
$result = $client->GetBalance($params);
```

Пример на Perl

```
$params = [716240];
$result = $client->GetBalance($params);
```

См. также
GetBalance

GetBannerPhrases

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = [54654, 46546, 878545]
result = proxy.GetBannerPhrases(params = params)
```

Пример на PHP NuSOAP

```
$params = array(54654, 46546, 878545);
$result = $client->call('GetBannerPhrases', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(54654, 46546, 878545);
$result = $client->GetBannerPhrases($params);
```

Пример на Perl

```
$params = [54654, 46546, 878545];
$result = $client->GetBannerPhrases($params);
```

См. также
GetBannerPhrases

GetBannerPhrasesFilter

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'BannerIDS': [6664636, 6664637, 6664638],
          'FieldsNames': ['IsRubric', 'BannerID',
'StatusPhraseModerate'],
          'ConsiderTimeTarget': 'No'
         }

result = proxy.GetBannerPhrasesFilter(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
              'BannerIDS' => array(6664636, 6664637, 6664638),
              'FieldsNames' => array('IsRubric', 'BannerID',
'StatusPhraseModerate'),
              'ConsiderTimeTarget' => 'No'
          );
$result = $client->call('GetBannerPhrases', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
              'BannerIDS' => array(6664636, 6664637, 6664638),
              'FieldsNames' => array('IsRubric', 'BannerID',
'StatusPhraseModerate'),
              'ConsiderTimeTarget' => 'No'
          );
$result = $client->GetBannerPhrases($params);
```

Пример на Perl

```
$params = {
           'BannerIDS' => [6664636, 6664637, 6664638],
           'FieldsNames' => ['IsRubric', 'BannerID',
'StatusPhraseModerate'],
           'ConsiderTimeTarget' => 'No'
          };
$result = $client->GetBannerPhrases($params);
```

См. также
GetBannerPhrasesFilter

GetBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'BannerIDS': [510183],
          'Filter': {
                     'StatusBannerModerate': ['Pending'],
                     'StatusPhrasesModerate': ['Pending', 'Yes'],
                    },
         }

result = proxy.GetBanners(params = params)
```

или

```
params = {
          'CampaignIDS': [28456],
          'Filter': {
                     'StatusPhoneModerate': ['Yes'],
                     'StatusModerate':      ['Yes', 'Sent']
                    }
         }
result = proxy.GetBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
              'BannerIDS' => array(6661972, 6661973, 6661974)
              'Filter' => array(
                                'StatusPhoneModerate' =>
array('Yes'),
                            'StatusModerate' =>      array('Yes',
'Sent')
                          )
          );
$result = $client->call('GetBanners', array('params' =>
$params));
```

или

```
$params = array(
              'CampaignIDS' => array(1987518, 1987522)
              'Filter' => array(
                                'StatusBannerModerate' =>
array('Pending'),
                                'StatusPhrasesModerate' =>
array('Pending', 'Yes'),
                          ),
          );
$result = $client->call('GetBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
              'BannerIDS' => array(6661972, 6661973, 6661974)
              'Filter' => array(
                                'StatusPhoneModerate' =>
array('Yes'),
                            'StatusModerate' =>      array('Yes',
'Sent')
                          )
          );
$result = $client->GetBanners($params);
```

или

```
$params = array(
              'CampaignIDS' => array(1987518, 1987522)
              'Filter' => array(
                                 'StatusBannerModerate' =>
array('Pending'),
                                 'StatusPhrasesModerate' =>
array('Pending', 'Yes'),
                            ),
          );
$result = $client->GetBanners($params);
```

Пример на Perl

```
$params = {
           'BannerIDS' => [6661972, 6661973, 6661974)
           'Filter' => {
                        'StatusPhoneModerate' => ['Yes'],
                        'StatusModerate' =>      ['Yes', 'Sent']
                       }
          };
$result = $client->GetBanners($params);
```

или

```
$params = {
           'CampaignIDS' => array(1987518, 1987522)
           'Filter' => {
                        'StatusBannerModerate' => ['Pending'],
                        'StatusPhrasesModerate' => ['Pending',
'Yes'],
                       },
          };
$result = $client->GetBanners($params);
```

См. также
GetBanners

GetCampaignParams

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 2670805,
}
result = proxy.GetCampaignParams(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 2670805,
);
$result = $client->call('GetCampaignParams', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 2670805,
);
$result = $client->GetCampaignParams($params);
```

Пример на Perl

```
$params = {
   CampaignID => 2670805,
};
$result = $client->GetCampaignParams($params);
```

См. также
GetCampaignParams

GetCampaignsList

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = ['login1', 'login2']
result = proxy.GetCampaignsList(params = params)
```

Пример на PHP NuSOAP

```
$params = array('login1', 'login2');
$result = $client->call('GetCampaignsList', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array('login1', 'login2');
$result = $client->GetCampaignsList($params);
```

Пример на Perl

```
$params = ['login1', 'login2'];
$result = $client->GetCampaignsList($params);
```

См. также
GetCampaignsList

GetCampaignsListFilter

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'Logins': ['login1', 'login2'],
          'Filter': {
                      'StatusActivating': ['Pending'],
                      'StatusModerate': ['Yes','No','Sent']
                     },
         }
result = proxy.GetCampaignsListFilter(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
              'Logins' => array('login1', 'login2'),
              'Filter' => array(
                                'StatusActivating' =>
array('Pending'),
                                'StatusModerate' =>
array('Yes','No','Sent')
                            ),
          );
$result = $client->call('GetCampaignsListFilter', array('params'
=> $params));
```

Пример на PHP SoapClient

```
$params = array(
                'Logins' => array('login1', 'login2'),
                'Filter' => array(
                                  'StatusActivating' =>
array('Pending'),
                                  'StatusModerate' =>
array('Yes','No','Sent')
                              ),
          );
$result = $client->GetCampaignsListFilter($params);
```

Пример на Perl

```
$params = {
           'Logins' => ['login1', 'login2'],
           'Filter' => {
                        'StatusActivating' => ['Pending'],
                        'StatusModerate' => ['Yes','No','Sent']
                       },
          }
$result = $client->GetCampaignsListFilter($params);
```

См. также
GetCampaignsListFilter

GetClientInfo

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = ['login1', 'login2']
result = proxy.GetClientInfo(params = params)
```

Пример на PHP NuSOAP

```
$params = array('login1', 'login2');
$result = $client->call('GetClientInfo', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array('login1', 'login2');
$result = $client->GetClientInfo($params);
```

Пример на Perl

```
$params = ['login1', 'login2'];
$result = $client->GetClientInfo($params);
```

См. также
GetClientInfo

GetClientsList

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetClientsList()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetClientsList', array());
```

Пример на PHP SoapClient

```
$result = $client->GetClientsList();
```

Пример на Perl

```
$result = $client->GetClientsList();
```

См. также
GetClientsList

GetClientsUnits

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = ['login1', 'login2']
result = proxy.GetClientsUnits(params = params)
```

Пример на PHP NuSOAP

```
$params = array('login1', 'login2');
$result = $client->call('GetClientsUnits', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array('login1', 'login2');
$result = $client->GetClientsUnits($params);
```

Пример на Perl

```
$params = ['login1', 'login2'];
$result = $client->GetClientsUnits($params);
```

См. также
GetClientsUnits

GetForecast

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetForecast(params = 36)
```

Пример на PHP NuSOAP

```
$result = $client->call('GetForecast', array('params' => 36));
```

Пример на PHP SoapClient

```
$result = $client->GetForecast(36);
```

Пример на Perl

```
$result = $client->GetForecast(36);
```

См. также
GetForecast

GetForecastList

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetForecastList()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetForecastList', array());
```

Пример на PHP SoapClient

```
$result = $client->GetForecastList();
```

Пример на Perl

```
$result = $client->GetForecastList();
```

См. также
GetForecastList

GetKeywordsSuggestion

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'Keywords': ['keyword1', 'keyword2'],
         }
result = proxy.GetKeywordsSuggestion(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
                'Keywords' => ['keyword1', 'keyword2']
);
$result = $client->call('GetKeywordsSuggestion', array('params'
=> $params));
```

Пример на PHP SoapClient

```
$params = array(
                'Keywords' => ['keyword1', 'keyword2']
);
$result = $client->GetKeywordsSuggestion($params);
```

Пример на Perl

```
$params = {
            'Keywords' => ['keyword1', 'keyword2']
          };
$result = $client->GetKeywordsSuggestion($params);
```

См. также
GetKeywordsSuggestion

GetRegions

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetRegions()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetRegions', array());
```

Пример на PHP SoapClient

```
$result = $client->GetRegions();
```

Пример на Perl

```
$result = $client->GetRegions();
```

См. также
GetRegions

GetReportList

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetReportList()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetReportList', array());
```

Пример на PHP SoapClient

```
$result = $client->GetReportList();
```

Пример на Perl

```
$result = $client->GetReportList();
```

См. также
GetReportList

GetRubrics

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetRubrics()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetRubrics', array());
```

Пример на PHP SoapClient

```
$result = $client->GetRubrics();
```

Пример на Perl

```
$result = $client->GetRubrics();
```

См. также
GetRubrics

GetStatGoals

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 2670805,
}
result = proxy.GetStatGoals(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 2670805,
);
$result = $client->call('GetStatGoals', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 2670805,
);
$result = $client->GetStatGoals($params);
```

Пример на Perl

```
$params = {
   CampaignID => 2670805,
};
$result = $client->GetStatGoals($params);
```

См. также
GetStatGoals

GetSubClients

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {'Login': 'my-client-1'}
childs = proxy.GetSubClients(params = params)
```

Пример на PHP NuSOAP

```
$createShortUserInfo = array(
    'Login' => "my-client-1"
);
$childs = $client->call('GetSubClients', array('params' =>
$createShortUserInfo));
```

Пример на PHP SoapClient

```
$params = array(
    'Login' => "my-client-1"
);
$childs = $client->GetSubClients($params);
```

Пример на Perl

```
$childs = $client->GetSubClients({Login => "my-client-1"});
```

См. также
GetSubClients

GetTimeZones

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetTimeZones()
```

Пример на PHP NuSOAP

```
$time_zones = $client->call("GetTimeZones", array());
```

Пример на PHP SoapClient

```
$time_zones = $client->GetTimeZones();
```

Пример на Perl

```
my $time_zones = $client->GetTimeZones();
```

См. также
GetTimeZones

GetVersion

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetVersion()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetVersion', array());
```

Пример на PHP SoapClient

```
$result = $client->GetVersion();
```

Пример на Perl

```
$result = $client->GetVersion();
```

См. также
GetVersion

GetWordstatReport

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetWordstatReport(params = 36)
```

Пример на PHP NuSOAP

```
$result = $client->call('GetWordstatReport', array('params' =>
36));
```

Пример на PHP SoapClient

```
$result = $client->GetWordstatReport(36);
```

Пример на Perl

```
$result = $client->GetWordstatReport(36);
```

См. также
GetWordstatReport

GetWordstatReportList

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.GetWordstatReportList()
```

Пример на PHP NuSOAP

```
$result = $client->call('GetWordstatReportList', array());
```

Пример на PHP SoapClient

```
$result = $client->GetWordstatReportList();
```

Пример на Perl

```
$result = $client->GetWordstatReportList();
```

См. также
GetWordstatReportList

ModerateBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 123121,
          'BannerIDS': [54654, 46546, 878545]
         }
result = proxy.ModerateBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->call('ModerateBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->ModerateBanners($params);
```

Пример на Perl

```
$params = {
   CampaignID => 123121,
   BannerIDS => [54654, 46546, 878545]
};
$result = $client->ModerateBanners($params);
```

См. также
ModerateBanners

PingAPI

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
result = proxy.PingAPI()
```

Пример на PHP NuSOAP

```
$result = $client->call('PingAPI', array());
```

Пример на PHP SoapClient

```
$result = $client->PingAPI();
```

Пример на Perl

```
$result = $client->PingAPI();
```

См. также
PingAPI

ResumeBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 123121,
          'BannerIDS': [54654, 46546, 878545]
         }
result = proxy.ResumeBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->call('ResumeBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->ResumeBanners($params);
```

Пример на Perl

```
$params = {
   CampaignID => 123121,
   BannerIDS => [54654, 46546, 878545]
};
$result = $client->ResumeBanners($params);
```

См. также
ResumeBanners

ResumeCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 54654564,
}
result = proxy.ResumeCampaign(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->call('ResumeCampaign', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->ResumeCampaign($params);
```

Пример на Perl

```
$params = {
   CampaignID => 54654564,
};
$result = $client->ResumeCampaign($params);
```

См. также
ResumeCampaign

SetAutoPrice

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 1477023,
          'Mode': 'SinglePrice',
          'SinglePrice': 0.8
         }
result = proxy.SetAutoPrice(params = params)

params = {
          'CampaignID': 1477023,
          'Mode': 'Wizard',
          'PriceBase': 'pmin',
          'ProcBase': 'value',
          'MaxPrice': 30,
          'Proc': 60,
          'UpdatePhrases': 'Yes',
          'UpdateCategories': 'Yes'
         }
result = proxy.SetAutoPrice(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
              'CampaignID' => 1477023,
              'Mode' => 'SinglePrice',
              'SinglePrice' => 0.8
          );
$result = $client->call('SetAutoPrice', array('params' =>
$params));

$params = array(
              'CampaignID' => 1477023,
              'Mode' => 'Wizard',
              'PriceBase' => 'pmin',
              'ProcBase' => 'value',
              'MaxPrice' => 30,
              'Proc' => 60,
              'UpdatePhrases' => 'Yes',
              'UpdateCategories' => 'Yes'
          );

$result = $client->call('SetAutoPrice', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
              'CampaignID' => 1477023,
              'Mode' => 'SinglePrice',
              'SinglePrice' => 0.8
          );
$result = $client->SetAutoPrice($params);

$params = array(
              'CampaignID' => 1477023,
              'Mode' => 'Wizard',
              'PriceBase' => 'pmin',
              'ProcBase' => 'value',
              'MaxPrice' => 30,
              'Proc' => 60,
              'UpdatePhrases' => 'Yes',
              'UpdateCategories' => 'Yes'
          );
$result = $client->SetAutoPrice($params);
```

Пример на Perl

```
$params = {
           'CampaignID' => 1477023,
           'Mode' => 'SinglePrice',
           'SinglePrice' => 0.8
          };
$result = $client->SetAutoPrice($params);

$params = {
           'CampaignID' => 1477023,
           'Mode' => 'Wizard',
           'PriceBase' => 'pmin',
           'ProcBase' => 'value',
           'MaxPrice' => 30,
           'Proc' => 60,
           'UpdatePhrases' => 'Yes',
           'UpdateCategories' => 'Yes'
          };
$result = $client->SetAutoPrice($params);
```

См. также
SetAutoPrice

StopBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 123121,
          'BannerIDS': [54654, 46546, 878545]
         }
result = proxy.StopBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->call('StopBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->StopBanners($params);
```

Пример на Perl

```
$params = {
   CampaignID => 123121,
   BannerIDS => [54654, 46546, 878545]
};
$result = $client->StopBanners($params);
```

См. также
StopBanners

StopCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 54654564,
}
result = proxy.StopCampaign(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->call('StopCampaign', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->StopCampaign($params);
```

Пример на Perl

```
$params = {
   CampaignID => 54654564,
};
$result = $client->StopCampaign($params);
```

См. также
StopCampaign

UnArchiveBanners

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
          'CampaignID': 123121,
          'BannerIDS': [54654, 46546, 878545]
         }
result = proxy.UnArchiveBanners(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->call('UnArchiveBanners', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
        CampaignID => 123121,
        BannerIDS => array(54654, 46546, 878545)
    );
$result = $client->UnArchiveBanners($params);
```

Пример на Perl

```
$params = {
   CampaignID => 123121,
   BannerIDS => [54654, 46546, 878545]
};
$result = $client->UnArchiveBanners($params);
```

См. также
UnArchiveBanners

UnArchiveCampaign

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = {
    'CampaignID': 54654564,
}
result = proxy.UnArchiveCampaign(params = params)
```

Пример на PHP NuSOAP

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->call('UnArchiveCampaign', array('params' =>
$params));
```

Пример на PHP SoapClient

```
$params = array(
    CampaignID => 54654564,
);
$result = $client->UnArchiveCampaign($params);
```

Пример на Perl

```
$params = {
   CampaignID => 54654564,
};
$result = $client->UnArchiveCampaign($params);
```

См. также
UnArchiveCampaign

UpdatePrices

- Пример на Python
- Пример на PHP NuSOAP
- Пример на PHP SoapClient
- Пример на Perl

Пример на Python

```
params = [
    {
        'CampaignID' : 635646,
        'BannerID' : 824972 ,
        'PhraseID' : 11124877,
        'Price' : '1.14',
        'ContextPrice' : '0.91',
        'AutoBroker' : 'No',
        'AutoBudgetPriority' : 'Low',
    }, {
        'CampaignID' : 635646,
        'BannerID' : 824952,
        'PhraseID' : 11141888,
        'AutoBroker' : 'No',
    }
]
result = proxy.UpdatePrices(params = params)
```

Пример на PHP NuSOAP

```
$updatePricesData = array(
                        array(
                              "CampaignID" => 1717240,
                              "BannerID" => 4773917,
                              "PhraseID" => 48953559,
                              "Price" => 10,
                              "ContextPrice" => 8,
                              "AutoBudgetPriority" => High,
                        ),
                        array(
                              "CampaignID" => 1717240,
                              "BannerID" => 4773917,
                              "PhraseID" => 48953558,
                              "Price" => 10,
                              "ContextPrice" => 8,
                              "AutoBudgetPriority" => Low,
                        )
                    );

$result = $client->call('UpdatePrices', array('params' =>
$updatePricesData));
```

Пример на PHP SoapClient

```
$params = array(
              array(
                    "CampaignID" => 1717240,
                    "BannerID" => 4773917,
                    "PhraseID" => 48953559,
                    "Price" => 10,
                    "ContextPrice" => 8,
                    "AutoBudgetPriority" => Medium,
              ),
              array(
                    "CampaignID" => 1717240,
                    "BannerID" => 4773917,
                    "PhraseID" => 48953558,
                    "Price" => 10,
                    "ContextPrice" => 8,
                    "AutoBudgetPriority" => Low,
              )
          );
$result = $client->UpdatePrices($params);
```

Пример на Perl

```
$params = [
    {
         CampaignID => 1717240,
         BannerID => 4773917,
         PhraseID => 48953559,
         Price => 10,
         ContextPrice => 8,
         AutoBudgetPriority => 'Low',
    },
    {
         CampaignID => 1717240,
         BannerID => 4773917,
         PhraseID => 48953558,
         Price => 10,
         ContextPrice => 8,
         AutoBudgetPriority => 'Medium',
    }
];
$result = $client->UpdatePrices($params);
```

См. также
UpdatePrices

Примеры приложений

Perl

Создание рекламной кампании на Perl (SOAP::Lite)

Иллюстрирует создание рекламной кампании на Яндекс.Директе и размещение в ней баннера с использованием **SOAP::Lite**.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примеров не производится.

Совет:
Пример доступен для скачивания: example-create-camp.pl (6,51 кБ).

Демонстрирует использование методов:

- CreateOrUpdateCampaign
- CreateOrUpdateBanners

```
#!/usr/bin/perl -w

=head1 NAME

    Пример работы с API сервиса Яндекс.Директ
        на языке Perl с использованием библиотеки SOAP::Lite

=head1 DESCRIPTION

   В примере использован рекомендуемый синтаксис для работы с API
Яндекс.Директа
        на языке Perl с использованием библиотеки SOAP::Lite

    Обращаем внимание, что все текстовые данные должны быть в
кодировке UTF-8

=cut

use strict;

use SOAP::Lite;
use Crypt::SSLeay;

# Information for authorization
$ENV{HTTPS_CA_DIR} = '/path/to/certificates/';
$ENV{HTTPS_CERT_FILE} = $ENV{HTTPS_CA_DIR} . 'cert.crt';
$ENV{HTTPS_KEY_FILE}  = $ENV{HTTPS_CA_DIR} . 'private.key';
$ENV{HTTPS_CA_FILE}   = $ENV{HTTPS_CA_DIR} . 'cacert.pem';
```

```
# Login of owner for new ad campaign
my $new_campaign_login = 'client-login';

# Initializing soap object
my $client = SOAP::Lite
                    -> proxy('https://soap.direct.yandex.ru/api/
v3/')
                    -> uri('API')
                    -> on_fault(\&process_soap_fault);

# For debug mode
# $client->on_debug(sub {print @_});

print "Creating a new advertising campaign: ";

# Ad campaign parameters
my $CAMPAIGN_PARAMS = get_campaign_params($new_campaign_login);

# Call method API - CreateOrUpdateCampaign
my $new_campaign_id = $client-
>CreateOrUpdateCampaign( $CAMPAIGN_PARAMS )->result;

# Processing result
print "CampaignID = $new_campaign_id";

if ($new_campaign_id) {
    print "\nCreating a new ad: ";

    # Ad parameters
    my $BANNER_PARAMS = get_banner_params($new_campaign_id);

    # Call method API - CreateOrUpdateBanners
    my $new_banner_ids = $client-
>CreateOrUpdateBanners([ $BANNER_PARAMS ])->result;

    # Processing result
    print "BannerID = ".join(", ", @$new_banner_ids);
}


=head2 process_soap_fault

    Функция для обработки ошибок

=cut

sub process_soap_fault
{
    my ($soap, $request) = @_;

    my $err_string = "";
    if (ref $request) {
        $err_string =  "Error:"
                . join("\n", ""
                            , "Code: ".$request->faultcode
                            , "Describe error: ". $request-
>faultstring
                            , "Details: ".$request->faultdetail
                            , ""
                );
    } else {
        $err_string = $soap->transport->status;
    }

    die $err_string;
}

=head2 get_campaign_params
```

```
    Функция возвращает пример параметров для новой рекламной
кампании

=cut

sub get_campaign_params
{
    my $campaign_login = shift;

    # Example of parameters for new ad campaign

    return {
        'CampaignID' => 0,
        'Login' => $campaign_login,

        'Name' => 'New ad campaign 1',
        'FIO' => 'Ivanov Ivan Ivanovitch',
        'StartDate' => '2009-01-15',

        'TimeTarget' => {
            'TimeZone' => 'Europe/Moscow',
            'DaysHours' => [
                {
                    'Hours' => [8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],
                     'Days' => [1, 2, 3, 4, 5]
                }
            ],
        },

        'ContextLimit' => 'Limited',
        'ContextLimitSum' => 90,
        'ContextPricePercent' => 100,
        'StatusContextStop' => 'No',

        'Autobudget' => 'Yes',
        'AutobudgetSum' => 100,
        'AutobudgetMaxPrice' => 5,

        'EmailNotification' => {
            'Email' => 'user@ya.ru',
            'MoneyWarningValue' => 1,
            'WarnPlaceInterval' => 15,
            'SendWarn' => 'Yes',
            'SendAccNews' => 'Yes'
        },

        'SmsNotification' => {
            'MetricaSms' => 'Yes',
            'ModerateResultSms' => 'Yes',
            'MoneyInSms' => 'Yes',
            'MoneyOutSms' => 'Yes',
            'SmsTimeFrom' => '09:15',
            'SmsTimeTo' => '22:15'
        },

        'StatusMetricaControl' => 'Yes',
        'ConsiderTimeTarget' => 'No',
        'StatusOpenStat' => 'No',
        'AutoOptimization' => 'Yes',
        'StatusBehavior' => 'Yes',

        'DisabledDomains' => 'domain1.ru,domain2.ru',
        'DisabledIps' => '64.234.23.21',

        'MinusKeywords' => ['keyword1', 'keyword2', 'keyword3'],

        'AddRelevantPhrases' => 'Yes',
        'RelevantPhrasesBudgetLimit' => 50
```

```
    };
}

=head2 get_banner_params

    Функция возвращает пример параметров для нового объявления

=cut

sub get_banner_params
{
    my $campaign_id = shift;

    return {
        'CampaignID' => $campaign_id
        , 'BannerID' => 0
        , 'Title' => 'Ad Title'
        , 'Text' => 'Ad text'
        , 'Href' => 'http://your-site.ya.ru/'
        , 'Geo' => '1, 2, -214'

        , 'Phrases' => [
            {
                'IsRubric' => 'No'
                , 'Phrase' => 'Test phrase 1'
                , 'PhraseID' => 0
                , 'AutoBroker' => 1
                , 'Price' => 0.11
                , 'AutobudgetPriority' => 'Medium'

            }, {
                'IsRubric' => 'No'
                , 'Phrase' => 'Test phrase 2'
                , 'PhraseID' => 0
                , 'AutoBroker' => 1
                , 'Price' => 0.11
                , 'AutobudgetPriority' => 'Medium'
            }, {
                'IsRubric' => 'Yes'
                , 'Phrase' => '99'
                , 'PhraseID' => 0
                , 'AutoBroker' => 1
                , 'Price' => 0.11
                , 'AutobudgetPriority' => 'Medium'
            }
        ]

        # Контактная информация
        , 'ContactInfo' => {
            'CompanyName' => 'Your Company Name'

            # address
            , 'Country' => 'Россия'
            , 'City' => 'Москва'
            , 'Street' => 'улица Центральная'
            , 'House' => 1
            , 'Build' => 2
            , 'Apart' => 123

            # phone
            , 'CountryCode' => '+7'
            , 'CityCode' => '495'
            , 'Phone' => '123-45-67'
            , 'PhoneExt' => '1234'

            # additional information
            , 'ContactEmail' => 'email@ya.ru'
            , 'ContactPerson' => 'Ivanov Ivan'
            , 'ExtraMessage' => 'Detailed description of your
```

```
product / service'
            , 'IMClient' => 'jabber'
            , 'IMLogin' => 'email@ya.ru'

            # worktime
            , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
        }

        # Дополнительные ссылки
        , 'Sitelinks' => [
            {
                'Title' => 'Дополнительная ссылка 1'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
            }
          , {
                'Title' => 'Дополнительная ссылка 2'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
            }
          , {
                'Title' => 'Дополнительная ссылка 3'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
            }
        ]
    };
}
```

См. также
Протокол SOAP
http://www.soaplite.com/

Пример работы с JSON на Perl

Иллюстрирует обращение к сервису с использованием библиотеки JSON и авторизацией по токену.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примеров не производится.

Совет:
Пример доступен для скачивания: example-auth-by-token.pl (2,65 кБ).

Демонстрирует использование методов:

- GetClientInfo

```
#!/usr/bin/perl -w

=head1 NAME

    Пример работы с API сервиса Яндекс.Директ
        на языке Perl с использованием библиотеки JSON
        и авторизацией по токену

=head1 DESCRIPTION

   В примере использован рекомендуемый синтаксис для работы с API
сервиса Яндекс.Директ
        на языке Perl с использованием протокола JSON и
авторизацией по токену.

    Обращаем внимание, что все текстовые данные должны быть в
кодировке UTF8

    Подробнее про получение токена - читайте в документации:
        http://api.yandex.ru/direct/

    В качестве примера рассмотрен вызов метода GetClientInfo,
возвращающий информацию об указанном логине.

=cut


use strict;

use JSON qw/to_json from_json/;
use Encode qw/decode_utf8/;

use HTTP::Request::Common;
use HTTP::Response;
use LWP::UserAgent;

use Data::Dumper;

# Данные для авторизации, полученные в сервисе oauth.yandex.ru
my $client_id = 'YANDEX-OAUTH-CLIENT-ID';
my $token = 'YANDEX-OAUTH-LOGIN-TOKEN';
my $login = 'YANDEX-LOGIN';

my $url = "https://soap.direct.yandex.ru/json-api/v3/";

# Подготовка данных для запроса
my $data = {};
$data->{param} = [$login];
$data->{method} = 'GetClientInfo';

$data->{application_id} = $client_id;
$data->{token} = $token;
$data->{login} = $login;

my $json_data = to_json($data);

my $ua = new LWP::UserAgent( timeout => 600 );

my $request = HTTP::Request->new('POST', $url, undef,
$json_data);

# Запрос на сервер
my $response = $ua->request($request);

# for debug
```

```
# warn $request->as_string, "\n";
# warn $response->as_string, "\n";

if ($response->is_success) {

    my $result = from_json(Encode::decode_utf8($response-
>decoded_content())) || {};

    if (defined $result->{error_code}) {

        # Если ошибку вернул сервер API Яндекс.Директа
        die join "\n"
                    , "API error:"
                    , "Code: ".$result->{error_code}
                    , "Describe: ".$result->{error_str}
                    , "Detail:".($result->{error_detail} || '');

    } elsif (defined $result->{data}) {

        print Dumper {result => $result};

    }

} else {

    # Если ошибка произошла при отправке запроса
    die "HTTP error: ".$response->status_line;

}
```

См. также
Формат JSON
Описание формата JSON

PHP

Создание рекламной кампании на PHP (NuSOAP)

Иллюстрирует создание рекламной кампании на Яндекс.Директе и размещение в ней баннера с использованием NuSOAP.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примера не производится.

Пример демонстрирует использование следующих методов:

- CreateOrUpdateCampaign
- CreateOrUpdateBanners

```
<?php

#
# Пример работы с API службы Яндекс.Директ
# на языке PHP с использованием NuSOAP.
#
# Для запуска примера необходим пакет NuSOAP
#

# Информация о клиенте Яндекс.Директа:
$new_campaign_login = 'client-login';
$new_campaign_fio = 'ФИО';
$new_campaign_email = 'user@domain';

$cert = '/home/path_to_cert_dir/cert.crt';        # файл,
содержащий сертификат пользователя
$private = '/home/path_to_cert_dir/private.key';  # файл,
содержащий приватный ключ
$cacert = '/home/path_to_cert_dir/cacert.pem';    # файл,
содержащий корневой сертификат

################################################################
################

require_once('nusoap073/lib/nusoap.php');

$proxyhost = '';
$proxyport = '';
$proxyusername = '';
$proxypassword = '';

# Создание объекта для подключения к SOAP
$wsdlurl = 'http://soap.direct.yandex.ru/wsdl/v3/';
$client = new nusoap_client($wsdlurl, 'wsdl', $proxyhost,
$proxyport, $proxyusername, $proxypassword);
$client->authtype = 'certificate';
$client->decode_utf8 = 0;
$client->soap_defencoding  = 'UTF-8';
$client->certRequest['sslcertfile'] = $cert;
$client->certRequest['sslkeyfile']  = $private;
$client->certRequest['cainfofile']  = $cacert;
```

```
# Пример параметров, необходимых для создания кампании
$params = array(
    'CampaignID' => 0,
    'Login' => 'user_login',
    'StatusContextStop' => 'No',
    'Name' => 'Новая кампания',
    'FIO' => 'Иванов Иван Иванович',
    'StartDate' => '2009-01-15',

    'AutoOptimization' => 'No',

    'DisabledDomains' => 'domain1.ru,domain2.ru',
    'DisabledIps' => '64.234.23.21',

    'StatusBehavior' => 'Yes',

    'TimeTarget' => array(
        'TimeZone' => 'Europe/Moscow',
        'DaysHours' => array(
            array(
                'Hours' => array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12),
                'Days' => array(1, 2, 3, 4)
            )
        ),
    ),
    'StatusMetricaControl' => 'Yes',
    'ConsiderTimeTarget' => 'No',
    'StatusOpenStat' => 'No',
    'ContextPricePercent' => 10,

    'ContextLimit' => 'Limited',
    'ContextLimitSum' => 70,

    'Autobudget' => 'Yes',
    'AutobudgetSum' => 76,
    'AutobudgetMaxPrice' => 17,

    'EmailNotification' => array(
        'Email' => 'user@ya.ru',
        'MoneyWarningValue' => 1,
        'WarnPlaceInterval' => 15,
        'SendWarn' => 'Yes',
        'SendAccNews' => 'Yes'
    ),

    'SmsNotification' => array(
        'MetricaSms' => 'Yes',
        'ModerateResultSms' => 'Yes',
        'MoneyInSms' => 'Yes',
        'MoneyOutSms' => 'Yes',
        'SmsTimeFrom' => '09:15',
        'SmsTimeTo' => '22:15'
    ),

    'MinusKeywords' => array('keyword1', 'keyword2', 'keyword3'),

    'AddRelevantPhrases' => 'Yes',
    'RelevantPhrasesBudgetLimit' => 50
);

# Вызов SOAP-метода CreateOrUpdateCampaign
$result = $client->call("CreateOrUpdateCampaign", array('params'
=> $params));
print("Create new company: ");
if ($client->fault) {
    $newCampaignId = 0;
    print "SOAP Fault: ";
```

```
} else {
    $newCampaignId = $result;
}
print "$result\n";
if ($newCampaignId) {

    # Параметры создания баннера
    $params = array(
        array(
            'CampaignID' => $newCampaignId
            , 'BannerID' => 0
            , 'Title' => 'Заголовок объявления'
            , 'Text' => 'Текст объявления'
            , 'Href' => 'http://market-click2.ya.ru/?q=qu'
            , 'Geo' => '2,183,3,5'
            , 'Phrases' => array(
                array(
                      'IsRubric' => 0
                    , 'Phrase' => 'Текст фразы'
                    , 'PhraseID' => 0
                    , 'AutoBroker' => 0
                    , 'Price' => 0.11
                    , 'AutobudgetPriority' => 'Medium'
                ),
                array(
                      'IsRubric' => 0
                    , 'Phrase' => 'Текст еще одной фразы'
                    , 'PhraseID' => 0
                    , 'AutoBroker' => 1
                    , 'Price' => 0.11
                    , 'AutobudgetPriority' => 'Medium'
                ),
            )

            # Контактная информация
            , 'ContactInfo' => array(
                  'Apart' => 2
                , 'Build' => 2
                , 'City' => 'Москва'
                , 'CityCode' => 495
                , 'ContactEmail' => 'email@ya.ru'
                , 'ContactPerson' => 'контактное лицо'
                , 'Country' => 'Россия'
                , 'CountryCode' => '+7'
                , 'ExtraMessage' => 'подробнее о товаре/услуге'
                , 'House' => '2'
                , 'IMClient' => 'jabber'
                , 'IMLogin' => 'email@ya.ru'
                , 'CompanyName' => 'ФИО2'
                , 'Phone' => '23455'
                , 'PhoneExt' => '223'
                , 'Street' => 'улица'
                , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
            )

            # Дополнительные ссылки
            , 'Sitelinks' => array(
                array(
                      'Title' => 'Дополнительная ссылка 1'
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
                ),
                array(
                      'Title' => 'Дополнительная ссылка 2'
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
                ),
                array(
                      'Title' => 'Дополнительная ссылка 3'
```

```
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
                )
            )
        )
    );

    # Вызов метода CreateOrUpdateBanners
    $result = $client->call("CreateOrUpdateBanners",
array("params" => $params));
    print("Create new banner: ");
    if ($client->fault) {
        print "SOAP Fault: " . $result;
    }
    print $result;
}
?>
```

См. также
Протокол SOAP
http://sourceforge.net/projects/nusoap/
http://ru2.php.net/manual/en/book.soap.php

Создание рекламной кампании на PHP (SOAPClient)

Иллюстрирует создание рекламной кампании на Яндекс.Директе и размещение в ней баннера с использованием библиотеки SOAPClient.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примера не производится.

Примечание:
Для программных приложений на языке PHP с использованием библиотеки SOAPClient необходимо соединить файл сертификата `cert.crt` и файл приватного ключа `private.key` в один файл. Путь к этому файлу необходимо указывать при инициализации SOAP-соединения.

Подробнее о файлах сертификата читайте в разделе Получение цифровых сертификатов.

Пример демонстрирует использование следующих методов:

- CreateOrUpdateCampaign
- CreateOrUpdateBanners

```
<?php

# -*- encoding: utf-8 -*-
#
# Пример работы с API сервиса Яндекс.Директ
# на языке PHP с использованием SOAPClient.
#
# Для запуска примера необходим интерпретатор PHP версии 5.2
#

# Адрес WSDL
$wsdlurl = "http://soap.direct.yandex.ru/wsdl/v3/";

# Путь с файлу, состоящему из сертификата пользователя и
приватного ключа
$local_cert = '/path_to_certs/solid-cert.crt';

# Информация о клиенте Яндекс.Директа
$new_campaign_login = 'client-login';
$new_campaign_fio = 'ФИО';
$new_campaign_email = 'user@domain';

# Настройка клиентского SOAP-приложения
ini_set("soap.wsdl_cache_enabled", "0");
################################################################
################

# Создание объекта SoapClient
$client = new SoapClient($wsdlurl,
    array(
        'trace'=> 1,
        'exceptions' => 0,
        'encoding' => 'UTF-8',
        'local_cert' => $local_cert,
        'passphrase' => ''
    )
);

# Пример параметров, необходимых для создания кампании
$params = array(
    'CampaignID' => 0,
    'Login' => 'user_login',
    'StatusContextStop' => 'No',
    'Name' => 'Новая кампания',
    'FIO' => 'Иванов Иван Иванович',
    'StartDate' => '2009-01-15',

    'AutoOptimization' => 'No',

    'DisabledDomains' => 'domain1.ru,domain2.ru',
    'DisabledIps' => '64.234.23.21',

    'StatusBehavior' => 'Yes',

    'TimeTarget' => array(
        'TimeZone' => 'Europe/Moscow',
        'DaysHours' => array(
            array(
                'Hours' => array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12),
                'Days' => array(1, 2, 3, 4)
            )
        ),
    ),
    'StatusMetricaControl' => 'Yes',
    'ConsiderTimeTarget' => 'No',
    'StatusOpenStat' => 'No',
    'ContextPricePercent' => 10,
```

```
    'ContextLimit' => 'Limited',
    'ContextLimitSum' => 70,

    'Autobudget' => 'Yes',
    'AutobudgetSum' => 76,
    'AutobudgetMaxPrice' => 17,

    'EmailNotification' => array(
        'Email' => 'user@ya.ru',
        'MoneyWarningValue' => 1,
        'WarnPlaceInterval' => 15,
        'SendWarn' => 'Yes',
        'SendAccNews' => 'Yes'
    ),

    'SmsNotification' => array(
        'MetricaSms' => 'Yes',
        'ModerateResultSms' => 'Yes',
        'MoneyInSms' => 'Yes',
        'MoneyOutSms' => 'Yes',
        'SmsTimeFrom' => '09:15',
        'SmsTimeTo' => '22:15'
    ),

    'MinusKeywords' => array('keyword1', 'keyword2', 'keyword3'),

    'AddRelevantPhrases' => 'Yes',
    'RelevantPhrasesBudgetLimit' => 50
);

# Вызов SOAP-метода CreateOrUpdateCampaign
$newCampaignId = 0;
print("Create new company: ");
$result = $client->CreateOrUpdateCampaign($params);
if (is_soap_fault($result)) {
    print "SOAP Fault: (faultcode: {$result->faultcode},
faultstring: {$result->faultstring})";
} else {
    $newCampaignId = $result;
    print $newCampaignId . "\n";
}

if ($newCampaignId) {

    # Параметры создания баннера
    $params = array(
        array(
              'CampaignID' => $newCampaignId
            , 'BannerID' => 0
            , 'Title' => 'Заголовок объявления'
            , 'Text' => 'Текст объявления'
            , 'Href' => 'http://market-click2.ya.ru/?
q=qu'
            , 'Geo' => '2,183,3,5'
            , 'Phrases' => array(
                array(
                      'IsRubric' => 0
                    , 'Phrase' => 'Текст фразы'
                    , 'PhraseID' => 0
                    , 'AutoBroker' => 0
                    , 'Price' => 0.11
                    , 'AutobudgetPriority' => 'Medium'

                ),
                array(
                      'IsRubric' => 0
                    , 'Phrase' => 'Текст еще одной фразы'
                    , 'PhraseID' => 0
                    , 'AutoBroker' => 1
```

```
                    , 'Price' => 0.11
                    , 'AutobudgetPriority' => 'Medium'
                ),
            )

            # Контактная информация
            , 'ContactInfo' => array(
                  'Apart' => 2
                , 'Build' => 2
                , 'City' => 'Москва'
                , 'CityCode' => 495
                , 'ContactEmail' => 'email@ya.ru'
                , 'ContactPerson' => 'контактное лицо'
                , 'Country' => 'Россия'
                , 'CountryCode' => '+7'
                , 'ExtraMessage' => 'подробнее о товаре/услуге'
                , 'House' => '2'
                , 'IMClient' => 'jabber'
                , 'IMLogin' => 'email@ya.ru'
                , 'CompanyName' => 'ФИО2'
                , 'Phone' => '23455'
                , 'PhoneExt' => '223'
                , 'Street' => 'улица'
                , 'WorkTime' =>'0;4;10;00;18;00;5;6;13;00;16;00'
            )

            # Дополнительные ссылки
            , 'Sitelinks' => array(
                array(
                      'Title' => 'Дополнительная ссылка 1'
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=1'
                ),
                array(
                      'Title' => 'Дополнительная ссылка 2'
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
                ),
                array(
                      'Title' => 'Дополнительная ссылка 3'
                    , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
                )
            )
        )
    );

    # Вызов метода CreateOrUpdateBanners
    print("Create new banner: ");
    $result = $client->CreateOrUpdateBanners($params);
    if (is_soap_fault($result)) {
        print "SOAP Fault: (faultcode: {$result->faultcode},
faultstring: {$result->faultstring})";
    } else {
        print_r($result);
        print "\n";
    }
}
?>
```

См. также
Протокол SOAP
http://sourceforge.net/projects/nusoap/
http://ru2.php.net/manual/en/book.soap.php

Проверка подключения. Простой скрипт на PHP (NuSOAP)

Иллюстрирует подключение к сервису Яндекс.Директ и проверку успешности авторизации с использованием NuSOAP.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примера не производится.

Пример демонстрирует использование следующих методов:

- PingAPI

```
<?php

require_once('nusoap073/lib/nusoap.php');

$cert = '/home/system-login/private/yandex-login/cert.crt';
$private = '/home/system-login/private/yandex-login/private.key';
$cacert = '/home/system-login/private/yandex-login/cacert.pem';

$proxyhost = '';
$proxyport = '';
$proxyusername = '';
$proxypassword = '';

$wsdlurl = 'http://soap.direct.yandex.ru/wsdl/v3/';
$client = new nusoap_client($wsdlurl, 'wsdl', $proxyhost,
$proxyport, $proxyusername, $proxypassword);

$client->authtype = 'certificate';
$client->decode_utf8 = 0;
$client->soap_defencoding = 'UTF-8';
$client->certRequest['sslcertfile'] = $cert;
$client->certRequest['sslkeyfile'] = $private;
$client->certRequest['cainfofile'] = $cacert;

$result = $client->call("PingAPI", array());
if ($client->fault) {
    echo '<p><b>Сбой: ';
    print_r($result);
    echo '</b></p>';
} else {
    // Проверяем, не произошла ли ошибка
    $err = $client->getError();
    if ($err) {
        // Отображаем ошибку
        echo '<p><b>Ошибка: ' . $err . '</b></p>';
    } else {
        // Отображаем результат
        print_r($result);
    }
}

// ОТЛАДКА и ПОИСК ошибок

// Отображаем запрос и ответ
echo '<h2>Запрос</h2>';
echo '<pre>' . htmlspecialchars($client->request, ENT_QUOTES) .
```

```
'</pre>';
echo '<h2>Ответ</h2>';
echo '<pre>' . htmlspecialchars($client->response, ENT_QUOTES) .
'</pre>';
// Отображаем отладочные сообщения
echo '<h2>Отладка</h2>';
echo '<pre>' . htmlspecialchars($client->debug_str, ENT_QUOTES) .
'</pre>';

?>
```

См. также
Протокол SOAP
http://sourceforge.net/projects/nusoap/
http://ru2.php.net/manual/en/book.soap.php

Пример работы с JSON на PHP

Иллюстрирует обращение к сервису с использованием библиотеки JSON и авторизацией по токену.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примеров не производится.

Совет:
Пример доступен для скачивания: example-auth-by-token.php (2,68 кБ).

Демонстрирует использование методов:

- GetClientInfo

```
<?php

/*
    Пример программного кода для работы с API сервиса
Яндекс.Директ

     В примере использован рекомендуемый синтаксис для работы с
API сервиса Яндекс.Директ
         на языке PHP с использованием протокола JSON и
авторизацией по токенам.

     Обращаем внимание, что все текстовые данные должны быть в
кодировке UTF8

     Подробнее про получение токена читайте в документации:
         http://api.yandex.ru/direct/
*/

require_once "HTTP/Request.php";

// Важно: данные отправляем POST-методом
$req =& new HTTP_Request("https://soap.direct.yandex.ru/json-api/
v3/");
```

```
$req->setMethod(HTTP_REQUEST_METHOD_POST);

// Инициализация параметров для авторизации
$data = array(
    token => "YOUR-YANDEX-OAUTH-TOKEN",
    application_id => "YOUR-YANDEX-OAUTH-CLIEND-ID",
    login => "YOUR-YANDEX-LOGIN"
);

// Параметры для запроса метода GetClientInfo
$data['method'] = "GetClientInfo";
$data['param'] = array(YOUR-YANDEX-LOGIN);

/*
    Если ваша версия php не поддерживает встроенной функции
json_encode/json_decode,
    можно воспользоваться библиотекой: http://pear.php.net/
package/Services_JSON
*/

require_once 'Classes/json.php';
$json = new Services_JSON;
$json_data = $json->encode($data);

/*
    Если встроенные функции json_encode/json_decode
поддерживаются, то программа может выглядеть так:
        $json_data = json_encode($data);
        $decoded_result = json_decode($json_data);

*/

$req->addRawPostData($json_data);

$response = $req->sendRequest();
$errmsg = PEAR::isError($response);

if (! $errmsg) {
     $result = $req->getResponseBody();
     $decoded_result = $json->decode($result);

     if (isset($decoded_result->data)) {

         // Обработка ответа метода
         print_r($decoded_result);

     } else if ($decoded_result->error_code) {
         // Если ошибку вернул сервер API
         echo "Error: code = ".$decoded_result->error_code
                     ., str = ".$decoded_result->error_str
                     ., detail = ".$decoded_result-
>error_detail;
     } else {
         echo "Unknown error";
     }

} else {
    // Если ошибка произошла при попытке запроса
    echo "Request error: ".$errmsg;
}

?>
```

См. также
Формат JSON
Описание формата JSON

Python

Создание рекламной кампании на Python (SOAPpy)

Иллюстрирует создание рекламной кампании на Яндекс.Директе и размещение в ней баннера с использованием библиотеки **SOAPpy**.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примера не производится.

Пример демонстрирует использование следующих методов:

- CreateOrUpdateCampaign
- CreateOrUpdateBanners

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

#
# Пример работы с API сервиса Яндекс.Директ
# на языке Python с использованием SOAPpy
#
# Для запуска примера необходим модифицированный пакет SOAPpy
#

import sys

from datetime import date

import SOAPpy


CERT_FILE = '/path_to_certs/cert.crt'             # файл,
содержащий сертификат пользователя
PKEY_FILE = '/path_to_certs/private.key'          # файл,
содержащий приватный ключ
client_login = u'login-client'
client_email = u'user@login'
client_fio = u'ФИО'

#################################################################
################

# Создание объекта для подключения к SOAP-сервису
url = 'http://soap.direct.yandex.ru/wsdl/v3/';
config = SOAPpy.Config
config.debug = 0                                  # 1 - для вывода
запросов и ответов сервера
config.SSL.key_file = PKEY_FILE
config.SSL.cert_file = CERT_FILE
proxy = SOAPpy.WSDL.Proxy(url, config)

# Пример параметров, необходимых для создания кампании
params = {
    'CampaignID': 0,
    'Login' : u'user_login',
```

```
    'StatusContextStop': 'No',
    'Name': u'Новая кампания',
    'FIO': u'Иванов Иван Иванович',
    'StartDate': '2009-01-15',

    'AutoOptimization': 'No',

    'DisabledDomains': 'domain1.ru,domain2.ru',
    'DisabledIps': '64.234.23.21',

    'StatusBehavior': 'Yes',

    'TimeTarget':
    {
        'TimeZone': 'Europe/Moscow',
        'DaysHours': [
            {
                'Hours': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12],
                'Days': [1, 2, 3, 4]
            }
        ],
    },
    'StatusMetricaControl': 'Yes',
    'ConsiderTimeTarget': 'No',
    'StatusOpenStat': 'No',
    'ContextPricePercent': 10,

    'ContextLimit': 'Limited',
    'ContextLimitSum': 70,

    'Autobudget': 'Yes',
    'AutobudgetSum': 76,
    'AutobudgetMaxPrice': 17,

    'EmailNotification': {
        'Email': 'user@ya.ru',
        'MoneyWarningValue': 1,
        'WarnPlaceInterval': 15,
        'SendWarn': 'Yes',
        'SendAccNews': 'Yes'
    },

    'SmsNotification': {
        'MetricaSms': 'Yes',
        'ModerateResultSms': 'Yes',
        'MoneyInSms': 'Yes',
        'MoneyOutSms': 'Yes',
        'SmsTimeFrom': '09:15',
        'SmsTimeTo': '22:15'
    },

    'MinusKeywords': {'keyword1', 'keyword2', 'keyword3'},

    'AddRelevantPhrases': 'Yes',
    'RelevantPhrasesBudgetLimit': 50
}

newCampaignId = 0

# Вызов SOAP-метода CreateOrUpdateCampaign
try:
    print "Create new company: "
    newCampaignId = proxy.CreateOrUpdateCampaign(params = params)
    print newCampaignId, "\n"
except Exception, e:
    print 'SOAP Fault: ', e

if newCcampaignId:
```

```
    # Параметры создания баннера
    params = [
    {
          'CampaignID': newCampaignId
        , 'BannerID': 0
        , 'Title': u'Заголовок объявления'
        , 'Text': u'Текст объявления'
        , 'Href': 'http://market-click2.ya.ru/?q=qu'
        , 'Geo': '2,183,3,5'
        , 'Phrases': [
            {
                  'IsRubric': 0
                , 'Phrase': u'Текст фразы'
                , 'PhraseID': 0
                , 'AutoBroker': 0
                , 'Price': 0.11
                , 'AutobudgetPriority': 'Medium'

            },
            {
                  'IsRubric': 0
                , 'Phrase': u'Текст еще одной фразы'
                , 'PhraseID': 0
                , 'AutoBroker': 1
                , 'Price': 0.11
                , 'AutobudgetPriority': 'Medium'
            },
        ]

        # Контактная информация
        , 'ContactInfo': {
              'Apart': 2
            , 'Build': 2
            , 'City': u'Москва'
            , 'CityCode': 495
            , 'ContactEmail': 'email@ya.ru'
            , 'ContactPerson': u'контактное лицо'
            , 'Country': u'Россия'
            , 'CountryCode': '+7'
            , 'ExtraMessage': u'подробнее о товаре/услуге'
            , 'House': '2'
            , 'IMClient': 'jabber'
            , 'IMLogin': 'email@ya.ru'
            , 'CompanyName': u'ФИО2'
            , 'Phone': '23455'
            , 'PhoneExt': '223'
            , 'Street': u'улица'
            , 'WorkTime':'0;4;10;00;18;00;5;6;13;00;16;00'
        }

        # Дополнительные ссылки
        , 'Sitelinks': {
            {
                'Title': 'Дополнительная ссылка 1'
              , 'Href': 'http://market-click2.ya.ru/?
q=qu&action=1'
            }
          , {
                'Title' => 'Дополнительная ссылка 2'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=2'
            }
          , {
                'Title' => 'Дополнительная ссылка 3'
              , 'Href' => 'http://market-click2.ya.ru/?
q=qu&action=3'
            }
        }
```

```
    }]

    # Вызов метода CreateOrUpdateBanners
    try:
        print "Create new banners: \n"
        newBannerIds = proxy.CreateOrUpdateBanners(params =
params)
        print newBannerIds, "\n"
    except Exception, e:
        print 'SOAP Fault: ', e
```

См. также
Протокол SOAP
http://pywebsvcs.sourceforge.net/soappy.txt

Работа с отчетами

Получение файла отчета (PHP, cURL)

Иллюстрирует подключение к сервису Яндекс.Директ и получение файла по протоколу https с использованием библиотеки cURL.

Примечание:
Приведенный ниже пример кода предоставляется «как есть» и исключительно в целях обучения. Техническая поддержка кода примера не производится.

Совет:
Пример доступен для скачивания: example-get-report.php (1,45 кБ).

```
<?php

/*

    Пример скрипта для получения файла по протоколу https с
использованием curl
    Возможно использование для получения отчетов по статистике
кампании в API Яндекс.Директа

*/

    function get_https_file($url)
    {
        // create a new cURL resource
        $ch = curl_init();

        # !CHANGE THE PATH TO SSL CERTIFICATES!
        $path = "my/path/to/directory/with/certs";

        // set URL and other options
        curl_setopt($ch, CURLOPT_URL, $url);

        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
        curl_setopt($ch, CURLOPT_CAPATH, $path);
        curl_setopt($ch, CURLOPT_CAINFO, $path."/cacert.pem");
        curl_setopt($ch, CURLOPT_SSLCERT, $path."/cert.crt");
        curl_setopt($ch, CURLOPT_SSLKEY, $path."/private.key");

        $result = curl_exec($ch);

        if( curl_errno($ch) != 0 ) {
            die('CURL_error: ' . curl_errno($ch) . ', ' .
curl_error($ch));
        }

        //  close the cURL resource and free the system resources
        curl_close($ch);
```

```
        return $result;
    }

    // адрес отчета из метода GetReportList
    // ! ссылка на wsdl указана в качестве примера
    $url = "https://soap.direct.yandex.ru/wsdl/v2/";
    print get_https_file($url);

?>
```

См. также
http://www.php.net/manual/en/book.curl.php
GetReportList — описание метода
GetReportList — примеры вызова метода

Пример файла отчета по статистике рекламной кампании

Содержит пример XML-файла отчета по статистике рекламной кампании.

Пример файла отчета:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <report>
   <reportID>1234</reportID>
   <campaignID>1234567</campaignID>
   <startDate>2006-01-01</startDate>
   <endDate>2007-01-01</endDate>

   <phrasesDict>
        <phrase type="phrase" phraseID="1" value="фраза1" />
        <phrase type="phrase" phraseID="2" value="фраза2" />
         ...
   </phrasesDict>

    <stat>
        <row [ bannerID="123456" ]
             [ rubricID="1234" | phraseID="2" ]
             [ regionID="1" ]
             [ statDate="2006-02-02" ]
             [ placeName="Яндекс" ]
             [ placeType="search" ]
             sum="12" shows="1234" clicks="123"
             [ sum_search="10" shows_search="1000"
clicks_search="100" ]
             [ sum_context="2" shows_context="123"
clicks_context="23" ]
             [ goal_id="18565" session_depth="9.35"
goal_conversion="25.91" goal_cost="1.54" ]
             [ position_type="premium" ]
        />
        ...
    </stat>
</report>
```

Примечания:

1. В теге `row` будут перечислены только те поля, которые были указаны в параметре GroupByColumns при запросе отчета с помощью метода CreateNewReport.
2. В теге `row` не могут одновременно присутствовать атрибуты `rubricID` и `phraseID`.
3. При запросе `phraseID` текст самой фразы следует искать в словаре фраз (между тегами`<phrasesDict>..</phrasesDict>`, в начале отчета).
4. Название рубрики каталога можно получить с помощью метода GetRubrics.
5. Атрибут `placeType` появляется только в том случае, если при запросе отчета была указана группировка по площадке (`clPage`).
6. Атрибуты `*_search` и `*_context` содержат данные по поисковым и контекстным площадкам соответственно, но только при условии, что не задан фильтр `filter.pageType = 'all'` .
7. Атрибуты `goal_id`, `session_depth`, `goal_conversion`, `goal_cost` появляются только в том случае, если в статистике рекламной кампании есть данные по целям счетчиков Метрики.

См. также
CreateNewReport

Предметный указатель

A
AddRelevantPhrases 7, 40, 63
AdWarnings 7, 51
AgencyLogin 90
AgencyName 40, 43, 46
Apart 52
application_id 10, 13, 14
ArchiveBanners 81, 101
ArchiveCampaign 77, 102
AutoBroker 57, 61, 68, 74, 75, 100
Autobudget 8, 41, 63
AutobudgetMaxPrice 8, 41, 63
AutoBudgetPriority 8, 55, 60, 69, 74, 75
AutobudgetSum 8, 41, 63
AutoOptimization 41, 63
AutoPriceInfo 71

B
Banner 25
BannerID 24, 50, 55, 60, 67, 70, 73, 74, 80–83, 98, 99
BannerIDS 49, 50, 55, 59, 80–83
BannerInfo 50, 67
BannerPhraseInfo 55, 60, 68
BannerPhrasesFilterRequestInfo 59
BannerPriceInfo 74
BannersFilterInfo 50
BorderDate 95
Build 52

C
CampaignBalanceInfo 48
CampaignBidsInfo 80–83
CampaignID 8, 17, 24, 28, 40, 43, 46, 48, 50, 55, 60, 63, 66, 67, 71, 73, 74, 76–83
CampaignIDInfo 28, 40, 76–79
CampaignIDS 49, 50
CampaignInfo 8, 40, 63
CampaignsFilterInfo 46
Categories 34
City 52
CityCode 52
clBanner 24
clDate 24
clGeo 24
Clicks 8, 37, 40, 44, 46, 56, 60
client_id 14
ClientID 91
ClientInfo 8, 85, 86
ClientsUnitInfo 87
clPage 24, 168
clPhrase 24
clPositionType 7, 24
clStatGoals 7, 24
COLUMNS 24
CompanyName 52
CompressReport 24
ConsiderTimeTarget 8, 41, 59, 64
ContactEmail 53
ContactInfo 51, 52, 68
ContactPerson 53
ContextCoverage 8, 56, 61
ContextLimit 8, 41, 64
ContextLimitSum 41, 64
ContextLowCTR 8, 56, 61
ContextPrice 8, 55, 60, 68, 69, 74, 75
ContextPricePercent 41, 64
Country 52
CountryCode 52
Coverage 56, 61
CoverageInfo 57, 61
CreateNewForecast 18, 34, 103
CreateNewReport 23
CreateNewReports 104
CreateNewSubclient 89, 105
CreateNewSubclientRequest 90
CreateNewSubclientResponse 91
CreateNewWordstatReport 29, 106
CreateOrUpdateBanners 66, 67, 107
CreateOrUpdateCampaign 63, 112
CTR 18, 37
cURL 166
CurrentOnSearch 56, 60

D
DateCreate 85
day 24
Days 66
DaysHours 66
DeleteBanners 83, 116
DeleteCampaign 78, 117
DeleteReport 26, 118
detail 14
diff 72
DisabledDomains 8, 41, 64
DisabledIps 8, 41, 64
Discount 7, 85
Domain 51
Done 26, 30, 36

E
Email 8, 65, 85, 91
EmailNotification 41, 63
EmailNotificationInfo 65
EndDate 24
error_code 10
error_detail 10
error_str 10
ExtraMessage 53

F
faultcode 14
faultstring 14
FieldsNames 59
Filter 24, 46, 49
FIO 40, 63, 85, 88, 91
FirstPlaceClicks 37
FirstPlaceCTR 37

FixedOnModeration 7, 52
ForecastID 18, 35–37
ForecastInfo 34
ForecastStatusInfo 36

G
Geo 25, 37, 51, 68
GeoID 29, 31, 34
GetAvailableVersions 94, 119
GetBalance 47, 119
GetBannerPhrases 54, 120
GetBannerPhrasesFilter 58, 120
GetBanners 49, 121
GetBannersInfo 49
GetCampaignParams 40, 123
GetCampaignsInfo 45
GetCampaignsList 43, 124
GetCampaignsListFilter 45, 124
GetClientInfo 9, 11, 13, 85, 125
GetClientsList 86, 126
GetClientsUnits 87, 127
GetForecast 18, 36, 127
GetForecastList 35, 128
GetKeywordsSuggestion 32, 128
GetRegions 9, 12, 92, 129
GetReportList 25, 130
GetRubrics 92, 130
GetStatGoals 27, 131
GetSubClients 88, 131
GetTimeZones 9, 12, 95, 132
GetVersion 94, 133
GetWordstatReport 31, 133
GetWordstatReportList 30, 134
GMTOffset 96
goal_conversion 168
goal_cost 168
goal_id 168
GoalID 28
GroupByColumns 24
GroupByDate 24

H
HolidayShowFrom 66
HolidayShowTo 66
Hours 66
House 52
Href 50, 51, 68, 69
Href 68
HTTPS 18

I
IMClient 53
IMLogin 53
IsActive 8, 42, 44, 46, 47, 50, 51
IsRubric 37, 55, 60, 68

J
JSON 149, 160

K
keyout 21
Keywords 32, 33
KeywordsSuggestionInfo 32

L
Limit 24
LinkWSDL 95
locale 7, 9, 12
Login 8, 40, 43, 46, 63, 85, 87, 88, 90, 91
Logins 46
LowCTR 56, 61
LowCTRWarning 56, 61

M
ManagerName 40, 43, 46
MapPoint 53
Market 37, 56, 60
Max 37, 38, 56, 60, 61
MaxPrice 72
method 9
MetricaSms 8, 65
Min 37, 38, 56, 60, 61
MinPrice 56, 61
MinusKeywords 7, 41, 64
Mode 71
ModerateBanners 79, 134
ModerateRejectionReasons 7, 51
ModerateResultSms 8, 65
MoneyInSms 8, 65
MoneyOutSms 8, 65
MoneyWarningValue 8
month 24

N
Name 8, 28, 40, 43, 46, 63, 90, 96
NewReportFilterInfo 25
NewReportInfo 24
NewWordstatReportInfo 29
No 72
NonResident 8, 86
nuSOAP 19

O
Offset 24
OGRN 7, 52
OrderBy 24
out 21

P
PageName 25
PageType 25
param 9, 39
ParentID 92, 93
Password 91
Pending 26, 30, 36
Perl 145, 149
Phone 52, 85
PhoneExt 52
PHP 152, 155, 158, 160, 166
Phrase 25, 31, 32, 37, 55, 60, 68
PhraseID 55, 60, 68, 73–75, 99
PhrasePriceInfo 73
Phrases 29, 34, 68
PingAPI 97, 135

placeType 168
pmax 72
pmin 72
PointOnMap 7, 53
PositionType 25
PremiumClicks 37, 38
PremiumCTR 37
PremiumMax 37, 56, 60, 61
PremiumMin 37, 38, 56, 60, 61
Price 55, 57, 60, 61, 68, 69, 73–75
PriceBase 72
Prices 56, 61
Probability 57, 61
Proc 72
ProcBase 72
Python 162

R

RegionID 92
RegionInfo 92
RegionName 92
ReguRubrics 9, 12
RejectReason 53
RelevantPhrasesBudgetLimit 7, 41, 64
ReportID 18, 23, 25–27, 29–31, 98
ReportInfo 26
RequestPrices 7, 59
Rest 8, 40, 44, 46, 48
result 27
ResumeBanners 81, 136
ResumeCampaign 76, 136
Role 8, 85, 88, 89
row 168
RubricFullName 93
RubricID 34, 93
RubricInfo 93
RubricName 93

S

SearchedAlso 32
SearchedWith 31
SendAccNews 8, 65, 86
SendNews 8, 86
SendWarn 8, 65, 86
session_depth 168
SetAutoPrice 71, 137
ShortCampaignInfo 43, 46
ShortClientInfo 88
ShowOnHolidays 66
Shows 8, 32, 37, 40, 44, 46, 56, 60
SinglePrice 71
Sitelink 69
Sitelinks 7, 50, 68
SmsNotification 41, 64
SmsNotificationInfo 65
SmsPhone 8, 65
SmsSendModerate 8
SmsSendWarnMoneyIn 8
SmsSendWarnMoneyOut 8
SmsTimeFrom 8, 65
SmsTimeHourFrom 8
SmsTimeHourTo 8
SmsTimeMinFrom 8
SmsTimeMinTo 8
SmsTimeTo 8, 65
SOAP-запроса 19
SOAP::Lite 19, 145
SOAPClient 19, 155
SOAPpy 19, 162
StartDate 8, 24, 40, 43, 46, 64
StatGoalInfo 28
StatGoals 25
Status 8, 42, 44, 46, 64
StatusActivating 8, 42, 44, 46, 47, 50, 51
StatusArch 85
StatusArchive 8, 42, 44, 46, 47, 50, 51
StatusBannerModerate 50, 51
StatusBehavior 42, 64
StatusContextStop 8, 42, 64
StatusForecast 36
StatusMetricaControl 8, 42, 65
StatusModerate 8, 42, 44, 46, 47
StatusOpenStat 8, 42, 65
StatusPhoneModerate 50, 52
StatusPhraseModerate 56, 60
StatusPhrasesModerate 50, 52
StatusReport 23, 26, 30
StatusShow 8, 42, 44, 46, 47, 50, 52
StatusSitelinksModerate 7, 52
STDOUT 19
StopBanners 80, 139
StopCampaign 76, 140
Street 52
string 40, 63
Sum 8, 40, 44, 46, 48
Surname 90

T

Text 50, 53, 68
TimeTarget 8, 42, 65
TimeTargetInfo 65
TimeTargetItem 66
TimeZone 66, 96
TimeZoneInfo 96
Title 50, 68, 69
token 10, 13
Type 53
TypeResultReport 24

U

UnArchiveBanners 82, 140
UnArchiveCampaign 78, 141
UnitsRest 87
UpdateCategories 72
UpdatePhrases 72
UpdatePrices 73, 142
URL 18

V

value 72
VersionDesc 95

VersionNumber 95

W

WarnPlaceInterval 8, 65
week 24
Wizard 71
WordstatItem 32
WordstatReportInfo 31
WordstatReportStatusInfo 30
WorkTime 53

X

x 53
x1 53
x2 53
XML-код 18

Y

y 53
y1 53
y2 53
Yes 72

Б

бюджет рекламной кампании 18

К

код ошибки 98

О

отчет по статистике 17

Р

рекламная кампания 17
рубрика каталога 92

С

сертификат 19

Т

токен 149, 160

Ф

формат файла отчета 167

Я

Яндекс.Каталог 93

Яндекс

API Яндекс.Директа

Руководство разработчика

5.05.2011