WEB-Дизайн и разработка

Вузовский отборочный открытый чемпионат Оренбургского государственного университета по стандартам WorldSkills 2019  
День 1. Модуль RESTful API

## содержание

Данный тестовый проект состоит из следующих файлов:

1. Задание.docx

## введение

К вам обратился директор колледжа с просьбой принять участие в разработке нового сайта его образовательной организации. Сайт должен стать своеобразным информационным посредником между сотрудниками колледжа и обучающимися.

От вас требуется написать RESTful API для сайта, всем остальным будут заниматься обучающиеся колледжа, так как технология RESTful API слишком сложна для них.

Функционал сайта будет разделен на две роли:

* сотрудник колледжа (администратор)
* обучающийся

Функциональные возможности обучающегося:

* авторизация
* просмотр новостей сайта и комментариев к ним
* комментирование новостей сайта
* поиск новостей по тегу

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

* создание, редактирование и удаление новостей колледжа
* удаление комментариев к новости колледжа

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 администратора
* password  
  пароль администратора

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 администратора

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 создания новостей на сайте** должен быть доступен только **администратору** POST-запросом на адрес {API}/news, а также принимать следующие обязательные параметры:

* title  
  название новости, не пустой, уникальный
* anons  
  анонс новости, не пустой
* text  
  полный текст новости, не пустой
* tags  
  тэги новости через запятую, не обязательное поле
* image  
  изображение отражающее суть новости, не пустой, разрешенные форматы: jpg, png, gif. Максимальный размер: 2 мегабайта

При успешном добавлении новости на сайт, должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true
  + news\_id  
    уникальный идентификатор новости

После добавления новости, изображение должно загружаться на сервер в папку {API}/news\_images.

При безуспешном добавлении поста должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Creating 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”: {

“title”: “already exists”,

“image”: “invalid file format”

}

}

**API редактирования новостей** должен быть доступен только **администратору** POST-запросом на адрес {API}/news/<NEWS\_ID>, а также принимать следующие параметры:

* title  
  название новости, уникальный
* anons  
  анонс новости
* text  
  полный текст новости
* tags  
  тэги новости через запятую
* image   
  изображение отражающее суть новости, разрешенные форматы: jpg, png. Максимальный размер: 2 мегабайта

При успешном редактировании новости должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true
  + news
    - title  
      название новости
    - datatime  
      дата и время создания новости в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
    - anons  
      анонс новости
    - text  
      полный текст новости
    - tags  
      массив из тэгов новости ([“tag1”, “tag2”])
    - image  
      ссылка на изображение новости

При безуспешном добавлении новости 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

При попытке редактирования несуществующей новости, ответ должен возвращаться в следующем виде:

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

**API удаления новостей** должен быть доступен только **администратору** DELETE-запросом на адрес {API}/news/<NEWS\_ID>.

При успешном удалении новости должен возвращаться ответ в следующем виде:

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

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

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

**API просмотра новостей колледжа** должен быть доступен GET-запросом на адрес {API}/news. Ответ должен содержать массив объектов постов, содержащие следующие параметры:

* title  
  название новости
* datatime  
  дата и время создания новости в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* anons  
  анонс новости
* text  
  полный текст новости
* tags  
  массив из тэгов новости ([“tag1”, “tag2”])
* image  
  ссылка на изображение новости

Ответ должен иметь status code “200” и status text “List news”.

**API просмотра одной новости колледжа** должен быть доступен GET-запросом на адрес {API}/news/<NEWS\_ID>.

Ответ должен содержать следующие параметры:

* title  
  название новости
* datatime  
  дата и время создания новости в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* anons  
  анонс новости
* text  
  полный текст новости
* tags  
  массив из тэгов новости ([“tag1”, “tag2”])
* image  
  ссылка на изображение новости
* comments  
  массив из объектов комментариев новости колледжа содержащие следующие параметры:
  + comment\_id  
    уникальный идентификатор комментария
  + datatime  
    дата и время в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
  + author  
    имя автора, если автором комментария является администратор, то парметр содержит текст “admin”
  + comment  
    текст комментария

Ответ должен иметь status code “200” и status text “View post”.

При попытке просмотра комментариев к несуществующей новости, ответ должен возвращаться в следующем виде:

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

**API добавления комментария** к новостям сайта должен быть доступным POST-запросом на адрес {API}/news/<NEWS\_ID>/comments и содержать следующие обязательные параметры:

* author  
  имя комментатора, обязательное только для гостя
* comment  
  обязательное, максимум 255 символов

При успешном добавлении комментария должен возвращаться ответ в следующем виде:

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

При безуспешном добавлении комментария должен возвращаться ответ в следующем виде:

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

При попытке добавить комментарий к несуществующей новости, ответ должен возвращаться в следующем виде:

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

**API удаления комментария** к новостям колледжа должен быть доступным только **администратору** DELETE-запросом на адрес {API}/news/<NEWS\_ID>/comments/<COMMENT\_ID>.

При успешном удалении комментария должен возвращаться ответ в следующем виде:

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

При попытке удаления комментария к несуществующему посту должен возвращаться ответ в следующем виде:

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

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

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

**API поиска новостей** по тегу должен быть доступен GET-запросом на адрес {API}/news/tag/<TAG\_NAME>.

В ответ должен возвращаться ответ в следующем виде:

* status code: 200
* status text: Found news
* body  
  массив из объектов постов, содержащие данный тэг:
  + title  
    название новости
  + datatime  
    дата и время создания новости в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
  + anons  
    анонс новости
  + text  
    текст новости
  + tags  
    массив из тэгов новости([“tag1”, “tag2”])
  + image  
    ссылка на изображение новости

Авторизация пользователя при использовании административных функциональных возможностей 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, который должен быть доступен только администратору, должен возвращаться ответ в следующем виде:

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

Под {API} подразумевается адрес <webX.worldskills.osu.ru/m1/api>, где X – номер участника (т.е. webX – ваш логин), а m1 означает 1 модуль.

Логин администратора admin, пароль – uc2019.

**ИНСТРУКЦИя ДЛЯ КОНКУРСАНТА**

Готовая система должна быть доступна по адресу:  [webX.worldskills.osu.ru/m1/api](file:///C:\Users\runner\Desktop\Заднания%20WS%202019\WEB_UC_OSU_2019\webX.worldskills.osu.ru\m1\api), где X – номер участника (т.е. webX – ваш логин), а m1 означает 1 модуль.

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

**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** |