

Модуль В. Разработка Веб-приложения на стороне сервера

Время на выполнение модуля: 3 часа

К Вам обратилась компания «BreezeResort», которая имеет свою сеть отелей в разных странах мира. Так как компания хочется расширяться, им необходимо разработать свою информационную систему для администрирования своих отелей. Для начала им необходимо разработать REST API. К сожалению, предыдущая компании не исполнила свой контракт на разработку и отказалась предоставлять резервную копию базы данных, единственное, что осталось, - список HTTP-запросов с описанием работы API. Вам необходимо создать базу данных и создать REST API.

Ваша задача – создать базу данных и реализовать REST API заданной структуры. В примерах будет использоваться переменная `{{host}}`, которая обозначает адрес `http://xxxxxx-m1.wsr.ru/resortapi`, где `xxxxxx` - логин участника.

Общие требования к API

Идентификацию пользователя организуйте посредством Bearer Token.

При написании API вам необходимо разрешить кросс-доменные запросы (CORS) для обращения с другого домена.

При попытке доступа к защищенным авторизацией функциям системы во всех запросах необходимо возвращать ответ следующего вида:

Status: 401

Content-Type: application/json

Body:

```
{
  "error": {
    "message": "Unauthorized"
  }
}
```

В случае ошибок, связанных с валидацией данных во всех запросах, необходимо возвращать следующее тело ответа:

```
{
  "message": "Validation error",
  "errors": {
    <key>: [ <error message> ]
  }
}
```

В свойстве 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:

username – обязательно поле

password – обязательное поле

то тело ответа будет следующим:

```
{
  "message": "The given data was invalid.",
  "errors": {
    "username": [
      "The username field is required."
    ],
    "password": [
      "The password field is required."
    ]
  }
}
```

В значениях свойств errors вы можете использовать любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

Разработанное API должно быть доступно по адресам, указанным в требованиях к API.

Вы можете использовать любой из представленных фреймворков:

Laravel 8.5.x

Yii 2.0.x

NodeJS(express, cors, mysql2, typeorm, reflect-metadata, typescript, cors-env, dotenv, config)

Регистрация администратора системы

При успешной аутентификации возвращается сообщение.

Поля: username, password – обязательные, при этом username должен быть уникальным.

Request	Response
URL: {{host}} /signup Method: POST Headers - Content-Type: application/json	Успех Status: 200 Content-Type: application/json Body: {

Body: <pre>{ "username": "admin ", "password": "admin" }</pre>	<pre>"data": { "message": "Administrator created" }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>
--	--

Аутентификация администратора системы

При успешной аутентификации возвращается сгенерированный токен зарегистрированного администратора системы.

Request	Response
URL: {{host}} /login Method: POST Headers - Content-Type: application/json Body: <pre>{ "username": "admin@admin.ru", "password": "qwerty123456" }</pre>	<pre>----- УСПЕХ -----</pre> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "token_user": <сгенерированный token> } }</pre> <p>Неправильные логин или пароль</p> <p>Status: 401</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "message": "Unauthorized", "errors": { "login": "invalid credentials" } }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>

Последующий функционал доступен только авторизованному администратору. Идентификация пользователя должно быть организована посредством Bearer Token.

Номер

Добавление

Для добавления новой необходимо указать два поля, оба поля являются обязательными, название должно быть уникальным.

Request	Response
<p>URL: {{host}} /room</p> <p>Method: POST</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "name": "Название", "desc_data": "Описание", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "message": "Created" } }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>

Получение списка всех

Возвращается массив list, содержащий данные, которые есть в системе

Request	Response
<p>URL: {{host}} /rooms</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "list": { { "id": 1, "name": "Название", } } }</pre>

	<pre> "desc_data": "Описание" }, { "id": 2, "name": "Название", "desc_data": "Описание" } } </pre>
--	--

Удаление

Request	Response
<p>URL: {{host}} /room/{id}</p> <p>Method: DELETE</p> <p>Headers</p> <ul style="list-style-type: none"> - Content-Type: application/json - Authorization: Bearer {token} 	<p>----- УСПЕХ -----</p> <p>Status: 204</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre> { "data": { "message": "Deleted" } } </pre> <p>Попытка удалить несуществующую</p> <p>Status: 403</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre> { "error": { "message": "Not found" } } </pre>

Клиент

Добавление нового

При отправке запроса необходимо передать объект со следующими свойствами:

fio – обязательное и уникальное поле, строка;

email – обязательное и уникальное поле, электронная почта, строка;

phone – обязательное и уникальное поле, строка;

birth_date - обязательное, формат даты (ГГГГ-ММ-ДД);

id_chiilddata – обязательное поле, id существующего номера.

Request	Response
<p>URL: {{host}} /register Method: POST</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "fio": "Ivanov Ivan Ivanovich", "email": "ivan@wsr.ru", "phone": "89001234567", "id_chiilddata": 1, "birth_date": "2000-03-12", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 201 Content-Type: application/json Body:</p> <pre>{ "data": { "message": "Created" } }</pre> <p>Ошибки валидации полей Формат ответа из общих требований</p>

Редактирование информации

Request	Response
<p>URL: {{host}} /userdata/{id} Method: PATCH</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "phone": "89231234567", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "data": { "id": "5", "message": "Updated" } }</pre>

Удаление

При удалении из системы необходимо производить id в системе.

Request	Response
URL: {{host}} /userdata/{id} Method: DELETE Headers - Content-Type: application/json - Authorization: Bearer {token}	Успех Status: 204 Content-Type: application/json Body: { "data": { "message": "Deleted" } } Попытка удалить несуществующего Status: 403 Content-Type: application/json Body: { "error": { "message": "Not found" } } Примечание: {id} - идентификатор в системе

Изменение номера комнаты у клиента

Request	Response
---------	----------

URL: {{host}} /room/{id}/userdata/{iduser} Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token} Примечание: { id } – идентификатор номера в системе { iduser } – идентификатор клиента в системе	<p>----- УСПЕХ -----</p> Status: 200 Content-Type: application/json Body: <pre>{ "data": { "message": "Changed" } }</pre> Попытка указать несуществующие данные Status: 403 Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>
---	--

Получение списка номеров и клиентов в них

Request	Response
URL: {{host}} /usersinroom Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> Status: 200 Content-Type: application/json Body: <pre>{ "data": [{ "name": "Название", "userdata": [{ "fio": "Иванов Иван Иванович", "phonenumber": "79001234567" }] }] }</pre>

	<pre>] } Ошибки валидации полей Формат ответа из общих требований</pre>
--	---

Отель

Добавление

При отправке запроса необходимо передать объект со следующими свойствами:

name- обязательное и уникальное

number – обязательное и уникальное

Request	Response
URL: {{host}} /hotel Method: POST Headers - Content-Type: application/json - Authorization: Bearer {token} Body: <pre>{ "name": "Название", "number": 000, }</pre>	<pre>----- УСПЕХ ----- Status: 201 Content-Type: application/json Body: { "data": { "id": 18, "name": "Название", "number": 000 } }</pre> <p>Ошибки валидации полей Формат ответа из общих требований</p>

Получение всех

Возвращается массив list, содержащий список данных в системе

Request	Response
<p>URL: {{host}} /hotels</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "list": [{ "name": "Название", "number": 0001 }, { "name": "Название", "number": 0002 }] }</pre>

Удаление

При удалении из системы необходимо производить id в системе.

Request	Response
<p>URL: {{host}} /hotel/{id}</p> <p>Method: DELETE</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Примечание:</p> <p>{id} - идентификатор в системе</p>	<p>Успех</p> <p>Status: 204</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "message": "Deleted" } }</pre> <p>Попытка удалить несуществующие данные</p> <p>Status: 403</p>

	Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>
--	---

Добавление комнаты в отель

При добавлении комнаты в определенный отель, необходимо передать id комнаты и id отеля.

Request	Response
URL: {{host}} /hotel/{id}/room/{idroom} Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token} Примечание: { idroom } - идентификатор комнаты в системе { id } - идентификатор отеля в системе	<pre>----- УСПЕХ -----</pre> Status: 201 Content-Type: application/json Body: <pre>{ "data": { "name": "Название", "title ": "Название" } }</pre> Попытка указать несуществующие данные Status: 403 Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>

Получение списка всех отелей, закрепленных за ними номеров и клиентов, которые входят в этот номер

Request	Response
<p>URL: {{host}} /roomsinhotels</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none"> - Content-Type: application/json - Authorization: Bearer {token} 	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": [{ "title": "Название", "number": 3, "data_children": [{ "name": "Название", "userdata": [{ "fio": "Иванов Иван Иванович", "phonenumber": "79001234567" }] }], }, { "name": "Название", "userdata": [] }] }, { "title": "Название", "number": 3, "data_children": [] }]</pre> <p>----- Validation error -----</p> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>