



AMBER
DATA

ДОКУМЕНТАЦИЯ API

ПО «ВИ КРУТИЛКА» (AmberData)

Сведения о правообладателе

Общество с ограниченной ответственностью «Амбердата»
127521, г. Москва, 12-й проезд Марьиной рощи, д.9, корп. 1, пом. VI, ком. 1
ИНН/КПП: 7703096683/771701001
тел.: (499) 938-43-78
e-mail: info@amberdata.ru

Содержание

Список терминов	3
Документируемые системы	4
Администрирование	5
Управление	5
Описание системы	6
Стэк технологий	6
Цель	6
Описание модулей системы	7
1 api	7
2 placementxml	7
3 motherofmodels	7
4 jamiefoxx	7
5 app_models	7
6 dashboard_provider	7
Диаграмма сущностей	8
Описание сущностей	9
Основные сущности	9
Сервисные сущности	10
Кейсы использования API	12
Работа с API	13
1 Фильтры	13
2 Пагинация	13
3 Поиск	13
4 Используемые нумерованные значения полей	14
5 Авторизация	15
6 Лимиты запросов	16
7 Формат запроса	17
Инструменты разграничения контекста	18
Основные точки	19
Служебные точки	34
Auth	34
Password management	34
Администрирование	34
Кейсы	35

Список терминов

- Rest

<https://ru.wikipedia.org/wiki/REST>

- CRUD

<https://ru.wikipedia.org/wiki/CRUD>

- API

<https://ru.wikipedia.org/wiki/API>

- DMP

<https://ru.wikipedia.org/wiki/DMP> (платформа управления данными)

- CSRF

[https://ru.wikipedia.org/wiki/Межсайтовая подделка запроса](https://ru.wikipedia.org/wiki/Межсайтовая_подделка_запроса)

(Запрещает POST запросы к формам со сторонних площадок)

- CORS

https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

(Запрещает встраивание)

- OAuth

https://en.wikipedia.org/wiki/OAuth#OAuth_2.0

- Таксономия – операционные данные DMP – несервисные информационные сущности, операбельные в процессе работы DMP

- ORM

<https://ru.wikipedia.org/wiki/ORM>

Документируемые системы

- Amberdata API - основной программный интерфейс системы DMP.
- DMG-Lite - вспомогательный сервис, служащий для составления профиля пользователя (см. пункт “Используемые термины” и “Кейсы”).

Управление

Запуск

Для запуска системы в продакшн используется любой uwsgi провайдер, к примеру <https://gunicorn.org>. Он настраивается на запуск файла конфигурации uwsgi из модуля jamiefoxx (подробнее см пункт “Структура модулей”).

Если первый запуск выполняется для системы инсталлированной не путем разворачивания готового образа/контейнера рабочей копии, то первоначальную инициализацию приложения сжит выполнять путем запуска скрипта.

```
sh prod_init.sh
```

Обновление

Обновление системы производится путем запуска корневого скрипта.

```
sh prod_rebuild.sh
```

Описание системы

Стэк технологий

- DB: PostgreSQL - В качестве базы хранения таксономии и сервисных сущностей;
- ~~- REDIS - В качестве системы хранения кэшей;~~
- Python 3.6 - Язык разработки;
- Django - фреймворк, модели данных, авторизации;
- Django Rest Framework - фреймворк, отвечающий за представление REST API.

Цель

1. Объединить в рамках единого API данные:
 - SelfService UI для Клиентов - паблишеров / провайдеров данных / агентств / DSP
 - Back-office UI сотрудников DMP для управления бизнес-сущностями (контрагенты, тарифные планы, сегменты, цены и тд)
 - конфигураций модулей / компонентов DMP - для Администрирования и отдачи DMG, процессинга
2. Унификация
 - API для:
 - Клиентов:
 - агентств;
 - DSP;
 - Проектов:
 - CJ;
 - BDP;
 - DMG;
 - AMG?;
 - UI;
 - Предоставление интерфейсов:
 - Back-office UI;
 - Self-service UI для:
 - паблишеров;
 - провайдеров данных;
 - агентств;
 - DSP;
 - Единый интерфейс авторизации.

Описание модулей системы

1 api

Описание правил работы CRUD REST API и система генерации описания точек входа на основе их метаданных (api).

2 placementxml

Система выгрузки конфигурации для запуска DMG, отдаваемую в виде xml, а также механизм авторизации, для доступа к получению этой конфигурации (placementxml)

Доступ к /placement_xml/ по legacy-token авторизации. (Смотри “API”)

3 motherofmodels

Управляющие механизмы, абстрактные классы для использования в других модулях.

4 jamiefoxx

Настройки для запуска родительского Django приложения. Точка входа uwsgi находится в файле

```
jamiefoxx/uwsgi.py.
```

5 app_models

Структура моделей данных системы, описанных в терминах Django-Orm, а также описания интерфейса администратора для доступа к ним.

6 dashboard_provider

Описание правил работы системы-конструктора предоставления отчетности.

Диаграмма сущностей

Рис 1. Диаграмма сущностей таксономии

Описание сущностей

Основные сущности

1 Пользователь системы

Аккаунт пользователя – учетная запись сотрудника контрагента для авторизации и разграничения прав доступа.

2 Проект (контрагент)

Контрагент

Contraгент, контрагент (медиа-партнер) - компания, с которой существуют какие-либо договорные отношения (например, контрагент м.б. поставщиком или потребителем данных). На данный момент в нашей системе регистрируются контрагенты, являющиеся нашими непосредственными партнерами, поэтому регистрация отношений вида "контрагент1 - контрагент2" пока не предусматривается.

Проект

Контрагент = Проект. Под каждую площадку партнера создается отдельный проект.

Проект включает в себя коды, наборы данных, системы и категории, оперируемые в рамках текущей площадки.

3 Код

Embedded code, код (тэга) - код определенной информационной системы, обеспечивающий обращение к сервису этой системы и размещаемый (встраиваемый) на стороне других внешних систем (например: на страницах сайтов, в креативах, в приложениях, на сторонних серверах и т.п.) с целью интеграции для обмена какими-либо данными.

4 Отчет

Сущность, репрезентативное представление которой содержит агрегированные данные в машино-читабельном виде согласно настройкам отчета. Стандартный отчет содержит данные о демографии, географии, охватах категорий в проекте.

5 Лицензия

Сущность, содержащая сведения о выгрузке сегмента пользователей из DMP в систему лицензирования (Google Ya Mail)/промежуточного хранения (Redis) - а именно - о цене/валюте, названии, временных ограничениях и тд.

6 Категория

Сущность, содержащая правила построения набора пользователей (сегмента)

Сформированный по заданным правилам именованный набор профилей пользователей.

7 Сегмент

Сформированный по заданным правилам категории именованный набор профилей пользователей.

8 Профиль пользователя

Список категорий, сформированный на основании вхождения в них пользователя.

Сервисные сущности

9 Набор данных

Dataset, Датасет, Набор данных – набор данных, принадлежащий контрагенту, наполняемый из / выгружаемый в определенную систему

Набор данных отчетов – является связкой между набором данных профилей и отчетом

Набор данных профилей – содержит категории и служит условным маркером деления данных при процессинге.

10 Система

System, система (платформа) - Информационная (программная / программно-аппаратная) система (платформа), используемая контрагентами для технологической интеграции в рамках их бизнес отношений. Для системы определяется один контрагент - владелец, предоставляющий другим контрагентам возможности использования этой системы в качестве ее абонентов.

Кейсы использования API

Получение:

- Проектов
- Кодов
- Категорий
- Систем
- Наборов данных
- Лицензий

Управление (CRUD):

- Проектами
- Кодами
- Категориями
- Лицензиями
- Аккаунтом

Работа с API

1 Фильтры

Точные

/<collection>/?<key>=<val1>

По нескольким значениям

/<collection>/?<key>=<val1>,<val2>

Промежуток

/<collection>/?<key>__range=<start>,<end>

Format 2018-10-18 15:31:41

Булевые

/<collection>/?<key>__bool=True/False

Пустого значения

/<collection>/?<key>__isnull=True/False

Пустого массива

/<collection>/?<key>__isempty=True/False

Пустой строки

/<collection>/?<key>__isemptystr=True/False

2 Пагинация

Ключи-параметры пагинации

/<collection>/?limit=100&offset=100

Ограничения

Желательно не использовать лимиты больше 150. Относится, в основном, к категориям. Иначе спи не успевают отдавать результат до истечения тайм-аута Nginx.

3 Поиск

/<collection>/?search=[string]

4 Используемые нумерованные значения полей

Статусы

ARCHIVE = 0

ACTIVE = 1
DEACTIVE = 2

Типы категорий

RAW = 0
STREAM = 1
AUDIENCE = 2
LAL = 3
KEYWORDS = 4
URLS = 5
URLMASK = 6
URLPART = 7

5 Авторизация

Правовой контекст

Задачи:

1 разграничить видимость точек в зависимости от варианта использования api

2 Пользовательское разграничение

2.1 Лимиты

2.1.1 Trottle

2.1.2 CRUD

2.2 Разграничение видимости

2.2.1 по типам операций

2.2.2 отображения полей

2.2.3 доступности точек

2.2.3 доступности объектов

Типы авторизации

1) Placementxml

2) Admin

Form-based authorisation & CSRF

3) API (OAuth2):

implicit

grant

(неявное

предоставление токена)

Стандартный способ авторизации для пользователей

Используются credentials приложения интерфейса — можно узнать через административную панель системы

Password-based

Используется при интегрировании API вместе с системами внутреннего контура клиента

6 Лимиты запросов

Для обычных обращений пользователей ограничение установлено в 5000 запросов в час. Для получающих токен по приложению через password-based авторизацию (внутренние сервисы компании) – нет ограничения. Настройки в

jamiefoxx/configs/rest_framework.py

7 Формат запроса

`/api/codes/?state=2,1&ordering=-state`

`/api/codes/?state=2,1&search=TEST`

Инструменты разграничения контекста

- User type
- Object
- Object Ownership
- Custom
- Application
- OAuth2 Scopes
- Application name
- Auth type

Основные точки

/api/me/

- получение информации о текущем пользователе

Методы

GET

/api/ownerships/

Методы

GET POST PATCH DELETE OPTIONS

Поля

user user__username user__email contragent
can_create can_update can_delete email admin pk

Фильтры

contragent

Сортировка

contragent user pk user__username

Запрос создания

[/api/events/](#)

Методы

GET POST PATCH DELETE OPTIONS

Поля

code name description state pk

Фильтры

pk code code__dataset__contragent_owner

Поиск и сортировка

name description pk code

Запрос создания

/api/goals/

Методы

GET OPTIONS

Поля

name	type	description	goal_codes	version	hint
state	version	pk			

/api/reports/

Методы

GET OPTIONS

Поля

pk name

Фильтры

contragents

/api/codes/

Методы

GET POST PATCH DELETE OPTIONS

Поля

name goal create_time description group dataset
dataset__name state pk

Фильтры

pk state goal group dataset
systems__contragent_owner systems
dataset__contragent_owner

Поиск

id name description pathname dataset__name

Сортировка

pk create_time name state goal goal__name
description group, dataset__name

Запрос создания

/api/reports_datasets/

Методы

GET OPTIONS

Поля

name system description expired_ttl contragents
codes profiles_dataset state pk

Фильтры

pk contragent_owner contragents system
expired_ttl profiles_dataset

Поиск

id name description system__id

Сортировка

pk name description state

/api/outboxes/

Методы

GET OPTIONS

Поля

name system description system_space client_id
google_product contragents contragents licenses
state pk

Фильтры

pk licenses client_id contragent_owner
contragents system system_space google_product

Поиск

id name description system__id system_space__id

Сортировка

pk name description state client_id

/api/currencies/

Методы

GET OPTIONS

Поля

code name symbol pk

/api/profiles_datasets/

Методы

GET OPTIONS

Поля

name system description expired_ttl contragents
system_space categories codes state pk

Фильтры

pk contragent_owner contragents system
expired_ttl codes system_space apply_to_all
fill_with_batches default auto_approved

Поиск

id name description system_space__id system__id

Сортировка

pk name description state

/api/systems/

Методы

GET OPTIONS

Поля

pk name description system_codes contragents
system_upid_datasets system_profilesdatasets
system_codeserviceconfigs systems_codes
system_codes state pk

Фильтры

pk contragent_owner contragents

Поиск

id name description

/api/licenses/

Методы

GET POST PATCH DELETE OPTIONS

Поля

name price currency description segment_id
category outbox client_id begin end
category__state category__category_type
category__dataset__name category__dataset
category__groups category__pathname
category__reach outbox__name outbox__system_id
outbox__system_space_id outbox__client_id
outbox__google_product private
outbox__system__name state pk

Фильтры

pk state category outbox category__category_type
category__state category__dataset__name
category__dataset_id outbox__system_id
outbox__system_space_id outbox__client_id
category__dataset__contragent_owner
outbox__system__name outbox__system_id
outbox__system__contragent_owner segment_id

Поиск

id name description segment_id client_id
category__dataset__name category__pathname
outbox__name

Сортировка

pk name description state price category dataset
client_id segment_id begin end category__state
category__category_type category__dataset_id
category__dataset__name category__pathname
category__reach dataset__name

Запрос создания

/api/categories/

Методы

GET POST PATCH DELETE OPTIONS

Поля

pk name description create_time modify_time groups licenses
source_categories dataset_name dataset category_type ttl price
quick_process expired_date currency deep audience_rule raw_rule
stream_rule lal_rule keywords_rule urlmask_rule urls_rule urls_part_rule
parent pathname reach reach_timestamp reach_meta parents
audience_descendants has_children expired can_edit state pk

Фильтры

pk dataset_contragents dataset_contragent_owner contragents deep reach_timestamp
dataset_rule parent licenses groups source_categories expired_date_range
archiving_date_range create_time_range modify_time_range reach_range
price_range rule_segmentationaudiencerule_is_direct_link
rule_segmentationstreamrule_is_direct_tag rule_keywordrule_approved
rule_urlrule_approved rule_urlmaskrule_approved rule_urlpartrule_approved
dataset_profilesdataset_fill_with_batches quick_process archiving_date
expired_date reach rule_keywordrule_urls rule_urlrule_urls
rule_urlpartrule_urls category_type state rule_lalrule_rule_state

Поиск

id name description pathname dataset_name

Сортировка

pk name state description dataset_rule reach pathname category_type
dataset_name create_time expired_date

Запрос создания

/api/contragents/

Методы

GET POST PATCH DELETE OPTIONS

Поля

name contragent_systems contragent_owner_systems
description users own_reports_datasets
own_raw_datasets own_profiles_datasets
shared_reports_datasets shared_raw_datasets
shared_profiles_datasets own_codes limit logo
use_logo admin_mails state pk

Фильтры

pk name

Запрос создания

Служебные точки

Auth

ВИДНЫ ВСЕМ ИЗВНЕ

/o/authorize/

- auth url

/o/token/

- token url

/o/revoke_token/

- удаление токена

Password management

ВИДНЫ ВСЕМ ИЗВНЕ

/password_reset/

Форма для ввода реквизитов пользователя

/reset/

сюда переходят из письма

Администрирование

/admin/

/o/applications

список приложений пользователя

/o/authorized_tokens/

СПИСОК ТОКЕНОВ С ВОЗМОЖНОСТЬЮ ОТОЗВАТЬ
ТОКЕН

Кейсы

1. Управление сессией -> Авторизация

Решение должно поддерживать авторизацию системных пользователей для обеспечения интеграций.

Должна быть реализована аутентификация по защищенному соединению.

Интеграции

Решение должно обеспечивать возможность управления интеграциями системными пользователями.

Https + CSRF + CORS + Oauth

Решение должно поддерживать по-пользовательскую авторизацию и разграничение прав системных пользователей от непривилегированных.

Разграничение пользователей

Решение должно поддерживать по-пользовательскую авторизацию и разграничение прав системных пользователей от непривилегированных.

2. Управление целевыми группами -> Управление "категориями"

Решение должно предоставлять API для создания/удаления кастомных К, сформированных на основании уже существующих К с применением логических правил и правил частоты/глубины.

3. Управления сегментами -> Получение "сегментов"

Выгрузка

Решение должно предоставлять механизмы получения "данных о пользователях, входящих в категорию" ("сегмента").

Поиск

Решение должно предоставлять возможность поиска и получения информации категорий.

4. Управление контрагентами -> Получения "профиля"

Решение должно предоставлять механизмы получения "набора категорий на основании входящего в них идентификатора пользователя" ("профиля пользователя").