# Техническое описание проекта

## 1. Введение

Сервис состоит из следующих основных компонент:

- Gateway сервис, реализующий механизм аутентификации и авторизации

пользователей с использованием токенов доступа

- RESTful API сервис для взаимодействия с пользователями

- Сервис логики, реализующий механизм приема, расчета и распределения заявок на обслуживание маломобильных пассажиров

- WEB-сервис, реализующий интерфейс взаимодействия пользователя с внутренними сервисами.

## 2. Требования к аппаратному и программному обеспечению

Аппаратное обеспечение:

- Сервер: Процессор: одноядерный, 1.4 ГГц или выше. · ОЗУ: 2 Гб, Диск: 10 ГБ. · ОС: Ubuntu 20.04 LTS

Программное обеспечение:

- СУБД SqLite

- ReactJS

## 3. Технологический стек

ReactJS – front-end

Golang – back-end API-сервис

Rust – back-end Gateway сервис и сервис логики

SqLite - СУБД

## 4. Архитектура системы

- API Gateway выступает в качестве центральной точки входа для внешних запросов и обеспечивает единый интерфейс для доступа к различным его компонентам. Он решает такие задачи, как маршрутизация запросов, аутентификация, авторизация, ограничение частоты запросов и логирование всех действий пользователей. API Gateway упрощает разработку, повышает производительность и общую безопасность приложения. Так же он поддерживает автоматическую регистрацию новых сервисов.

- API сервис для взаимодействия с пользователями обеспечивает обмен информацией Gateway сервиса, сервиса логики и front-end c базой данных.

## 5. Инструкции по использованию

### 5.1. API Service - описание доступных API методов и их параметров

#### Назначение

* API функционал сервера, работа с данными сотрудников, пассажиров.
* API сервис реализован на Golang и связан следующими зависимостями:

go get -u github.com/lib/pq

go get -u github.com/gorilla/mux

go get -u modernc.org/sqlite

go get -u github.com/samborkent/uuidv7

go get -u github.com/rs/cors

go get -u golang.org/x/crypto/blake2b

#### Api –методы

*  [metod post /employee/set](Метод регистрации нового сотрудника метрополитена)
*  [metod post /employee/update](Метод изменения информации о сотруднике)
*  [metod get /employee/list](Метод получения списка всех сотрудников, работающих по заявкам)
*  [metod delete /employee/delete](Метод удаления сотрудника)
*  [metod post /passenger/set](Метод регистрации нового пассажира)
*  [metod post /passenger/update](Метод изменения информации о пассажире)
*  [metod get /passenger/list](Метод получения списка всех зарегистрированных пассажиров)
*  [metod get /passenger](Метод поиска пассажира по ФИО или по номеру телефона)
*  [metod delete /passenger/delete](Метод удаления пассажира)
*  [metod post /category/set](Метод создания новой категории пассажиров)
*  [metod get /category/list](Метод получения списка всех всех категорий пассажиров для выбора при заполнении заявки)
*  [metod get /uchastok/list](Метод получения списка всех участков обслуживания сотрудниками)
*  [metod get /rank/list](Метод получения списка всех должностей сотрудников)
*  [metod post /account/set](Метод создания аккаунта для сотрудника)
*  [metod get /account](Метод получения Id сотрудника при запросе по логину и паролю)
*  [metod get /account/exist](Метод проверки логина на уникальность, существует или нет такой логин в БД)
*  [metod get /order/set](Метод создания новой заявки на обслуживание пассажиров)
*  [metod post /order/update](Метод изменения заявки на обслуживание пассажиров)
*  [metod get /order/list](Метод получения списка всех заявок на обслуживание пассажиров)
*  [metod delete /order/delete](Метод удаления заявки на обслуживание пассажиров)
*  [metod get /order/state/list](Метод получения списка всех статусов заявок)
*  [metod get /role/list](Метод получения списка всех значений ролей пользователей)
*  [metod post /workday/set](Метод регистрации рабочего дня сотрудника, выполняющего заявки)
*  [metod post /workday/upgate](Метод внесения изменений в рабочий день сотрудника, выполняющего заявки)
*  [metod get /roworkday/list](Метод получения списка всех рабочих дней для сотрудника с заданным id)

##### /employee/set

входные параметры:

json в body:

{

"date": "25.04.2024",

"time\_work": "20:00-08:00",

"fio": "Тестовых В.И.",

"uchastok": "018fe823-9419-7fe0-a8cb-19db345ec1ac",

"smena": "2Н",

"rank": "ЦИ",

"sex": "Мужской",

"phone\_work": "84959112233",

"phone\_personal": "89119876543",

"tab\_number": "46573645",

"type\_work": "лёгкая работа"

}

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"id": "019010b6-da44-7602-8294-cf8e529f7680",

"message": "Добавлен сотрудник: Тестовитов В.И."

}

}

при повторном добавлении сотрудника статус 417 и ошибка:

Ошибка добавления сотрудника: constraint failed: UNIQUE constraint failed: employees.id (1555)

##### /employee/update

входные параметры:

id - уникальный идентификатор сотрудника

и json в body:

{

"id": "019010b6-da44-7602-8294-cf8e529f7680",

"date": "25.04.2024",

"time\_work": "20:00-08:00",

"fio": "Тестовcкий В.И.",

"uchastok": "018fe823-9419-7fe0-a8cb-19db345ec1ac",

"smena": "2Н",

"rank": "ЦИ",

"sex": "Женский",

"phone\_work": "84959112233",

"phone\_personal": "89119876543",

"tab\_number": "46573645",

"type\_work": "лёгкая работа"

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "018fe832-eb8f-7d95-b1d3-e22306c5bf23"

}

где id - уникальный идентификатор сотрудника

при попытке изменить ФИО сотрудника на ФИО, уже имеющийся в БД, будет статус 417 и ошибка:

Ошибка изменения информации о сотруднике: constraint failed: UNIQUE constraint failed: employees.fio (2067)

##### /employee/list

входные параметры: limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"total\_count": 118,

"page\_count": 6,

"details": [

{

"date": "23.04.2024",

"time\_work": "20:00-08:00",

"id": "018fee07-b2d2-7eff-8e2e-c140187a569b",

"fio": "Болотов Г.Е.",

"uchastok": "ЦУ-3 (Н)",

"smena": "2Н",

"rank": "ЦСИ",

"sex": "Мужской",

"phone\_work": "",

"phone\_personal": "",

"tab\_number": "",

"type\_work": ""

},

{

"date": "23.04.2024",

"time\_work": "20:00-08:00",

"id": "018fee07-b3b1-7f37-b8be-079ffbfd136e",

"fio": "Шварева М.В.",

"uchastok": "ЦУ-3 (Н)",

"smena": "2Н",

"rank": "ЦСИ",

"sex": "Женский",

"phone\_work": "",

"phone\_personal": "",

"tab\_number": "",

"type\_work": ""

}

]

}

}

##### /employee/delete

входные параметры:

id - уникальный идентификатор сотрудника

на выходе при успешном выполнении статус 200 и json:

{

"id": "018fe832-eb8f-7d95-b1d3-e22306c5bf24"

}

где id - уникальный идентификатор сотрудника

при попытке удалить сотрудника с несуществующим id будет статус 417 и ошибка:

Ошибка изменения информации о сотруднике: constraint failed: UNIQUE constraint failed: employees.fio (2067)

##### /passenger/set

на входе:

{

"id": "018fe832-eb8f-7d95-b1d3-e22306c5bf23",

"fio": "Пассажиркин П.П.",

"phone": "79871234567",

"category": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"sex": "Мужской",

"description": "",

"eks": 0

}

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"id": "019010a5-ff0a-7d0c-b3dc-974a0ba5a1e7",

"message": "Добавлен пассажир: Пассажирова П.П."

}

}

при повторном добавлении пассажира статус 417 и ошибка:

Ошибка добавления пассажира: constraint failed: UNIQUE constraint failed: passengers.fio, passengers.phone (2067)

##### /passenger/update

входные параметры:

json в body:

{

"id": "018fe832-eb8f-7d95-b1d3-e22306c5bf23",

"fio": "Пассажиркин П.П.",

"phone": "79871234567",

"category": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"sex": "Мужской",

"description": "",

"eks": 0

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "018fe864-84b3-7656-a270-6a57f43821c4"

}

где id - уникальный идентификатор пассажира

при попытке изменить ФИО или телефон пассажира на ФИО или телефон, уже имеющиеся в БД, будет статус 417 и ошибка:

Ошибка изменения информации о пассажире: constraint failed: UNIQUE constraint failed: passengers.fio, passengers.phone (2067)

##### /passenger/list

входные параметры: limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"total\_count": 2,

"page\_count": 1,

"details": [

{

"id": "018fe864-84b3-7656-a270-6a57f43821c4",

"fio": "Пассажиров П.П.",

"phone": "79871234567",

"category": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"sex": "Мужской",

"eks": 0

},

{

"id": "018fe866-ecc1-7ee5-90d6-7f6c84b1d4a1",

"fio": "Пассажиркин П.П.",

"phone": "79871234567",

"category": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"sex": "Мужской",

"description": "Дополнительная информация",

"eks": 0

}

]

}

}

##### /passenger

входные параметры: fio - ФИО пассажира и/или phone - телефон пассажира

на выходе при успешном выполнении статус 200 и json:

{

"id": "018fe864-84b3-7656-a270-6a57f43821c4",

"fio": "Пассажиров П.П.",

"phone": "79871234567",

"category": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"sex": "Мужской",

"description": "",

"eks": 0

}

если пассажир не найден в бд статус 417 и ошибка:

Нет такого пассажира в БД: sql: no rows in result set

##### /passenger/delete

входные параметры:

id - уникальный идентификатор пассажира

на выходе при успешном выполнении статус 200 и json:

{

"id": "018fe864-84b3-7656-a270-6a57f43821c4"

}

где id - уникальный идентификатор пассажира

##### /category/set

входные параметры: cat - наименование категории пассажиров

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"state": "018fe8e5-6b4c-7e88-a0b6-0f200e209ba7",

"message": "Добавлена категория пассажиров: Наименование категории пассажиров"

}

}

при повторном добавлении одного и того же пассажира статус 417 и ошибка:

Ошибка добавления категории пассажиров: constraint failed: UNIQUE constraint failed: category\_dictionary.category (2067)

##### /category/list

входных параметров нет

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"details": [

{

"id": "018fe832-ed6a-7150-8aae-cc3596b11ec9",

"category": "ИЗТ – Инвалид по зрению (тотальный, сопровождение по метрополитену)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b12ec9",

"category": "ИЗ – Инвалид по зрению с остаточным зрением (слабовидящий, сопровождение по метрополитену)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b13ec9",

"category": "ИС – Инвалид по слуху (в основном помощь в ориентировании)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b14ec9",

"category": "ИК – Инвалид колясочник (передвижение в инвалидной коляске)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b15ec9",

"category": "ИО – Инвалид опорник (необходима поддержка при передвижении и/или на лестницах/эскалаторах)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b16ec9",

"category": "ДИ – Ребенок инвалид (зачастую передвижение в инвалидной коляске)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b17ec9",

"category": "ПЛ – Пожилой человек (необходима поддержка при передвижении и/или на лестницах/эскалаторах)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b18ec9",

"category": "РД – Родители с детьми (сопровождение ребенка)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b19ec9",

"category": "РДК – Родители с детскими колясками (помощь с детской коляской)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b20ec9",

"category": "ОГД – Организованные группы детей (сопровождение по метрополитену)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b21ec9",

"category": "ОВ – Временно маломобильные (после операции, переломы и прочее)"

},

{

"id": "018fe832-ed6a-7150-8aae-cc3596b22ec9",

"category": "ИУ – Люди с ментальной инвалидностью"

},

{

"id": "018fe8e5-6b4c-7e88-a0b6-0f200e209ba7",

"category": "Наименование категории пассажиров"

}

]

}

}

##### /uchastok/list

входных параметров нет

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"details": [

{

"id": "018fe822-8929-7491-a897-e960f85d78fe",

"uchastok": "ЦУ-3 (Н)"

},

{

"id": "018fe823-68f8-781e-ac9b-5223c8c5f1d1",

"uchastok": "ЦУ-4 (Н)"

},

{

"id": "018fe823-7156-73c8-9c1a-c1f2a98fa72e",

"uchastok": "ЦУ-2"

},

{

"id": "018fe823-7bea-78fb-853c-eee11bfab2ae",

"uchastok": "ЦУ-1"

},

{

"id": "018fe823-84d5-7327-bf76-21ef41682534",

"uchastok": "ЦУ-8"

},

{

"id": "018fe823-8bc0-751c-a0aa-02fec47656e1",

"uchastok": "ЦУ-3"

},

{

"id": "018fe823-9419-7fe0-a8cb-19db323ec1ac",

"uchastok": "ЦУ-4"

},

{

"id": "018fe823-a158-7f75-96e9-bf36b4aeb76a",

"uchastok": "ЦУ-5"

}

]

}

}

##### /rank/list

входных параметров нет

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"details": [

{

"id": "018fed9c-d0f4-7a6c-af67-71bafee99c03",

"rank": "ЦСИ"

},

{

"id": "018fed9c-d1d6-7dd4-9b63-3a2090a1ad5c",

"rank": "ЦИ"

},

{

"id": "018fed9c-d2cd-7bc5-a444-bdcb0e6280c0",

"rank": "ЦУ"

}

]

}

}

##### /account/set

входные параметры:

json в body:

{

{

"id": "01901686-bfc4-71ea-a7e3-08b4851fb1e9",

"login": "loginQWE@26",

"pass": "passQWE@26"

}

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "01901686-bfc4-71ea-a7e3-08b4851fb1e9"

}

при повторном добавлении аккаунта статус 417 и ошибка:

Ошибка добавления аккаунта: constraint failed: UNIQUE constraint failed: accounts.login (2067)

##### /account

входные параметры:

json в body:

{

"id": "019010d7-d57b-70b1-9398-f2613602beaf"

}

В БД пароль хранится зашифрованным blake2b

В случае если сотрудник с таким логином и паролем найден в БД тогда на выходе статус 200 и json:

{

"id": "018fee07-bbfb-7446-aca3-6c0a66505937",

"fio": "Бушуев А.А.",

"role": "018ffd91-fcc5-70a9-bf36-f67ca6df0ca0"

}

Если один или оба параметра в запросе не были заданы тогда на выходе статус 400 и ошибка:

Parameters not found!

где id - это уникальный идентификатор сотрудника

Если сотрудник с таким логином и паролем не найден (отсутствует в БД) тогда на выходе статус 417 и ошибка:

Неверный логин или пароль: sql: no rows in result set

Если для найденного в БД сотрудника был указан неверный пароль тогда на выходе статус 417 и ошибка:

Неверный пароль!

##### /account/exist

входные параметры:

login - логин сотрудника (в открытом виде)

В случае если такой логин уже существует тогда статус 200 и json:

{

"id": "018fee07-bbfb-7446-aca3-6c0a66505937"

}

где id - это уникальный идентификатор сотрудника

Если параметр в запросе не был задан тогда на выходе статус 400 и ошибка:

Parameter not found!

Если такой логин не найден (отсутствует в БД) тогда на выходе статус 417 и ошибка:

Указанный логин не найден в БД: sql: no rows in result set

##### /order/set

входные параметры:

json в body:

{

"id": "",

"id\_pas": "11058234",

"datetime": "24.04.2024 7:30:00",

"time3": "07:13:52",

"time4": "07:51:11",

"cat\_pas": "ИЗТ",

"status": "Заявка закончена",

"tpz": "15.03.2024 22:48:43",

"INSP\_SEX\_M": "0",

"INSP\_SEX\_F": "1",

"TIME\_OVER": "00:52:20",

"id\_st1": "5",

"id\_st2": "97"

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "019010d7-d57b-70b1-9398-f2613602beaf"

}

при повторном добавлении заявки с тем же id статус 417 и ошибка:

Ошибка добавления заявки: constraint failed: UNIQUE constraint failed: orders.id (1555)

##### /order/update

входные параметры:

json в body:

{

"id": "019010d7-d57b-70b1-9398-f2613602beaf",

"id\_pas": "11058234",

"datetime": "14.06.2024 7:30:00",

"time3": "07:13:52",

"time4": "07:51:11",

"cat\_pas": "018fe832-ed6a-7150-8aae-cc3596b14ec9",

"status": "018ffea0-3cff-7738-8f27-ed3c9bc843a9",

"tpz": "15.03.2024 22:48:43",

"INSP\_SEX\_M": "3",

"INSP\_SEX\_F": "1",

"TIME\_OVER": "00:52:20",

"id\_st1": "5",

"id\_st2": "97"

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "019010d7-d57b-70b1-9398-f2613602beaf"

}

где id - уникальный идентификатор заявки

##### /order/list

входные параметры: limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"total\_count": 3,

"page\_count": 1,

"details": [

{

"id": "477354234",

"id\_pas": "11058234",

"datetime": "477354234",

"time3": "07:13:52",

"time4": "07:51:11",

"cat\_pas": "ИЗТ",

"status": "Заявка закончена",

"tpz": "15.03.2024 22:48:43",

"insp\_sex\_m": "0",

"insp\_sex\_f": "1",

"time\_over": "00:52:20",

"id\_st1": "5",

"id\_st2": "97"

},

{

"id": "477354235",

"id\_pas": "11058234",

"datetime": "477354235",

"time3": "07:13:52",

"time4": "07:51:11",

"cat\_pas": "ИЗТ",

"status": "Заявка закончена",

"tpz": "15.03.2024 22:48:43",

"insp\_sex\_m": "0",

"insp\_sex\_f": "1",

"time\_over": "00:52:20",

"id\_st1": "5",

"id\_st2": "97"

},

{

"id": "477354236",

"id\_pas": "11058234",

"datetime": "477354236",

"time3": "07:13:52",

"time4": "07:51:11",

"cat\_pas": "ИЗТ",

"status": "Заявка закончена",

"tpz": "15.03.2024 22:48:43",

"insp\_sex\_m": "0",

"insp\_sex\_f": "1",

"time\_over": "00:52:20",

"id\_st1": "5",

"id\_st2": "97"

}

]

}

}

##### /order/delete

входные параметры:

id - уникальный идентификатор заявки

на выходе при успешном выполнении статус 200 и json:

{

"id": "019010dc-3a7f-7868-ba1d-6d2e636da0ea"

}

где id - уникальный идентификатор удаленной заявки

##### /order/state/list

входных параметров нет

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"details": [

{

"id": "018ffea0-3a7e-7741-b55b-3efacd17ae2c",

"state": "Не подтверждена"

},

{

"id": "018ffea0-3c19-7a7c-8b1d-9461cc0742a6",

"state": "В рассмотрении"

},

{

"id": "018ffea0-3cff-7738-8f27-ed3c9bc843a9",

"state": "Принята"

},

{

"id": "018ffea0-3dd1-70bc-b23d-cbbe44fd1a5f",

"state": "Инспектор выехал"

},

{

"id": "018ffea0-3f22-7c74-82e1-daec00bcfa44",

"state": "Инспектор на месте"

},

{

"id": "018ffea0-4018-7666-b68a-c223bfe90e16",

"state": "Поездка"

},

{

"id": "018ffea0-416e-77f6-acba-cc8cba4ac2ff",

"state": "Заявка закончена"

},

{

"id": "018ffea0-429a-719e-898a-291e7585a408",

"state": "Выявление"

},

{

"id": "018ffea0-43ad-7927-b60f-7414c005eddb",

"state": "Лист Ожидания"

},

{

"id": "018ffea0-4505-7bbf-801f-74c98e1bcd02",

"state": "Отмена"

},

{

"id": "018ffea0-45d8-72d8-b3d5-8a46ba6a75ca",

"state": "Отказ"

},

{

"id": "018ffea0-46db-7ebe-ac13-1e6be465807c",

"state": "Пассажир опаздывает"

},

{

"id": "018ffea0-47f9-754f-884f-41ae979b33c3",

"state": "Инспектор опаздывает"

}

]

}

}

##### /role/list

входные параметры: limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"details": [

{

"id": "018ffd91-effd-7a98-8289-78f327a6903e",

"role": "Администратор"

},

{

"id": "018ffd91-fb97-7cf1-a105-43cd25b783e2",

"role": "Специалист"

},

{

"id": "018ffd91-fc38-759a-9d8c-32b6b0419de4",

"role": "Оператор"

},

{

"id": "018ffd91-fcc5-70a9-bf36-f67ca6df0ca0",

"role": "Сотрудник"

}

]

}

}

##### /workday/set

входные параметры:

json в body:

{

"id": "",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "14.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "01901161-1f38-7c64-a748-d56267cca716"

}

##### /workday/update

входные параметры:

json в body:

{

"id": "01901160-bf0b-7c72-ba93-2158a7694cb8",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "14.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

}

на выходе при успешном выполнении статус 200 и json:

{

"id": "01901160-bf0b-7c72-ba93-2158a7694cb8"

}

где id - уникальный идентификатор рабочего дня сотрудника

##### /workday/list

входные параметры: id - уникальный id сотрудника limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"total\_count": 9,

"page\_count": 1,

"details": [

{

"id": "01901160-bf0b-7c72-ba93-2158a7694cb8",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "10.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "выходной",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "01901160-e4cb-72c7-8611-9c4d95b5469a",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "11.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "01901160-fb3e-7af6-8e47-993a064a36ad",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "12.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "01901161-0cc8-78c2-a100-4cb11699bf1c",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "13.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "01901161-1f38-7c64-a748-d56267cca716",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "14.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019011ef-d7cc-7a2c-bb85-d58778039f8c",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "09.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019011ef-ee38-7829-bf16-07073b0cafab",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "08.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019011f0-03bc-7dda-b50b-caa9e8f4258f",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "07.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019011f0-18e2-73a1-98a2-9e3aacb14d67",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "06.06.2024 0:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

}

]

}

}

##### /workday/date/list

входные параметры: date - дата выборки limit - ограничение выборки (по умолчанию 20 записей) off - смещение выборки (по умолчанию 0)

на выходе при успешном выполнении статус 200 и json:

{

"document": {

"total\_count": 3,

"page\_count": 1,

"details": [

{

"id": "01901160-bf0b-7c72-ba93-2158a7694cb8",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "2024-06-10T00:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "выходной",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019016e3-a390-7f4f-ab3a-be8ba782eb2c",

"employee\_id": "018fee07-b8fb-7350-8147-d6d2925d6873",

"date\_work": "2024-06-10T00:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

},

{

"id": "019016e5-4a54-73b8-aa48-2960e1170e70",

"employee\_id": "018fee07-b41e-77a5-aaa3-df70941f8837",

"date\_work": "2024-06-10T00:00:00",

"time\_work": "07:00-19:00",

"state\_wd": "",

"date\_dop\_smena": "",

"date\_ucheba": "",

"date\_change": "",

"intern": ""

}

]

}

}