**ВВЕДЕНИЕ**

**Технологии этого модуля:** REST API

**Время на выполнение**: 3 часа

К вам обратилась компания «Мегаполис» - новая ЖД компания, предоставляющая услуги пассажирских ЖД переездов. Вам необходимо использовать все имеющиеся навыки в серверной разработке для создания REST API.

Заказчик хочет, чтобы api можно было легко поддерживать, поэтому использование фреймворков будет плюсом.

|  |
| --- |
| **ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!** |

Обратите внимание, что компания пока что работает только в пределах Центрального округа, поэтому данные о времени вылета и прилета находятся в пределах одного часового пояса.

|  |
| --- |
| **В медиафайлах вам предоставляется sql дамп с готовой базой данных. База данных уже содержит набор данных, которые НЕ должны быть изменены! Структуру БД менять также запрещается. Любое изменение или удаление предоставленных данных в БД будет влиять на вашу оценку.** |

## 

## **ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ**

Ваша задача – реализовать REST API, которое будет отвечать требованиям заказчика.

Для вашего удобства, во всех URL будет использоваться переменная {host} которая обозначает адрес http://xxxxxx-m2.wsr.ru/, где xxxxxx - логин участника.  
В случае ошибок связанных с валидацией данных во всех запросах необходимо возвращать следующее тело ответа:  
{  
 “error”: {  
 “code”: <code>,  
 “message”: <message>,  
 “errors”: {  
 <key>: [ <error message>]  
 }  
 }  
}  
  
Обратите внимание, что вместо <code> и <message> необходимо указывать соответствующее значение, определенное в описании ответа на соответствующий запрос. В свойстве error.errors необходимо перечислить те свойства, которые не прошли валидацию, а в их значениях указать массив с ошибками валидации.  
  
Например если отправить пустой запрос на сервер, где проверяется следующая валидация:

* phone – обязательно поле
* password – обязательное поле

то тело ответа должно быть следующим  
{  
 “error”: {  
 “code”: 422,  
 “message”: “Validation error”,  
 “errors”: {  
 phone: [ “field phone can not be blank” ],  
 password: [ “field password can not be blank” ]  
 }  
 }  
}  
  
Учтите, что code и message могут быть определены иначе, если в запросе указано иное. В значениях свойств errors вы можете использовать любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

**Регистрация**

Запрос для регистрации нового пользователя в системе. При отправке запроса необходимо передать объект со следующими свойствами:

* first\_name – обязательное поле, строка
* last\_name – обязательное поле, строка
* phone – обязательное и уникальное поле, строка
* document\_number - обязательное, строка из 10 цифр (может быть с ведущим нулем)
* password – обязательное поле, строка

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/register  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {  “first\_name”: “Ivan”,  “last\_name”: “Ivanov”,  “phone”: “89001234567”,  “document\_number”: “7567999222”,  “password”: “paSSword”  } | **------------------------- Successful --------------------------**  **Status:** 204  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

**Аутентификация**

Запрос для аутентификации пользователя в системе. При отправке запроса необходимо передать объект с логином и паролем. Если клиент отправил корректные данные, то необходимо вернуть сгенерированный токен, а иначе сообщение об ошибке.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/login  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {  “phone”: “89001234567”,  “password”: “paSSword”  } | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  “data”: {  “token”: <сгенерированный token>  }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  }  **----------------------- Unauthorized -------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 401,  “message”: “Unauthorized”,  “errors”: {  “phone”: [ “phone or password incorrect” ]  }  }  } |

**Список станций**

Запрос на поиск станций по названию города. Поиск без учета регистра.

При отправке запроса обязательно нужно передать параметр query, который может содержать одно из следующих значений:

* название города (полное название или часть названия)
* название станции (полное название или часть названия)

Все следующие запросы должны вернуть в результатах поиска станций, т.к. все варианты подходят:

* GET {host}/station?query=oscow
* GET {host}/station?query=MOSCOW
* GET {host}/station?query=Moscow
* GET {host}/station?query=Kazansky
* GET {host}/station?query=Kaz

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/station  **Method:** GET  **Query string (GET parameters):**  - query | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  “data”: {  “items”: [  {  “id”: 1  “name”: “Kazansky”  }  ]  }  }  **---------------------------- No results -----------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  “data”: {  “items”: []  }  } |
|  |  |

**Поиск рейсов**

Запрос на поиск рейсов по указанным параметрам. Должна быть возможность передать следующие GET параметры:

* from – id станции отъезда, обязательно, должен существовать
* to - id станции прибытия, обязательно, должен существовать
* date1 - дата отъезда туда, обязательно, в формате YYYY-MM-DD
* date2 - дата возвращения обратно, не обязательно, в формате YYYY-MM-DD
* passengers - число пассажиров (от 1 до 4 включительно), обязательно

В ответе на запрос должен быть список найденных рейсов из from в to, на которые еще остались места в заданные даты.

В поле data.dispatch\_to должны быть рейсы из from в to.

Если указана дата возвращения (data2), то в поле data.dispatch\_back должны быть обратные рейсы (из to в from), а иначе пустой массив.

В базе данных вам предоставлены рейсы и станции. Дата рейсов не указана, это означает, что **рейсы осуществляются ежедневно**.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/dispatch  **Method:** GET  **Query string (GET parameters):**  - from *(ID)*  - to *(ID)*  - date1 *(2020-10-01)*  - date2 *(2020-10-13)*  - passengers *(2)* | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "dispatch\_to": [  {  "dispatch\_id": 2,  "dispatch\_code": "FP1200",  "from": {  "city": "Moscow",  "station": "Kazansky",  "station\_id": 1,  "date": "2020-10-01",  "time": "12:00"  },  "to": {  "city": "Krasnoyarsk",  "station": "Krasnoyarsk Center",  "station\_id": 31,  "date": "2020-10-01",  "time": "13:35"  },  "cost": 9500  },  ],  "dispatches\_back": [  {  "dispatch\_id": 1,  "dispatch\_code": "FP2100",  "from": {  "city": "Moscow",  "station": " Kazansky ",  "station\_id": 1,  "date": "2020-10-10",  "time": "08:35"  },  "to": {  "city": "Krasnoyarsk",  "station": "Krasnoyarsk Center",  "station\_id": 31,  "date": "2020-10-10",  "time": "10:05"  },  "cost": 10500  },  ]  }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

**Оформление бронирования**

При оформлении бронирования клиент должен передать на сервер идентификаторы рейсов из базы данных, даты рейсов (в формате YYYY-MM-DD), а также список пассажиров. Каждый пассажир должен содержать следующие поля:

* first\_name – обязательно поле, строка
* last\_name – обязательно поле, строка
* birth\_date – обязательно поле, дата в формате YYYY-MM-DD
* document\_number – обязательное поле, строка из цифр длиною в 10 символов

При создании бронирования необходимо также проверить, что на выбранных рейсах есть свободные места. Если на каком-то из рейсов недостаточно мест, то всё бронирование не может быть оформлено.

В случае успешного создания бронирования, с сервера должен вернуться уникальный код бронирования, который может состоять из пяти латинских символов.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/booking  **Method:** POST  **Body:**  {  "dispatch\_from": {  "id": 1,  "date": "2020-09-20"  },  "dispatch\_back": {  "id": 2,  "date": "2020-09-30"  },  "passengers": [  {  "first\_name": "Ivan",  "last\_name": "Ivanov",  "birth\_date": "1990-02-20",  "document\_number": "1234567890"  },  {  "first\_name": "Ivan",  "last\_name": "Gorbunov",  "birth\_date": "1990-03-20",  "document\_number": "1224567890"  }  ]  } | **------------------------- Successful --------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {  “data”: {  “code”: “QSASE”  }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

**Информация о бронировании**

Получить информации о бронировании можно по коду бронирования.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/booking/{code}  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "АKIJF",  "cost": 40000,  "dispatches": [  {  "dispatch\_id": 1,  "dispatch\_code": "FP2100",  "from": {  "city": "Moscow",  "station": " Kazanskyi ",  "station\_id": 1,  "date": "2020-10-01",  "time": "08:35"  },  "to": {  "city": "Krasnoyarsk",  "station": "Krasnoyarsk Center",  "station\_id": 32,  "date": "2020-10-01",  "time": "10:05"  },  "cost": 10500  },  ],  "passengers": [  {  "id": 1,  "first\_name": "Ivan",  "last\_name": "Ivanov",  "birth\_date": "1990-02-20",  "document\_number": "1234567890",  "place\_from": "7B",  "place\_back": null  },  {  "id": 2,  "first\_name": "Ivan",  "last\_name": "Larin",  "birth\_date": "1990-03-20",  "document\_number": "1224567890",  "place\_from": null,  "place\_back": null  }  ]  }  } |

place\_from и place\_back должны быть null пока не выбрано место.

**Получение занятых мест в вагоне**

Данный запрос должен возвращать список занятых мест в вагоне. Если обратного рейса нет то occupied\_back должен содержать пустой массив.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/booking/{code}/seat  **Method:** GET | **------------------------ Success --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "occupied\_from": [  {  "passenger\_id": 1,  "place": "4"  }  ],  "occupied\_back": []  }  } |

**Выбор места в салоне**

Данный запрос должен позволять изменить место в салоне поезда на выбранный рейс для определенного пассажира.

При отправке запроса клиент должен указать ID пассажира, выбранное место и тип рейса (from/back).

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/booking/{code}/seat  **Method:** PATCH  **Headers**  **- Content-Type:** application/json  **Body:**  {  "passenger": 1,  "seat": "7",  "type": "from/back"  } | **--------------------------- Success ----------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 1,  "first\_name": "Ivan",  "last\_name": "Ivanov",  "birth\_date": "1990-02-20",  "document\_number": "1234567890",  "place\_from": "7B",  "place\_back": null  }  }  **------------------------ Seat is occupied --------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 422,  "message": "Seat is occupied"  }  }  **------------------------ Forbidden --------------------------**  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": "Passenger does not apply to booking"  }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

**Получение своих бронирований**

Данный запрос должен возвращать все бронирования пользователя. Соотнести бронирования с аутентифицированным пользователем можно по номеру документа.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/user/booking  **Method:** GET  **Headers**  **- Content-Type:** application/json  **- Authorization:** Bearer {token} | **----------------------------- Success --------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "items": [  {  "code": "MGERS",  "cost": 40000,  "dispatches": [  {  "dispatch\_id": 1,  "dispatch\_code": "FP2100",  "from": {  "city": "Moscow",  "station": "Kazanskyi",  "station\_id": "1",  "date": "2020-10-01",  "time": "08:35"  },  "to": {  "city": "Krasnoyarsk",  "station": "Krasnoyarsk Center",  "station\_id": "32",  "date": "2020-10-01",  "time": "10:05"  },  "cost": 10500  },  ],  "passengers": [  {  "id": 1,  "first\_name": "Ivan",  "last\_name": "Ivanov",  "birth\_date": "1990-02-20",  "document\_number": "1234567890",  "place\_from": null,  "place\_back": null  },  {  "id": 2,  "first\_name": "Ivan",  "last\_name": "Sergeev",  "birth\_date": "1990-03-20",  "document\_number": "1224567890",  "place\_from": null,  "place\_back": null  }  ]  }  ]  }  }  **-------------------------- Unauthorized ----------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 401,  "message": "Unauthorized"  }  } |

**Получение информации о пользователе**

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/user  **Method:** GET  **Headers**  **- Content-Type:** application/json  **- Authorization:** Bearer {token} | **------------------------------ Success -------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "first\_name": "Ivan",  "last\_name": "Ivanov",  "phone": "89001234567",  "document\_number": "1224567890"  }  **------------------------ Unauthorized --------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 401,  "message": "Unauthorized"  }  } |

Заказчик допускает возможность изменения базы данных в будущем, поэтому вам необходимо подготовить свой вариант схемы базы данных и сохранить его в корне с модулем. Сохраните файл с названием DB.png.

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

Разработанное API должно быть доступно по адресу http://xxxxxx-m2.wsr.ru/, где xxxxxx - логин участника (указан на индивидуальной карточке).

Форматы запросов и ответов, а также форматы дат и времени должен соответствовать примерам из задания.

**Проверяются только работы, загруженные на сервер.**

|  |
| --- |
| **В медиафайлах вам предоставляется sql дамп с готовой базой данных. База данных уже содержит набор данных, которые НЕ должны быть изменены! Структуру БД менять также запрещается. Любое изменение или удаление предоставленных данных в БД будет влиять на вашу оценку.** |

**СИСТЕМА ОЦЕНКИ**

|  |  |  |
| --- | --- | --- |
| **Секция** | **Критерий** | **Сумма** |
| A | Организация работы и управление | 2,00 |
| B | Коммуникация и навыки межличностного общения | 2,00 |
| C | Графический дизайн | 0,00 |
| D | Верстка | 0,00 |
| E | Программирование на стороне клиента | 0,00 |
| F | Программирование на стороне сервера | 14,00 |
| G | CMS | 0,00 |
| **Всего** |  | 18,00 |