

Инструкция по эксплуатации

Программное обеспечение «Рейтинг»

Содержание

Термины и сокращения.....	3
Введение	3
Идентификатор запроса - RequestId.....	3
Доступ к API - PingAPI.....	4
Информация для авторизации	6
Получение результата применения модели – GetStreetClientScore	7
Коды ошибок.....	9
Описание предупреждений.....	11

Термины и сокращения

API	Application Programming Interface — Программный Интерфейс Приложения
ПО	Программное обеспечение Рейтинг

Введение

Доступ к API позволяет автоматизировать процесс обмена информацией в рамках ПО Рейтинг.

Адрес сервиса внутри виртуальной машины: <http://127.0.0.1:8080/score/>

Взаимодействие с API происходит по протоколу HTTP с помощью любого подходящего клиента. Запросы отправляются методом POST, тело запроса в формате json, ответ также в формате json

Идентификатор запроса - RequestId

Каждому запросу в API назначается уникальный идентификатор, который возвращается в ответе на запрос. В случае возникновения проблем с конкретными запросами стоит указывать идентификаторы в тексте обращения в службу поддержки. Пример RequestId:

```
$ curl -X POST -H "Content-Type: application/json" -d '{"Method": "PingAPI"}'  
http://127.0.0.1:8080/score/  
{ "Data": 1, "RequestId": "f3de4ee012752f611d90fb6a872dba7d" }
```

Партнерский RequestId:

Партнер может генерировать RequestId на своей стороне. Значение нужно передать в HTTP-заголовке *X-Request-ID*. Пример:

```
$ curl -X POST -H 'X-Request-ID: custom-request-id' -H "Content-Type: application/json" -d '{"Method": "PingAPI"}' http://127.0.0.1:8080/score/ {"Data":1,"RequestId":"custom-request-id"}
```

Доступ к API - PingAPI

Для доступа к API необходимо составить POST запрос, содержащий в поле data JSON с информацией о запросе. Метод PingAPI позволяет проверить доступность сервиса для запросов. JSON запроса должен содержать только одно поле 'Method' со значением 'PingAPI'. Метод PingAPI не требует авторизационных данных.

Запрос в формате JSON:

JSON example

```
{
  'Method': 'PingAPI'
}
```

Параметры:

Параметр	Значения	Примечание	Обязательно
Method	строка	метод, к которому необходимо получить доступ	да

Возвращаемое значение:

PingAPI response

```
{
  "Data": 1,
  "RequestId": "f3de4ee012752f611d90fb6a872dba7d"
}
```

1 означает, что сервис доступен для обращений.

Пример реализации клиента для запросов на языке Python:

Client python example

```
import urllib2
import ujson

URL = 'http://127.0.0.1:8080/score/'

def main():
    request = ujson.dumps({
        'Method': 'PingAPI',
    })
    req = urllib2.Request(URL)
    data = urllib2.urlopen(req, data=request).read()
    print ujson.loads(data)

if __name__ == '__main__':
    main()
```

curl-команда для аналогичного запроса:

```
curl -X POST -H 'Content-Type: application/json' -d '{"Method":"PingAPI"}'
http://127.0.0.1:8080/score/
```

Информация для авторизации

Для доступа к другим методам API необходимо в JSON запроса указать дополнительную информацию для авторизации.

Параметр	Значения	Примечание	Обязательно
ApplicationName	строка	имя приложения, к которому необходимо получить доступ	да

Method	строка	метод, к которому необходимо получить доступ	да
Param	JSON с параметрами метода	необходимо добавить в запрос только при наличии параметров, необходимых для работы метода	нет
Login	строка	имя клиентского приложения, осуществляющего доступ	да
Token	строка из 32 цифр '0-9' и латинских букв 'A-E'	генерируемая заранее строка для осуществления доступа к приложению	да
ProductName	строка	имя продукта, к данным которого необходимо получить доступ	нет

Поле 'param' также может содержать дополнительную информацию, передаваемую в метод.

Auth data

```
{
  'ApplicationName': 'Application',
  'Method': 'Method',
  'Param': {
    'param1': 'value1',
    'param2': 'value2',
    ...
  },
  'ProductName': 'Product',
  'Login': 'PartnerLogin',
  'Token': '0123456789ABCDEF0123456789ABCDEF'
}
```

Получение результата применения модели - GetStreetClientScore

Для получения значения score необходимо вызвать метод GetStreetClientScore с параметрами. Метод рассчитывает актуальное значение score на основе указанных данных.

Параметр	Значения	Примечание	Обязательно
Model	строка	Имя модели	да
Phones	список строк	Телефоны клиента в формате '+7xxxxxxxxx' или '8xxxxxxxxx'	да (в случае отсутствия Emails), максимум 3 элемента
Emails	список строк	Список email'ов клиента	да (в случае отсутствия Phones)
BD	строка или None	Дата рождения клиента в формате yyyymmdd, None при отсутствии	нет
Sex	0, 1 или None	Пол клиента: 0="жен", 1="муж", None="неизвестно"	нет
FirstName	строка или None	Имя клиента в utf-8	нет
MiddleName	строка или None	Отчество клиента в utf-8	нет
SecondName	строка или None	Фамилия клиента в utf-8	нет

Примечание: для полей Phones и Emails для расчета возвращаемого значения учитываются все возможные валидные значения.

При наличии ошибок в необязательных полях метод работает без учета этих полей (см описание 'warning').

Пример запроса:

GetStreetClientScore

```
{
  'ApplicationName': 'Application',
  'Method': 'GetStreetClientScore',
  'Param': {
    'Model': 'Model',
    'Phones': ['+79261112233','89262221133'],
    'Emails': ["ab@d3.re"],
    'Sex': 1,
    'BD': "19571122",
    'FirstName': "Иван",
    'MiddleName': "Иванович"
  },
  'ProductName': 'Product',
  'Login': 'PartnerLogin',
  'Token': '0123456789ABCDEF0123456789ABCDEF'
}
```

Возвращаемое значение:

GetStreetClientScore response

```
{
  'Score': 0.777,
  'MatchingLevel': 5,
  'Date': '20140522',
  'RequestId': 'bba17ee012752116fd90fb6a872dba7d'
}
```

Параметр	Значение	Примечание
Score	действительное число	величина, возвращаемая моделью
Date	Дата в формате "YYYYMMDD"	дата актуальности Score
MatchingLevel	Опционально. Целое положительное число	Принимает значения от 0 до 5

Коды ошибок

При работе с API могут возникнуть ошибки.

При недоступности сервиса API возвращаются стандартные HTTP коды ошибок, описанные в RFC.

При возникновении ошибок при работе с API сервис возвращает HTTP-response с кодом 200 и JSON с описанием ошибки в поле 'error', например:

errors

```
{
  "error":
  {
    "message": "fields required",
    "code": 405,
    "data": ["Login", "ApplicationName", "Token"]
  },
  "RequestId": "1aab7ee012752116fd90fb6a872dff56"
}
```

Возвращаемое значение:

Параметр	Значение	Примечание
message	строка	тип ошибки
code	целое положительное число	код ошибки
description	структура с полем msg и служебным полем matching (можно игнорировать)	описание ошибки, может отсутствовать
data	зависит от типа ошибки	дополнительная информация об ошибке

Возможны следующие варианты ошибок:

code	message	описание
500	internal error	ошибка сервиса, необходимо сообщить о возникновении такой ошибки
400	bad request	не получилось прочитать тело запроса или json внутри не парсится
401	wrong access key	неверные авторизационные данные
403	forbidden	запрещен доступ
404	data not found	необходимая информация по клиенту не найдена в базе данных
405	fields required	в запросе не указаны обязательные поля, поле data содержит список полей, которые необходимо добавить в запрос, например "data": ["Login", "ApplicationName", "Token"]
406	fields with bad values	в указанных полях в запросе указаны неверные значения, поле data содержит список полей с ошибками и кратким описанием каждой ошибки, например "data": [{"BD", "'BD' field must be in format '%Y%m%d'"}, {"SecondName", "'SecondName' field has bad value"}]

Описание предупреждений

При работе с сервисом некоторые запросы могут игнорировать наличие ошибок в некоторых полях и возвращать данные.

Поля, в которых возникли ошибки и которые были проигнорированы, указываются в поле 'warning', например:

```

{
  "warning":
  {
    "message": "fields with bad values",
    "data": [["BD", "BD field has bad value"],["Sex", "Sex field must be 0
or 1"]]
  },
  "RequestId": "1aab7ee012752116fd90fb6a872dff56"
}

```

Возвращаемое значение:

Параметр	Значение	Примечание
message	строка	тип ошибки
data	зависит от типа ошибки	дополнительная информация об ошибке

Ответ от сервиса в этом случае полностью выглядит следующим образом. Либо это успешный ответ с дополнительным полем 'warning', либо это ответ с ошибкой 'error' и дополнительным полем 'warning'.

Пример успешного ответа с полем 'warning':

```

{
  "Date": "20160526",
  "Data": 0.6556898354,
  "warning":
  {
    "message": "fields not valid",
    "data": [["BD", "BD field has bad value"],["Sex", "Sex field must be 0
or 1"]]
  },
  "RequestId": "1aab7ee012752116fd90fb6a872dff56"
}

```

Пример ответа с ошибкой и полем 'warning'

```

{

```

```
"warning":
{
    "message":"fields not valid",
    "data":[["BD","BD field has bad value"],["Sex","Sex field must be 0
or 1"]]
},
"error":
{
    "message":"data not found",
    "code":404,
    "description":
    {
        "msg":"data not found"
    }
},
"RequestId":"1aaddfb012752116fd90fb6a872dff56"
}
```