# 

# 

# 

# Сервис расписания доставки сим-карт Подключи

**[Сервис расписания доставки сим-карт Подключи](#_1v447qz8hlec) 1**

**[1. Руководство по развертыванию сервиса](#_ed5dr4rzxuwy) 2**

[1.1. Создание окружения](#_ma2t9izne7m) 2

[1.2. Установка зависимостей](#_y60e6m9lni49) 2

[1.4. Создание таблиц бд](#_g5neqo9a19ui) 3

[1.5. Запуск](#_43b30ztuc070) 3

[1.5.1. Запуск сервера](#_798kkele4afi) 3

[1.5.2. Запуск модульных тестов загрузки файлом (тесты разметки):](#_bv7xz0y399il) 3

**[2. Руководство по использованию API](#_no5vfqqap30p) 4**

[2.1. Доступ](#_xj8k5uw8rc) 4

[2.2. Методы API](#_9g4ve3kbnrgc) 4

[2.2.3. Загрузить слоты доставки файлом](#_q0nk2sqgm17v) 4

[2.2.4. Получить список временных слотов для доставки](#_t09h1vk1h0iw) 5

[2.2.5. Получить слоты доставки файлом](#_pzeeqpuwv9el) 5

[2.2.6. Пример ответов с ошибкой (во всех методах):](#_kgpvxqh9xtph) 5

# 

# 1. Руководство по развертыванию сервиса

## 1.1. Создание окружения

sudo -E python3 -m venv ./

## 1.2. Установка зависимостей

sudo -E yum install gmp-devel

sudo rm -rf /tmp/pip-\*

export LD\_LIBRARY\_PATH="/usr/lib64:$LD\_LIBRARY\_PATH"

sudo -E yum install python-devel

sudo -E bin/python -m pip install -r requirements.txt

1.3. Настройки

создать файл [settings.py](http://settings.py/) в корне проекта

пример файла настроек с необходимыми параметрами:

# settings.py

DEBUG = True

PROCESSES\_NUM = 1

DB\_PATH = 'postgresql://<db-user>:<password>@<db\_ip>:<db\_port>/<db\_name>'

SERVER\_SCHEMA = 'https'

SERVER\_HOST = '<server url>'

PORT = SERVER\_PORT = 9008

SERVER\_URL = '{}://{}'.format(SERVER\_SCHEMA, SERVER\_HOST)

SERVER\_AUTH\_TOKENS = {

'<токен для чтения>': 'read',

'<токен для записи>': 'write',

}

## 1.4. Создание таблиц бд

bin/python sync\_db.py

## 1.5. Запуск

### 1.5.1. Запуск сервера

bin/python server.py

### 1.5.2. Запуск модульных тестов загрузки файлом (тесты разметки):

bin/python tests.py

# 2. Руководство по использованию API

## 2.1. Доступ

Методы API вызываются с заголовком ServerAuthorization, в котором должен быть токен доступа на запись либо на чтение

Пример вызова:

curl -X POST -H "ServerAuthorization: "<access\_token>" -H "Content-Length: 116" -d '{"region\_id": "45000000000", "local\_time": "2021.04.01T16:10:00", "subject": "Москва"}' 'http://10.77.35.96:9028/api/v1/delivery\_date\_times'

## 2.2. Методы API

### 2.2.3. Загрузить слоты доставки файлом

POST /api/v1/upload\_dates.xls

->

<xls или xlsx файл>

<-

{

updated\_delivery\_dates\_count: int, // обновлено записей

added\_delivery\_dates\_count: int // добавлено записей

}

### 2.2.4. Получить список временных слотов для доставки

POST /api/v1/delivery\_date\_time

->

{

region\_id: string // код окато (необязательный параметр) # '41000000000'

local\_time: string // день для которого необходимо расписание # '2021.02.14T16:10:00'

subject: string // наименование города (необязательный параметр) # 'Москва'

}

Без уточнения региона и города (subject, region\_id):

<-

{

"result": 1,

"error": 0,

"delivery\_info\_text": {okato: text, ...}, // текст для отображения

"delivery\_prices": {okato: price, '''}, // цены на доставку

"default\_region": string, // Код ОКАТО по-умолчанию

"delivery\_regions": [{'id' : string, 'name': string}] // список регионов с наименованиями городов

}

С регионом и городом (subject, region\_id):

<-

{

"result": 1,

"error": 0,

"delivery\_times": {"DD.MM.YYYY": ["HH:MM", "HH:MM", ...], ...}, // cлоты доставки

"delivery\_regions": [{'id' : string, 'name': string}] // список регионов с наименованиями городов

'price': int // цена дотавки в указанном регионе

}

### 2.2.5. Получить слоты доставки файлом

POST /api/v1/dates.xls

<- <xlsx файл>

### 2.2.6. Пример ответов с ошибкой (во всех методах):

{

'error': 1,

'reply': 'Текст ошибки'

}