



## ТЕХНИЧЕСКОЕ ЗАДАНИЕ №1

№	РЕАЛИЗУЕМАЯ ФУНКЦИОНАЛЬНОСТЬ
1.1	<p><b>Добавить запрос на получение списка городов</b></p> <p>Добавить обработку запроса GET city_list</p> <p>Запрос:</p> <pre>{     *без данных* }</pre> <p>Ответ:</p> <p>cities – массив городов</p> <pre>{     city_id – идентификатор города (тип данных – целое число),     title – название (тип данных – текст),     archive – архивность (тип данных – логическое значение) }</pre> <p>В ответе возвращаются все города.</p>
1.2	<p><b>Добавить запрос на получение списка мест выдачи и возврата</b></p> <p>Добавить обработку запроса GET place_list</p> <p>Запрос:</p> <pre>{     *без данных* }</pre> <p>Ответ:</p> <p>places – массив мест выдачи и возврата</p> <pre>{     place_id – идентификатор города (тип данных – целое число),     title – название (тип данных – текст),     city_id – идентификатор города (тип данных – целое число),     delivery_cost – стоимость доставки/получения автомобиля (тип данных – число),     archive – архивность (тип данных – логическое значение) }</pre> <p>В ответе возвращаются все места выдачи и возврата.</p>
1.3	<p><b>Добавить запрос на получение списка свободных автомобилей</b></p>

	<p>Добавить обработку запроса GET car_free_list</p> <p>Запрос:</p> <pre>{     city_id – идентификатор города (тип данных – целое число),     begin – начало периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     end – окончание периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     include_reserves – флаг «учитывая брони» (тип данных – логическое значение)     include_idles – флаг «учитывая сервисы» (тип данных – логическое значение) }</pre> <p>Ответ:</p> <pre>{     cars – массив идентификаторов автомобилей (тип данных – массив целых чисел) }</pre> <p>В ответе возвращаются все автомобили, которые находятся в выбранном городе в кластерах этого города в течение всего выбранного периода и при этом в выбранном периоде нет неудаленных аренд, броней (при условии include_reserves) и сервисов (при условии include_idles).</p>
<b>1.4</b>	<p><b>Добавить запрос на получение списка автомобилей с краткой информацией</b></p> <p>Добавить обработку запроса GET car_list</p> <p>Запрос:</p> <pre>{     *без данных* }</pre> <p>Ответ:</p> <p>cars – массив автомобилей</p> <pre>{     car_id – идентификатор автомобиля (тип данных – целое число),     model – марка и модель (тип данных – текст),     number – гос. номер (тип данных – текст),     city_id – идентификатор города (тип данных – целое число),     year – год выпуска (тип данных – целое число),     photo_guid – guid фотографии (тип данных – текст) }</pre> <p>В ответе возвращаются все автомобили, которые в момент обращения находятся в кластере.</p>
<b>1.5</b>	<p><b>Добавить запрос на получение списка тарифов</b></p> <p>Добавить обработку запроса GET tariff_list</p> <p>Запрос:</p> <pre>{     car_id – идентификатор автомобиля (тип данных – целое число), необязательное поле }</pre>

	<p>Ответ: cars – массив автомобилей</p> <pre>{     car_id – идентификатор автомобиля (тип данных – целое число),     deposit – стоимость залога (тип данных – число),     tariffs – массив тарифов     {         min – минимальное кол-во дней в периоде (тип данных – целое число),         max – максимальное количество дней в периоде (тип данных – целое число),         cost – стоимость аренды за один день (тип данных – число)     } }</pre> <p>В ответе возвращаются тарифы в тарифной сетке, помеченной галочкой «Тарифная сетка для сайта», для указанного автомобиля, если идентификатор автомобиля передан, либо, если идентификатор автомобиля не передан, для всех автомобилей, которые в момент обращения находятся в кластере.</p>
1.6	<p><b>Добавить запрос на получение списка периодов, в которых автомобиль свободен</b></p> <p>Добавить обработку запроса GET car_period_list</p> <p>Запрос:</p> <pre>{     car_id – идентификатор автомобиля (тип данных – целое число),     begin – начало периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     end – окончание периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     include_reserves – флаг «учитывая брони» (тип данных – логическое значение)     include_idles – флаг «учитывая сервисы» (тип данных – логическое значение) }</pre> <p>Ответ: car_periods – массив периодов</p> <pre>{     begin – начало периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     end – окончание периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ) }</pre> <p>В ответе возвращаются все периоды автомобиля, в которые нет неудаленных аренд, броней (при условии include_reserves) и сервисов (при условии include_idles).</p>
1.7	<p><b>Добавить запрос на получение списка типов услуг</b></p> <p>Добавить обработку запроса GET service_type_list</p> <p>Запрос:</p> <pre>{     *без данных* }</pre> <p>Ответ: service_types – массив типов услуг</p>

	<pre>{     service_type_id – идентификатор типа услуги (тип данных – целое число),     title – название (тип данных – текст) }</pre> <p>В ответе возвращаются все типы услуг.</p>
<b>1.8</b>	<p><b>Добавить запрос на получение списка услуг</b></p> <p>Добавить обработку запроса GET service_list</p> <p>Запрос:</p> <pre>{     *без данных* }</pre> <p>Ответ:</p> <p>services – массив услуг</p> <pre>{     service_id – идентификатор услуги (тип данных – целое число),     title – название (тип данных – текст),     cost – стоимость (тип данных – число),     service_type_id – идентификатор типа услуги (тип данных – целое число),     archive – архивность (тип данных – логическое значение) }</pre> <p>В ответе возвращаются все услуги, кроме системных услуг «Перепробег 1 км» и «Повреждение».</p>
<b>2.0</b>	<p><b>Добавить запрос на получение стоимости заявки</b></p> <p>Добавить обработку запроса GET bid_cost</p> <p>Запрос:</p> <pre>{     car_id – идентификатор автомобиля (тип данных – целое число),     begin – начало периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     end – окончание периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),     begin_place_id – идентификатор места выдачи (тип данных – целое число),     end_place_id – идентификатор места возврата (тип данных – целое число),     services – массив идентификаторов услуг (тип данных – массив целый чисел),     необязательное поле }</pre> <p>Ответ:</p> <pre>{     cost – стоимость заявки (тип данных – число),     prepay – предоплата (тип данных – число),     deposit – залог (тип данных – число),     error_message – сообщение об ошибке (тип данных – текст) }</pre>

	<p>В ответе возвращается стоимость заявки на аренду указанного автомобиля в указанном периоде для тарифной сетки, помеченной галочкой «Тарифная сетка для сайта», включая указанные услуги, а также услуги по доставке автомобиля в пункт выдачи и получении автомобиля в пункте возврата, если в настройках мест выдачи/возврата задана стоимость; размер предоплаты с учетом настройки «Процент стоимости для предоплаты по заявке с сайта» тарифной сетки; депозит.</p> <p>Варианты сообщения об ошибке:          «incorrect_car» – если нет автомобиля с указанным идентификатором;          «incorrect_dates» – если дата начала периода больше или равна дате окончания;          «incorrect_start_place» – если нет места выдачи/возврата с указанным идентификатором;          «incorrect_end_place» – если нет места выдачи/возврата с указанным идентификатором;          «incorrect_services» – если нет хотя бы одной услуги в массиве указанных идентификаторов;          «no_charge_type» – если нет тарифной сетки, помеченной галочкой «Тарифная сетка для сайта».</p>
<b>2.1</b>	<p><b>Добавить запрос на создание заявки</b></p> <p>Добавить обработку запроса PUT bid_create</p> <p>Запрос:          Параметры из опубликованных полей формы (multipart/form-data).          {              fio – ФИО клиента,              phone – телефон клиента,              car_id – идентификатор автомобиля (тип данных – целое число),              begin – начало периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),              end – окончание периода (тип данных – дата+время, ГГГГ-ММ-ДД ЧЧ:ММ),              begin_place_id – идентификатор места выдачи (тип данных – целое число)              end_place_id – идентификатор места возврата (тип данных – целое число),              services – массив идентификаторов услуг (тип данных – массив целый чисел),          }          необязательное поле,              prepayment – размер предоплаты (тип данных – число), необязательное поле,              files – файлы клиента, каждый файл добавляется отдельным параметром «files»          }</p> <p>Ответ:          {              bid_id – идентификатор заявки (тип данных – целое число),              bid_number – номер заявки (тип данных – целое число),              error_message – сообщение об ошибке (тип данных – текст)          }</p> <p>Результатом выполнения метода является создание нового клиента, если в базе не будет обнаружен клиент с переданными ФИО и номером телефона, прикрепление файлов к новому или существующему клиенту и создание заявки.</p> <p>Варианты сообщения об ошибке:          «incorrect_car» – если нет автомобиля с указанным идентификатором;          «incorrect_dates» – если дата начала периода больше или равна дате окончания;          «incorrect_start_place» – если нет места выдачи/возврата с указанным идентификатором;          «incorrect_end_place» – если нет места выдачи/возврата с указанным идентификатором;          «incorrect_services» – если нет хотя бы одной услуги в массиве указанных идентификаторов;</p>

	«no_charge_type» – если нет тарифной сетки, помеченной галочкой «Тарифная сетка для сайта»; «no_account» – если нет счета, помеченной галочкой «Счет для сайта», при наличии предоплаты.
--	---

Для аутентификации используются следующий механизм:

1. Авторизация. Получение токена.

POST /api/v1/tokens/signin

Тело запроса

```
{
    "UserName": "<UserLogin>",
    "PasswordHash": "<HashPassword>",
    "LongToken": <LongToken>
}
```

UserName - логин пользователя

PasswordHash - хэш пароля (Sha512)

LongToken - необязательное поле. Ожидается логическое значение. Если поле не указано, то значение - false. Используется для получения токенов с большим временем жизни.

Ответ

```
{
    "accessToken": "...",
    "refreshToken": "....",
    "expires": 1614845896
}
```

accessToken - токен, которым нужно подписывать запросы

refreshToken - токен, необходимый для обновления основного токена

expires - Кол-во секунд с 1970-01-01. Время жизни accessToken. Если LongToken имеет значение false, то время жизни равно 10 минутам. Если true, то 100 лет

2. Обновление токена

POST /api/v1/tokens/refresh

Тело запроса

```
{
    "Token": "<refreshToken>"
}
```

Ответ

```
{
    "accessToken": "...",
    "refreshToken": "....",
    "expires": ...
}
```

3. Все запросы должны быть подписаны полученным токеном. Для этого в заголовке запроса необходимо прописывать:

Authorization: "Bearer <accessToken>"

user\_list

GET ⌵ [http://{{server}}/data\\_api/user\\_list](http://{{server}}/data_api/user_list) Params

Authorization Headers (1) Body Pre-request Script Tests

Key	Value	Description
<input checked="" type="checkbox"/> Authorization	Bearer	
New key	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfmFtZSI6WyJXU19HVUVTVCIsIlldTX0dVRVNUlI0slmp0aSI6IjUyZDdmM2ZhYzdmODRlZGZiMzVjNzQ3YzlwZmM2NmZjliwibmJmIjoxNjE2NjU3NjI0LCJleHAiOjE2MTY2NTgyMjQslmIhdCI6MTYxNjY1NzYyNCwiaXNzIjoib29ya2Zsb3dUZWNobm9sb2d5In0.qhmjBiYtK5NpV-0n_pKmN-6jN8NcUF3T2sFhw9TdjnA	Description
Response		