

# Outsourcing API ver.1

- [1. Общие правила аутентификации](#)
- [2. Method: GET](#)
- [3. Method: PUT](#)
- [4. Method: POST](#)
- [5. Method: DELETE](#)
- [6. Аутентификация пользователей](#)
  - [a. authLogin](#)
  - [b. authLogout/{id}](#)
- [7. Модуль работы с АК](#)
  - [a. persons](#)
  - [b. units](#)
  - [c. Companies](#)
  - [d. specialties](#)
  - [e. Accounts](#)
  - [f. fias](#)
- [8. Модуль работы с клиентами](#)
  - [a. Clients](#)
  - [b. ClientsGroup](#)
  - [c. ClientsObject](#)
  - [d. ClientsDepartment](#)
  - [e. ClientsRate](#)
  - [f. ClientsReg](#)
  - [g. TypeUnits](#)
- [9. Модуль работы с линейным персоналом \(исполнителями\)](#)

## Содержание:

1. [Общие правила аутентификации](#)
2. [Описание метода GET](#)
3. [Описание метода PUT](#)
4. [Описание метода POST](#)
5. [Описание метода DELETE](#)
6. Аутентификация пользователей
  - a. [authLogin](#) - Вход в CRM
  - b. [authLogout](#) - Выход из CRM
7. Модуль работы с АК
  - a. [persons](#) - пользователи CRM
  - b. [units](#) - роли пользователей
  - c. [Companies](#) - организации АК
  - d. [specialties](#) - специализации

- e. [Accounts](#) - счета АК
- f. [fias](#) - справочник адресов
- 8. Модуль работы с клиентами
  - a. [Clients](#) - клиенты АК
  - b. [ClientsGroup](#) - группы объектов клиентов АК
  - c. [ClientsObject](#) - объекты клиентов АК
  - d. [ClientsDepartment](#) - отделы объектов клиентов
  - e. [ClientsRate](#) - ставки
  - f. [ClientsReq](#) - базовые требования к ЛП
  - g. [ClientsTypeUnits](#) - единицы измерений штучных работ
- 9. Модуль работы с линейным персоналом (исполнителями)
  - a. [IpMan](#) - физические лица
  - b. [IpWorker](#) - исполнители
  - c. [IpContracts](#) - договора
  - d. [IpSpeciality](#) - специализации
  - e. [IpAccount](#) - счета исполнителя
  - f. [IpDoc](#) - документы физического лица
  - g. [IpNote](#) - комментарии к физическому лицу
- 10. Модуль работы с заказами
  - a. [orders](#) - заказы клиентов
  - b. [ordersReq](#) - требования к ЛП к заявкам
  - c. [requests](#) - заявки клиентов
  - d. [shifts](#) - Смены
- 11. Модуль работы с платежами
  - a. [payPay](#) - выплаты
  - b. [payBalance](#) - денежные операции начислений
  - c. [payTypes](#) - типы операций
  - d. [payDetail](#) - детализация операций

## 1. Общие правила аутентификации

Все запросы должны формироваться с HTTP заголовком Authorization: Bearer {токен}, где {токен} формируется в виде случайной последовательности символов длиной 40 символов при первичной аутентификации с помощью логина/пароля (см. authLogin). Токен хранится в БД и удаляется при разлогинировании.

## 2. Method: GET

**Получение множественных данных**

**URL:**

/V1/{модуль}/{функция}All?pref={id\_АК}&session\_id={id\_пользователя}&{дополнительные параметры}

**Пример:** /V1/Accounts/AccountAll?pref=1\_&session\_id=3

MySQL запрос данных указан в файле /V1/{модуль}/{функция}All.php

**Ответ в случае положительного результата:**

```
{
  "result": "Ok",
  "method": "GET",
  "data": {
    "0": {
      "имя поля 1": [строка 1 ],
      "имя поля 2": [строка 2 ],
      .
      .
      .
      "имя поля n": [строка n ]
    }
    "1": [строка 2 ],
    .
    .
    .
    "n": [строка n-1 ],
  },
  "data2": {
    "0": {
      "имя поля 1": [строка 1 ],
      "имя поля 2": [строка 2 ],
      .
      .
      .
      "имя поля n": [строка n ]
    }
    "1": [строка 2 ],
    .
    .
    .
    "n": [строка n-1 ],
  },
  "sort": [строка сортировки данных],
  "search": [строка для поиска данных по полям],
  "colspan": кол-во столбцов
}
```

Массив Data формируется путем выполнения MySQL запроса , указанного в конкретном файле функции API и содержит данные из перекрестных таблиц.

Массив Data содержит данные только из опрашиваемой таблицы без замены их на данные из других таблиц, содержит все столбцы таблицы с выборкой и сортировкой аналогичной массива Data.

### Пример ответа:

```
Array
(
    [result] => Ok
    [method] => GET
    [data] => Array
        (
            [0] => Array
                (
                    [id] => 1
                    [name] => ГиперРЫБА
                    [address] => Рязанская область, Рыбновский район, г. Рыбное, ул. Рыбаков, д. 3А, стр. 6
                    [id_group] => Гипермаркеты
                    [url_info] => <a href="client_object_info.dhtml?id=1">Инфо.</a>
                )

            [1] => Array
                (
                    [id] => 2
                    [name] => Объект 555
                    [address] => Москва
                    [id_group] => Группа 2
                    [url_info] => <a href="client_object_info.dhtml?id=2">Инфо.</a>
                )

            [2] => Array
                (
                    [id] => 3
                    [name] => Магнит 11бр
                    [address] =>
                    [id_group] => Супермаркеты Магнитов
                    [url_info] => <a href="client_object_info.dhtml?id=3">Инфо.</a>
                )
        )
    [data2] => Array
        (
            [0] => Array
                (
                    [id] => 12
                    [name] => Клиент, которого не может добавить Даня
                    [note] => тест
                    [id_fias] => 8
                    [contact] => -
                    [www] => -
                    [id_person] => 4
                    [state] => y
                    [close_period] =>
                    [dt_close] => 2023-04-20 09:48:17
                    [id_person_close] => 0
                )

            [1] => Array
```

```

        (
            [id] => 9
            [name] => ЛЕНТА3
            [note] =>
            [id_fias] => 3
            [contact] =>
            [www] => lenta.ru
            [id_person] => 0
            [state] => y
            [close_period] =>
            [dt_close] => 2023-04-13 15:41:02
            [id_person_close] => 0
        )
    )
[sort] => Array
(
    [0] => <a href="clients_object.dhtml?sort=id" class="upmenu">ID</a>
    [1] => <a href="clients_object.dhtml?sort=name" class="upmenu">Название</a>
    [2] => <a href="clients_object.dhtml?sort=address" class="upmenu">Адрес</a>
    [3] => <a href="clients_object.dhtml?sort=id_group" class="upmenu">Группа</a>
    [4] =>
)

[search] => Array
(
    [0] =>
    [1] => <input class=tR type=text name="search[text][tbl1.name]" id="search[text][tbl1.name]" size=50
value=""></input>
    [2] => <input class=tR type=text name="search[text][tbl1.address]" id="search[text][tbl1.address]" size=50
value=""></input>
    [3] => <select class=tR name="search[int][id_group]" style="width:50;"
onChange="this.form.submit();"><option value="">Все</option><option
value="1">Гипермаркеты</option><option value="2">СуперЛЕНТА</option><option value="3">Группа
2</option><option value="4">Супермаркеты Магнитов</option></select>
)

[colspan] => 5
)

```

## Получение данных по конкретной записи

### URL:

/V1/{модуль}/{функция}Info/{id}?pref={id\_AK}&session\_id={id\_пользователя}

**Пример:** /V1/Accounts/AccountInfo/1?pref=1\_&session\_id=3

MySQL запрос выводит все поля таблицы и значения этих полей для указанного {id}

### Ответ в случае положительного результата:

```

{
    "result": "Ok",
    "method": "GET",
    "data": {
        "id": "...",
        "name1": "...",

```

```
    "name2": "...",
    "name n": "..."
  }
}
```

#### Пример ответа:

```
{
  "result": "Ok",
  "method": "GET",
  "data": {
    "id": "1",
    "name": "Касса",
    "type": "n",
    "card": "",
    "bank_name": "",
    "bank_bik": "",
    "account": "",
    "note": "По умолчанию у каждой АК есть такой"
  }
}
```

## 3.Method: PUT

### Изменение данных

**URL:** /v1/{модуль}/{функция}InfoPut/{id}

#### Массив данных:

```
Array
(
  [input] => Array
    (
      [поле таблицы id] => ...
      [поле таблицы 1] => ...
      [поле таблицы 2] => ...

      [поле таблицы n] => ...
    )
  [pref] => {id_АК}_
  [session_id] => {id_пользователя}
)
```

MySQL запрос формируется по полям из массива данных [input].

**Пример:** /v1/Clients/ClientInfoPut/5

```
Array
(
    [input] => Array
        (
            [id] => 5
            [name] => магазин Петрович
            [www] => www
            [id_fias] =>
            [contact] =>
            [state] => y
            [id_person] => 4
            [close_period] => 2023-04-03
            [note] => dwqd
            [dt_close] => 2023-04-13 17:04:56
            [id_person_close] => 0
        )
    [pref] => 1_
    [session_id] => 3
)
```

**Ответ в случае положительного результата:**

```
{
  "result": "Ok",
  "method": "POST"
  "data": {
    "result": "Ok"
  }
}
```

## 4.Method: POST

**Добавление данных**

**URL:** /v1/{модуль}/{функция}Add

**Массив данных:**

```
Array
(
    [input] => Array
        (
            [поле таблицы id] => ...
            [поле таблицы 1] => ...
            [поле таблицы 2] => ...
        )
)
```

```

        [поле таблицы n] => ...
    )
    [pref] => {id_AK}_
    [session_id] => {id_пользователя}
)

```

MySQL запрос формируется по полям из массива данных [input].

**Пример:** /v1/clientsDepartment/clientDepartmentAdd

```

Array
(
    [input] => Array
        (
            [id] => 5
            [name] => мясной отдел
            [id_object] => 5
            [note] => test
        )
    [pref] => 1_
    [session_id] => 3
)

```

**Ответ в случае положительного результата:**

```

{
  "result": "Ok",
  "method": "POST"
  "data": {
    "result": "Ok",
    "id": {id}
  }
}

```

## 5.Method: DELETE

**Удаление конкретной записи**

**URL:**

/v1/{модуль}/{функция>Delete/{id}?pref={id\_AK}\_&session\_id={id\_пользователя}

**Пример:** /v1/Accounts/AccountDelete/1?pref=1\_&session\_id=3

MySQL запрос на удаление записи с ID={id}

**Ответ в случае положительного результата:**



```

{
  "result": "Ok",
  "method": "DELETE"
  "data": {
    "result": "Ok",
    "Id": {id}
  }
}

```

## 6. Аутентификация пользователей

### a. authLogin

Аутентификация пользователя. Вход в CRM

#### GET authLogin

Передаваемые данные:

```

(
  'login'=>{логин},
  'password'=>{пароль},
  'remember'=>{1- если пользователь хочет дальше авторизовываться через
Cookie},
  'remote_addr'=>{IP-адрес клиента} - опционально
);

```

Результат при положительной аутентификации:

[result] => Ok	
[method] => GET	
[data] => Array	
(	
[auth] => 1	
[id] => 3	ID пользователя (person)
[login] => anna	Логин
[id_unit] => 3	Роль пользователя
[realname] => Анна	Имя
[script_name] =>	Скрипт, переадресации при аутентификации
[id_owner] => 1	ID АК
[timezone] => 5	Часовой пояс относительно UTC+0
[token] => DFmkxjr%fdkmw4	Token ключ для API запросов
[cookie] =>	Cookie ключ, если пользователь выбрал
"Запомнить меня"	
[timeout] => 3600	Время жизни сессии в сек.
[edit] => 1	Общие права на редактирование данных
(1-разрешено)	

```
[permis] => Array                               Массив прав на скрипты
(
  [world/permission.dhtml] => 0
  [world/persons.dhtml] => 0
  [world/orders.dhtml] => 1
  [world/clients.dhtml] => 1
)
)
```

Результат при отрицательной аутентификации:

```
(
  [result] => Ok
  [method] => GET
  [data] => Array
  (
    [auth] => 0
    [message] => 401
  )
)
```

Аутентификация пользователя через переменные Cookie.

Возможно в случае, если пользователь при вводе логина/пароля поставил галочку “Запомнить меня”. При этом в БД сохраняется случайно сгенерированный код длиной 8 символов.

### **GET authLogin**

Передаваемые данные:

```
(
  'login'=>{$_COOKIE[login]},
  'cookie'=>{$_COOKIE[key]},
);
```

Результаты такие же, как и при проверке логина/пароля.

## **b. authLogout/{id}**

Выход из CRM.

В БД у пользователя удаляются данные из полей cookie и token.

# 7. Модуль работы с АК

## a. persons

Пользователи CRM. Пароль хранится в зашифрованном виде (Hash)

**Структура таблицы:**

```

`id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(32) NOT NULL,
`password` varchar(60) NOT NULL,
`email` varchar(32) NOT NULL,
`id_unit` smallint(5) unsigned DEFAULT NULL,
`realname` varchar(64) DEFAULT NULL,
`tel` varchar(12) DEFAULT NULL COMMENT 'мобильный телефон сотрудника',
`script_name` varchar(255) DEFAULT NULL,
`ip` varchar(15) DEFAULT NULL,
`id_group` int(11) DEFAULT 0,
`id_owner` int(11) DEFAULT 0,
`cookie` varchar(8) DEFAULT NULL,
`note` text DEFAULT NULL,
`timezone` smallint(6) NOT NULL DEFAULT 3,
`remember_token` varchar(256) DEFAULT NULL

```

#### Доступные методы:

- i. GET personInfo
- ii. GET personAll

```

(
  [id] => 1
  [realname] => Иванов Иван
  [name] => test
  [tel] =>
  [email] =>
  [id_group] => сотрудник
  [id_unit] => Администратор CRM
  [url_info] => Инфо.
)

```

- iii. PUT personInfoPut
- iv. POST personAdd
- v. DELETE personDelete

#### b. units

Роли пользователей CRM

#### Структура таблицы:

```

`id` int(11) NOT NULL AUTO_INCREMENT,
`name` varchar(50) DEFAULT NULL,
`script_name` varchar(255) DEFAULT NULL,

```

``note` text DEFAULT NULL`

**Доступные методы:**

- i. GET unitInfo
  - ii. GET unitAll
- (
- [id] => 1
  - [name] => Администратор CRM
  - [url\_info] => Инфо.
- )
- iii. PUT unitInfoPut
  - iv. POST unitAdd
  - v. DELETE unitDelete

### c. Companies

Описывает организации АК

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`state` smallint(6) NOT NULL DEFAULT 1,  
`type` enum('y','n') NOT NULL DEFAULT 'y',  
`name` varchar(256) DEFAULT NULL,  
`realname` varchar(256) DEFAULT NULL,  
`inn` varchar(12) DEFAULT NULL,  
`kpp` varchar(10) DEFAULT NULL,  
`ogrn` varchar(13) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL,
```

**Доступные методы:**

- i. GET CompanyInfo
  - ii. GET CompanyAll
- {
- [id] => "id"
  - [name] => "название организации"

[type] => “тип организации физ/юр”  
[inn] => “ИНН”  
[url\_info] => Инфо.

}

- iii. PUT CompanyInfoPut
- iv. POST CompanyAdd
- v. DELETE CompanyDelete

## d. specialties

Специализации АК

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`name` varchar(256) DEFAULT NULL,  
`note` text DEFAULT NULL
```

### Доступные методы:

- i. GET specialityInfo
- ii. GET specialityAll

{

[id] => 1  
[name] => кассир-продавец  
[url\_info] => Инфо.

}

- iii. PUT specialityInfoPut
- iv. POST specialityAdd
- v. DELETE specialityDelete

## e. Accounts

Описывает счета АК

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_company` int(11) DEFAULT NULL,
```

```

`name` varchar(256) DEFAULT NULL,
`type` enum('n','b') DEFAULT 'b',
`card` varchar(20) DEFAULT NULL,
`bank_name` varchar(256) DEFAULT NULL,
`bank_bik` varchar(9) DEFAULT NULL,
`account` varchar(30) DEFAULT NULL,
`note` text DEFAULT NULL,
`state` enum('y','n') DEFAULT 'y',
`main` enum('y','n') DEFAULT 'y',
`dt_change` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
`id_person_change` int(11) DEFAULT NULL

```

#### Доступные методы:

- i. GET AccountInfo
- ii. GET AccountAll

```

(
  [id] => "Id"
  [id_company] => "название организации"
  [main] => "основной да/нет"
  [name] => "название счета"
  [type] => "тип (нал/безнал)"
  [card] => "№ карты"
  [bank_name] => "название банка"
  [url_info] => Инфо.
)

```

- iii. PUT AccountInfoPut
- iv. POST AccountAdd
- v. DELETE AccountDelete

#### f. fias

Справочник адресов, формируемый по API с ресурса [dadata](#). Необходим для того, чтобы не обращаться к [dadata](#) каждый раз, когда нужен адрес.

#### Структура таблицы:

```

`id` int(11) NOT NULL AUTO_INCREMENT,
`address` varchar(256) NOT NULL,
`postal_code` varchar(6) DEFAULT NULL,

```

```
`federal_district` varchar(100) DEFAULT NULL,  
`region_fias_id` varchar(40) DEFAULT NULL,  
`region_with_type` varchar(100) DEFAULT NULL,  
`area_fias_id` varchar(40) DEFAULT NULL,  
`area_tith_type` varchar(100) DEFAULT NULL,  
`city_fias_id` varchar(40) DEFAULT NULL,  
`city_with_type` varchar(100) DEFAULT NULL,  
`city_area` varchar(100) DEFAULT NULL,  
`city_district_fias_id` varchar(40) DEFAULT NULL,  
`city_district_with_type` varchar(100) DEFAULT NULL,  
`street_fias_id` varchar(40) DEFAULT NULL,  
`street_with_type` varchar(100) DEFAULT NULL,  
`house_fias_id` varchar(40) DEFAULT NULL,  
`house_type` varchar(20) DEFAULT NULL,  
`house` varchar(30) DEFAULT NULL,  
`geo_lat` float DEFAULT NULL,  
`geo_lon` float DEFAULT NULL,  
`level` smallint(6) DEFAULT NULL,  
`json` text DEFAULT NULL,  
`fias_id` varchar(40) DEFAULT NULL
```

#### Доступные методы:

- i. GET fiasInfo
- ii. GET fiasAll

```
(  
  [id] => 7  
  [address] => г Улан-Удэ  
  [fias_id] => 9fdcc25f-a3d0-4f28-8b61-40648d099065  
  [url_info] => Инфо.  
)
```

- iii. GET fiasSearch с дополнительными параметрами

```
{  
  'search': {строка для поиска адреса}  
  'level': {уровень адреса ФИАС 1-8 (для города 4, для дома 8)}  
}
```

Результат запроса:

```
(  
  [9b968c73-f4d4-4012-8da8-3dacd4d4c1bd] => г Красноярск  
  [7dfa745e-aa19-4688-b121-b655c11e482f] => г Краснодар  
  [63fcf18a-365e-451f-baee-8d09ac50b773] => Московская обл, г Красногорск  
  [59b1c816-7f98-4707-b5e2-c6193fff8923] => Свердловская обл, г Краснотурьинск  
  [cced82af-3bdf-4a91-ab68-d654804767c0] => г Санкт-Петербург, г Красное Село
```

```
[3c9d5e9e-2922-4e9e-9e49-dbe6faf01fb2] => Забайкальский край, г  
Краснокаменск  
)
```

iv. PUT fiasInfoPut

v. POST fiasAdd с дополнительными параметрами

```
{  
  'fias_id': {ID ФИАС, полученный методом fiasSearch, или введенный  
вручную}  
  'client_id': {ID клиента, которому присваивается ID созданного адреса} -  
опционально  
  'object_id': {ID объекта, которому присваивается ID созданного адреса и  
координаты geo_lat, geo_lon} - опционально  
  'man_id': {ID физического лица, которому присваивается ID созданного  
адреса и координаты geo_lat, geo_lon} - опционально  
}
```

Результат, в случае удачного запроса:

```
(  
  [result] => Ok  
  [Id] => 8  
  [client update] => Ok  
)
```

vi. DELETE fiasDelete

## 8. Модуль работы с клиентами

### a. Clients

Описывает клиентов АК

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`name` varchar(256) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`id_fias` int(11) DEFAULT NULL,
```



```
`contact` varchar(256) DEFAULT NULL,  
`www` varchar(255) DEFAULT NULL,  
`id_person` int(11) DEFAULT NULL,  
`state` enum('y','n') DEFAULT 'y',  
`close_period` date DEFAULT NULL,  
`dt_close` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp(),  
`id_person_close` int(11) DEFAULT NULL
```

#### Доступные методы:

- i. GET ClientInfo
- ii. GET ClientAll

```
(  
  [id] => "Id"  
  [name] => "название компании"  
  [address] => "адрес"  
  [id_person] => "ответственный менеджер"  
  [url_info] => Инфо.  
)
```

- iii. PUT ClientInfoPut
- iv. POST ClientAdd
- v. DELETE ClientDelete

## b. ClientsGroup

Описывает группы объектов клиентов АК

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`name` text DEFAULT NULL,  
`id_person` int(11) DEFAULT NULL,  
`state` smallint(6) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`id_client` int(11) DEFAULT NULL
```

#### Доступные методы:

- i. GET ClientGroupInfo
  - ii. GET ClientGroupAll
- ```
(  
    [id] => "id"  
    [name] => "Название"  
    [state] => "Активный"  
    [id_client] => "Название клиента"  
    [id_person] => "Ответственный менеджер"  
    [url_info] => Инфо.  
)
```
- iii. PUT ClientGroupInfoPut
  - iv. POST ClientGroupAdd
  - v. DELETE ClientGroupDelete

### c. ClientsObject

Описывает объекты клиентов АК

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`name` varchar(256) DEFAULT NULL,  
`address` varchar(256) DEFAULT NULL,  
`id_fias` int(11) DEFAULT NULL,  
`geo_lat` varchar(10) DEFAULT NULL,  
`geo_lon` varchar(10) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`note2` text DEFAULT NULL,  
`id_group` int(11) DEFAULT NULL,  
`timezone` smallint(6) NOT NULL DEFAULT 0
```

#### Доступные методы:

- i. GET ClientObjectInfo
  - ii. GET ClientObjectAll
- ```
(  
    [id] => "id"  
    [name] => "название объекта"  
    [address] => "Адрес текстовый"  
    [id_group] => "Название группы"  
    [url_info] => Инфо.  
)
```

- iii. PUT ClientObjectInfoPut
- iv. POST ClientObjectAdd
- v. DELETE ClientObjectDelete

## d. ClientsDepartment

Описывает отделы объектов клиентов АК

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_object` int(11) DEFAULT NULL,  
`name` varchar(256) DEFAULT NULL,  
`note` text DEFAULT NULL
```

### Доступные методы:

- i. GET ClientDepartmentInfo
- ii. GET ClientDepartmentAll

```
(  
  [id] => "id"  
  [name] => "название отдела"  
  [id_object] => "название объекта"  
  [url_info] => Инфо.  
)
```

- iii. PUT ClientDepartmentInfoPut
- iv. POST ClientDepartmentAdd
- v. DELETE ClientDepartmentDelete

## e. ClientsRate

Описывает ставки

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_speciality` int(11) DEFAULT NULL,  
`id_department` int(11) DEFAULT NULL,  
`rate_hour` int(11) DEFAULT NULL,
```

```
`hours` int(11) DEFAULT NULL,  
`rate_day` int(11) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`id_req` int(11) DEFAULT NULL,  
`id_type_unit` int(11) DEFAULT NULL,  
`rate_unit` int(11) DEFAULT NULL
```

**Доступные методы:**

- i. GET ClientRateInfo
- ii. GET ClientRateAll

```
(  
    [id] => "id"  
    [id_speciality] => "специализация"  
    [id_client] => "клиент"  
    [id_object] => "объект"  
    [id_department] => "отдел"  
    [url_info] => Инфо  
)
```

- iii. PUT ClientRateInfoPut
- iv. POST ClientRateAdd
- v. DELETE ClientRateDelete

## f. ClientsReq

Описывает базовые требования к ЛП

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`gender` enum('m','w') DEFAULT NULL,  
`age_after` int(11) DEFAULT NULL,  
`age_before` int(11) DEFAULT NULL,  
`national` enum('y','n') DEFAULT NULL,  
`passport` enum('y','n') DEFAULT NULL,  
`med` enum('y','n') DEFAULT NULL,  
`note` text DEFAULT NULL
```

**Доступные методы:**

- i. GET ClientReqInfo
- ii. GET ClientReqAll

```
(  
    [id] => "id"  
    [gender] => "пол"  
    [age_after] => "возраст от"  
    [age_before] => "возраст до"  
    [national] => "Национальность РФ Да/Нет"  
    [passport] => "Наличие паспорта Да/Нет"  
    [med] => "Наличие мед.книжки Да/Нет"  
    [url_info] => Инфо.  
)
```

- iii. PUT ClientReqInfoPut
- iv. POST ClientReqAdd
- v. DELETE ClientReqDelete

## g. TypeUnits

Описывает единицы измерений штучных работ, т.к. часы, коробки, и т.п.

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`state` enum('y','n') NOT NULL DEFAULT 'y',  
`name_full` varchar(256) DEFAULT NULL,  
`name` varchar(20) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL
```

### Доступные методы:

- i. GET typeUnitInfo
- ii. GET typeUnitAll

```
(  
    [id] => 1  
    [name_full] => Час  
    [name] => ч  
    [note] => 95% всех услуг  
    [url_info] => Инфо.  
)
```

- iii. PUT typeUnitInfoPut
- iv. POST typeUnitAdd
- v. DELETE typeUnitDelete

## 9. Модуль работы с линейным персоналом (исполнителями)

### IpMan

Описывает физические лица

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`sname` varchar(50) DEFAULT NULL,  
`name` varchar(50) DEFAULT NULL,  
`mname` varchar(50) DEFAULT NULL,  
`gender` enum('m','w') NOT NULL DEFAULT 'm',  
`bd` varchar(10) DEFAULT NULL,  
`foto` varchar(256) DEFAULT NULL,  
`tel` varchar(100) DEFAULT NULL,  
`tel2` varchar(100) DEFAULT NULL,  
`mail` varchar(70) DEFAULT NULL,  
`address` varchar(256) DEFAULT NULL,  
`id_fias` int(11) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`geo_lat` varchar(50) DEFAULT NULL,  
`geo_lon` varchar(50) DEFAULT NULL,
```

#### Доступные методы:

##### GET IpManInfo

**GET IpManAll** с дополнительным параметром id\_worker={id\_worker}

```
{  
  [id] => "id"  
  [fio] => "ФИО"  
  [tel] => "Телефон"  
  [url_info] => Инфо.  
}
```

##### PUT IpManInfoPut

**POST IpManAdd**  
**DELETE IpManDelete**

## IpWorker

Описывает исполнителей

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_man` int(11) DEFAULT NULL,  
`number` varchar(8) NOT NULL,  
`dt_begin` varchar(10) DEFAULT NULL,  
`dt_pause` varchar(10) DEFAULT "",  
`dt_end` varchar(10) DEFAULT NULL,  
`dt_candidate` varchar(10) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`note2` varchar(256) DEFAULT NULL
```

### Доступные методы:

**GET IpWorkerInfo**

**GET IpWorkerAll**

```
{  
    [id] => "id"  
    [id_man] => "ФИО физлица"  
    [number] => "Учетный номер"  
    [dt_begin] => "дата начала работы"  
    [dt_pause] => "Заморожен до"  
    [dt_candidate] => "Кандидат дата"  
    [url_info] => Инфо.  
}
```

**PUT IpWorkerInfoPut**

**POST IpWorkerAdd**

**DELETE IpWorkerDelete**

## IpContracts

Договора исполнителя

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`state` smallint(6) NOT NULL DEFAULT 1,  
`id_worker` int(11) DEFAULT NULL,  
`type` smallint(6) DEFAULT NULL,  
`edo` enum('y','n') NOT NULL DEFAULT 'n',
```

```
`dt_begin` date DEFAULT NULL,  
`dt_end` date DEFAULT NULL,  
`dt_contract` date DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL,  
`id_company` int(11) DEFAULT NULL
```

#### Доступные методы:

##### GET IpContractInfo

GET IpContractAll с дополнительным параметром id\_worker={id\_worker} опционально

```
{  
  [id] => 1  
  [id_company] => ОНО  
  [type] => Оферта  
  [active] => Да  
  [dt_begin] => 2023-01-01  
  [dt_end] => 2023-12-31  
  [url_info] => Инфо.  
}
```

PUT IpContractInfoPut

POST IpContractAdd

DELETE IpContractDelete

## IpSpeciality

Специализация исполнителя

#### Структура таблицы:

```
`id_worker` int(11) NOT NULL,  
`id_speciality` int(11) NOT NULL,  
`note` text DEFAULT NULL,  
UNIQUE KEY `ind1` (`id_speciality`,`id_worker`)
```

#### Доступные методы:

##### GET IpSpecialityInfo

GET IpSpecialityAll с дополнительным параметром id\_worker={id\_worker} опционально

```
{  
  [id_speciality] => 1  
  [name] => кассир-продавец  
  [note] => просто кассир  
  [url_info] => Инфо.  
}
```

PUT IpSpecialityInfoPut

POST IpSpecialityAdd



**DELETE IpSpecialityDelete**

# IpAccount

Описывает счета исполнителя

## Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_worker` int(11) DEFAULT NULL,  
`card` varchar(20) DEFAULT NULL,  
`bank_name` varchar(256) DEFAULT NULL,  
`bank_bik` varchar(9) DEFAULT NULL,  
`account` varchar(30) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`type` enum('b','n') NOT NULL DEFAULT 'b',  
`main` enum('y','n') NOT NULL DEFAULT 'y',  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL
```

## Доступные методы:

### GET IpAccountInfo

**GET IpAccountAll** с дополнительным параметром id\_worker={id\_worker}

```
{  
    [id] => "id"  
    [main] => "главный Да/Нет"  
    [type] => "Тип Наличн/Безнал"  
    [card] => "№ карты"  
    [account] => "№ счета"  
    [url_info] => Инфо.  
}
```

### PUT IpAccountInfoPut

**POST IpAccountAdd** с дополнительным параметром id\_worker={id\_worker}

(опционально)

### DELETE IpAccountDelete

# IpDoc

Документы физического лица

## Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`type` smallint(6) NOT NULL DEFAULT 1,  
`id_man` int(11) DEFAULT NULL,  
`ser` varchar(6) DEFAULT NULL,
```

```
`number` varchar(10) DEFAULT NULL,  
`kem` varchar(256) DEFAULT NULL,  
`dt` varchar(10) DEFAULT NULL,  
`podr` varchar(7) DEFAULT NULL,  
`mesto` varchar(256) DEFAULT NULL,  
`reg` varchar(256) DEFAULT NULL,  
`dt_reg` varchar(10) DEFAULT NULL,  
`reg_vr` varchar(256) DEFAULT NULL,  
`dt_reg_vr` varchar(10) DEFAULT NULL,  
`dt_bd` varchar(10) DEFAULT NULL,
```

#### Доступные методы:

##### GET IpDocInfo

**GET IpDocAll** с дополнительным параметром id\_man={id\_man}

```
{  
    [id] => 1  
    [type] => Паспорт  
    [ser] => 1234  
    [number] => 567890  
    [url_info] => Инфо.  
}
```

##### PUT IpDocInfoPut

**POST IpDocAdd** с дополнительным параметром id\_man={id\_man} (опционально)

##### DELETE IpDocDelete

## IpNote

Комментарии к физическому лицу

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`id_man` int(11) NOT NULL,  
`importance` smallint(6) DEFAULT NULL,  
`dt` timestamp NULL DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL
```

#### Доступные методы:

##### GET IpNoteInfo

**GET IpNoteAll** с дополнительным параметром id\_man={id\_man}

```
{  
    [id] => 1  
    [importance] => Обычный  
    [id_man] => Пупкин Василий
```

```
[note] => первый комментарий
[url_info] => Инфо.
```

```
}
```

**PUT IpNoteInfoPut**

**POST IpNoteAdd** с дополнительным параметром id\_man={id\_man} (опционально)

**DELETE IpNoteDelete**

## orders

Заказы

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`id_person_change` int(11) DEFAULT NULL,
`dt_change` timestamp NULL DEFAULT NULL,
`state` smallint(6) DEFAULT NULL,
`note` text DEFAULT NULL,
`id_client` int(11) NOT NULL DEFAULT 0
```

**Доступные методы:**

**GET OrderInfo**

**GET OrderAll**

```
{
```

```
[id] => 1
[state] => Да
[id_client] => магазин Петрович
[note] => кассир на график 2/2
[url_info] => Инфо.
```

```
}
```

**PUT OrderInfoPut**

**POST OrderAdd**

**DELETE OrderDelete**

## ordersReq

Требования к ЛП по заявкам

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`state` enum('y','n') NOT NULL DEFAULT 'y',
`id_order` int(11) DEFAULT NULL,
`id_object` int(11) DEFAULT NULL,
`attention` enum('y','n') NOT NULL DEFAULT 'n',
`dt_begin` datetime DEFAULT NULL,
`dt_end` datetime DEFAULT NULL,
```

```
`cnt` int(11) DEFAULT NULL,  
`gender` enum('m','w') DEFAULT NULL,  
`id_rate` int(11) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) NOT NULL DEFAULT 0
```

#### Доступные методы:

**GET OrderReqInfo**

**GET OrderReqAll**

```
{  
    [id] => 1  
    [id_request] => Заявка №1 заказа №1 ГиперРЫБА  
    [gender] => муж.  
    [age_after] => 18  
    [age_before] => 45  
    [national] => Нет  
    [passport] => Да  
    [med] => Да  
    [url_info] => Инфо.  
}
```

**PUT OrderReqInfoPut**

**POST OrderReqAdd**

**DELETE OrderReqDelete**

## requests

Заявки

#### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`state` enum('y','n') NOT NULL DEFAULT 'y',  
`id_order` int(11) DEFAULT NULL,  
`id_object` int(11) DEFAULT NULL,  
`attention` enum('y','n') NOT NULL DEFAULT 'n',  
`dt_begin` datetime DEFAULT NULL,  
`dt_end` datetime DEFAULT NULL,  
`cnt` int(11) DEFAULT NULL,  
`gender` enum('m','w') DEFAULT NULL,  
`id_rate` int(11) DEFAULT NULL,  
`note` text DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) NOT NULL DEFAULT 0,  
`note2` text DEFAULT NULL,
```

#### Доступные методы:

**GET RequestInfo**

**GET RequestAll**

```
{
  [id] => 1
  [state] => Нет(удален)
  [note2] => Внимание!
  [id_order] => 1
  [id_object] => ГиперРЫБА
  [dt_begin] => 2023-03-30 17:08:43
  [dt_end] => 2023-03-30 17:08:43
  [cnt] => 3
  [id_rate] => кассир-продавец / касса
  [url_info] => Инфо.
}
```

**PUT RequestInfoPut**

**POST RequestAdd**

**DELETE RequestDelete**

# shifts

Смены

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`state` enum('y','n') NOT NULL DEFAULT 'y',
`id_request` int(11) DEFAULT NULL,
`id_contract` int(11) DEFAULT NULL,
`dt_begin` datetime DEFAULT NULL,
`dt_end` datetime DEFAULT NULL,
`cnt_hour` int(11) DEFAULT NULL,
`cost_hour` int(11) DEFAULT NULL,
`cnt` int(11) DEFAULT NULL,
`note` text DEFAULT NULL,
`dt_change` timestamp NULL DEFAULT NULL,
`id_person_change` int(11) NOT NULL DEFAULT 0,
`dt_plan` timestamp NULL DEFAULT NULL,
`id_person_plan` int(11) DEFAULT NULL,
`dt_remove` timestamp NULL DEFAULT NULL,
`id_person_remove` int(11) DEFAULT NULL,
`id_type_unit` int(11) DEFAULT NULL
```

**Доступные методы:**

**GET shiftsInfo**

**GET shiftsAll**

```
{
```

```
[id] => 1
[state] => Да
[id_request] => 2
[id_contract] => Оферта Пупкин Василий
[url_info] => Инфо.
```

```
}
```

**PUT shiftsInfoPut**

**POST shiftsAdd**

**DELETE shiftsDelete**

## payPay

Выплаты исполнителям

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`state` enum('y','n') DEFAULT 'y',
`id_account` int(11) DEFAULT NULL,      -ссылка на счет АК
`id_lp_account` int(11) DEFAULT NULL,   -ссылка на счет исполнителя
`sum` int(11) DEFAULT NULL,
`note` varchar(256) DEFAULT NULL,
`dt_pay` timestamp NULL DEFAULT NULL,
`id_person_pay` int(11) DEFAULT NULL,
`dt_signature` timestamp NULL DEFAULT NULL,
`id_person_signature` int(11) DEFAULT NULL,
```

### Доступные методы:

#### GET payPayInfo

Дополнительно в {data} выводится детализация оплат, на которые использовалась данная выплата в формате:

[detail] => Array

```
(
  [0] => Array
    (
      [0] => id операции баланса
      [1] => дата операции
      [2] => активность
      [3] => тип оплаты
      [4] => договор
      [5] => сумма оплаты
      [6] => расшифровка оплаты
      [7] => объект
      [8] => сумма, учтенная из выплаты на оплату
      [9] => дополнительные данные (опционально)
    )
)
```

)

### GET payPayAll

```
{
  [id] => 1
  [dt_pay] => 2023-04-07 10:37:40
  [id_account] => Касса
  [id_lp_account] => 00000001 Пупкин Василий на руки
  [sum] => 1500
  [note] => 1500р за две смены (950р. + 550р частично от второй, сколько
было в кошельке))
  [url_info] => Инфо.
}
```

### PUT payPayInfoPut

### PUT payPaySignature

Подтверждение выплаты. Меняются id\_person\_signature на текущего пользователя и dt\_signature на текущую дату/время

### POST payPayAdd

### DELETE payPayDelete

## payBalance

Денежные операции начислений

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`dt` datetime DEFAULT NULL,
`state` enum('y','n') DEFAULT 'y',
`id_type` int(11) DEFAULT NULL,
`id_contract` int(11) DEFAULT NULL,
`id_operation` int(11) DEFAULT NULL,
`name_operation` varchar(50) DEFAULT NULL,
`sum` int(11) DEFAULT 0,
`note` text DEFAULT NULL,
`dt_change` timestamp NULL DEFAULT NULL,
`id_person_change` int(11) DEFAULT NULL,
`sum2` int(11) DEFAULT 0,
`dt_signature` timestamp NULL DEFAULT NULL,
`id_person_signature` int(11) DEFAULT NULL,
`id_rate` int(11) DEFAULT NULL
```

### Доступные методы:

### GET payBalanceInfo

**GET payBalanceNow** с дополнительным параметром id\_man={id\_man}.

Показывает текущий баланс исполнителя .

```
{
```

```
[balance] => 1350
}
```

**GET payBalanceRate** с дополнительным параметром id\_rate={id ставки}, id\_shift={id смены}.

Вычисляет сумму вознаграждения по отработанной смене в зависимости от ставки.

```
{
  [summ] => 1350
}
```

**GET payBalanceAll**

```
{
  [id] => 1
  [dt] => 2023-04-04 09:02:04
  [state] =>
  [id_type] => Вознаграждение (Исполнителю за смену)
  [id_operation] => Смена №1 2023-03-31 08:00:00 - 2023-04-15 16:00:00
  [id_contract] => Оферта Пупкин Василий
  [sum] => 1000
  [url_info] => Инфо.
}
```

**PUT payBalanceInfoPut**

**PUT payBalanceSignature**

Заверение операции начисления. Меняются id\_person\_signature на текущего пользователя и dt\_signature на текущую дату/время

**POST payBalanceAdd**

**DELETE payBalanceDelete**

## payTypes

Типы денежных операций

**Структура таблицы:**

```
`id` int(11) NOT NULL AUTO_INCREMENT,
`state` enum('y','n') NOT NULL DEFAULT 'y',
`type` smallint(6) NOT NULL DEFAULT 1,
`name` varchar(256) DEFAULT NULL,
`note` text DEFAULT NULL,
`dt_change` timestamp NULL DEFAULT NULL,
`id_person_change` int(11) DEFAULT NULL,
`name_operation` varchar(50) DEFAULT NULL
```

**Доступные методы:**

**GET payTypeInfo**

**GET payTypeAll**

```
{
  [id] => 1
  [state] => Да
}
```



[type] => Увеличение  
[name] => Вознаграждение (Исполнителю за смену)  
[url\_info] => Инфо.

}

**PUT payTypeInfoPut**

**POST payTypeAdd**

**DELETE payTypeDelete**

## payDetail

Детализация денежных операций

### Структура таблицы:

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`state` enum('y','n') NOT NULL DEFAULT 'y',  
`id_pay_pay` int(11) DEFAULT NULL,  
`id_pay_balance` int(11) DEFAULT NULL,  
`sum` int(11) DEFAULT NULL,  
`dt_change` timestamp NULL DEFAULT NULL,  
`id_person_change` int(11) DEFAULT NULL
```

### Доступные методы:

**POST payTypeAdd**