WEB-ДИЗАЙН И РАЗРАБОТКА

V Вузовский чемпионат НовГУ имени Ярослава Мудрого по стандартам WorldSkills Russia 2021

День 3. Программирование на стороне сервера

СОДЕРЖАНИЕ

Данный тестовый проект состоит из следующих файлов:

1. WSR_UC_TP17_3_MODULE.docx – Задание

ВВЕДЕНИЕ

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

Заказчик хочет, чтобы api можно было легко поддерживать, поэтому должны быть использованы фреймворки

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.

Время на выполнение модуля: 4 часа

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-m3.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. При сохранении все номера приводятся к единому формату: 11 цифр. Восьмёрка, стоящая вначале заменяется на +7. Номера, которые содержат недостаточно цифр выдают ошибку.
- document_number - обязательное, строка из 10 цифр (может быть с ведущим нулем)
- password – обязательное поле, строка. Не менее 6 символов, должны присутствовать символы верхнего и нижнего регистра (только латиница) и цифры.

Request	Response
URL: {host}/api/register **Method:** POST **Headers**	---------- **Successful** ---------- **Status:** 204

- Content-Type: application/json
 	
Body:
{
“first_name”: “Ivan”,
“last_name”: “Ivanov”,
“phone”: “89001234567”,
“document_number”: “7567999222”,
“password”: “paSSword”
}
	---------- **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.

Полученный телефонный номер приводится к единому формату: 11 цифр. Восьмёрка, стоящая вначале заменяется на +7. Номера, которые содержат недостаточно цифр выдают ошибку.

Request	Response

URL: {host}/api/login
Method: POST
 	
Headers
- Content-Type: application/json
 	
Body:
{
“phone”: “+79001234567”,
“password”: “paSSw0rd”
}
	---------- **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 аэропортов

Запрос на поиск аэропортов по названию города или по IATA-коду. Поиск без учета регистра.

При отправке запроса можно передать параметр query, который может содержать одно из следующих значений. Если параметр query отсутствует, возвращается полный список аэропортов.

- название города (полное название или часть названия)
- название аэропорта (полное название или часть названия)
- IATA-код аэропорта (SVO, KZN, DME и т.д).

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

- GET {host}/api/airport?query=oscow
- GET {host}/api/airport?query=MOSCOW
- GET {host}/api/airport?query=Moscow

- GET {host}/api/airport?query=Sheremetyevo
- GET {host}/api/airport?query=Shere
- GET {host}/api/airport?query=SVO

Request	Response

URL: {host}/api/airport
Method: GET
 	
Query string (GET parameters):
- query
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	“data”: {
	“items”: [
	{
	“name”: “Sheremetyevo”,
	“iata”: “SVO”
	}
	]
	}
	}
	
	---------- **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 - iata-код аэропорта вылета, обязательно, должен существовать
- to - iata-код аэропорта назначения, обязательно, должен существовать
- date1 - дата вылета туда, обязательно, в формате YYYY-MM-DD. Дата вылета должна быть не ранее даты обращения.

- date2 - дата возвращения обратно, не обязательно, в формате YYYY-MM-DD. Дата возвращения должна быть не ранее даты вылета туда.
- passengers - число пассажиров (от 1 до 8 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.flights_to должны быть рейсы из from в to.
Если указана дата возвращения (data2), то в поле data.flights_back должны быть обратные рейсы (из to в from), а иначе пустой массив.
В базе данных вам предоставлены рейсы и аэропорты. Дата рейсов не указана, это означает, что **рейсы осуществляются ежедневно**.

Request	Response

URL: {host}/api/flight
Method: GET
 	
Query string (GET parameters):
- from *(SVO)*
- to *(KZN)*
- date1 *(2020-10-01)*
- date2 *(2020-10-13)*
- passengers *(2)*
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"flights_to": [
	{
	"flight_id": 2,
	"flight_code": "FP1200",
	"from": {
	"city": "Kazan",
	"airport": "Kazan",
	"iata": "KZN",
	"date": "2020-10-01",
	"time": "12:00"
	},
	"to": {
	"city": "Moscow",
	"airport": "Sheremetyevo",
	"iata": "SVO",
	"date": "2020-10-01",
	"time": "13:35"
	},
	"cost": 9500,
	"availability": 156
	},
	{
	"flight_id": 14,
	"flight_code": "FP 1201",

```
            "from": {
                "city": "Kazan",
                "airport": "Kazan",
                "iata": "KZN",
                "date": "2020-10-01",
                "time": "08:35"
            },
            "to": {
                "city": "Moscow",
                "airport": "Sheremetyevo",
                "iata": "SVO",
                "date": "2020-10-01",
                "time": "10:05"
            },
            "cost": 10500,
            "availability": 156
        }
    ],
    "flights_back": [
        {
            "flight_id": 1,
            "flight_code": "FP 2100",
            "from": {
                "city": "Moscow",
                "airport": "Sheremetyevo",
                "iata": "SVO",
                "date": "2020-10-10",
                "time": "08:35"
            },
            "to": {
                "city": "Kazan",
                "airport": "Kazan",
                "iata": "KZN",
                "date": "2020-10-10",
                "time": "10:05"
            },
            "cost": 10500,
            "availability": 156
        },
        {
            "flight_id": 13,
            "flight_code": "FP 2101",
            "from": {
```

	"city": "Moscow",
	"airport": "Sheremetyevo",
	"iata": "SVO",
	"date": "2020-10-10",
	"time": "12:00"
	},
	"to": {
	"city": "Kazan",
	"airport": "Kazan",
	"iata": "KZN",
	"date": "2020-10-10",
	"time": "13:35"
	},
	"cost": 12500,
	"availability": 156
	}
	]
	}
	}
	
	---------- **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. Должен быть уникален в пределах бронирования.

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}/api/booking
Method: POST
 	
Body:
{
"flight_from": {
"id": 1,
"date": "2020-09-20"
},
"flight_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":
	---------- **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>
	}
	}
	}

"Gorbunov", "birth_date": "1990-03-20", "document_number": "1224567890" }] }	

Информация о бронировании

Получить информации о бронировании можно по коду бронирования.

Request	Response

URL:
{host}/api/booking/{code}
Method: GET
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"code": "AKIJF",
	"cost": 40000,
	"flights": [
	{
	"flight_id": 1,
	"flight_code": "FP2100",
	"from": {
	"city": "Moscow",
	"airport": "Sheremetyevo",
	"iata": "SVO",
	"date": "2020-10-01",
	"time": "08:35"
	},
	"to": {
	"city": "Kazan",
	"airport": "Kazan",
	"iata": "KZN",
	"date": "2020-10-01",
	"time": "10:05"
	},
	"cost": 10500,

```
            "availability": 56
        },
        {
            "flight_id": 2,
            "flight_code": "FP1200",
            "from": {
                "city": "Kazan",
                "airport": "Kazan",
                "iata": "KZN",
                "date": "2020-10-12",
                "time": "12:00"
            },
            "to": {
                "city": "Moscow",
                "airport": "Sheremetyevo",
                "iata": "SVO",
                "date": "2020-10-12",
                "time": "13:35"
            },
            "cost": 9500,
            "availability": 56
        }
    ],
    "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 самолете
Данный запрос должен возвращать список занятых мест в самолете. Если обратного рейса нет то occupied_back должен содержать пустой массив.

Request	Response

URL:
{host}/api/booking/{code}/seat
Method: GET
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"occupied_from": [
	{
	"passenger_id": 1,
	"place": "7B"
	}
	],
	"occupied_back": []
	}
	}

Выбор места в салоне
Данный запрос должен позволять изменить место в салоне воздушного судна 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}/api/booking/{code}/seat
Method: PATCH
 	
Headers
- Content-Type:
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"id": 1,

application/json

Body:
{
"passenger": 1,
"seat": "7B",
"type": "from/back"
}

"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,
	"flights": [
	{
	"flight_id": 1,
	"flight_code": "FP2100",
	"from": {
	"city": "Moscow",
	"airport": "Sheremetyevo",
	"iata": "SVO",
	"date": "2020-10-01",
	"time": "08:35"
	},
	"to": {
	"city": "Kazan",
	"airport": "Kazan",
	"iata": "KZN",
	"date": "2020-10-01",
	"time": "10:05"
	},
	"cost": 10500,
	"availability": 58
	},
	{

```
                "flight_id": 2,
                "flight_code": "FP1200",
                "from": {
                    "city": "Kazan",
                    "airport": "Kazan",
                    "iata": "KZN",
                    "date": "2020-10-12",
                    "time": "12:00"
                },
                "to": {
                    "city": "Moscow",
                    "airport": "Sheremetyevo",
                    "iata": "SVO",
                    "date": "2020-10-12",
                    "time": "13:35"
                },
                "cost": 9500,
                "availability": 58
            }
        ],
        "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"
	}
	}

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное API должно быть доступно по адресу http://xxxxxx-m3.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	СУДЕЙСКАЯ	ОБЪЕКТИВНАЯ	СУММА
A	Организация работы и управление	0.00	2.00	2.00
B	Коммуникация и навыки межличностного общения	1.00	0.00	1.00
C	Дизайн	0.00	0.00	0.00
D	Верстка	0.00	0.00	0.00
E	Программирование на стороне клиента	0.00	16.00	16.00
F	Программирование на стороне сервера	0.00	0.00	0.00
G	Проектирование	0.00	0.00	0.00
Всего		**1.00**	**18.00**	**19.00**