

**Интеграция с
Data integration service(DIS)
MDMP for Vendor**

Процессы получения данных от партнёров сопряжены с процессами сетевого взаимодействия(получение, отправка данных). Продукт **App EC OMNICORE** для интеграции использует сервисы которые взаимодействуют с сервисами заказчика и занимаются трансляцией либо проксированием данных, запросов, а также хранением и обменом документами.

Основные интеграционные сервисы:

1. **Data integration service(DIS)** - сервис получения данных
2. **Document exchange service(DES)** - сервис обмена документами

Сетевое взаимодействие между сервисами и **App EC OMNICORE** происходит по протоколу **HTTPS**, для получения/передачи данных используется **REST API**. Сервисы поддерживают формат передачи данных **JSON**.

Для интеграции партнеру необходимо подготовить ряд методов для **DIS**, которые позволят **App EC OMNICORE** получать данные(справочники, номенклатуру).

Data integration service(DIS)

Для сборки самодостаточных единиц номенклатуры на стороне **App EC OMNICORE** необходимо получение справочников. Получение справочников происходит по **REST API**. Набор справочников для генерации и сопровождения товара, торгового предложения на стороне **App EC OMNICORE**:

- Цвета
- Категории(optional)
- Типы продуктов(optional)
- Сезоны
- Размеры
- Материалы(optional)
- Магазины(склады)
- Типы половой принадлежности
- Бренды

Справочник цветов:

Method GET url: {host}/v1/{location}/colors

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response:

```
[  
  {  
    "id": "c701e3f8-f6dc-11e7-80f6-000c297fcd5f",  
    "value": "Чорний"  
  },  
  ...  
]
```

Справочник категорий(optional):

Method GET url: {host}/v1/{location}/categories

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response:

```
[
  {
    "id": "3522d7d4-7ff9-11e6-9540-000c296c9139",
    "value": "Одежда"
  },
  ...
]
```

Справочник типов продуктов(optional):

Method GET url: {host}/v1/{location}/product-types

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства

value	string	значение свойства
-------	--------	-------------------

Response:

```
[
  {
    "id": "3522d7d4-7ff9-11e6-9540-000c296c9139",
    "value": "Джинсы"
  },
  ...
]
```

Справочник сезонов:

Method GET url: {host}/v1/{location}/seasons

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response:

```
[
  {
    "id": "0201690e-fe07-11e4-85a2-000c29177ad9",
    "value": "SS18"
  },
  ...
]
```

Справочник размеров:

Method GET url: {host}/v1/{location}/sizes

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response:

```
[  
  {  
    "id": "bc194517-d423-11e7-80f1-000c29f274dd",  
    "value": "XS"  
  },  
  ...  
]
```

Справочник материалов(optional):

Method GET url: {host}/v1/{location}/materials

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства

value	string	значение свойства
-------	--------	-------------------

Response:

```
[
  {
    "id": "1765020c-4527-11e2-9bb2-1c6f6540240a",
    "value": "100% акрил"
  },
  ...
]
```

Справочник магазинов:

Method GET url: {host}/v1/{location}/stores

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
name	string	название
city	string	город
address	string	адрес
phone	string	телефон
email	string	e-mail

Response:

```
[
  {
    "id": "3ad57bec-8d44-11e1-80b8-1c6f6540240a",
    "name": "ТЦ \"Плазма\" (пр. С.Бандери)",

```

```
"city": "Київ",
"address": "м.Київ, пр-т Степана Бандери, 20-б",
"phone": "+38 (044) 485-00-88",
"email": ""
},
...
]
```

Справочник полов:

Method GET url: {host}/v1/{location}/genders

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response example:

```
[
{
  "id": "363c6080-fd18-11e7-80fa-000c297bd59f",
  "value": "women"
},
...
]
```


Справочник брендов:

Method GET url: {host}/v1/{location}/brands

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Type	Description
id	string	uuid свойства
value	string	значение свойства

Response example:

```
[
  {
    "id": "363c6080-fd18-11e7-80fa-000c297bd59f",
    "value": "Oodji"
  },
  ...
]
```

Получение продуктов:

Method GET url: {host}/v1/{location}/products

Parameters:

Name	Description
host	IP, domain, port
location	path to method location

Response Fields:

Name	Description
id	uuid

name	наименование
article	артикул
brandId	бренд(uuid)
genderId	пол(uuid)
categoryId(optional)	категория(uuid)
typeId(optional)	тип продукта(uuid)
compositionId(optional)	состав(uuid)
seasonId	сезон(uuid)
skus.id	uuid торгового предложения
skus.sizeId	размер торгового предложения(uuid)
skus.colorId	цвет торгового предложения(uuid)
skus.barcode	штрихкод торгового предложения

Response:

```
[
{
  "id": "cf2f5911-8e0d-11e1-80b8-1c6f6540240a",
  "name": "Шкарпетки 3 пара Sneaker sock ",
  "article": "3.20.00001",
  "brandId": "00821781-6653-11e3-9507-000c29177ad9",
  "genderId": "dfbc2fd5-18d0-11e0-85ab-1c6f6540240a",
  "categoryId": "53dd0b39-0c98-11e2-8d10-1c6f6540240a",
  "typeId": "d764202e-ec87-11e7-80f6-000c297fcd5f",
  "compositionId": "cf2f5919-8e0d-11e1-80b8-1c6f6540240a",
  "seasonId": "a4c80954-9342-11e0-8248-1c6f6540240a",
  "skus": [
    {
      "id": "cf2f5916-8e0d-11e1-80b8-1c6f6540240a",
      "sizeId": "f904437b-d423-11e7-80f1-000c29f274dd",
      "colorId": "00000000-0000-0000-0000-000000000000",
      "barcode": "8715344127420"
    },
    ...
  ]
}
```

```
},  
...  
]
```

Получение цен:

Method POST url: {host}/{db}/hs/ref/price

Parameters:

Name	Description
host	IP, domain, port
location	path to method location
skus	array sku ids

Request:

```
{  
  "skus":  
  [  
    "5c8f53af-38e9-11e9-812b-bdad5ac4ac0a",  
    "5c8f53af-38e9-11e9-812b-bdad5ac4ac47",  
    "5c8f53af-38e9-11e9-812b-bdad5ac4ac11",  
    ...  
  ]  
}
```

Response Fields:

Name	Type	Description
id	string	uuid
barcode	string	штрихкод
firstPrice	Int	розничная цена
salePrice	Int	цена со скидкой (отпускная)
discountPercentage	Int	номинал скидки

Response:

```
[
  {
    "id": "2a1155eb-b1d0-11e9-ba99-0050568ddb4",
    "barcode": "8433997772148",
    "firstPrice": 2990,
    "salePrice": 2990,
    "discountPercentage": 0
  },
  ...
]
```

Получение остатков:

Method POST url: {host}/{db}/hs/ref/quantity

Parameters:

Name	Description
host	IP, domain, port
db	database name
store	uuid магазина, может быть пустым при заполненном skus
skus	optional - array sku ids, может быть пустым. При заполнении отдает остатки запрашиваемых торговых предложений во всей сети

Request:

```
{
  "store": "d18e0a9b-29f2-11e9-812b-e29f7d4c5931",
  "skus":
  [

  ]
}
```

Response Fields:

Name	Type	Description
id	string	uuid
barcode	string	штрихкод
stores.id	string	uuid магазина
stores.quantity	Int	количество торговых предложений

Response:

```
[
  {
    "id": "410712d7-d38c-11e8-8129-82cf444d6142",
    "barcode": "8718655542500",
    "stores": [
      {
        "store_id": "d18e0a9b-29f2-11e9-812b-e29f7d4c5931",
        "quantity": 2
      }
    ]
  },
  ...
]
```

Получение данных происходит по регламенту. Справочники и продукты обновляются раз в день. Продукты для получения **DIS** ставятся в очередь по признакам:

- новый
- модификация

Те продукты которые были ранее переданы в **DIS** не принимают участие в последующих обменах.

Получение остатков происходит каждые 2 часа. Регламенты получения данных могут меняться в зависимости от интенсивности работы.

Сервис **DIS** для интеграции с партнёром использует **Driver**, транслирует в **App EC OMNICORE** с помощью **SDK**.