# **Описание проекта**

Сервис, содержащий базу данных с людьми, возможностью добавления новых людей, редактирования и удаления.

Сервис базируется на API проекта HelpTense.

Тестирование функционала проводится путем операций с БД, по работе с пользователями

# **База данных**

База данных выглядит следующим образом:

**Таблица Person**:

Здесь содержится вся информация о пользователе, ФИО, ссылка на Адрес, а также такие поля как пол, номер телефона, почта и дата рождения.

**Таблица Address**:

Содержит полный адрес, имеет возможность повторного использования, у разных людей может быть один адрес.

# **Функционал API**

Для обращений к API прописаны следующие адреса:

**Для людей**:

/person/create/ - создание человека

/person/get/ - получение информации о человеке по ID  
/person/getAll/ - получение информации о всех людях  
/person/update/ - обновление человека  
/person/delete/ - удаление человека  
/person/search/ - поиск человека

Теперь рассмотрим каждый запрос подробнее.

## **/person/create/ - создание человека**

**Через POST-запрос передаются**:

- First\_Name  
- Last\_Name  
- Patronimic\_Name  
- Address\_Line\_1  
- Address\_Line\_2

- City

- Country

- ZIP

- Phone

- DOB

- Gender

**Варианты ответа (JSON)**:

При успешном добавлении:

{

'status': true,  
'message': 'person successfully created'

}

Если человек уже существует, а именно – пара: ФИО+Mail:

{

'status': false,  
'message': ‘person created error'

}

**Пример переданных данных**:

Данные:

{‘First\_Name’ => ‘John’, ‘Last\_Name’ => ‘Doe’, ‘Patronimic\_Name’ => ‘Zombie’, ‘Address\_Line\_1’ => ‘Capital st’, ‘Address\_Line\_2’ => ’54-21’, ‘City’ => ’London’, ‘Phone’ => ’3242354’, ‘DOB’ => ’1987-12-11’, ‘Gender’ => ’M’}

Ответ:

{‘status’: true, ‘message’: ‘person successfully created’}

## **/person/get/ - получение глагола**

**Через POST-запрос передаются**:

- id

**Варианты ответа (JSON)**:

При успешном выполнении:

{

'status': true,  
'message': ‘mail found'  
‘data’:[…]

Если человек с запрашиваемым mail не найден:

{

'status': false,  
'message': 'mail not found'

}

**Пример переданных данных**:

Данные:

{‘mail’ => ‘sher@gmail.com’}

Ответ:

{"status": true, "message": "mail found", "data": {"Person\_ID":"2","Name\_ID":"2","Address\_ID":"2","Phone":"12431897","Mail":"sher@gmail.com","DOB":"1865-06-03"}]}

## **/person/getAll/ - получение полного списка людей**

**Варианты ответа (JSON)**:

{

'status': true,  
'message': ‘'  
‘data’:[…]

**Пример:**

Ответ:

{"status": true, "data": [{"Person\_ID":"1","Name\_ID":"1","Address\_ID":"1","Phone":"12831247","Mail":"grimm@index.com","DOB":"1967-12-09"},{"Person\_ID":"2","Name\_ID":"2","Address\_ID":"2","Phone":"12431897","Mail":"sher@gmail.com","DOB":"1865-06-03"}]}

## **/person/update/ - обновление человека**

**Через POST-запрос передаются**:

- id

+ одно или несколько из нижеперечисленных:

- First\_Name  
- Last\_Name  
- Patronimic\_Name  
- Address\_Line\_1  
- Address\_Line\_2

- City

- Country

- ZIP

- Phone

- DOB

- Gender

**(JSON)**:

При успешном выполнении:

{

'status': true,  
'message': ‘person update’

}

При ошибке выполнения:

{

'status': false,  
'message': 'update error'

}

**Пример переданных данных**:

Данные: {‘id’ => ‘1’, ‘DOB’ => ‘2000-08-09’}

Ответ: {"status": true, "message": "person update"}

## **/verbs/delete/ - удаление человека**

**Через POST-запрос передаются**:

- id

**Варианты ответа (JSON)**:

При успешном выполнении:

{

'status': true,  
'message': ' person deleted'

}

При ошибке выполнения:

{

'status': false,  
'message': ' person deleted error'

}

**Пример переданных данных**:

Данные: {‘id’ => ‘1’ }

Ответ: {"status": true, "message": "person deleted"}