



---

# ΠΛΑΤΦΟΡΜΑ LENNUF

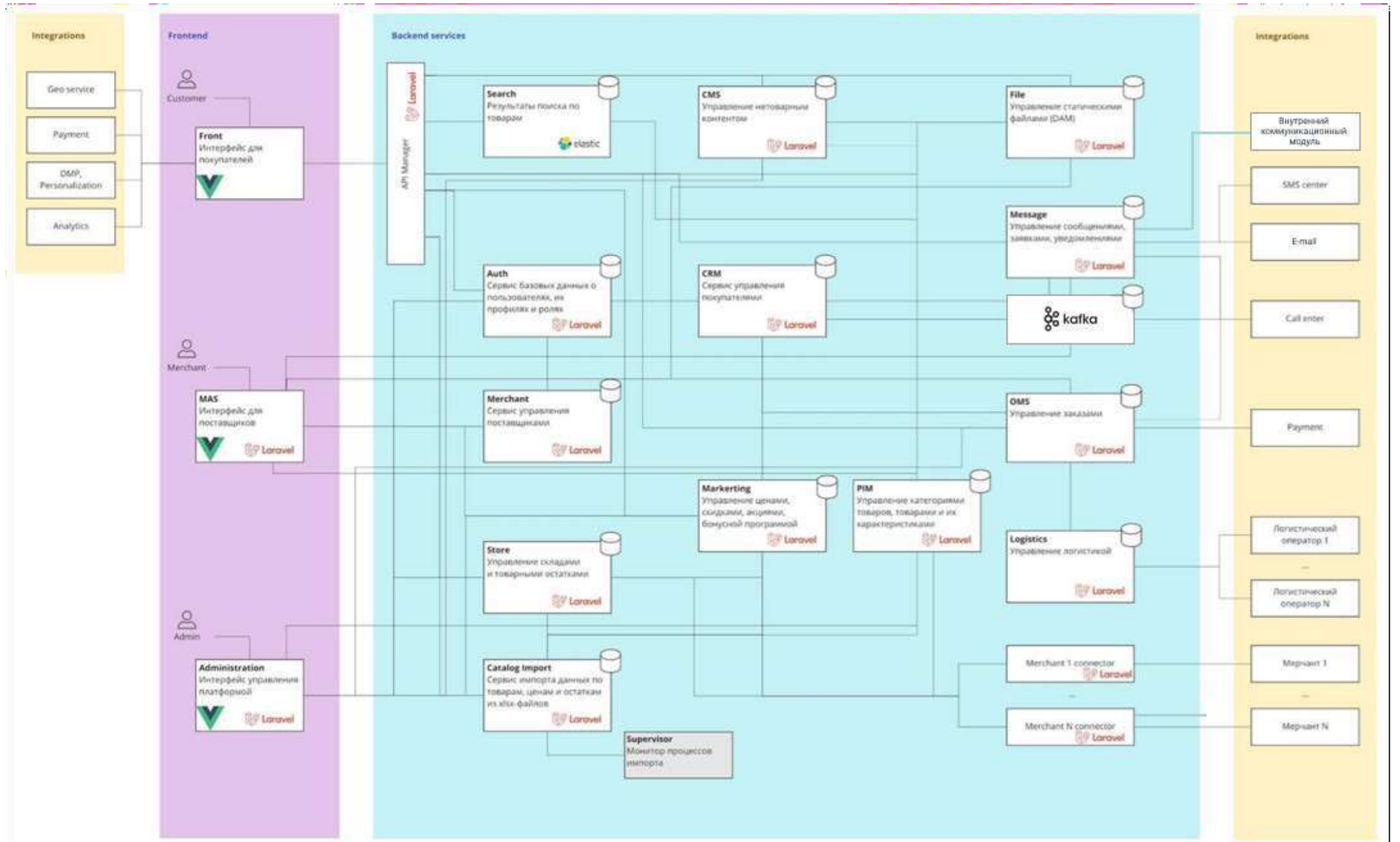
# СОДЕРЖАНИЕ

1. Архитектура платформы LENNUF.
2. Библиотеки сервисов LENNUF.
3. Библиотека CommonLib.
4. Кратко о каждом сервисе LENNUF.
5. Технологии реализации MAS и Admin.

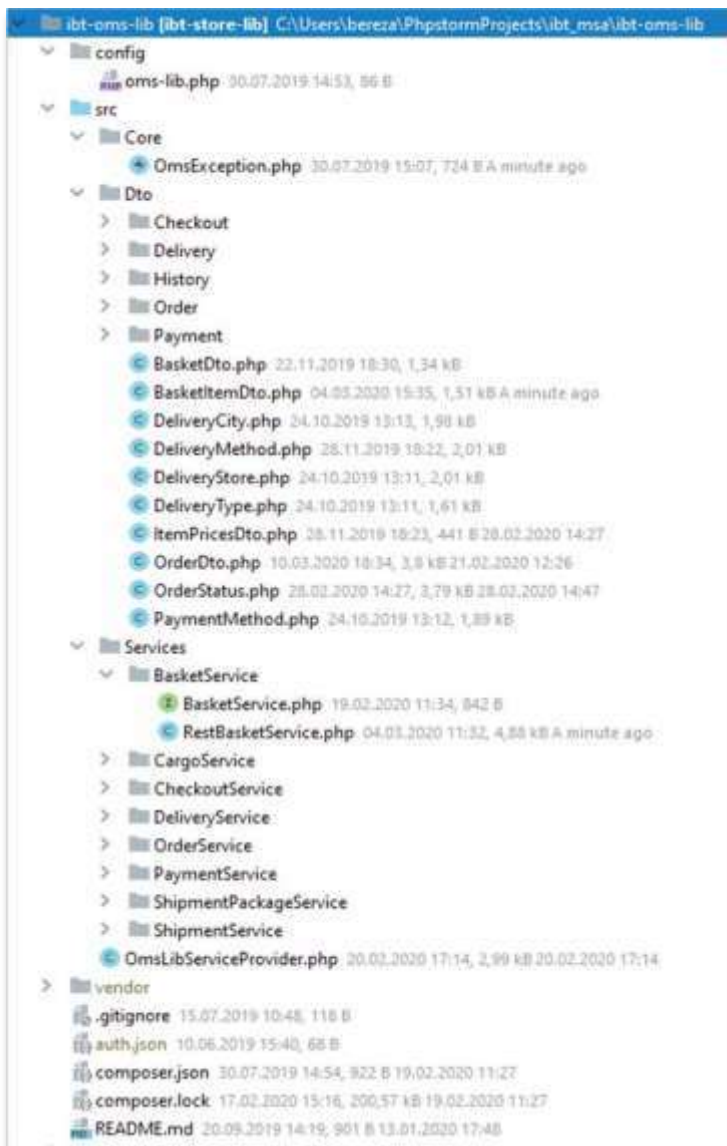


# Архитектура Платформы

---



# БИБЛИОТЕКИ СЕРВИСОВ ПЛАТФОРМЫ LENNUF



Это composer-библиотеки со следующей общей структурой:

- config/<service>-lib.php - конфигурационный файл
- src
  - Core/<Service>Exception.php - класс для исключений ответов REST API сервиса
  - Dto - классы с DTO библиотеки
  - Services - классы с сервис-провайдерами библиотеки, каждый метод которого обычно реализует работу с одним endpoint'ом REST API сервиса
  - <Service>LibServiceProvider.php - сервис-провайдер библиотеки, подключаемый в config/app.php сервиса-клиента
- .gitignore
- composer.json
- composer.lock
- README.md - справка по библиотеке

# БИБЛИОТЕКИ СЕРВИСОВ ПЛАТФОРМЫ LENNUF

OrderDto.php

```
11 /**
12  * //todo Перенести в namespace Order
13  * DTO для сущности "Заказы"
14  * Class OrderDto
15  * @package Greensight\Oms\Dto
16  *
17  * @property int $customer_id - id покупателя
18  * @property int $basket_id - id корзины
19  *
20  * @property float $cost - стоимость (расчитывается автоматически)
21  * @property float $price
22  * @property int $spent_bonus
23  * @property int $added_bonus
24  * @property string $promocode
25  * @property int $certificate
26  *
27  * @property int $delivery_type - тип доставки (одним отправлением, несколькими отправлениями)
28  * @property float $delivery_cost - стоимость доставки (BT)
29  * @property string $delivery_comment - комментарий к доставке
30  *
31  * @property int $status - статус
32  * @property Carbon|null $status_at - дата установки статуса заказа
33  * @property int $payment_status - статус оплаты
34  * @property Carbon|null $payment_status_at - дата установки статуса оплаты
35  * @property int $is_problem - флаг, что заказ проблемный
36  * @property Carbon|null $is_problem_at - дата установки флага проблемного заказа
37  * @property int $is_canceled - флаг, что заказ отменен
38  * @property Carbon|null $is_canceled_at - дата установки флага отмены заказа
39  * @property int $is_require_check - флаг, что заказ требует проверки
40  * @property string $manager_comment - комментарий менеджера
41  *
42  * @property string $number - номер
43  *
44  *
45  * @property-read Collection|DeliveryDto[] $deliveries
46  */
47 class OrderDto extends AbstractDto
48 {
49     /** @var int - не оплачено */
50     public const PAYMENT_STATUS_NOT_PAID = 1;
51     /** @var int - оплачено */
52     public const PAYMENT_STATUS_PAID = 2;
53     /** @var int - просрочено */
```

OrderDto.php

RestOrderService.php

```
39 /**
40  * @inheritDoc
41  * @return RestQuery
42  */
43 public function newQuery(): DataQuery
44 {
45     return new RestQuery();
46 }
47
48 //ЗАКАЗЫ
49 /**
50  * @inheritDoc
51  * @throws OmsException
52  * @throws Exception
53  */
54 public function orders(?DataQuery $dataQuery = null, ?int $merchantId = null): Collection
55 {
56     $dataQuery = $dataQuery ?? $this->newQuery();
57
58     /** @var RestQuery $dataQuery */
59     if ($merchantId) {
60         $dataQuery->setFilter('merchant_id', $merchantId);
61     }
62
63     try {
64         $response = $this->client->get('/api/v1/orders', $dataQuery->toQuery());
65     } catch (GuzzleException $e) {
66         throw new OmsException( message: 'rest client error', code: 0, $e);
67     }
68
69     return (new RestResult($response))
70         ->expectStatus(200)
71         ->unexpectedStatus(function (ResponseInterface $response) {
72             throw new OmsException( message: "get /api/v1/orders return code = {$response->getStatusCode()}");
73         })
74         ->wrapResult(OrderDto::class, true, 'items')
75         ->value(OmsException::class);
76 }
```

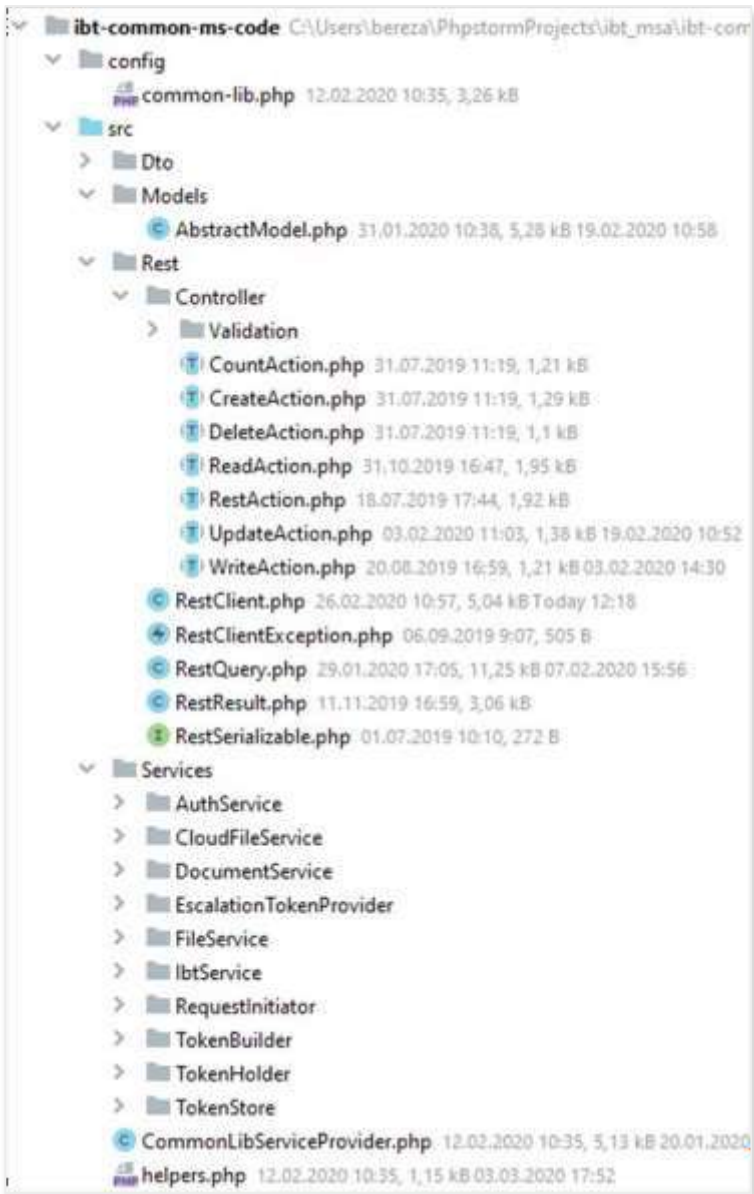
RestOrderService.php

# БИБЛИОТЕКИ СЕРВИСОВ ПЛАТФОРМЫ LENNUF

```
OmsLibServiceProvider.php
27 * @package Greensight\Store
28 */
29 class OmsLibServiceProvider extends ServiceProvider
30 {
31     public function boot()
32     {
33         $this->publishes([
34             __DIR__ . '/../config/oms-lib.php' => config_path( path: 'oms-lib.php'),
35         ]);
36         $this->mergeConfigFrom(
37             path: __DIR__ . '/../config/oms-lib.php', key: 'oms-lib'
38         );
39
40         $container = app();
41         $omsHost = config( key: 'oms-lib.omsHost');
42         if ($omsHost) {
43             $container->instance( abstract: 'oms-rest-client', new RestClient($omsHost, $container->make( abstract: TokenHolder::class)));
44         }
45
46         $container->when([
47             RestOrderService::class,
48             RestBasketService::class,
49             RestPaymentService::class,
50             RestDeliveryService::class,
51             RestShipmentService::class,
52             RestShipmentService::class,
53             RestShipmentPackageService::class,
54             RestCargoService::class,
55             RestCheckoutService::class,
56         ])
57         ->needs( abstract: RestClient::class)
58         ->give( implementation: 'oms-rest-client');
59
60         $container->singleton( abstract: OrderService::class, concrete: RestOrderService::class);
61         $container->singleton( abstract: BasketService::class, concrete: RestBasketService::class);
```

OmsLibServiceProvider.php

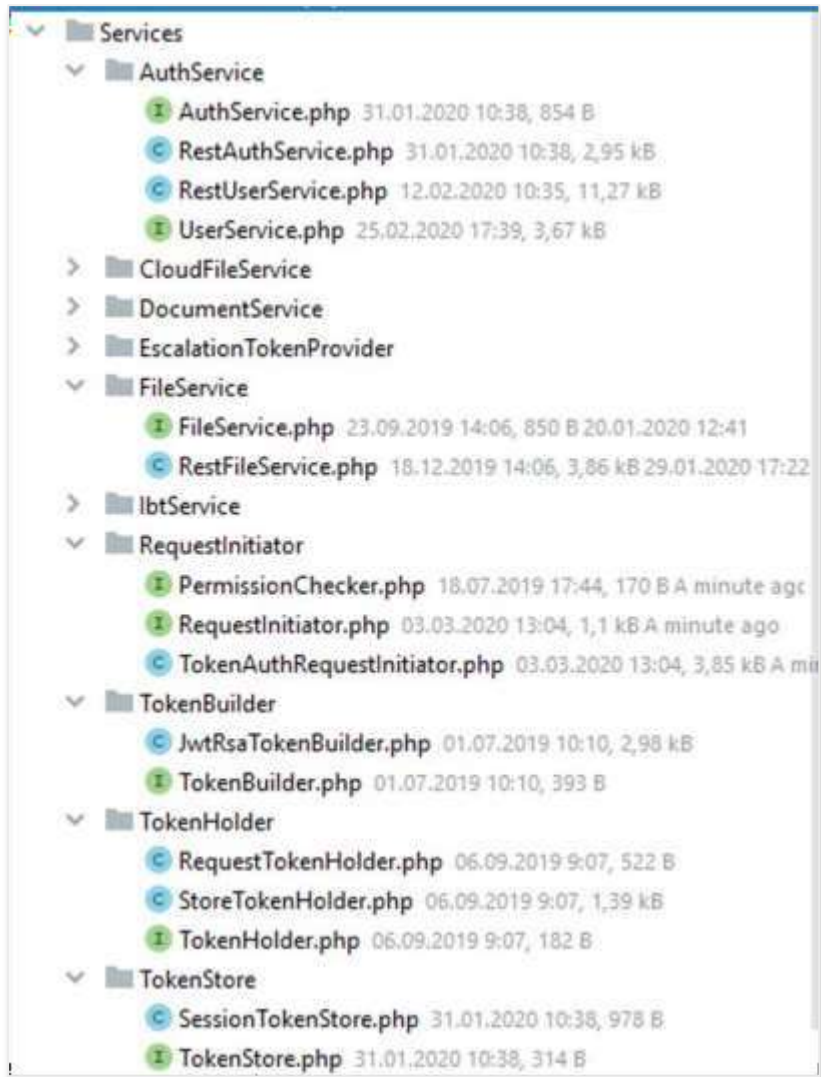
# БИБЛИОТЕКА CommonLib



- src
  - Models/AbstractModel.php - базовая модель, от которой наследуются почти все модели сервисов Lennuf
  - Rest
    - Controller - трейты для реализации CRUD-операций над сущностями сервисов
    - RestClient.php - единый Rest-клиент для всех сервисов Lennuf
    - RestQuery.php - построитель запросов в соответствие со спецификацией JSON:API <https://jsonapi.org/>
    - RestResult.php - класс обработки результатов от REST API сервисов



# БИБЛИОТЕКА CommonLib



- src
  - Services
    - AuthService - сервисы авторизации для фронтов
    - FileService - сервисы для работы с файлами
    - RequestInitiator - сервисы для работы с инициатором REST-запроса (получение user\_id, проверка прав доступа и т.д.)
    - TokenBuilder - сервисы для создания токена аутентификации REST-запросов
    - TokenHolder - сервисы для получения токена аутентификации REST-запросов
    - TokenStore - сервисы для хранения токена аутентификации REST-запросов

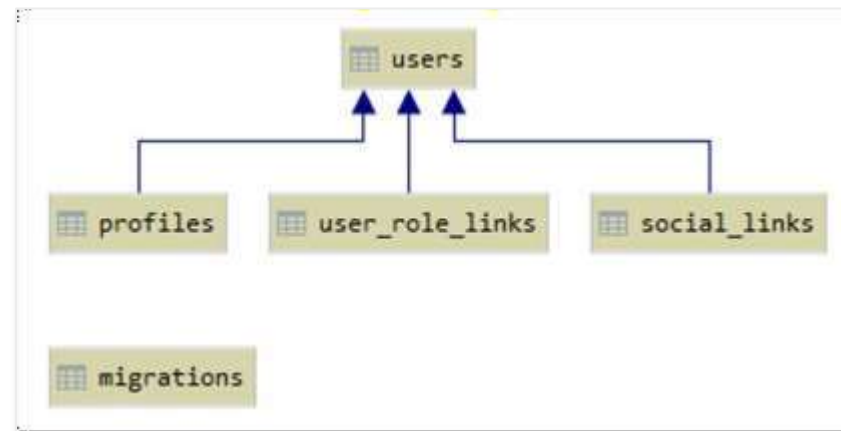
# СЕРВИС Auth



Назначение: управление общими данными всех пользователей платформы.

Основная функциональность:

- управление и авторизация пользователей;
- управление персональными данными пользователей;
- управление ролями пользователей;
- управление ссылками на социальные сети пользователей.



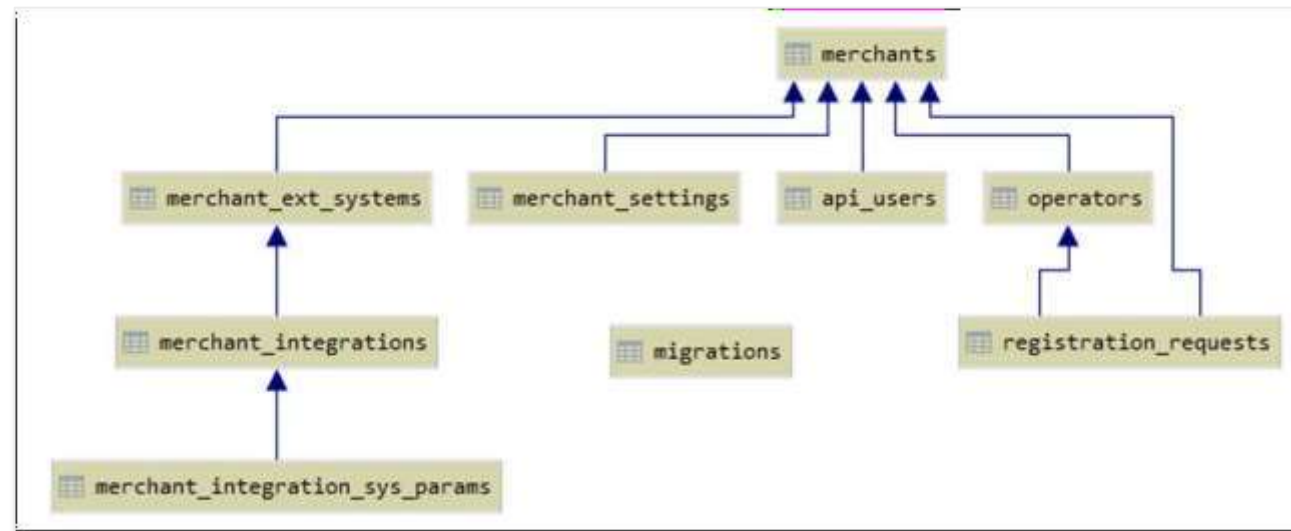
# СЕРВИС MMS



Назначение: управление мерчантами.

Основная функциональность:

- управление мерчантами;
- управление сотрудниками (операторами) мерчантов;
- управление настройками интеграций с внешними системами мерчантов;
- управление настройками мерчантов.



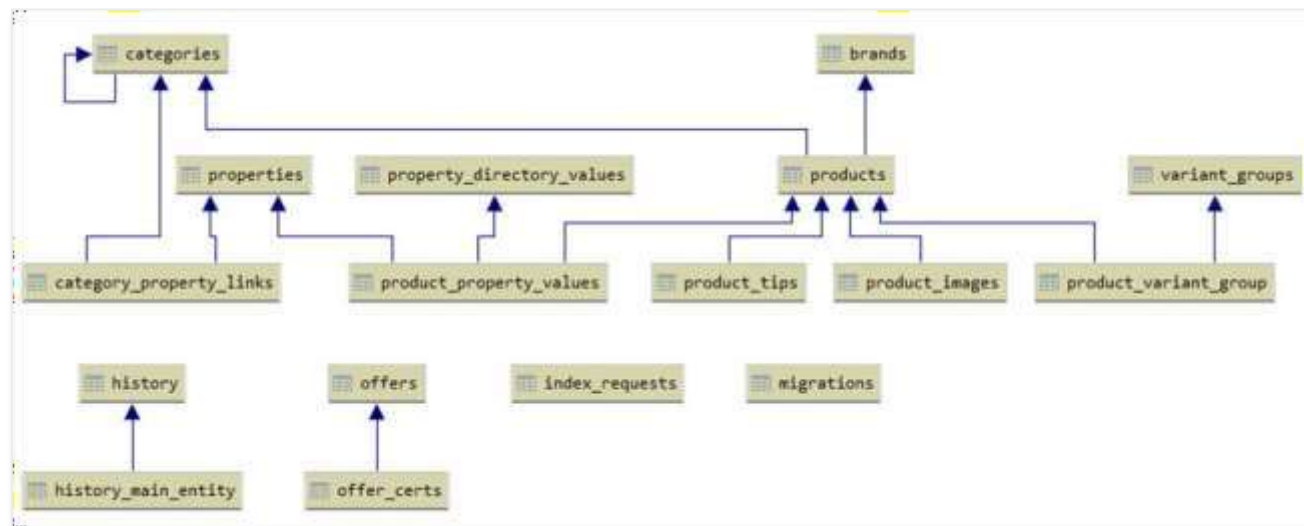
# СЕРВИС PIM



Назначение: управление товарным контентом.

Основная функциональность:

- управление категориями и свойствами (характеристиками) товаров в них, вариантами значений этих свойств (инфомодель);
- управление брендами;
- управление товарами, предложениями мерчантов и подборками (Bundles) товаров;
- логирование истории изменения всех сущностей.



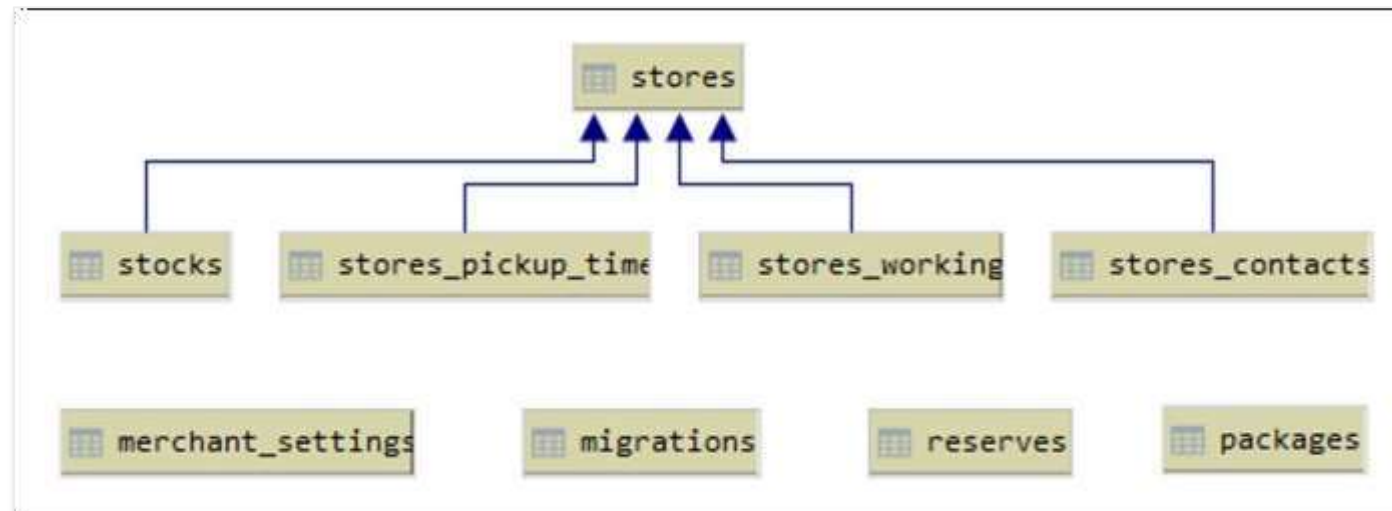
# СЕРВИС Store



Назначение: управление складами мерчантов и товарным остатками.

Основная функциональность:

- управление складами мерчантов;
- управление товарными остатками (стоками) мерчантов;
- управления типами коробок для упаковки товаров заказов.



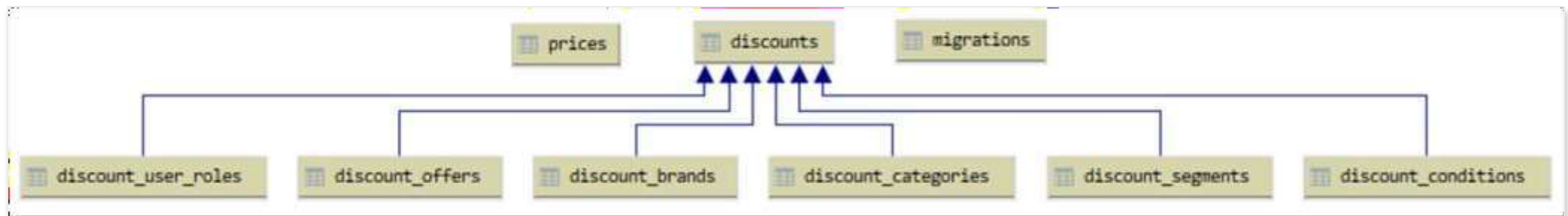
# СЕРВИС Marketing



Назначение: управление ценами, скидками и бонусной программой.

Основная функциональность:

- управление ценами на предложения мерчантов;
- управление скидками для:
  - конкретных ролей пользователей;
  - конкретных предложений мерчантов;
  - конкретных брендов;
  - конкретных категорий товаров;
  - конкретных сегментов покупателей;
  - дополнительных условий (первый заказ, заказ от определенной суммы и т.д.).



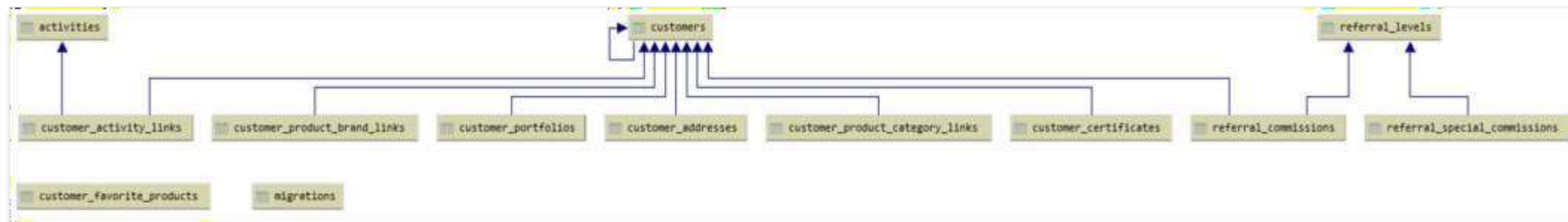
# СЕРВИС СМ



Назначение: управление покупателями.

Основная функциональность:

- управление покупателями и их:
  - профессиональными активностями;
  - любимыми брендами и товарными категориями;
  - портфолио;
  - адресами доставки;
  - избранным;
  - сертификатами и т.д.
- управление реферальными партнерами и их комиссиями.



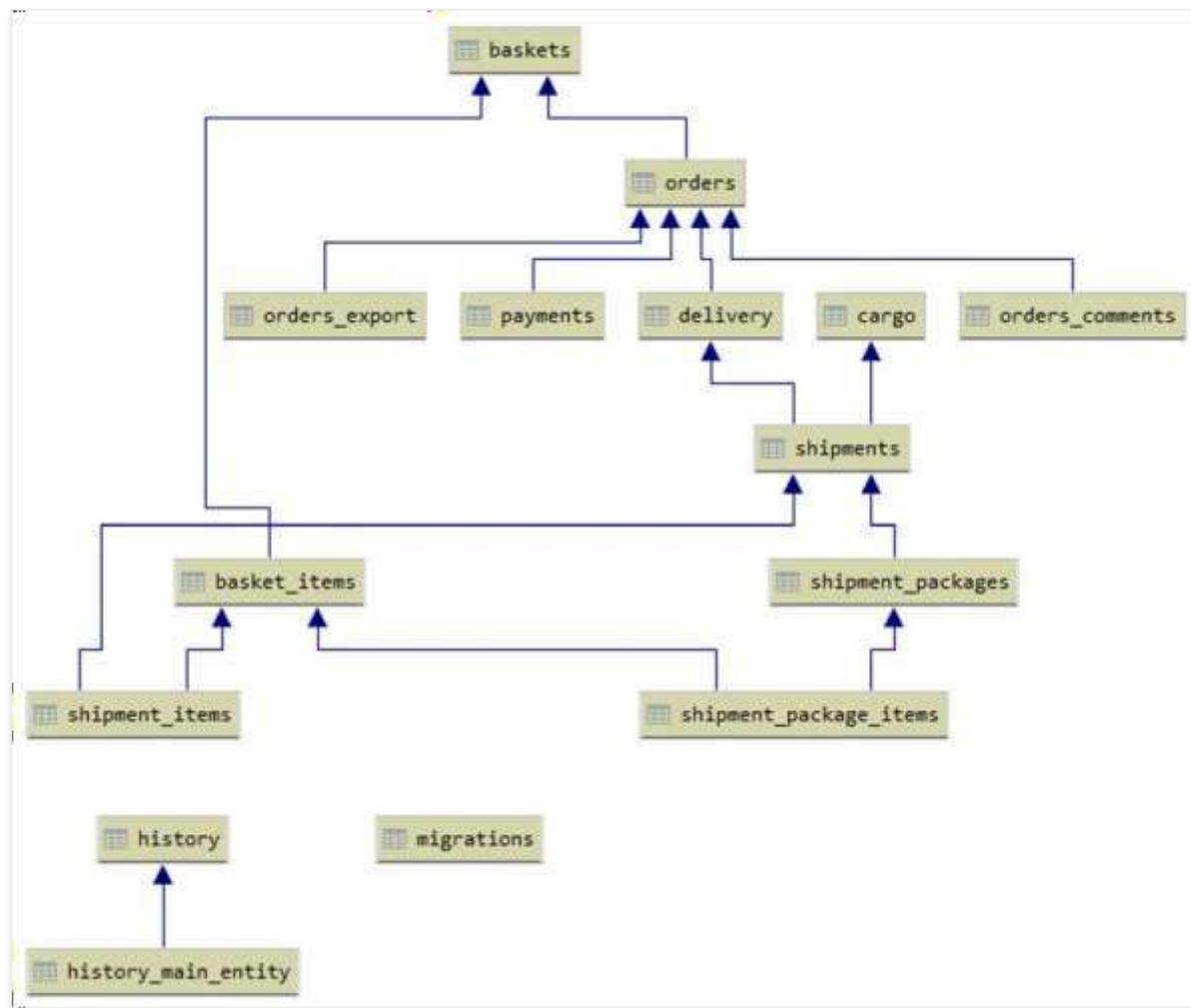
# СЕРВИС OMS



Назначение: управление заказами и их доставкой.

Основная функциональность:

- управление корзинами;
- управлениями заказами и их доставкой (грузы, доставки, отправления, коробки);
- управление оплатами;
- логирование истории изменения всех сущностей





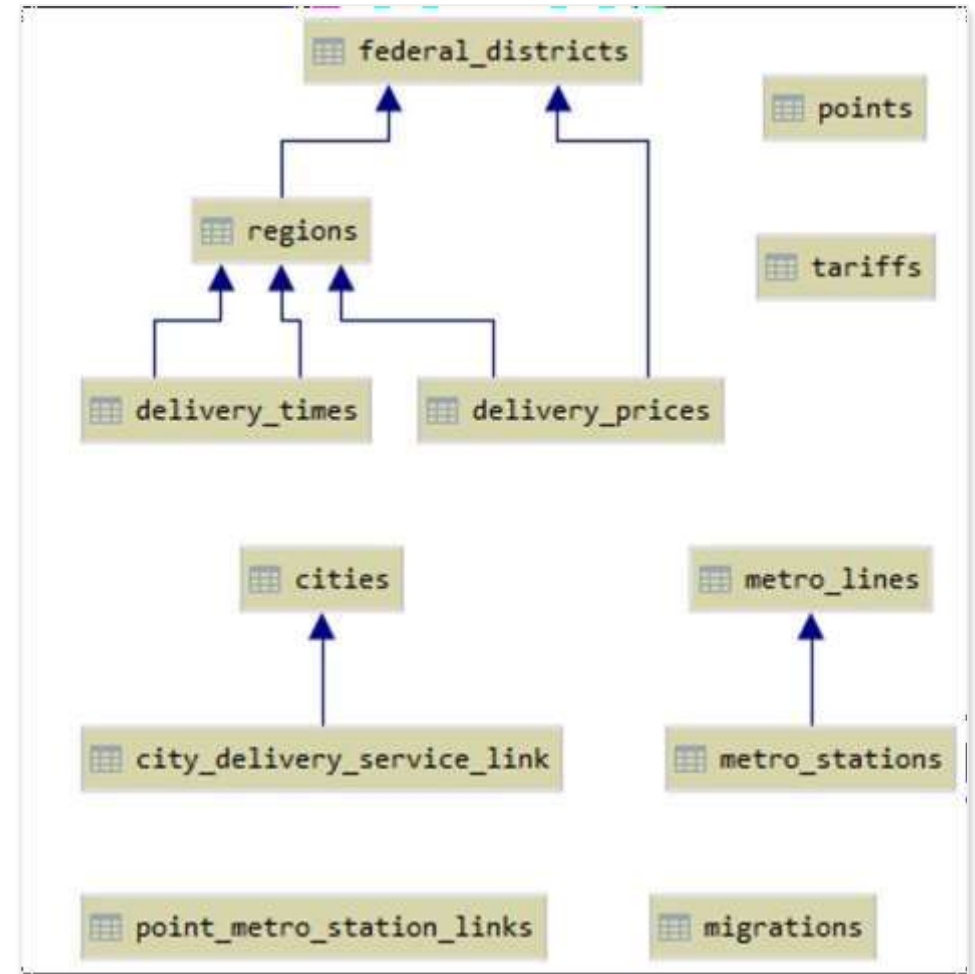
# СЕРВИС Logistics



Назначение: интеграция с логистическими операторами.

Основная функциональность:

- расчет вариантов доставки по срокам и стоимости;
- получение штрихкодов для коробок заказа;
- управления заявками на вызов курьеров логистических операторов для забора товара у мерчанта (нулевая миля);
- управления заказами на доставку у логистических операторов (“основная” и последняя миля) + получение статуса доставки;
- получение информации по адресам, метро и точкам самовывоза (ПВЗ и постаматам)



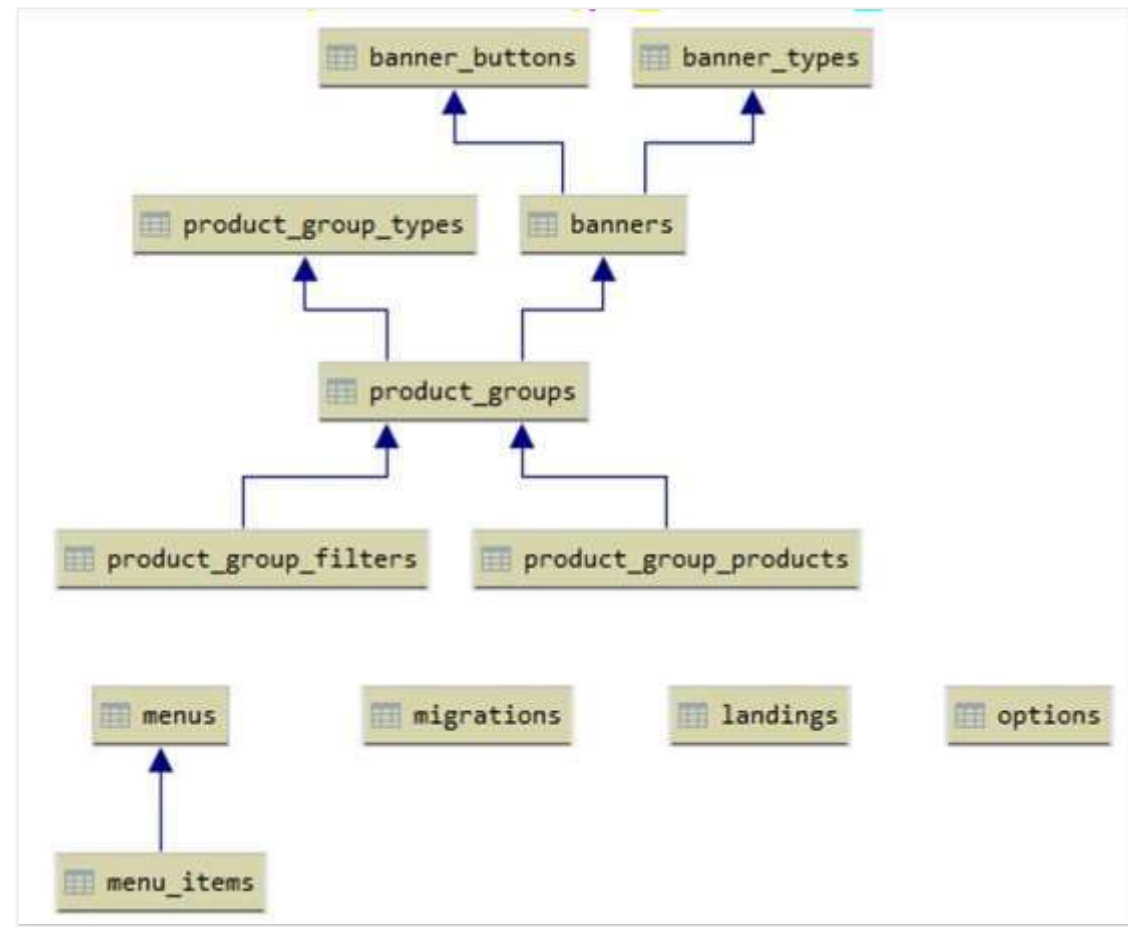
# СЕРВИС CMS



Назначение: управление нетоварным контентом.

Основная функциональность:

- управление меню;
- управление баннерами;
- управление товарными подборками;
- управления лендингами;
- управление общими настройками (key-value хранилище).



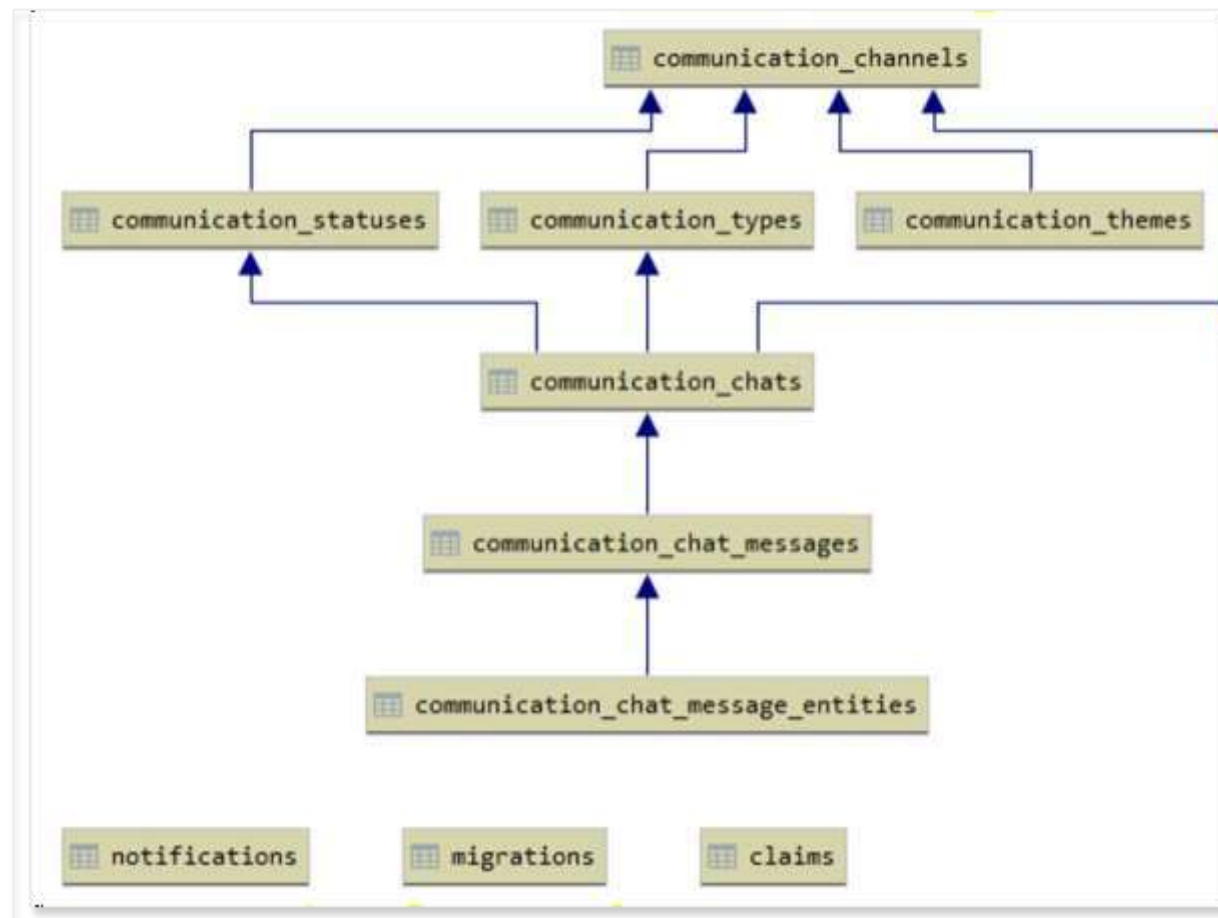
# СЕРВИС Message



Назначение: управление сообщениями, заявками и уведомлениями.

Основная функциональность:

- управление коммуникациями между пользователями маркетплейса;
- управление заявками;
- управление сообщениями (e-mail, sms);
- управление уведомлениями.



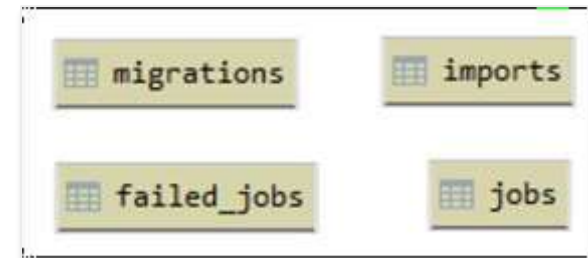
# СЕРВИСЫ ИНТЕГРАЦИИ

Сервис CatalogImport



**Назначение:** импорт товаров, цен и остатков из xlsx-файлов мерчантов.

---



Сервис MoySklad



**Назначение:** импорт цен и остатков из 1С / MoySklad в маркетплейс и экспорт заказов обратно.

---

Сервисы интеграции с самописными ERP мерчантов



**Назначение:** импорт цен и остатков из ERP в маркетплейс и экспорт заказов обратно.

# СЕРВИС API



Назначение: предоставление единого API для фронта маркетплейса.

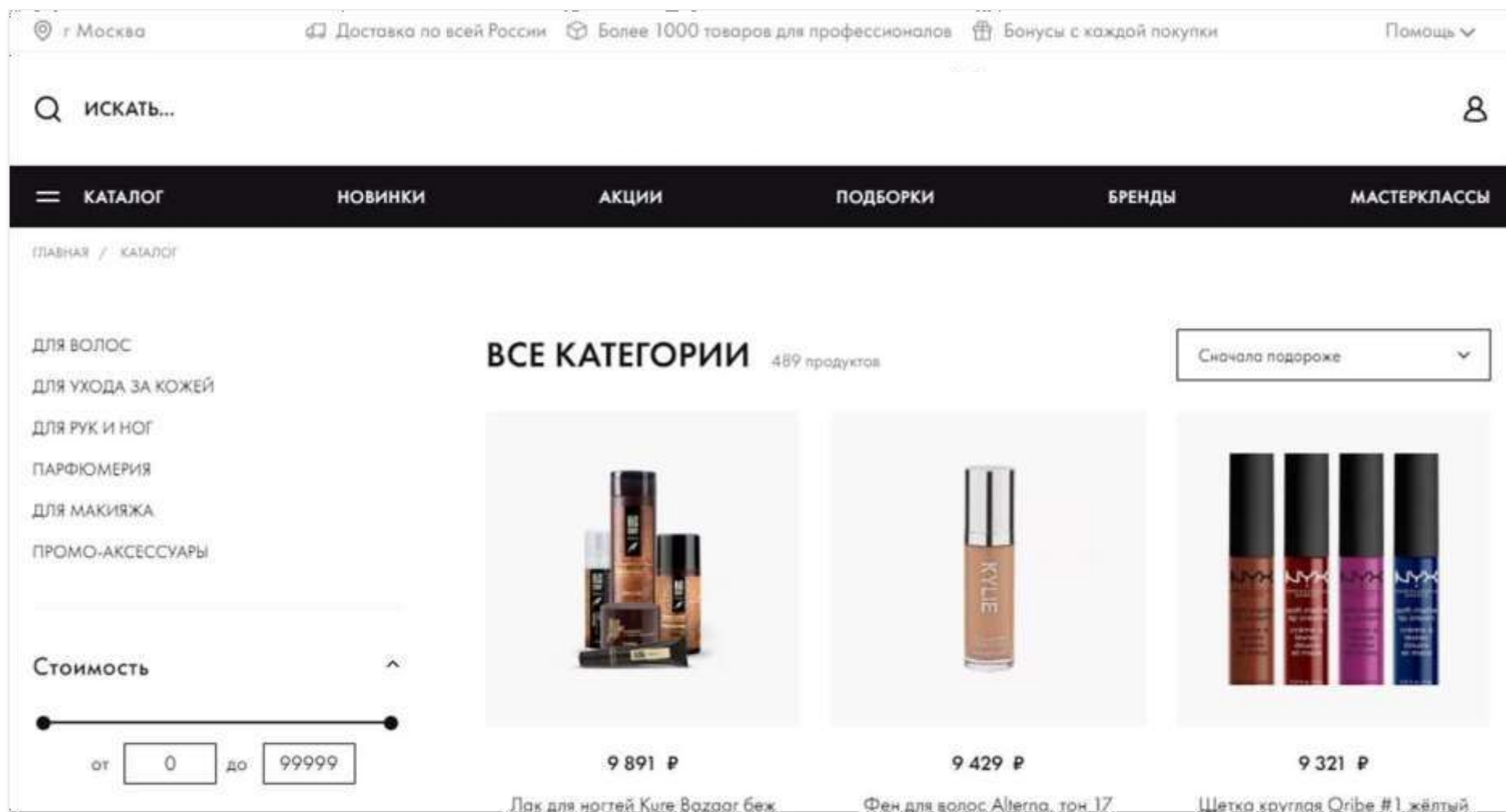
The screenshot shows the Swagger UI for an API titled "iBT. Авторизация" (iBT. Authorization). The interface includes a "Select a definition" dropdown menu currently set to "Авторизация". Below the title, there are version tags "1.0.0" and "OAS3". A "Servers" dropdown is set to "/v1". The main content area displays a "login" endpoint under the "Авторизация" category. It lists three endpoints:

- POST** `/auth/loginByPassword` Авторизация по логину и паролю
- POST** `/auth/logout` Выход из системы
- GET** `/auth/is-login` Проверка авторизован пользователь или нет

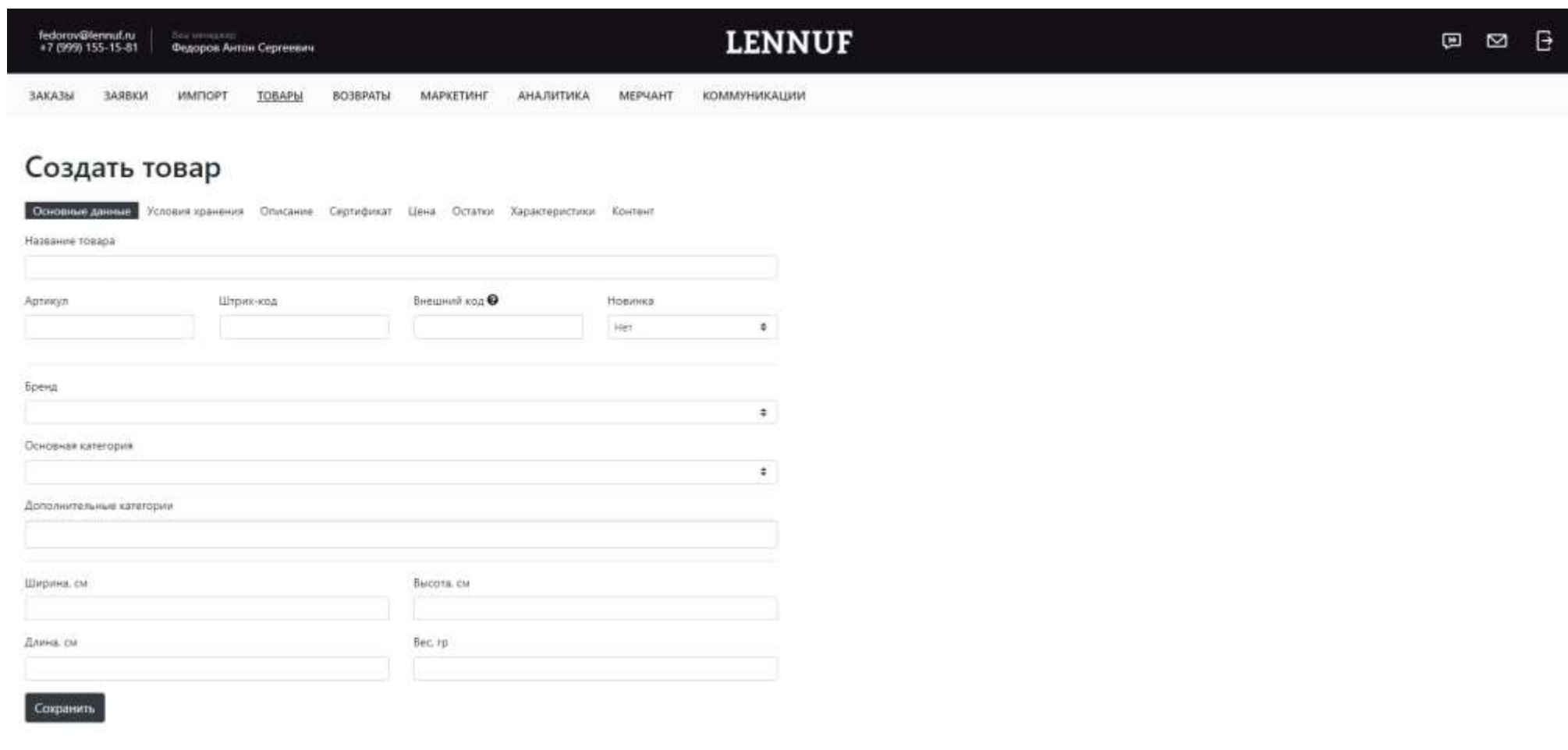
# СЕРВИС FRONT



Назначение: витрина маркетплейса для покупателей.



Назначение: фронт маркетплейса для мерчантов.



The screenshot shows the 'Создать товар' (Create product) form in the LENNUF merchant interface. The form is divided into several sections:

- Header:** Includes contact information (fedorov@lennuf.ru, +7 (999) 155-15-81), the name of the merchant (Федоров Антон Сергеевич), and the company name (LENNUF). Navigation icons for chat, email, and home are also present.
- Navigation:** A menu with items: ЗАКАЗЫ, ЗАЯВКИ, ИМПОРТ, ТОВАРЫ, ВОЗВРАТЫ, МАРКЕТИНГ, АНАЛИТИКА, МЕРЧАНТ, КОММУНИКАЦИИ.
- Form Title:** 'Создать товар' (Create product).
- Steps:** A series of tabs: Основные данные (selected), Условия хранения, Описание, Сертификат, Цена, Остатки, Характеристики, Контент.
- Fields:**
  - Название товара (Product name): A single-line text input.
  - Артикул (SKU): A single-line text input.
  - Штрих-код (Barcode): A single-line text input.
  - Внешний код (External code): A single-line text input with a help icon.
  - Новинка (New): A dropdown menu with 'Нет' (No) selected.
  - Бренд (Brand): A dropdown menu.
  - Основная категория (Main category): A dropdown menu.
  - Дополнительные категории (Additional categories): A multi-line text input.
  - Ширина, см (Width, cm): A single-line text input.
  - Высота, см (Height, cm): A single-line text input.
  - Длина, см (Length, cm): A single-line text input.
  - Вес, гр (Weight, g): A single-line text input.
- Buttons:** A 'Сохранить' (Save) button is located at the bottom left of the form.

# СЕРВИС ADMIN



Назначение: админка маркетплейса.

**LENNUF** | Administration System 111 ✉ 📧 Выйти

[ТОВАРЫ](#) [ВЕБ-ВИТРИНА](#) [ЗАКАЗЫ](#) [ВОЗВРАТЫ](#) [ЛОГИСТИКА](#) [КЛИЕНТЫ](#) [РЕФЕРАЛЫ](#) [МЕРЧАНТЫ](#) [МАРКЕТИНГ](#) [КОММУНИКАЦИИ](#) [УСЛУГИ](#) [ДОСТУПЫ](#) [АНАЛИТИКА](#) [РЕДИРЕКТ](#) [НАСТРОЙКИ](#)

## Категории

Фильтр

Название

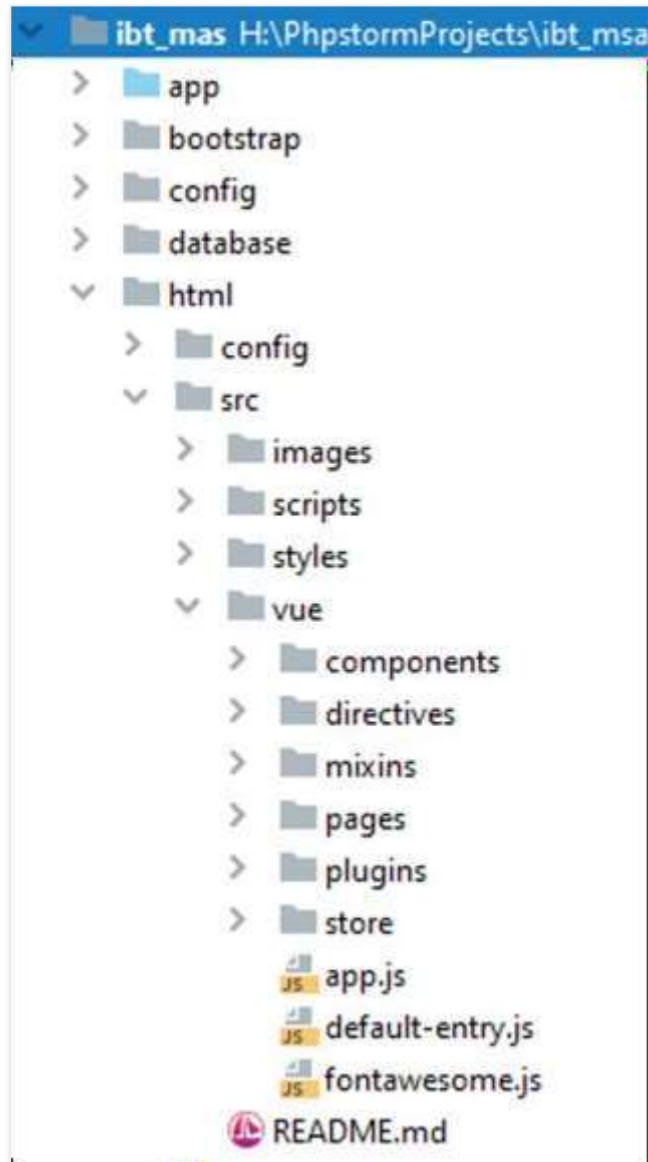
[Применить](#) [Сбросить](#)

[Создать категорию](#) [Шаблоны SEO](#)

Название категории	Код	Активна	Действия
Shampoo	shampoo	Да	<a href="#">✎</a> <a href="#">🔍</a>
UZB Patriot ▾	uzb-patriot	Да	<a href="#">✎</a> <a href="#">🔍</a>
Аксессуары	akssesuary	Да	<a href="#">✎</a> <a href="#">🔍</a>
Алмазные коронки	almaznye-koronki-2	Да	<a href="#">✎</a> <a href="#">🔍</a>
Безопасность ▾	bezopasnost	Да	<a href="#">✎</a> <a href="#">🔍</a>
Блески и тинты	bleski-i-tinty	Да	<a href="#">✎</a> <a href="#">🔍</a>



# ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ MAS И ADMIN



1. Бэкенд - Laravel, фронтенд - Vue.
2. Все запросы с фронта передаются AJAX в
  - HTTP-контроллеры Laravel, а из Laravel
  - отправляются к REST API нужных сервисов.

# КОНТАКТЫ

- +7 495 128 00 03
- +7 906 093 44 63 – Гордон Юлиана, Генеральный директор
- +7 916 989 85 22 – Бурмистров Илья, Партнер
- <https://www.lennuf.ru>
- [GO@lennuf.ru](mailto:GO@lennuf.ru)