

## A7: WEB RESOURCES DOCUMENTATION- LEARNIt (Q&A)

---

The architecture of the web application to develop is documented indicating the catalogue of resources and the properties of each resource, including: references to the graphical interfaces, and the format of JSON responses.

This page presents a documentation for **Learn It**, including the following operations over data: create, read, update, and delete.

### 1. OVERVIEW

An overview of the web application to implement is presented in this section, where the modules are identified and briefly described. The web resources associated with each module are detailed in the individual documentation of each module.

<b>M01: Authentication</b>	Web resources associated with user authentication, includes the following system features: login/logout, registration and recover password.
<b>M02: Static pages</b>	Web resources associated with immutable content pages, such as 404, about, terms, contacts.
<b>M03: Users</b>	Web resources associated with users, such as individual profile management: view and edit personal information, manage their followers/following, their settings. Furthermore, this module also includes the notifications management.
<b>M04: Main content (questions, answers and comments)</b>	Web resources associated with the categories, includes creating, viewing, editing and deleting questions, answers and comments.
<b>M05: Feed and search</b>	Web resources associated with the feed and search. This module includes the main feed and search features.
<b>M06: Administration</b>	Web resources associates with member management, specifically: view and search members, ban member accounts, promote members to moderator, manage reports, add new categories and faqs (Admin, moderator, faq).

## 2. PERMISSIONS

This section defines the permissions used in the modules to establish the conditions of access to resources.

<b>PUB</b>	Public	Group of users without privileges.
<b>MEM</b>	Member	Group of authenticated users that can create and search for content.
<b>OWN</b>	Owner	Users that are owners of information (profile, questions, answers, comments) and can update their content.
<b>ADM</b>	Administrator	System administrator.
<b>MOD</b>	Moderator	Group of members that manage the website.

## 3. MODULES

This section documents each web resource of each module, indicating the URL, HTTP methods, the possible parameters, interfaces with the user (referring to the A3 artefact) — or JSON responses in the event of APIs.

### 3.1. MODULE M01: AUTHENTICATION

These are the endpoints available in the authentication module:

- **R101:** Login Form */login*
- **R102:** Login Action */login*
- **R103:** Logout Action */logout*
- **R104:** Register Form */register*
- **R105:** Register Action */register*
- **R106:** RecoverPassword Form */password/recover*
- **R107:** RecoverPassword Action */password/recover*

#### R101: Login Form

<b>URL</b>	<i>/login</i>
<b>Description</b>	Page with a form to login to a user account.
<b>Method</b>	GET
<b>UI</b>	UI01
<b>SUBMIT</b>	R102
<b>Permissions</b>	PUB

**R102: Login Action**

<b>URL</b>	<b>/login</b>	
<b>Description</b>	This web resource logs the user into the system. Redirects to the member's feed or to admin or moderator page on success and to login form on failure.	
<b>Method</b>	POST	
<b>Request Body</b>	+username: string or +email: string	Username or Email
	+password: string	Password
<b>Redirects</b>	R502	Success, if member
	R601	Success, if admin
	R602	Success, if moderator
	R101	Error
<b>Permissions</b>	PUB	

**R103: Logout Action**

<b>URL</b>	<b>/logout</b>	
<b>Description</b>	This web resource logs out the authenticated user, the moderator or the administrator of the system. Redirects to the login form.	
<b>Method</b>	POST	
<b>Redirects</b>	UI04	
<b>Permissions</b>	MEM or ADM or MOD	

**R104: Register Form**

<b>URL</b>	<b>/register</b>	
<b>Description</b>	Page with a form to register a new user account.	
<b>Method</b>	GET	
<b>UI</b>	UI02	
<b>SUBMIT</b>	R105	
<b>Permissions</b>	PUB	

**R105: Register Action**

<b>URL</b>	<b>/register</b>	
<b>Description</b>	This web resource inserts the user into the system. Redirects to the member's feed on success and to the sign-up form on failure.	
<b>Method</b>	POST	
<b>Request Body</b>	+username: string	Username
	+email: string	Email
	+password: string	Password
	+birthdate: date	Birthdate
<b>Redirects</b>	R502	Success
	R104	Error
<b>Permissions</b>	PUB	

**R106: RecoverPassword Form**

<b>URL</b>	<b>/password/recover</b>	
<b>Description</b>	Page with a form to request a token to reset the password.	
<b>Method</b>	GET	
<b>UI</b>	UI03	
<b>SUBMIT</b>	R107	
<b>Permissions</b>	USR, MOD, ADM	

**R107: RecoverPassword Action**

<b>URL</b>	<b>/password/recover</b>	
<b>Description</b>	Web resource that sends a reset password link to the specified email.	
<b>Method</b>	POST	
<b>Request Body</b>	+email: string	Email
<b>Redirects</b>	R102	Success
	R106	Error
<b>Permissions</b>	USR, MOD, ADM	

### 3.2. MODULE M02: STATIC PAGES

These are the endpoints available in the static pages module:

- **R201:** 404 Page */404*
- **R202:** Contact Page Form */contacts*
- **R203:** Contact Page Action */contacts*
- **R204:** About Page */about*
- **R205:** Terms Page */terms*
- **R206:** FAQ Page */faq*
- **R207:** Home Page */home*

#### R201: 404 Page

<b>URL</b>	<b>/404</b>
<b>Description</b>	Get the 404 page
<b>Method</b>	GET
<b>UI</b>	UI22
<b>Permissions</b>	PUB

#### R202: Contacts Page Form

<b>URL</b>	<b>/contacts</b>
<b>Description</b>	Get the contacts page.
<b>Method</b>	GET
<b>UI</b>	UI08
<b>SUBMIT</b>	R203
<b>Permissions</b>	PUB

#### R203: Contacts Page Action

<b>URL</b>	<b>/contacts</b>	
<b>Description</b>	This web resource sends a message to the site master.	
<b>Method</b>	POST	
<b>Request Body</b>	+username: string	Username
	+email: string	Email
	+subject: string	Subject
	+message: string	Message
<b>Redirects</b>	R204	Success
	R502	Success, if member
	R202	Error
<b>Permissions</b>	PUB	

**R204: About Page**

<b>URL</b>	<b>/about</b>
<b>Description</b>	Get about page.
<b>Method</b>	GET
<b>UI</b>	UI05
<b>Permissions</b>	PUB

**R205: Terms Page**

<b>URL</b>	<b>/terms</b>
<b>Description</b>	Get the terms of use page.
<b>Method</b>	GET
<b>UI</b>	UI06
<b>Permissions</b>	PUB

**R206: FAQ Page**

<b>URL</b>	<b>/faq</b>
<b>Description</b>	Get the faqs page.
<b>Method</b>	GET
<b>UI</b>	UI07
<b>Permissions</b>	PUB

**R207: Home Page**

<b>URL</b>	<b>/home</b>
<b>Description</b>	Get the Home page.
<b>Method</b>	GET
<b>UI</b>	UI04
<b>Permissions</b>	PUB

### 3.3. MODULE M03: USERS

These are the endpoints available in the user's module:

- **R301:** View Profile */profile/{username}*
- **R302:** Edit Profile Form */profile/edit*
- **R303:** Get Edit Profile Form API */api/members/{username}/edit-profile*
- **R304:** Edit Profile Action */profile/{username}*
- **R305:** View Followers */profile/{username}/followers*
- **R306:** View Following */profile/{username}/following*
- **R307:** Follow/ unfollow user API */api/member/{username}/toggle-follow*
- **R308:** View Settings */username/settings*
- **R309:** Get Settings API */api/members/{username}/settings*
- **R310:** Edit E-mail API */api/members/{username}/settings/email*
- **R311:** Edit Password API */api/members/{username}/settings/password*
- **R312:** Edit Google Account API */api/members/{username}/settings/google-account*
- **R313:** Edit Facebook Account API */api/members/{username}/settings/facebook-account*
- **R314:** Delete account */api/members/{username}/settings/delete-account*
- **R315:** Get Notifications API */api/members/{username}/notifications*
- **R316:** Mark a Notification as Read API */api/members/{username}/notifications/{notification\_id}*

#### R301: View Profile

<b>URL</b>	<i>/profile/{username}</i>	
<b>Description</b>	Page with a member's profile info.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>UI</b>	UI09	
<b>UI observations</b>	UI changes according to permissions. If member has OWN permissions, editing options are shown. If member has MEM permissions, follow option should be displayed on all profiles that are not one's own.	
<b>Permissions</b>	PUB (general access) or OWN (for editing)	

#### R302: Edit Profile Form

<b>URL</b>	<i>/profile/edit</i>
<b>Description</b>	Page with a form to edit one's profile info.
<b>Method</b>	GET
<b>UI</b>	UI10
<b>Ajax calls</b>	R203
<b>SUBMIT</b>	R204
<b>Permissions</b>	OWN

**R303: Get Edit Profile Form API**

<b>URL</b>	<b>api/members/{username}/edit-profile</b>	
<b>Description</b>	This resource gets member's information to use as default values in the edit form.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>Response Body</b>	JSON303	
<b>Permissions</b>	OWN	

**R304: Edit Profile Action**

<b>URL</b>	<b>/profile/{username}</b>	
<b>Description</b>	This web resource changes user profile info based on the input received. Redirects to the member profile on success and edit profile page on failure.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username
<b>Request Body</b>	+ username: string	username
	+ email: string	email
	+ password: string	password
	? bio	Bio description
	? picture	Profile photo
<b>Redirects</b>	R301	Success
	R303	Error
<b>Permissions</b>	OWN	

**R305: View Followers**

<b>URL</b>	<b>/profile/{username}/followers</b>	
<b>Description</b>	Page with the member's followers.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>UI</b>	UI11	
<b>Permissions</b>	PUB	



**R306: View Following**

<b>URL</b>	<b>profile/{username}/following</b>	
<b>Description</b>	Page with the members that another member follows.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>UI</b>	UI11	
<b>Permissions</b>	PUB	

**R307: Follow/ unfollow user API**

<b>URL</b>	<b>/api/member/{username}/toggle-follow</b>	
<b>Description</b>	This web resource toggles following of another member.	
<b>Method</b>	PUT	
<b>Request body</b>	+ username: string	Target member's username
<b>Returns</b>	200 Ok	The following status was updated
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**R308: View Settings**

<b>URL</b>	<b>/{username}/settings</b>
<b>Description</b>	Page with the member's settings.
<b>Method</b>	GET
<b>UI</b>	UI23
<b>Ajax Calls</b>	R309, R310, R311, R312, R313
<b>Permissions</b>	OWN

**R309: Get Settings API**

<b>URL</b>	<b>/api/members/{username}/settings</b>	
<b>Description</b>	Obtains a member's current settings.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username.
<b>Response body</b>	JSON309	
<b>Permissions</b>	OWN	

**R310: Edit e-mail API**

<b>URL</b>	<b>/api/members/{username}/settings/email</b>	
<b>Description</b>	This resource updates the member's e-mail account.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username.
<b>Request Body</b>	+ email: string	New email
<b>Returns</b>	200 Ok	The email was updated.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Response body</b>	JSON310	
<b>Permissions</b>	OWN	

**R311: Edit Password API**

<b>URL</b>	<b>/api/members/{username}/settings/password</b>	
<b>Description</b>	This resource updates the member's password.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username.
<b>Request Body</b>	+ password: string	New password
<b>Returns</b>	200 Ok	The password was updated.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**R312: Edit Google Account API**

<b>URL</b>	<b>/api/members/{username}/settings/google-account</b>	
<b>Description</b>	This resource updates the member's connected google account.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username.
<b>Request Body</b>	+ account: string	New google account
<b>Returns</b>	200 Ok	The account was updated.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**R313: Edit Facebook Account API**

<b>URL</b>	<b>/api/members/{username}/settings/facebook-account</b>	
<b>Description</b>	This resource updates the member's connected facebook account.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username.
<b>Request Body</b>	+ account: string	New facebook account.
<b>Returns</b>	200 Ok	The account was successfully updated.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**R314: Delete Account**

<b>URL</b>	<b>/api/members/{username}/settings/delete-account</b>	
<b>Description</b>	This resource deletes the member's account.	
<b>Method</b>	DELETE	
<b>Parameters</b>	+ username: string	Member's username.
	200 Ok	The account was successfully deleted.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**R315: Get Notifications API**

<b>URL</b>	<b>/api/members/{username}/notifications</b>	
<b>Description</b>	This resource gets the member's notifications.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>Response body</b>	JSON315	
<b>Permissions</b>	OWN	

**R316: Mark a Notification as Read API**

<b>URL</b>	<b>/api/members/{username}/notifications/{id_notification}</b>	
<b>Description</b>	This resource marks a notification as read.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username.
	+ id_notification: integer	Notifications primary key
<b>Returns</b>	200 Ok	The account was successfully deleted.
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	OWN	

**3.4. MODULE M04: MAIN CONTENT**

These are the endpoints available in the main content module:

Topic:

- **R401:** View Topic */topic/{name}*

Questions:

- **R402:** View Question */questions/{id\_question}*
- **R403:** Add Question From */questions/add*
- **R404:** Add Question Action */questions/add*
- **R405:** Edit Question API */api/questions/{id\_question}*
- **R406:** Delete Question API */api/questions/{id\_question}*
- **R407:** Rate Question API */api/questions/{id\_question}/rate*
- **R408:** Report Action API */api/questions/{id\_question}/report*

Answers:

- **R409:** Get Answer */questions/{id\_question}/answers/{id\_answer}*
- **R410:** Add Answer API */api/questions/{id\_question}/answers*
- **R411:** Edit Answer API */api/questions/{id\_question}/answers/{id\_answer}*
- **R412:** Delete Answer API */api/questions/{id\_question}/answers/{id\_answer}*
- **R413:** Rate Answer API */api/questions/{id\_question}/answers/{id\_answer}/rate*
- **R414:** Report API */api/questions/{id\_question}/answers/{id\_answer}/report*
- **R415:** Assign/unassign as best answer API */api/{id\_question}/answers/{id\_answer}/best-answer*

Comment:

- **R416:** Get Comment API */api/questions/{id\_question}/comments/{id\_comment}*
- **R417:** Add Comment API */api/questions/{id\_question}/comments*
- **R418:** Edit Comment API */api/questions/{id\_question}/comments/{id\_comment}/edit*

- **R419:** Delete Comment API *api/questions/{id\_question}/comments/{id\_comment}*
- **R420:** Rate Comment API  
*api/questions/{id\_question}/comments/{id\_comment}/rate*
- **R421:** Report Comment API  
*api/questions/{id\_question}/comments/{id\_comment}/rate*

**R401: View Topic**

<b>URL</b>	<b>/topics /{name}</b>	
<b>Description</b>	Displays the topic's page.	
<b>Method</b>	GET	
<b>Parameters</b>	+ name: String	Topic name
<b>UI</b>	UI19	
<b>Permissions</b>	PUB	

**R402: View Question**

<b>URL</b>	<b>/questions/{id_question}</b>	
<b>Description</b>	Displays the question's page.	
<b>Method</b>	GET	
<b>Parameters</b>	+ id: integer	Question primary key
<b>Ajax Calls</b>	R307, R403, R405, R406, R407, R408, R409, R410, R411, R412, R413, R414, R415, R416, R417, R418, R419, R420, R421	
<b>UI</b>	UI21	
<b>Ui observations</b>	UI changes according to permissions. If viewer has PUB permissions, no content options should be displayed. If MEM, report, rating and comment/ answer options. If OWN, editing options only for the member's content.	
<b>Permissions</b>	PUB (general access) or MEM (rating, reporting content, comment) or OWN (for editing one's content)	

**R403: Add Question Form**

<b>URL</b>	<b>/questions/add</b>	
<b>Description</b>	Page with a form to add a new question.	
<b>Method</b>	GET	
<b>Parameters</b>	+ category: string	Question's category.
	+ text: string	Question
	? description: string	Question's description.
	? photo: string	Question's photo or file.
<b>UI</b>	UI18	
<b>SUBMIT</b>	R303	
<b>Permissions</b>	MEM	

**R404: Add Question Action**

<b>URL</b>	<b>/questions/add</b>	
<b>Description</b>	This web resource adds a new question.	
<b>Method</b>	POST	
<b>Request Body</b>	+ id_author: integer	Member primary key
	+ category: string	Question's category.
	+ text: string	Question
	? description: string	Question's description.
	? photo: string	Question's photo or file.
<b>Redirects</b>	R402	Success
	R403	Error
<b>Permissions</b>	MEM	

**R405: Edit Question API**

<b>URL</b>	<b>/api/questions/{id_question}</b>	
<b>Description</b>	This web resource updates a question.	
<b>Method</b>	PUT	
<b>Parameter</b>	+ id_question: integer	Question primary key
<b>Request Body</b>	+ category: string	Question's category.
	+ text: string	Question
	? description: string	Question's description.
	? photo: string	Question's photo or file.
<b>Response body</b>	JSON405	The question's information when successful, an error message when unsuccessful.
<b>Permissions</b>	OWN	

**R406: Delete Question API**

<b>URL</b>	<b>/api/questions/{id_question}</b>	
<b>Description</b>	This web resource deletes a question.	
<b>Method</b>	DELETE	
<b>Parameters</b>	+ id_question: integer	Question primary key
<b>Returns</b>	200 Ok	The question was successfully deleted.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No question with the specified primary key exists.
<b>Redirects</b>	R502	Success
<b>Permissions</b>	OWN or MOD	

**R407: Rate Question API**

<b>URL</b>	<b>/api/questions/{id_question}/rate</b>	
<b>Description</b>	This web resource rates a question.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_author: integer	User who rated the question.
<b>Returns</b>	200 Ok	The question was successfully rated.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No question with the specified primary key exists.
<b>Permissions</b>	MEM	

**R408: Report Question API**

<b>URL</b>	<b>/api/questions/{id_question}/report</b>	
<b>Description</b>	This web resource reports a question.	
<b>Method</b>	POST	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ reason: string	Reason
<b>Returns</b>	200 Ok	The question was successfully reported.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No question with the specified primary key exists.
<b>Permissions</b>	MEM	

**R409: Get Answer**

<b>URL</b>	<b>/questions{id_question}/answers/{id_answer}</b>	
<b>Description</b>	This web resource gets an answer.	
<b>Method</b>	GET	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer primary key
<b>Response Body</b>	JSON409	The answer's information when successful, an error message when unsuccessful
<b>Permissions</b>	MEM	

**R410: Add Answer API**

<b>URL</b>	<b>/api/questions/{id_question}/answers</b>	
<b>Description</b>	Page with a form to add a new answer.	
<b>Method</b>	POST	
<b>Parameters</b>	+ id_question: integer	Question primary key
<b>Request Body</b>	+ id_author: integer	Member primary key
	+ content: string	Content
<b>Response Body</b>	JSON410	The answer's information when successful, an error message when unsuccessful
<b>Permissions</b>	MEM	



**R411: Edit Answer API**

<b>URL</b>	<b>/api/questions/{id_question}/answers/{id_answer}</b>	
<b>Description</b>	This web resource updates an answer.	
<b>Method</b>	PUT	
<b>Parameter</b>	+ id_question: integer	Question primary key
<b>Request Body</b>	+ id_author: integer	Member primary key
	+ content: string	Content
<b>Response body</b>	JSON411	The answer's information when successful, an error message when unsuccessful.
<b>Permissions</b>	OWN	

**R412: Delete Answer API**

<b>URL</b>	<b>/api/questions/{id_question}/answers/{id_answer}</b>	
<b>Description</b>	This web resource deletes an answer.	
<b>Method</b>	DELETE	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer primary key
<b>Returns</b>	200 Ok	The answer was successfully deleted.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No question with the specified primary key exists.
<b>Permissions</b>	OWN or MOD	

**R413: Rate Answer API**

<b>URL</b>	<b>/api/questions/{id_question}/answers/{id_answer}/rate</b>	
<b>Description</b>	This web resource rates an answer.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer's primary key
	+ id_author: integer	User who rated the question.
<b>Returns</b>	200 Ok	The answer was successfully rated.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No answer with the specified primary key exists.
<b>Permissions</b>	MEM	

**R414: Report Answer API**

<b>URL</b>	<b>/api/questions/{id_question}/answers/{id_answer}/report</b>	
<b>Description</b>	This web resource reports an answer.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer primary key
<b>Request body</b>	+ id_user: integer	Member primary key
	+ reason: string	Reason
<b>Returns</b>	200 Ok	The answer was successfully reported.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No answer with the specified primary key exists.
<b>Permissions</b>	MEM	

**R415: Assign/unassign as best answer API**

<b>URL</b>	<b>/api/{id_question}/answers/{id_answer}/best-answer</b>	
<b>Description</b>	This web resource assign/ unassign as best answer	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
<b>Returns</b>	200 Ok	The answer was successfully Assign/unassign as best answer.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No answer with the specified primary key exists.
<b>Permissions</b>	OWN (question's author)	

**R416: Get Comment API**

<b>URL</b>	<b>/api/questions/{id_question}/comments/{id_comment}</b>	
<b>Description</b>	This web resource gets a comment.	
<b>Method</b>	GET	
<b>Parameters</b>	+ id_comment: integer	Comment primary key
<b>Response body</b>	JSON416	The comment's information when successful, an error message when unsuccessful
<b>Permissions</b>	MEM	

**R417: Add Comment API**

<b>URL</b>	<b>/api/questions/{id_question}/ comments</b>	
<b>Description</b>	This web resource adds a new comment.	
<b>Method</b>	POST	
<b>Parameters</b>	+ id_question: integer	Question primary key
<b>Request Body</b>	+ id_author: integer	Member primary key
	+ id_answer: integer	Answer primary key
	+ content: string	Content
<b>Response Body</b>	JSON417	The comment's information when successful, an error message when unsuccessful
<b>Permissions</b>	MEM	

**R418: Edit Comment API**

<b>URL</b>	<b>/api/questions/{id_question}/comments/{id_comment}/edit</b>	
<b>Description</b>	This web resource updates a comment.	
<b>Method</b>	PUT	
<b>Parameter</b>	+ id_comment: integer	Comment primary key
<b>Request Body</b>	+ id_author: integer	Member primary key
	+ content: string	Content
<b>Response body</b>	JSON418	The comment's information when successful, an error message when unsuccessful.
<b>Permissions</b>	OWN	

**R419: Delete Comments API**

<b>URL</b>	<b>/api/questions/{id_question}/ comments/{id_comment}</b>	
<b>Description</b>	This web resource deletes a comment.	
<b>Method</b>	DELETE	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer primary key
	+ id_comment: integer	Comment primary key
<b>Returns</b>	200 Ok	The answer was successfully deleted.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No question with the specified primary key exists.
<b>Permissions</b>	OWN or MOD	

**R420: Rate Comment API**

<b>URL</b>	<b>/api/questions/{id_question}/ comments/{id_comment}/rate</b>	
<b>Description</b>	This web resource rates an answer.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer's primary key
	+ id_author: integer	User who rated the question.
	+ id_comment: integer	Comment's primary key
<b>Returns</b>	200 Ok	The comment was successfully rated.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No comment with the specified primary key exists.
<b>Permissions</b>	MEM	

**R421: Report Comment API**

<b>URL</b>	<b>/api/questions/{id_question}/comments/{id_comment}/report</b>	
<b>Description</b>	This web resource reports a comment.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_question: integer	Question primary key
	+ id_answer: integer	Answer primary key
	+ id_comment: integer	Comment primary key
<b>Request body</b>	+ id_user: integer	Member primary key
	+ reason: string	Reason
<b>Returns</b>	200 Ok	The comment was successfully reported.
	400 Bad request	Error. Error message is specified via a HTTP header.
	404 Not found	Error. No comment with the specified primary key exists.
<b>Permissions</b>	MEM	

### 3.5. MODULE M05: FEED AND SEARCH

These are the endpoints available in the feed and search module:

- **R501:** View Feed */feed*
- **R502:** Get Feed API */api/members/{username}/feed*
- **R503:** Search API */api/search*

#### R501: View Feed

<b>URL</b>	<b>/feed</b>
<b>Description</b>	Page with the member's personal feed
<b>Method</b>	GET
<b>UI</b>	UI20
<b>Ajax Calls</b>	R307, R502
<b>Permissions</b>	MEM

#### R502: Get Feed API

<b>URL</b>	<b>/api/members/{username}/feed</b>	
<b>Description</b>	This resource gets the member's personal feed.	
<b>Method</b>	GET	
<b>Parameters</b>	+ username: string	Member's username
<b>Response Body</b>	JSON502	
<b>Permissions</b>	MEM	

#### R503: Search API

<b>URL</b>	<b>/api/search</b>	
<b>Description</b>	This web resource searches the system for all content that matches the search query and filters it accordingly.	
<b>Method</b>	GET	
<b>Request body</b>	+ query: string	Search query
	+ name: string	Category's content
	? filter_option: integer	Filters option
<b>Response body</b>	JSON503	
<b>Permissions</b>	PUB	

### 3.6. MODULE M06: ADMINISTRATION

These are the endpoints available in the administration module:

- **R601:** Admin Page */admin*
- **R602:** Moderator Page */moderator*
- **R603:** Search for Members API */api/members*
- **R604:** Add Category API */api/admin/add-category*
- **R605:** Edit Category API */api/admin/category/{id}*
- **R606:** Ban Member API */api/members/{username}/ban*
- **R607:** Promote Member API */api/members/{username}/promote*
- **R608:** Dismiss Moderator API */api/members/{username}/dismiss*

#### R601: Admin Page

<b>URL</b>	<b>/admin</b>
<b>Description</b>	Get admin page.
<b>Method</b>	GET
<b>UI</b>	UI12
<b>Ajax Calls</b>	R603 R604, R605, R606, R607, R608
<b>Permissions</b>	ADM

#### R602: Moderator Page

<b>URL</b>	<b>/moderator</b>
<b>Description</b>	Get moderator page.
<b>Method</b>	GET
<b>UI</b>	UI16
<b>Ajax Calls</b>	R603, R606
<b>Permissions</b>	MOD

#### R603: Search for Members API

<b>URL</b>	<b>/api/members</b>	
<b>Description</b>	This web resource searches the members whose data matches the search query.	
<b>Method</b>	GET	
<b>Request body</b>	+ query: string	Search query
<b>Response body</b>	JSON603	
<b>Permissions</b>	ADM or MOD	

**R604: Add Category API**

<b>URL</b>	<b>/api/admin/add-category</b>	
<b>Description</b>	This web resource adds a new category.	
<b>Method</b>	POST	
<b>Request Body</b>	+ name: string	Category's name
	+ icon: text	Category's icon
<b>Response body</b>	JSON604	The category's information when successful, an error message when unsuccessful
<b>Permissions</b>	ADM	

**R605: Edit Category API**

<b>URL</b>	<b>/api/admin/category/{id}</b>	
<b>Description</b>	This web resource updates a category.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ id_category: integer	Category primary key
<b>Request body</b>	+ name: string	Category's name
	+ icon: text	Category's icon
<b>Response body</b>	JSON606	The category's information when successful, an error message when unsuccessful
<b>Permissions</b>	ADM	

**R606: Ban Member API**

<b>URL</b>	<b>/api/members/{username}/ban</b>	
<b>Description</b>	This web resource bans a member.	
<b>Method</b>	POST	
<b>Parameters</b>	+ username: string	Member's username
<b>Returns</b>	200 Ok	The following status was updated
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	ADM or MOD	

**R607: Promote Member API**

<b>URL</b>	<b>/api/members/{username}/promote</b>	
<b>Description</b>	This web resource promotes a member to acquire moderator privileges.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username
<b>Returns</b>	200 Ok	The following status was updated
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	ADM or MOD	

**R608: Dismiss Moderator API**

<b>URL</b>	<b>/api/members/{username}/dismiss</b>	
<b>Description</b>	This web resource removes the moderator privileges of a member.	
<b>Method</b>	PUT	
<b>Parameters</b>	+ username: string	Member's username
<b>Returns</b>	200 Ok	The following status was updated
	400 Bad request	Error. Error message is specified via http header.
	404 Not found	Error. No member with the specified username exists.
<b>Permissions</b>	ADM	

**4. JSON/XML TYPES****JSON303: Edit profile (default values for form)**

```
{
  "data":
  {
    "id": 1,
    "username": "SmilingOwl",
    "bio_description": "Hi! I love this forum!",
    "profile_photo": "sophie.jpg"
  },
  "message": "Example error message when editing a profile"
}
```



**JSON309: Get current member settings**

```
{
  "data": {
    "id": 1,
    "email": "sophie2000@gmail.com",
    "google_account": "myfacebookaccount",
    "facebook_account": "mygoogleaccount"
  },
  "message": "Example error message when getting a member settings"
}
```

**JSON310: Edit email**

```
{
  "data": {
    "email": "sophie2000@gmail.com"
  },
  "message": "Example error message when editing an email"
}
```

**JSON315: Get notifications**

```
{
  "notifications": [
    {
      "id_notification": 1,
      "type": "comment",
      "date": "2017-04-23 01:57:00",
      "description": "Peter left a comment on your question:
'How do a black holes impact time?'",
      "view": true
    },
    {
      "id_notification": 2,
      "type": "answer",
      "date": "2017-04-23 01:57:00",
      "description": "Peter answered your question: 'How do a
black holes impact time?'",
      "view": true
    },
    {
      "id_notification": 1,
      "type": "follow",
      "date": "2017-05-31 02:35:13",
      "description": "Diogo start follow you",
      "view": false
    }
  ],
  "message": "Example error message when getting notifications"
}
```

**JSON405: Edit question**

```
{
  "DATA": {
    "ID_QUESTION": 4,
    "TITLE": "WHAT IS  $\sin(x) + \cos(x)$ ?",
    "DESCRIPTION": "PLEASE HELP ME!",
    "DATE": "2017-01-01 05:00:00",
    "VOTES": 1000,
    "PHOTO": "YVES.JPG",
    "DELETED": FALSE,
    "ID_CATEGORY": 5,
    "ID_USER": 20
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN GETTING A QUESTION"
}
```

**JSON409, JSON410 and JSON411: Get/ add or edit an answer**

```
{
  "DATA": {
    "ID_ANSWER": 4,
    "TEXT": "MAYBE 1",
    "DATE": "2018-01-02 12:35:01",
    "VOTES": 5,
    "PHOTO": NULL,
    "DELETED": "FALSE",
    "ID_QUESTION": 10,
    "USER_POST": "JOANA"
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN GETTING/ADDING/EDITING AN ANSWER"
}
```

**JSON416, JSON417 and JSON418: Get, add or edit a comment**

```
{
  "DATA": {
    "FIRSTANSWER": 1,
    "SECONDANSWER": 2,
    "SECONDANSWERDATA": {
      "ID_ANSWER": 4,
      "TEXT": "YOUR ANSWER HELP ME A LOT!",
      "DATE": "2018-01-02 12:35:01",
      "VOTES": 5,
      "PHOTO": NULL,
      "DELETED": "FALSE",
      "ID_QUESTION": 10,
      "USER_POST": "ANNA"
    }
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN GETTING/ADDING/EDITING A COMMENT"
}
```

**JSON502: Get feed**

```

{
  "FEED": {
    "QUESTIONS": [
      {
        "ID_QUESTION": 10,
        "TITLE": "WHAT IS  $\sin(x) + \cos(x)$ ?",
        "DESCRIPTION": "PLEASE HELP ME!",
        "DATE": "2017-01-01 05:00:00",
        "VOTES": 1000,
        "PHOTO": "YVES.JPG",
        "DELETED": FALSE,
        "ID_CATEGORY": 5,
        "ID_USER": 20
      },
      {
        "ID_QUESTION": 4,
        "TITLE": "WHAT IS REQUIRED TO START SCULPTING CLAY (TOOLS,
EXPERIENCE...)?",
        "DESCRIPTION": "PLEASE HELP ME!",
        "DATE": "2017-06-01 14:00:00",
        "VOTES": 500,
        "PHOTO": "YVES.JPG",
        "DELETED": FALSE,
        "ID_CATEGORY": 2,
        "ID_USER": 5
      }
    ],
    "ANSWERS": [
      {
        "ID_ANSWER": 4,
        "TEXT": "MAYBE 1",
        "DATE": "2018-01-02 12:35:01",
        "VOTES": 5,
        "PHOTO": NULL,
        "DELETED": "FALSE",
        "ID_QUESTION": 10,
        "USER_POST": "JOANA"
      },
      {
        "ID_ANSWER": 7,
        "TEXT": "WHEN YOU ARE SCULPTING IT IS IMPORTANT TO HAVE A WELL-
ORGANIZED.",
        "DATE": "2019-01-02 12:35:01",
        "VOTES": 5,
        "PHOTO": NULL,
        "DELETED": "FALSE",
        "ID_QUESTION": 4,
        "USER_POST": "PETER"
      }
    ]
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN GETTING FEED"
}

```

**JSON503: Search results**

```

{
  "SEARCH": {
    "QUESTIONS": [
      {
        "ID_QUESTION": 10,
        "TITLE": "WHAT IS  $\sin(x) + \cos(x)$ ?",
        "DESCRIPTION": "PLEASE HELP ME!",
        "DATE": "2017-01-01 05:00:00",
        "VOTES": 1000,
        "PHOTO": "YVES.JPG",
        "DELETED": FALSE,
        "ID_CATEGORY": 5,
        "ID_USER": 20
      },
      {
        "ID_QUESTION": 4,
        "TITLE": "WHAT IS REQUIRED TO START SCULPTING CLAY (TOOLS,
EXPERIENCE...)?",
        "DESCRIPTION": "PLEASE HELP ME!",
        "DATE": "2017-06-01 14:00:00",
        "VOTES": 500,
        "PHOTO": "YVES.JPG",
        "DELETED": FALSE,
        "ID_CATEGORY": 2,
        "ID_USER": 5
      }
    ],
    "ANSWERS": [
      {
        "ID_ANSWER": 4,
        "TEXT": "MAYBE 1",
        "DATE": "2018-01-02 12:35:01",
        "VOTES": 5,
        "PHOTO": NULL,
        "DELETED": "FALSE",
        "ID_QUESTION": 10,
        "USER_POST": "JOANA"
      },
      {
        "ID_ANSWER": 7,
        "TEXT": "WHEN YOU ARE SCULPTING IT IS IMPORTANT TO HAVE A WELL-
ORGANIZED.",
        "DATE": "2019-01-02 12:35:01",
        "VOTES": 5,
        "PHOTO": NULL,
        "DELETED": "FALSE",
        "ID_QUESTION": 4,
        "USER_POST": "PETER"
      }
    ]
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN SEARCHING"
}

```

**JSON603: Search user**

```
{
  "DATA": {
    "ID": 0,
    "USERNAME": "SOPHIE"
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE GETTING MATCHING USER"
}
```

**JSON604 and JSON606: Get, add or edit a category**

```
{
  "DATA": {
    "ID_CATEGORY": 2,
    "NAME": "BIOLOGY",
    "ICON": "BIOLOGY-01.svg"
  },
  "MESSAGE": "EXAMPLE ERROR MESSAGE WHEN GETTING/ADDING/EDITING A CATEGORY"
}
```

**GROUP1834:**

- João Augusto dos Santos Lima, up201605314@fe.up.pt
- José Diogo da Cunha Moreira Trindade Martins, up201504761@fe.up.pt
- Juliana Maria Cruz Marques, up201605568@fe.up.pt (Editor)
- Leonor Ribeiro e Sousa Mendes de Freitas, up201207603@fe.up.pt