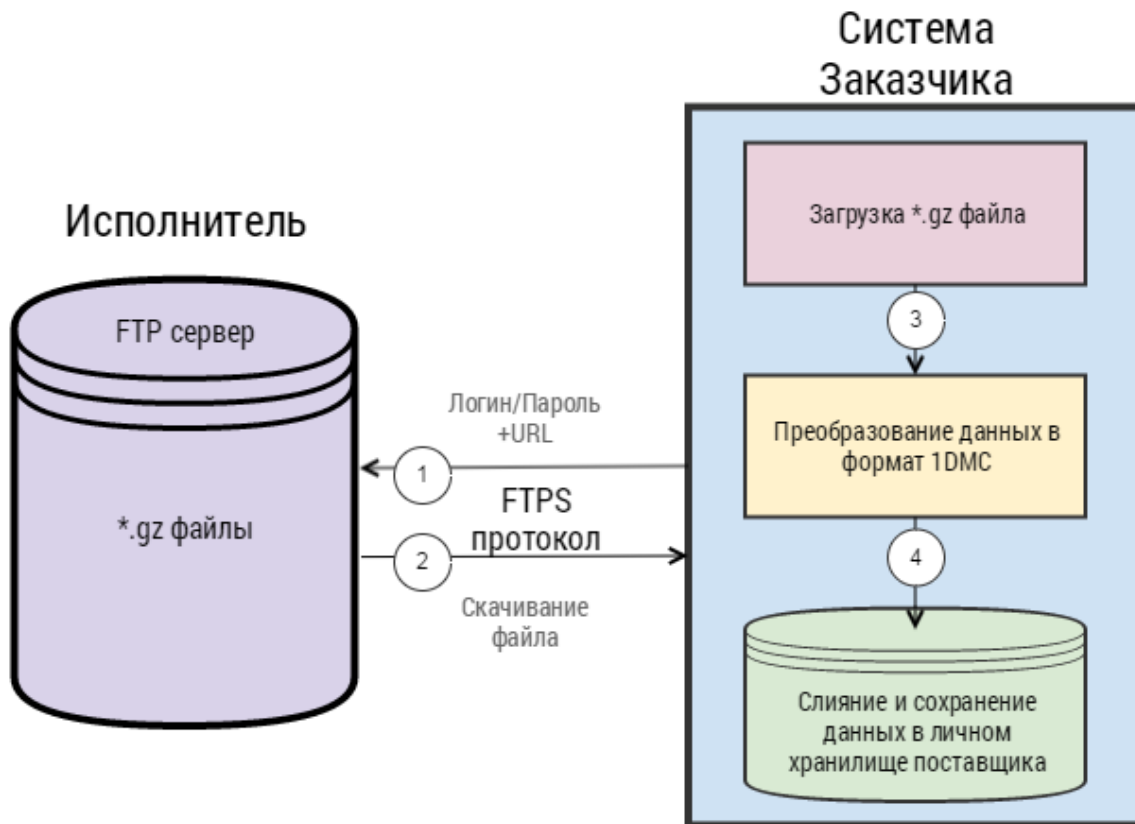


Типовые протоколы передачи данных

1. Типовой протокол передачи данных от Исполнителей (Поставщиков данных)

Процесс поставки данных от Исполнителя (Поставщика) в Систему Заказчика (1DMC) осуществляется посредством выкладки текстового файла, содержащего профили, в виде gzip архива на FTP сервере.

Процесс информационного взаимодействия Исполнителя (Поставщика) и Системы Заказчика (1DMC)



Первичная загрузка данных из хранилища Исполнителя (Поставщика) в Систему Заказчика (1DMC)

Обязанности Исполнителя:

1. Сформировать один или несколько текстовых файлов с кодировкой UTF-8, содержащие описания профилей. Каждая строка содержит JSON с идентификатором

профиля (ID) и его атрибутами или сегментами, к которым этот профиль принадлежит. Идентификаторы назначаются поставщиком самостоятельно.

Схема JSON строки:

```
{
  "id": [{"code": "string", "value": "string"}],
  "attributes": [{"code": "string"}, {"code": "string", "value": "string"}, {"code": "string", "value": "string", "conf": 0.8}]
}
```

Параметр	Тип	Описание
id	Массив	Идентификаторы профиля (сейчас содержит один элемент)
id[].code	Строка	Код идентификатора поля
id[].value	Строка	Значение идентификатора профиля
attributes	Массив	Список атрибутов профиля
attributes[].code	Строка	Код атрибута
attributes[].value	Строка	Значение атрибута (опциональное, может опускаться)
attributes[].conf	Float	Вероятность истинности значения, принимает значения от 0.0 до 1.0 (опциональное)

2. Заpackовать подготовленные файлы в gzip архив и выложить на FTP сервер.
3. Обеспечить IDMC доступ к FTP серверу. Предоставить логин и пароль для подключения.
4. Сообщить URL для скачивания файла данных в IDMC. Формат URL: \\[domain]\\[path]\\[files].gz.
5. Предоставить описание таксономии в свободной форме для возможности преобразования данных в формат IDMC.

Регулярная загрузка данных из хранилища Исполнителя (поставщика) в Систему Заказчика (IDMC)

Обязанности Исполнителя:

1. Ежедневно формировать текстовый файл с кодировкой UTF-8, содержащий измененные профили за последние 24 часа или все профили. Каждая строка содержит JSON с идентификатором профиля (ID) и его атрибутами или сегментами, к которым этот профиль принадлежит. Идентификаторы назначаются поставщиком самостоятельно.
2. Заpackовывать подготовленный файл с измененными профилями в gzip архив, имя которого указывает дату изменения профилей и имеет следующий формат: YYYY-MM-DD.gz, и выкладывать на FTP сервер.
3. Обеспечивать IDMC доступ к FTP серверу. Предоставить логин и пароль для подключения.

4. Сообщить URL для скачивания файла данных в 1DMS. Формат URL: \\[domain]\[path]\[YYYY-MM-DD].gz.
5. Обеспечивать хранение архивов с измененными профилями за последние две недели.

Пример файла с профилями для загрузки

```
{ "id": [{"code": "10005", "value": "537688512326"}], "attributes": [{"code": "106"}, {"code": "107"}, {"code": "114"}, {"code": "120", "value": "ABC"}] \n
{ "id": [{"code": "10005", "value": "538122059361"}], "attributes": [{"code": "106"}, {"code": "108"}, {"code": "109"}, {"code": "111"}] \n
{ "id": [{"code": "10005", "value": "538028514437"}], "attributes": [{"code": "111"}, {"code": "112"}, {"code": "120", "value": "ABC"}] \n
{ "id": [{"code": "10005", "value": "603784626612"}], "attributes": [{"code": "120", "value": "ABC"}] \n
```

2. Типовой протокол выгрузки данных (из Системы Исполнителя)

Сервис позволяет Потребителям самостоятельно выбрать нужные данные из Системы Исполнителя (1DMS), а также задать необходимые сложные правила выборки пользователей и получить результаты сделки в Личном кабинете или через REST API единоразово или на периодической основе.

Профилирование позволяет Потребителям расширить свою существующую аудиторию после ее анализа и добавления новых атрибутов, купленных у поставщиков.

2.1. Создание и исполнение сделок

Для создания и запуска сделок через REST API используйте следующие методы:

- Авторизация
- Список поставщиков
- Создание сделки
- Запуск сделки

Авторизация

Метод для авторизации, в ответе возвращается токен, который нужно передавать во всех последующих методах в HTTP-заголовке. Для вызова метода нужно передать:

- логин и пароль от Личного кабинета в GET-параметрах username и password соответственно
- Basic-авторизацию (логин и пароль для Basic-авторизации выдаются по запросу администратором Биржи)

Запрос:

```
POST
https://auth_user:auth_pass@api.1dms.io/v1/oauth/token?grant_type=password&username=my_user&password=my_pass
```

Здесь, `my_user` и `my_pass` - соответственно логин и пароль от Личного кабинета, `auth_user` и `auth_pass` - логин и пароль для Basic-авторизации (выдаются по запросу администратором Биржи)

Ответ:

```
{"access_token":"e6be553b-9d49-43c6-8315-81fd8f962c73","token_type":"bearer","refresh_token":"c4533f30-da54-4e7f-a298-3f8d83be60e5","expires_in":3599,"scope":"read write"}
```

Токен **e6be553b-9d49-43c6-8315-81fd8f962c73** нужно использовать в последующих методах в HTTP-заголовке следующим образом:

Пример HTTP-заголовка с авторизационным токеном:

```
Authorization: bearer e6be553b-9d49-43c6-8315-81fd8f962c73
```

Для стабильной работы по протоколу получения профилей в реальном времени, рекомендуется получать токен не на каждый запрос, а по мере истечения срока его действия. Когда срок действия токена заканчивается, его можно получить заново с помощью специального метода обновления токена. В этом методе, в параметр `refresh_token` нужно передать токен, полученный на этапе авторизации.

Пример запроса:

```
https://api.1dmc.io/v1/oauth/token?client_id={your_client_id}&grant_type=refresh_token&refresh_token={your_refresh_token}
```

Пример ответа:

```
{"access_token":"2df50ab0-88e1-4445-aec9-1fd25e56719b","token_type":"bearer","refresh_token":"fdf08a02-2464-40ca-b9e8-f90e72694503","expires_in":3599,"scope":"read write"}
```

Список поставщиков

Метод, который возвращает список доступных поставщиков. Не забывайте в HTTP-заголовке передавать токен, полученный при авторизации.

Запрос:

```
GET https://api.1dmc.io/v1/customers
```

Ответ:

```
[{
  "id": "4424a091-aa08-4ff9-93fe-47ab09fe8b22",
  "name": "alpha",
  "description": "Alpha",
  "created": 1455548825672,
  "roles": ["ROLE_PROVIDER"]
}, {
  "id": "d618333e-c6ce-4d0d-9c4d-8244eab98cf2",
  "name": "beta",
```

```

    "description": "Beta",
    "created": 1455548825705,
    "roles": ["ROLE_PLATFORM"]
  }, {
    "id": "f6310789-3df7-47f2-ab0e-efb5e5ebefa9",
    "name": "gamma",
    "description": "Gamma",
    "created": 1455548825722,
    "roles": ["ROLE_PROVIDER"]
  }
}

```

Идентификатор поставщика из поля `id` нужно использовать в дальнейшем при создании сделки.

Создание сделки

Метод для создания сделки. Не забывайте в HTTP-заголовке передавать токен, полученный при авторизации.

Запрос:

```
POST https://api.1dmc.io/v1/deals
```

Параметры тела запроса:

Параметр	Тип	Описание
<code>instrument</code>	string	Тип сделки: <ul style="list-style-type: none"> <code>INTEGRAL_PROFILE_ONE_OFF</code> - для одноразовой сделки. <code>INTEGRAL_PROFILE_PERIODIC</code> - для периодической сделки.
<code>name</code>	string	Название сделки.
<code>params</code>	object	Параметры сделки.
<code>params.billParams</code>	object	Параметры биллинга сделки (обязательный параметр)
<code>params.billParams.commissionParams</code>	object	Параметры комиссии Биржи данных: <ul style="list-style-type: none"> <code>type</code> - тип комиссии: <ul style="list-style-type: none"> <code>zero</code> - нулевая комиссия; <code>rate</code> - процент от суммы сделки; <code>fix_price</code> - фиксированный размер комиссии.

<p>params.billParams.dataPriceParams</p>	<p>object</p>	<p>Параметры цены сделки:</p> <ul style="list-style-type: none"> • unit - единица тарификации сделки. Принимает значения: <ul style="list-style-type: none"> ◦ <ul style="list-style-type: none"> ▪ attribute - тарификация по атрибутам; ▪ profile - тарификация по профилям; ▪ impression - тарификация по показам. • unitPrice - цена за единицу тарификации. <p>Если unit=impression, то цена указывается за 1000 показов (CPM). Например, при unitPrice=12, 1127 показов будет стоить $(1127/1000) * 12 = 13.524$.</p>
<p>params.execution</p>	<p>object</p>	<p>Настройки исполнения сделки:</p> <ul style="list-style-type: none"> • nextExecutionAfter - время, после которого сделка может начать исполняться. Реальный запуск может состояться не раньше этого времени. Отсутствие значения означает, что исполнения сделки не будет. • expireTime - время окончания выполнения сделки. <p>Настройки nextExecutionAfter и expireTime задаются сервером и пока не могут быть заданы клиентом.</p>
<p>parties</p>	<p>array</p>	<p>Массив из контрагентов сделки. Один из контрагентов является покупателем, второй - продавцом. Тип контрагента задается в параметре side.</p>
<p>parties[].side</p>	<p>string</p>	<p>Тип контрагента сделки (BUY - покупатель, SELL - продавец).</p>
<p>parties[].externalId</p>	<p>string</p>	<p>Идентификатор контрагента сделки (обязательный параметр). Для покупателя указывается собственный идентификатор, его можно узнать из Личного кабинета в разделе "Профиль", см. поле Client ID, или вызвав метод:</p> <pre>GET https://api.1dmc.io/v1/customers/self</pre> <p>Для поставщика указывается идентификатор выбранного поставщика, получен из предыдущего метода, возвращающего список поставщиков.</p>

parties[].params.condition.dataRule	string	<p>Условия для сегментирования аудитории поставщика (описание синтаксиса языка), параметр присутствует только у контрагента типа поставщик. Описание кодов атрибутов и их значений можно посмотреть в справочнике таксономии 1DMC. Справочник в Excel-формате предоставляется по запросу.</p> <p>Для получения справочника можно также использовать метод:</p> <pre>GET https://api.1dmc.io/v1/profile/attributes/ru/</pre>
parties[].params.attributes	array	Покупаемые атрибуты, возвращаются в результатах сделки.
state	string	<p>Параметр возвращается в ответе на запрос. Отражает текущее состояние сделки и может принимать следующие значения:</p> <ul style="list-style-type: none"> • UNREQUESTED - Сделка создана, но не запрошена у остальных участников. Конфигурацию сделки можно изменять. • REQUESTED - Сделка запрошена у всех участников. Конфигурацию сделки нельзя изменять. • AWAIT_EXECUTION - Все условия для исполнения сделки выполнены. Сделка ожидает исполнения. • EXECUTING - Сделка исполняется: как минимум одно исполнение сделки запущено в данный момент. • CANCELED - Сделка отменена. Более исполнения (deal execution) по этой сделке созданы не будут. • FINISHED - Сделка окончена. Более исполнения по этой сделке созданы не будут.

Пример тела запроса для создания единоразовой сделки:

```
{
  "instrument": "INTEGRAL_PROFILE_ONE_OFF",
  "name": "1st deal",
  "params": {
    "billParams": {
      "commissionParams": {
        "type": "rate",
        "value": 0.2
      },
      "dataPriceParams": {
        "unit": "attribute",
        "unitPrice": 42.35
      }
    }
  }
},
```

```
"parties": [{
  "externalId": "A",
  "side": "BUY"
},
{
  "externalId": "B",
  "params": {
    "attributes": [
      {
        "primary": 10001,
        "secondary": 10000
      },
      {
        "primary": 10002
      }
    ],
    "condition": {
      "dataRule": "exists(a[10001:10000]) AND (a[10004:-1]
contains (value > 100)) OR a[10008:10002] contains (value = 22)"
    }
  },
  "side": "SELL"
}
]
```

В ответе возвращается информация о созданной сделке. Сделка получает статус UNREQUESTED ("Не запрошена").

Пример ответа:

```
{
  "created": 1474447992670,
  "executions": [],
  "id": 230,
  "instrument": "INTEGRAL_PROFILE_ONE_OFF",
  "name": "1st deal",
  "params": {
    "billParams": {
      "commissionParams": {
        "type": "rate",
        "value": 0.2
      },
      "dataPriceParams": {
        "unit": "attribute",
        "unitPrice": 42.35
      }
    },
    "outputFormat": "cleverdata",
    "updateStorage": false
  },
  "parties": [
    {
      "approval": "REQUESTED",
      "externalId": "A",
      "id": 445,
```



```

        "params": {
            "mergeStrategy":
"ru.cleverdata.dmpkit.profile.integral.shared.job.merge.AddOnlyMergeStrategy
"
            },
            "side": "BUY"
        },
        {
            "approval": "UNREQUESTED",
            "externalId": "B",
            "id": 446,
            "params": {
                "attributes": [
                    {
                        "primary": 10001,
                        "secondary": 10000
                    },
                    {
                        "primary": 10002
                    }
                ],
                "condition": {
                    "dataRule": "exists(a[10001:10000]) AND (a[10004:-1]
contains (value > 100)) OR a[10008:10002] contains (value = 22)"
                }
            },
            "side": "SELL"
        }
    ],
    "state": "UNREQUESTED"
}

```

Пример тела запроса для создания периодической сделки:

```

{
    "instrument": "INTEGRAL_PROFILE_PERIODIC",
    "name": "1st deal",
    "params": {
        "billParams": {
            "commissionParams": {
                "type": "rate",
                "value": 0.2
            },
            "dataPriceParams": {
                "unit": "attribute",
                "unitPrice": 42.35
            }
        }
    },
    "parties": [{
        "externalId": "A",
        "side": "BUY"
    },
    {
        "externalId": "B",
        "params": {

```

```
        "attributes": [
            {
                "primary": 10001,
                "secondary": 10000
            },
            {
                "primary": 10002
            }
        ],
        "condition": {
            "dataRule": "exists(a[10001:10000]) AND (a[10004:-1]
contains (value > 100)) OR a[10008:10002] contains (value = 22)"
        }
    },
    "side": "SELL"
}
]
```

В ответе возвращается информация о созданной сделке. Сделка получает статус UNREQUESTED ("Не запрошена").

Пример ответа:

```
{
  "created": 1474447992670,
  "executions": [],
  "id": 230,
  "instrument": "INTEGRAL_PROFILE_PERIODIC",
  "name": "1st deal",
  "params": {
    "billParams": {
      "commissionParams": {
        "type": "rate",
        "value": 0.2
      },
      "dataPriceParams": {
        "unit": "attribute",
        "unitPrice": 42.35
      }
    },
    "execution": {
      "expireTime": 1475269199000 // Параметр выставляется в последнюю
секунду текущего месяца.
    },
    "outputFormat": "cleverdata",
    "updateStorage": false
  },
  "parties": [
    {
      "approval": "REQUESTED",
      "externalId": "A",
      "id": 445,
      "params": {
```

```
        "mergeStrategy":  
"ru.cleverdata.dmpkit.profile.integral.shared.job.merge.AddOnlyMergeStrategy"  
"  
        },  
        "side": "BUY"  
    },  
    {  
        "approval": "UNREQUESTED",  
        "externalId": "B",  
        "id": 446,  
        "params": {  
            "attributes": [  
                {  
                    "primary": 10001,  
                    "secondary": 10000  
                },  
                {  
                    "primary": 10002  
                }  
            ],  
            "condition": {  
                "dataRule": "exists(a[10001:10000]) AND (a[10004:-1]  
contains (value > 100)) OR a[10008:10002] contains (value = 22)"  
            }  
        },  
        "side": "SELL"  
    }  
],  
"state": "UNREQUESTED"  
}
```

Идентификатор контрагента с типом BUY (в примере - 445) необходимо использовать в дальнейшем в методе подтверждения сделки.

Подтверждение сделки

Метод для отправки поставщику запроса на подтверждение сделки. В строке запроса нужно указать идентификатор сделки и идентификатор контрагента (с типом BUY), полученный на этапе создания сделки.

Запрос:

```
PUT https://api.1dmc.io/v1/deals/{dealId}/parties/{partyId}
```

Тело запроса:

```
{"approval":"APPROVED", "side":"BUY", "externalId":"A"}
```

Пример ответа:

```
{  
    "id": 131,  
    "externalId": "A",  
    "side": "BUY",  
    "approval": "APPROVED",  
    "params": {
```

```

    "mergeStrategy":
"ru.cleverdata.dmpkit.profile.integral.shared.job.merge.AddOnlyMergeStrategy
"
    }
}

```

Проверка статуса исполнения сделки

После отправки сделки на исполнение нужно следить за ее завершением. Для этого используется метод, который возвращает список запусков сделки. Этот метод также необходим для получения идентификатора запуска (exec_id), который нам понадобится далее при получении результатов сделки.

Запрос:

```
GET https://api.1dmc.io/v1/deals/{dealId}/executions
```

Ответ:

```

[{"id":199,"state":"SUCCEEDED","params":{"buy":{"id":"f7a20cd8-7933-421a-963c-3313d28212d4"},"mergeStrategy":"ru.cleverdata.dmpkit.profile.integral.shared.job.merge.AddOnlyMergeStrategy"},"sell":{"parties":[{"id":"4424a091-aa08-4ff9-93fe-47ab09fe8b22"},"condition":{"dataRule":"EXISTS(a[10045:10001]) AND EXISTS(a[10211:10643]) AND a[10210:-1] CONTAINS (value >= 60000) AND (EXISTS(a[10056:10068]) OR EXISTS(a[10056:10067]))"},"attributes":[{"primary":10238,"secondary":null}, {"primary":10056,"secondary":null}, {"primary":10211,"secondary":null}, {"primary":10210,"secondary":null}]}]}]},"created":1468393518409,"updated":1468394208624}]

```

Здесь нас интересуют два параметра: идентификатор запуска сделки (параметр id) и статус запуска сделки (параметр state). На примере выше - запуск сделки имеет идентификатор 199, статус запуска - SUCCEEDED. Ниже представлен справочник возможных статусов. Необходимо периодически опрашивать статус запуска сделки до тех пор, пока он не получит один из финальных статусов (любой статус, кроме RUNNING).

Справочник статусов исполнения сделки

Статус	Описание
CANCELED	Сделка была отменена поставщиком или потребителем. Происходит корректная остановка выполнения сделки. Все обрабатываемые данные должны быть доставлены потребителю.
FAILED	Доставка данных не удалась, в результате чего сделка не выполнялась.
RUNNING	Доставка данных в процессе.
SUCCEEDED	Доставка данных от всех поставщиков успешно завершена.

2.2. Получение результатов

Результаты сделки можно загрузить программно, используя REST API Биржи. Вам потребуются три метода: авторизация, просмотр каталогов и загрузка файла. Метод авторизации был описан выше, если у вас уже есть авторизационный токен, метод авторизации вам не потребуется.

Просмотр каталога

Метод для просмотра содержимого каталога, в ответе возвращается список подкаталогов и файлов в каталоге. Не забывайте передавать токен в HTTP-заголовке.

Запрос:

```
GET
https://api.1dmc.io/v1/browser/deals/{dealId}/{execId}/cleverdata?op=LIST
```

Ответ:

```
{
  "nav": {
    "offset": 0,
    "limit": 100,
    "length": 2
  },
  "filter": {},
  "meta": {},
  "items": [{
    "path": "deals",
    "name": "101",
    "size": 0,
    "lastModified": 1465467617871,
    "directory": true
  }, {
    "path": "deals",
    "name": "102",
    "size": 0,
    "lastModified": 1465989024258,
    "directory": true
  }
]}
```

Загрузка результатов сделки

Метод для загрузки файлов с результатом сделки. Обратите внимание, в этом случае `access_token` передается как GET-параметр, а не HTTP-заголовок.

Запрос:

```
GET https://api.1dmc.io/v1/browser/deals/{dealId}/{execId}/cleverdata/part-
r-00001.gz?op=DOWNLOAD&access_token={access_token}
```

В ответе возвращается содержимое файла. Для загрузки результатов сделки необходимо сначала получить список файлов в каталоге (см. предыдущий метод), а затем

последовательно загрузить все файлы из списка. Файлы будут называться part-r-00000.gz, part-r-00001.gz, part-r-00002.gz и так далее.

В ответе возвращается содержимое результатов сделки.

2.3. Формат файла с результатом сделки

Результатом сделки является один или несколько Gzip-архивов. В архиве находится текстовый файл, в каждой строке файла находится интегральный профиль в JSON-формате. Пример строки файла:

```
{"id":[{"primary":10005,"secondary":null,"value":"65AE0957D07ABDF3C65B597E5F223772","confidence":1}], "attributes":[{"primary":10238,"secondary":10005,"value":true,"confidence":1}, {"primary":10238,"secondary":10006,"value":true,"confidence":1}, {"primary":10210,"secondary":null,"value":30000,"confidence":1}, {"primary":10211,"secondary":10643,"value":true,"confidence":1}]}
```

Тот же профиль, в beautified формате:

```
{
  "id": [
    {
      "primary": 10005, "secondary": null, "value": "65AE0957D07ABDF3C65B597E5F223772", "confidence": 1
    }
  ],
  "attributes": [
    {
      "primary": 10238, "secondary": 10005, "value": true, "confidence": 1
    },
    {
      "primary": 10238, "secondary": 10006, "value": true, "confidence": 1
    },
    {
      "primary": 10210, "secondary": null, "value": 50000, "confidence": 1
    },
    {
      "primary": 10211, "secondary": 10643, "value": true, "confidence": 1
    }
  ]
}
```

JSON содержит две группы - "id" и "attributes". Группа id содержит идентификаторы, в общем случае здесь находится один атрибут с кодом 10005 и со значением идентификатора покупателя. Группа "attributes" содержит сами атрибуты с данными.

Каждый атрибут включает в себя следующие поля:

- *primary* - код атрибута первого уровня, из таксономии CleverDATA.
- *secondary* - код атрибута второго уровня, из таксономии CleverDATA. Поле может опускаться.
- *value* - значение атрибута, для enum атрибутов содержит значение true, для остальных типов - значение соответствующего типа.
- *confidence* - уверенность или качество данных, значение от 0 до 1 (на текущий момент всегда заполнено 1.0).