**Back-end routes**

Authentication process:



After some research, I have found the best authentication guides available for REST API here: <http://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html>

<http://stackoverflow.com/questions/21706356/rest-api-authentication-how-to-prevent-man-in-the-middle-replays>

<https://www.owasp.org/index.php/REST_Security_Cheat_Sheet>

However, I am still not very satisfied. Steps 1-3 looks pretty secure even if performed via HTTP, but step 4 must be performed only via HTTPS with SSL/TLS encryption or it will be vulnerable for MITM attacks. Also, after step 4, cookie must be transferred only via HTTPS, again, to prevent possible MITM attacks. However, HTTPS overhead is not so large (<https://www.keycdn.com/blog/https-performance-overhead/>) and so it should be no problem to use it for security critical parts of application.

I haven’t implemented proposed authentication protocol, because I would spend all my 8 hours and maybe some more just on it alone.

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/auth/sign-in | - | User authentication via REST. Signing in. |
| **Method** | **Parameters** | |
| **GET:** Get nonce for user authentication. | username (string): name of user, who we want to authenticate .  See authentication process diagram above for description. | |
| **POST:** Authenticate user using cnonce, username and secret. | secret (string): hash that authenticates user and generated on client side using hashing of username, password and key, provided by server.  username (string): name of user, who we want to authenticate.  cnonce (string): hash that was generated by server part.  See authentication process diagram above for description. | |

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/auth/sign-out | Authenticated | User authentication via REST. Signing out. |
| **Method** | **Parameters** | |
| **GET:** Close current user’s session. |  | |

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/entities/tags | None for GET; Admin role for other operations. | CRUD operations for Tag entities. |
| **Method** | **Parameters** | |
| **GET:** Get tags list |  | |
| **POST:** Create new tag | tag: JSON stringified Tag entity. | |

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/entities/tags/  {slug} | None for GET; Admin role for other operations. | CRUD operations for Tag entities. |
| **Method** | **Parameters** | |
| **GET:** Get tag by slug | slug: Tag’s slug. | |
| **PUT:** Update tag by slug | slug: Tag’s slug.  article: Updated JSON stringified Tag entity. | |
| **DELETE:** Delete tag by slug | slug: Tag’s slug. | |

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/entities/articles | None for GET; Admin role for other operations. | CRUD operations for Article entities. |
| **Method** | **Parameters** | |
| **GET:** Get articles list | [Optional] offset (default = 0): Get articles starting from {offset}  [Optional] limit (default = 0): Get not more than {limit} articles  [Optional] sort\_field (default = created\_on): By what field to sort articles. Possible values are 'title', 'created\_on', 'updated\_on'.  [Optional] sort\_order (default = asc): Sort order. Possible values are 'asc', 'desc'.  [Optional] only\_published (default = true): If true, return only published articles. If false, return all articles. Works only when user has Admin role. | |
| **POST:** Create new article | article: JSON stringified Article entity. | |

|  |  |  |
| --- | --- | --- |
| **Path** | **Required role** | **Description** |
| /api/v0.1/entities/articles/{slug} | None for GET; Admin role for other operations. | CRUD operations for Article entities. |
| **Method** | **Parameters** | |
| **GET:** Get article by slug | slug: Article’s slug. | |
| **PUT:** Update article by slug | slug: Article’s slug.  article: Updated JSON stringified Article entity. | |
| **DELETE:** Delete article by slug | slug: Article’s slug. | |

**Front-end routes**

Path: /

Brief description: Home page. Display N latest published articles and pagination to reach other articles list.

Required role: -

--------------------------------------------

Path: /article/{slug}

Parameters:

- [GET, required] slug: Slug of article to display

Brief description: Article's content.

Required role: -

--------------------------------------------

Path: /tags

Brief description: List of tags.

Required role: -

--------------------------------------------

Path: /articles

Parameters:

- [GET, optional] tag: Slug of tag for which to get articles list

- [GET, optional] page: Index of page

Brief description: List of articles.

Required role: -

--------------------------------------------

Path: /sign-in

Brief description: Login screen.

Required role: -

--------------------------------------------

Path: /admin/

Brief description: Admin dashboard.

Required role: Admin

--------------------------------------------

Path: /admin/list/tags

Brief description: List of tags.

Required role: Admin

--------------------------------------------

Path: /admin/list/articles

Parameters:

- [GET, optional] tag: Slug of tag for which to get articles list

- [GET, optional] page: Index of page

Brief description: List of articles.

Required role: Admin

--------------------------------------------

Path: /admin/create/tags

Brief description: Create new tag.

Required role: Admin

--------------------------------------------

Path: /admin/create/articles

Brief description: Create new article.

Required role: Admin

--------------------------------------------

Path: /admin/edit/tags/{slug}

Brief description: Edit tag by slug.

Required role: Admin

--------------------------------------------

Path: /admin/edit/articles/{slug}

Brief description: Edit article by slug.

Required role: Admin