Article Hub Documentation

Purpose

This API provides the basic functions for creating a platform for sharing and discussing articles related to any subject. It implements the following business rules:

1. **Creating an article:** Users can create an article by providing both a title and the content they want to share.
2. **Modifying existing articles:** Users can modify their articles or remove them entirely from account library.
3. **Commenting:** Other users should be able to express their opinion regarding any article. They should also be able to express their opinion on others comments as well.
4. **Comment visibility:** Users can hide the content of their comments but the responses from other users should still be visible. Users can not entirely delete comments.
5. **Adding external resources to existing articles:** Users can add, to their own articles, links that lead to other websites, which are used for references.
6. **Ordering external resources:** The server should provide an indexing method for storing external resources in a certain order for the article.
7. **Categorizing articles:** articles can be assigned multiple categories for an easier understanding of the discussed topic.
8. **Creating a user:** Users can be created by providing a username, a password and a valid email.
9. **Accessing user data:** Users can access all of their articles and comments.
10. **Data size limiting:** The server needs a limit in the amount of content that it returns in a given request.

Controllers v0.1.0

Article Controller

GET *.../article/{id}*

- Returns an article by an id.

GET *.../article/{id}/accepted-categories/*

- Returns a list off all the categories of type ACCEPTED that are associated with the article.

GET *.../article/{id}/comments/*

- Returns all the comments posted on the article.

GET *.../article/{id}/external-resources/*

- Returns a list with all the resources used by the article.

GET *.../article/headers-list*

- Returns a list with a certain number of article headers.
- Article headers don't have the content, only details like title.

POST *.../article/new*

- Creates a new article with the data provided in the body.

PUT *.../article/{id}*

- Modifies an existing article with data provided as parameters.

DELETE *.../ article/{id}*

- Deletes an article.

Category Controller

GET *.../category/{id}*

- Returns a category by a given id.

GET *.../category/list*

- Returns a list with all the categories.

POST *.../category/new/{title}*

- Creates a new category with the given title and state will be the default IN_REVIEW.

PUT *.../category/{id}/*

- Modifies an existing category with data provided as parameters.

DELETE *.../ category/{id}*

- Deletes a category.

Comment Controller

GET *.../comment/{id}*

- Returns a comment by a given id.

GET *.../comment/{id}/comments*

- Returns a list with all the comments given on a certain comment.

POST *.../comment/new/*

- Creates a new comment with the data given in the body.
- Only one of the comment id or article id can be valid because it will be mapped to that object. Providing none or both ids will result in a bad request and the comment will not be saved.

PUT *…/comment/{id}/update-content*

- Modifies the content of a comment with the text given in the body.

PUT *…/comment/{id}/update-visibility*

- Modifies the visibility of a comment with the one given in the body.

External Resource Controller

GET *…/resource/{id}*

- Returns a resource given by an id.

POST *…/resource/new/*

- Creates a new resource for an article with the data provided in the body.
- Recommended: use the article index to store the resources in a certain order. They should be in increasing order as they appear in the article.

PUT *…/resource/{id}/*

- Modifies a resource with the new data provided in the body.

DELETE *…/ resource/{id}*

- Deletes a resource.

User Controller

GET *…/user/{id}*

- Retrieves user data, besides the password.

GET *…/user/email/{email}*

- Retrieves user data of the user that has the given email.

GET *…/user/{id}/articles*

- Returns a list with all the articles that belong to a user.
- The list returns article headers only.

GET *…/user/{id}/comments*

- Returns a list with all the comments written by the user.

GET *…/user/list*

- Returns a list with a certain number of users.

POST *…/user/new/*

- Creates a new user with the data provided in the body.
- The email has to be a new one that is not used by another account.

PUT *…/user/{id}/*

- Can update username, password and email with data received as optional parameters.

DELETE *…/ user/{id}*

- Deletes a user.