

Telerik Academy Forum

Task Description

Implement a server-side part of a articles application. Users can create articles, comments for the articles and like/dislike articles.

Description

Using a centralized server the users of the page can **register**, **login** and **logout**. Each registered user can create **articles** and **comments**. Registered users can view all articles, their comments and **likes** and can comment in every article. Each articles has 1 **categories** and 0 or more **tags**.

Additionally there is are **alerts** that display on the page.

Forum Architecture:

The architecture of the forum application consists of two parts: server and client. The server exposes RESTful API that the client consumes.

Your task is to implement the server part of the application

Application Logic

The application works as follows:

- Users register in the articles page / Users login in the articles page
 - They receive SessionKey for later authentication
- Users can view all articles and their comments
- Users can create a new article, and it will be added to the other articles
- Users can open an article and view its comments and likes
- Users can comment in any article
- Users can like/dislike any article
- When creating an article each word from the Title is added as a tag

RESTful API Overview

The RESTful API should expose web services to work with the server. The API has the following web service endpoints:

HTTP Method	Web service endpoint	Description
POST (public)	api/users/register	Registers a new user in the forum
POST (public)	api/users/login	Logs in a user in the forum

PUT	api/users/logout	Logs out a user from the forum
POST	api/articles	Creates a new thread, returns the article created so it can be loaded in the UI
GET (public)	api/articles	Gets top 10 articles, sorted by their date of creation
GET (public)	api/articles?page=P	Gets the articles at positions from $P*10$ to $(P+1)*10$. The articles sorted by date of creation and are at most 10.
GET	api/articles?category=[categoryName]	Gets top 10 articles in category "categoryName", sorted by their date of creation
GET	api/articles?category=[categoryName]&page=P	Gets the articles in category "categoryName" at positions from $P*10$ to $(P+1)*10$. The articles sorted by date of creation and are at most 10.
GET	api/articles/{articleID}	Get article with ID = articleID, with 10 comments sorted by date
POST	api/articles/{articleID}/comments	Creates a new comment for a given article
GET	api/categories	Gets all categories
GET	api/categories/{categoryID}	Gets all articles, sorted by date, from Category with ID = categoryID
GET	api/tags	Gets all tags
GET	api/tags/{tagID}	Gets all articles, sorted by date, that have Tag with ID = categoryID
POST	api/articles/like/{articleID}	Creates a like by the current user for the article with ID = articleID
PUT	api/articles/like/{articleID}	Removes a like by the current user for the article with ID = articleID
GET	api/alerts	Get all alerts with date of expiration later than today

1. Task 1 – Create a Database

Create a MS SQL database to hold the data about the forum application.

Use EntityFramework code-first or database first approach.

2. Create the WebAPI endpoints

Develop the RESTful API of the Articles application.

The services are as follows:

Users

Register

api/users/register		POST
Request	{ "Email": "evlogi@telerik.com", "Password": "123456", "ConfirmPassword": "123456" }	
Response	{ "" }	

Login

api/users/login		POST
Request	{ "grant_type": "password" "username": " evlogi@telerik.com", "authCode": "123456" }	
Response	{ "access_token": "deYtpuVHa4Ba-1t-..... _hdTI-1Pa-bNiASNQz", "token_type": "bearer", "expires_in": 1209599, "userName": "evlogi@telerik.com", ".issued": "Thu, 18 Sep 2014 07:05:31 GMT", ".expires": "Thu, 02 Oct 2014 07:05:31 GMT" }	

Logout

api/users/logout		PUT
Request	{ "Authorization": " Bearer deYtpuVHa4Ba-1t-..... _hdTI-1Pa-bNiASNQz "}	
Response	Null	

Articles

Create articles

api/articles		POST
Headers	{ "Authorization": "Bearer deYtpuVHa4Ba-1t-..... _hdTI-1Pa-bNiASNQz "}	
Request	{ "Title": "Some Random Title", }	

	<pre> "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "Tags": [] } </pre>
Response	<pre> { "ID": 1, "Title": "Some Random Title", "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "DateCreated": "2014-09-16T20:30:28.713", "Tags": [] } </pre>

Get articles

api/articles	GET
Headers	{ "Authorization": "Bearer deYtpuVHa4Ba-1t-....._hdTI-1Pa-bNiASNQz " }
Response	<pre> [{ "ID": 1, "Title": "Some Random Title", "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "DateCreated": "2014-09-16T20:30:28.713", "Tags": [] }, { "ID": 2, "Title": "Best Article Ever", </pre>

	<p>"Content": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?",</p> <p>"Category": "Testing",</p> <p>"DateCreated": "2014-09-17T15:33:10.91",</p> <p>"Tags": []</p> <p>}]</p>
--	--

Articles by page

api/articles?page=0		GET
Headers	{ "Authorization": "Bearer deYtpuVHa4Ba-1t-....._hdTI-1Pa-bNiASNQz "}	
Response	<pre>[{ "ID": 1, "Title": "Some Random Title", "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "DateCreated": "2014-09-16T20:30:28.713", "Tags": [] }, { "ID": 2, "Title": "Best Article Ever", "Content": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit</pre>	

	<pre> esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?", "Category": "Testing", "DateCreated": "2014-09-17T15:33:10.91", "Tags": [] }]</pre>
--	---

Articles with paging and category (example with full page)

api/articles?category=testing&page=0		GET
Headers	{ "Authorization": "Bearer deYtpuVHa4Ba-1t-....._hdTI-1Pa-bNiASNQz " }	
Response	<pre> [{ "ID": 1, "Title": "Some Random Title", "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "DateCreated": "2014-09-16T20:30:28.713", "Tags": [] }, { "ID": 2, "Title": "Best Article Ever", "Content": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?", "Category": "Testing", "DateCreated": "2014-09-17T15:33:10.91", "Tags": [] }]</pre>	

Article details

api/articles/1		GET
Headers	{ "Authorization": "Bearer deYtpuVHa4Ba-1t-....._hdTI-1Pa-bNiASNQz " }	

Response	<pre>{ "ID": 1, "Title": "Some Random Title", "Content": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum", "Category": "Testing", "DateCreated": "2014-09-16T20:30:28.713", "Tags": [], "Comments": [{ "ID": 1, "Content": "Some inappropriate comment!", "DateCreated": "2014-09-17T15:05:45.913", "AuthorName": "evlogi@telerik.com" }, { "ID": 2, "Content": "Another useless comment :)", "DateCreated": "2014-09-17T15:06:24.63", "AuthorName": "evlogi@telerik.com" }], "Likes": [] }</pre>
----------	--