WEB-Дизайн и разработка

lII Открытый региональный чемпионат Молодые профессионалы World Skills Russia Республики Хакасия 2019

День 1. Модуль RESTful API

## содержание

Данный тестовый проект состоит из следующих файлов:

1. Задание.pdf

## введение

К вам обратился владелец частной лингвистической компании с просьбой вступить в команду, которая разрабатывает для него форум.

От вас требуется написать RESTfulAPI для форума, всем остальным будет заниматься другие разработчики из вашей команды.

Функционал онлайн-сервис технической поддержки на веб-сайте будет разделен на две роли:

* администратор
* модератор
* гость

Функциональные возможности гостя:

* авторизация
* создание тем форума
* создание сообщений на форуме
* поиск по темам форума

Функциональные возможности модератора включают в себя функциональные возможности гостя, а также:

* удаление тем форума
* удаление сообщений на форуме

Функциональные возможности администратора включают в себя функциональные возможности модератора, а также:

* назначение и снятие модераторов

Данное задание состоит из одного модуля и рассчитано на 3 часа. Распределите свое время таким образом, чтобы успеть выполнить все поставленные задачи.

## описание проекта и задач

**API авторизации** должен быть доступен POST-запросом на адрес {API}/auth, а также принимать следующие обязательные параметры:

* login  
  логин пользователя
* password  
  пароль пользователя

При успешной авторизации должен возвращаться ответ в следующем виде:

* status code: 200
* status text: Successful authorization
* body:
  + status  
    true
  + token  
    bearer-токен для использования API

При безуспешной авторизации или ошибке в запросе должен возвращаться ответ в следующем виде:

* status code: 401
* status text: Invalid authorization data
* body:
  + status  
    false
  + message  
    Invalid authorization data

**API регистрации** должен быть доступен POST-запросом на адрес {API}/registration, а также принимать следующие обязательные параметры:

* login  
  логин пользователя, не пустой, уникальный
* password  
  пароль пользователя, не пустой

При успешном создании поста должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful registration
* body:
  + status  
    true
  + token  
    bearer-токен для использования API

При безуспешной регистрации должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Registration error
* body:
  + status  
    false
  + message  
    ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

Пример ответа при безуспешном добавлении поста:

{

“status”: false,

“message”: {

“login”: “already exists”,

“password”: “not found”

}

}

**API создания темы форума** должен быть доступенPOST-запросом на адрес {API}/posts/, а также принимать следующие параметры:

* title  
  название темы, не пустой
* text  
  текст темы
* tags  
  тэги темы через запятую

При успешном редактировании поста должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true

При безуспешном добавлении поста должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Editing error
* body:
  + status  
    false
  + message  
    ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

**API создания сообщений к теме** должен быть доступен только **POST**-запросом на адрес {API}/posts/<POST\_ID>/message.

При успешном создании собщения должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true

При попытке создания сообщения к несуществующей теме должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + status

false

* + message  
    Post not found

**API просмотра тем** форума должен быть доступен GET-запросом на адрес {API}/posts, а также принимать следующие параметры:

* title  
  название темы
* tags  
  тэги темы через запятую

Ответ должен содержать массив объектов тем, подходящий под условие выборки:

* tags

полностью совпадают с тегами темы форума

* title

**частично** или полностью совпадает с названием темы форума

Ответ должен содержать массив объектов тем содержащие следующие параметры:

* id

идентификатор темы

* title  
  название темы
* created\_at  
  дата и время создания темы в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* text  
  описание темы
* tags  
  массив из тэгов поста ([“tag1”, “tag2”])

Ответдолжениметь status code “200” и status text “List posts”.

**API просмотра одной темы** форума должен быть доступен GET-запросом на адрес {API}/posts/<POST\_ID>.

Ответ должен содержать следующие параметры:

* id

идентификатор темы

* title  
  название темы
* created\_at  
  дата и время создания поста в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* text  
  описание темы
* tags  
  массив из тэгов поста ([“tag1”, “tag2”])
* image  
  ссылка на изображение поста
* messages  
  массив из объектов сообщений содержащие следующие параметры:
  + comment\_id  
    уникальный идентификатор сообщения
  + created\_at  
    дата и время в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
  + author  
    имя автора
  + comment  
    текст комментария

Ответдолжениметь status code “200” и status text “View post”.

При попытке просмотра несуществующей темы должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

**API удаления темы форума должен быть доступен только модератору и администратору DELETE**-запросом на адрес {API}/posts/<POST\_ID>

При успешном удалении темы должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful delete
* body:
  + status  
    true

При попытке удаления несуществующей темы должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
     Post not found

**API удаления сообщений** должен быть доступен только **модератору и** **администратору** DELETE-запросом на адрес {API}/posts/<POST\_ID>/messages/<MESSAGE\_ID>.

При успешном удалении сообщения должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful delete
* body:
  + status  
    true

При попытке удаления сообщения к несуществующей теме должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

При попытке удаления несуществующего сообщения должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Message not found
* body:
  + message  
    Message not found

**API назначения модератора** должен быть доступен PUT-запросом на адрес {API}/users/<USER\_ID>/moderator.

В ответ должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful change role
* body
  + status  
    true

При попытке назначения модератором несуществующего пользователя должен возвращаться ответ в следующем виде:

* status code: 404
* status text: User not found
* body:
  + status

false

* + message  
    User not found

**API удаления модератора** должен быть доступен DELETE-запросом на адрес {API}/users/<USER\_ID>/moderator.

В ответ должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful change role
* body
  + status  
    true

При попытке удаления несуществующего модератора должен возвращаться ответ в следующем виде:

* status code: 404
* status text: User not found
* body:
  + status

false

* + message  
    User not found

Авторизация всех пользователей будет проверяться через bearer-token авторизацию. Т.е. к каждому такому запросу в headers будет дописываться bearer-token:

Authorization: Bearer xxxxxxxx

При безуспешной попытке пройти авторизацию по токену на любой из запросов, должен возвращаться ответ в следующем виде:

* status code: 401
* status text: Unauthorized
* body:
  + message  
    Unauthorized

При безуспешной попытке пройти авторизацию по токену на любой из запросов, на который не хватает прав текущей роли (ситуация, когда модератор делает запрос, доступный только администратору), должен возвращаться ответ в следующем виде:

* status code: 401
* status text: Permission denied
* body:
  + message  
    Permission denied

Под {API} подразумевается адрес <http://xxx-m1.wsr.ru/api>, где xxx - ваш логин.

Логин администратора admin, пароль - khakasia\_19.

**ИНСТРУКЦИя ДЛЯ КОНКУРСАНТА**

Готовая система должна быть доступна по адресу: <http://xxxxxx-m1.wsr.ru/>

Вам доступны фреймворки Yii2 и Laravel для работы с этим модулем.

Все ответы должны приходить в JSON формате.

Во время выполнения конкурсного задания можно использовать Postman, предложенный в медиа данных.

**СИСТЕМА ОЦЕНКИ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **секция** | **критерий** | **судейская** | **объективная** | **сумма** |
| A | Организация работы и управление | 0.80 | 0.00 | 0.80 |
| B | Коммуникация и навыки межличностного общения | 1.00 | 0.00 | 1.00 |
| С | Дизайн | 0.00 | 0.00 | 0.00 |
| D | Верстка | 0.00 | 0.00 | 0.00 |
| Е | Программирование на стороне клиента | 0.00 | 0.00 | 0.00 |
| F | Программирование на стороне сервера | 0.00 | 11.75 | 11.75 |
| G | CMS | 0.00 | 0.00 | 0.00 |
| **Всего** |  | **1.80** | **11.75** | **13.55** |