

Code Home

A7: Web Resources Documentation

The architecture of the web application to develop is documented below, 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 CodeHome, 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.

M01: Authenticated and Individual Profile	Web resources associated with user authentication and individual profile management. Includes all the typical user interaction features: login/logout, registration, credential recovery, view and edit personal profile information.
M02: Posts	Web resources associated with questions and includes all their management features: view questions and their answers, edit questions, edit answers, add questions, add answer, report question, report answers, remove questions, remove answers, search and list questions.
M03: User Administration and Static pages	Web resources associated with user management, specifically: view and search users, ban or unban user accounts, view and change user information, and view system access details for each user. Web resources with static content are associated with this module: about, contacts, services and FAQ.

2. Permissions

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

Type	Identifier	Description
PUB	Public	Group of users without privileges
USR	User	Authenticated user
OWN	Owner	Group of users that can update their profiles and have privileges regarding their posts
ADM	Administrator	Group of administrators

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.

Module M01: Authentication and Individual Profile

Endpoints of Module Authentication and Individual Profile

These are the endpoints available in the Authenticated and Individual Profile Module.

- R101: Login Form /login
- R102: Login Action /login
- R103: Logout Action /logout
- R104: Register Form /register
- R105: Register Action /register
- R106: View Homepage Logged In /homepage
- R107: View Homepage Not Logged In /homepage
- R108: View Profile /users/{id}
- R109: Edit Profile Form /users/{id}/edit
- R110: Edit Profile Action /users/{id}
- R111: Delete Profile Action /users/{id}/delete

R101: Login Form

URL	/login
Description	Page with a form to login to a user account.
Method	GET
UI	UI06
SUBMIT	R102
Permissions	PUB

R102: Login Action

URL	/login	
Description	This web resource logs the user into the system. Redirects to the user home page on success and the login form on failure.	
Method	POST	
Request Body	+username: string	Username
	+ pass_token: string	Password
Redirects	R106	Success
	R101	Error
Permissions	PUB	

R103: Logout Action

URL	/logout	
Description	This web resource logs out the authenticated user or admin.	
Method	POST	
Redirects	R107	Success
Permissions	USR, ADM	

R104: Register Form

URL	/signup
Description	Page with a form to register a new user account.
Method	GET
UI	UI07
SUBMIT	R105
Permissions	PUB

R105: Register Action

URL	/signup	
Description	This web resource inserts the user into the system. Redirects to the home page on success and the register form on failure.	
Method	POST	
Request Body	+username: string	Username
	+email: string	Email
	+description: string	Description
	+password: string	Password
	+confirmPassword: string	Password confirmation
	?picture: string	Profile picture path
Redirects	R106	Success
	R104	Error
Permissions	PUB	

R106: View Homepage Logged In

URL	/homepage
Description	Shows the homepage for an authenticated user.
Method	GET
UI	UI02
Permissions	USR

R107: View Homepage Not Logged In

URL	/homepage
Description	Shows the homepage for a visitor.
Method	GET
UI	UI01
Permissions	PUB

R108: View Profile

URL	/users/{id}	
Description	Shows the user individual profile page.	
Method	GET	
Parameters	+id: integer	user primary key
UI	UI11	
Permissions	USR	

R109: Edit Profile Form

URL	/users/{id}/edit	
Description	Page with a form to edit profile info.	
Method	GET	
Parameters	+id: integer	user primary key
UI	UI12	
SUBMIT	R110	
Permissions	USR	

R110: Edit Profile Action

URL	/users/edit/{id}	
Description	Web resource that changes user profile info based on the input received. Redirects to the user's profile on success and edit profile page on failure.	
Method	POST	
Parameters	+id: integer	user primary key
Request Body	?name: string	New name
	?picture: string	New profile picture path
	?username: string	username
	?description: string	description
	?password: string	password
	?confirm password: string	Confirm password
Redirects	R108	Success
	R109	Error
Permissions	USR	

R111: Delete Profile Action

URL	/users/{id}/delete	
Description	Web resource that delete a user's own profile. Redirects to the Homepage not logged in on success and view profile page on failure.	
Method	POST	
Parameters	+id: integer	user primary key
Redirects	R107	Success
	R108	Error
Permissions	USR	

Module M02: Questions and Answers

Endpoints of Module Questions and Answers

These are the endpoints available in the Questions and Answers Module.

- R201: View Question and Answers to that same question /questions/{id}
- R202: Vote on Post Action /voteOnPost/{id}
- R203: Post New Answer /postNewAnswer/{questionId}
- R204: Post New Question Form /postNewQuestion
- R205: Post New Question Action /postNewQuestion
- R206: Search Question /searchQuestion
- R207: Report Question /questions/{id}/reportQuestion
- R208: Delete Question /questions/{id}/deleteQuestion

R201: View Question and Answers to that same question

URL	/questions/{id}	
Description	Page with a question and the answers already given to it. The question and answers are posts. Each post has a form to vote on it. The bottom of the page has a form to post a new answer.	
Method	GET	
Parameters	+id: integer	user primary key
UI	UI09	
SUBMIT	R202	On post vote
	R203	On posting new answer
Permissions	USR	

R202: Vote on Post Action

URL	/voteOnPost/{id}	
Description	This web resource performs a request to vote on a post, using ajax, maintaining the user in the same web page without refreshing it.	
Method	POST	
Parameters	+id: integer	user primary key
Request Body	+postId: int	PostId
	+value: int	Value
Return	200 OK	The vote action took place successfully.
	400 Bad Request	Error. Error message is specified via a HTTP header.
	404 Not Found	Error. No post with the specified id exists.
Redirects	None	Success
	No redirect, but message pops up	Error
Permissions	USR	

R203: Post new Answer

URL	/postNewAnswer/{questionId}	
Description	This web resource allows a user to reply to an existing question.	
Method	POST	
Parameters	+questionId: integer	Question Id
Request Body	+content: string	Content
Return	200 OK	The post answer action took place successfully.
	400 Bad Request	Error. Error message is specified via a HTTP header.
	404 Not Found	Error. No question with the specified id exists.
Redirects	None	Success
	No redirect, but message pops up	Error
Permissions	USR	

R204: Post new Question Form

URL	/postNewQuestion/{questionId}	
Description	Page with form that allows to build a question and then post it.	
Method	GET	
Parameter	+questionId: integer	Question id
UI	UI10	
Submit	R205	
Permissions	USR	

R205: Post new Question Action

URL	/postNewQuestion/{questionId}	
Description	This web resource allows a user to request a post of a new question. It redirects to the newly posted question's page on success and the same page on failure.	
Method	POST	
Parameters	+questionId: integer	Question id
Request Body	+title: string	Title
	+content: string	Content
Return	200 OK	The post question action took place successfully.
	400 Bad Request	Error. Error message is specified via a HTTP header.
Redirects	UI09	Success
	No redirect, but message pops up	Error
Permissions	USR	

R206: Search for Question

URL	/searchForQuestion	
Description	This web resource allows a user to search for a question by text, tag or both.	
Method	GET	
Request Body	+text: string	Text
	+tags: list of strings	Tags
Response body	JSON201	
Redirects	None	Success
	No redirect, but message pops up	Error
Permissions	PUB	

R207: Report Question

URL	/questions/{id}/reportQuestion	
Description	This web resource allows a user to report a question if he thinks that it is not complying with the platform's conduct.	
Method	POST	
Parameters	+questionId: int	QuestionId
Request Body	+id: integer	user id
Return	200 OK	The report question took place successfully.
	400 Bad Request	Error. Error message is specified via a HTTP header.
	404 Not Found	Error. No question with the specified id exists.
Redirects	None	Success
	No redirect, but message pops up	Error
Permissions	PUB	

R208: Delete Question

URL	/questions/{id}/deleteQuestion	
Description	This web resource allows a user to delete a question. Only the question owner or the administrator can delete a question. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+questionId: int	QuestionId
Request Body	+id: integer	user id
Return	200 OK	The delete question took place successfully.
	400 Bad Request	Error. Error message is specified via a HTTP header.
	404 Not Found	Error. No question with the specified id exists.
Redirects	R106	Success
	No redirect, but message pops up	Error
Permissions	USR	

Module M03: User Administration and Static pages

Endpoints of Module User Administration and Static pages

These are the endpoints available in the User Administration and Static pages.

- R301: Ban User Action /ban/{id}
- R302: Unban User Action /unban/{id}
- R303: View User Information /users/{id}/admin
- R304: Edit User Information Form /users/edit/{id}/admin
- R305: Edit User Information Action /users/edit/{id}/admin
- R306: Search User Form /users/{?query}/admin
- R307: Search User Action /users/{?query}/admin
- R308: Remove Post Action /post/{id}/remove/admin
- R309: View Post Reports /post/{id}/reports
- R310: Close Question Action /question/{id}/close
- R311: Mark Answer as Correct Action /answers/{id}/correct
- R312: Edit Post Form /post/{id}/edit/admin
- R313: Edit Post Action /post/{id}/edit/admin
- R314: Edit Question Form /question/{id}/edit/admin
- R315: Edit Question Action /question/{id}/edit/admin
- R316: View Contacts List /contactsList
- R317: View Contact Page /contacts
- R318: View FAQ Page /faq
- R319: View AboutPage /about
- R320: View Error 404 Page /404

R301: Ban User Action

URL	/ban/{id}	
Description	This web resource allows an admin to ban a user. Only an administrator can ban a user. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Id of user to be banned
Request Body	+ StartDate: String	Start date of the ban
	? End Date: String	End date of the ban
	+ IsPermanent: Boolean	Whether the ban is permanent or not
	+ AdminID: Integer	Identifier of the administrator responsible for the ban
Return	200 OK	Successfully banned user
	404 Not Found	Error. User not found
	401 Invalid Permissions	Error. Not enough permissions to execute this action
	407 Invalid Header	Error. Header values are invalid
Permissions	ADM	

R302: Unban User Action

URL	/unban/{id}	
Description	This web resource allows an admin to unban a user. Only an administrator can unban a user. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Unbanned user id
Request Body	+ AdminID: Integer	Identifier of the administrator responsible for the unban
Return	200 OK	Successfully unbanned user
	404 Not Found	Error. User not found
	401 Invalid Permissions	Error. Not enough permissions to execute this action
Permissions	ADM	

R303: View User Information

URL	/users/{id}/admin	
Description	This web resource allows an admin to view all information regarding a user including login type, ban history and post history. Only an administrator can view user information. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+id: integer	user primary key
Request Body	+AdminID: Integer	Identifier of the administrator
Return Body	JSON303	
Permissions	ADM	

R304: Edit User Information Form

URL	/users/edit/{id}/admin	
Description	Page with a form that allows an admin to edit a user's personal information. Only an administrator can edit any user information. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+id: integer	user primary key
Request Body	+AdminID: Integer	Identifier of the administrator
Submit	R305	
Permissions	ADM	

R305: Edit User Information Action

URL	/users/edit/{id}/admin	
Description	Submit action of the form that allows an admin to edit a user's personal information. Only an administrator can edit any user information. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: Integer	user primary key
Request Body	+ AdminID: Integer	Identifier of the administrator responsible for editing the user's profile
	? Username: String	Username set for the user
	? Auth_Type: Integer	Authentication type set for the user
	? Email: String	Email set for the user
	? Description: String	Description set for the user
	? Img_path: String	Image set for the user
	? State: String	State set for the user
Return	200 OK	Successfully edited the user's profile
	404 Not Found	Error. User not found
	401 Invalid Permissions	Error. Not enough permissions to execute this action
	407 Invalid Header	Error. Header values are invalid
Permissions	ADM	

R306: Search User Form

URL	/users/{?query}/admin	
Description	Page with a form that allows an admin to search for users in the system, banned or not. Only an administrator can search for any user. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+query: String	Query String
Request Body	+AdminID: Integer	Identifier of the administrator
Submit	R307	
Permissions	ADM	

R307: Search User Action

URL	/users/{?query}/admin	
Description	Submit action for the page with a form that allows an admin to search for users in the system, banned or not. Only an administrator can search for any user. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+query: String	Query String
Request Body	+ AdminID: Integer	Identifier of the administrator
	+ Query	Username Query
Return Body	JSON302	
Permissions	ADM	

R308: Remove Post Action

URL	/post/{id}/remove/admin	
Description	This web resource allows an admin to delete a post. Only an administrator can delete any post. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Id of the post to be removed
Request Body	+ AdminID: Integer	Administrator id of the administrator removing the post
Return	200 OK	Successfully deleted post
	404 Not Found	Error. Post not found
	401 Invalid Permissions	Error. Not enough permissions to execute this action
Permissions	ADM	

R309: View Post Report

URL	/post/{id}/reports	
Description	This web resource allows an admin to view all the reports associated with a post. Only an administrator can view all post reports. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+id: integer	Id of the post to be reported
Request Body	+ AdminID: Integer	Identifier of the administrator
Return Body	JSON303	
Permissions	ADM	

R310: Close Question Action

URL	/Question/{id}/close	
Description	This web resource allows an admin to close a question. Only an administrator can close any reports. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Id of the question to be deleted
Request Body	+AdminID: Integer	Administrator id of the administrator closing the question
Return	200 OK	Successfully closed question
	404 Not Found	Error question not found
	401 Invalid Permissions	Error not enough permissions to execute this action
Permissions	ADM	

R311: Mark Answer as Correct Action

URL	/answers/{id}/correct	
Description	This web resource allows an admin to mark an answer as correct. Only an administrator can mark any question as correct. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Id of answer to be marked as correct
Request Body	+AdminID: Integer	Administrator id of the administrator marking the answer as correct
Return	200 OK	Successfully marked answer as correct
	404 Not Found	Error answer not found
	401 Invalid Permissions	Error not enough permissions to execute this action
Permissions	ADM	

R312: Edit Post Form

URL	/post/{id}/edit/admin	
Description	Page that allows an admin to edit a post. Only an administrator can edit any post. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+id: integer	Id of the post to be edited
Request Body	+ AdminID: Integer	Identifier of the administrator
Submits	R313	
Permissions	ADM	

R313: Edit Post Action

URL	/post/{id}/edit/admin	
Description	Submit action of the page that allows an admin to edit a post. Only an administrator can edit any post. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	If of the user to be edited
Request Body	+ AdminID: Integer	Identifier of the administrator responsible for editing the post
	? Content: String	Content set for the post
	? PosterID: Integer	Post author set for the post
	? isVisible: Boolean	Visibility set for the post
	? Date: String	Post date set for the post
Return	200 OK	Successfully edited post
	404 Not Found	Error post not found
	401 Invalid Permissions	Error not enough permissions to execute this action
	407 Invalid Header	Error header values are invalid
Permissions	ADM	

R314: Edit Question Form

URL	/question/{id}/edit/admin	
Description	Page that allows an admin to edit a question. Only an administrator can edit any question. To ensure such security, the verification is performed on server-side.	
Method	GET	
Parameters	+id: integer	Id of the question to be edited
Request Body	+AdminID: Integer	Identifier of the administrator
Submits	R315	
Permissions	ADM	

R315: Edit Question Action

URL	/question/{id}/edit/admin	
Description	Submit action of the page that allows an admin to edit a question. Only an administrator can edit any question. To ensure such security, the verification is performed on server-side.	
Method	POST	
Parameters	+id: integer	Id of the question to be edited
Request Body	+ AdminID: Integer	Identifier of the administrator responsible for editing the question
	? Title: String	Title set for the question
	? isClosed: Boolean	Whether or not the question is closed
	? Tags: String	Tags set for the question
Return	200 OK	Successfully edited question
	404 Not Found	Error question not found
	401 Invalid Permissions	Error not enough permissions to execute this action
	407 Invalid Header	Error header values are invalid
Permissions	ADM	

R316: View Contacts List

URL	/contactsList	
Description	This web resource allows an admin to view all contacts sent by users. Only an administrator can view contacts list. To ensure such security, the verification is performed on server-side.	
Method	GET	
Request Body	+ AdminID: Integer	Identifier of the administrator
Return Body	JSON304	
Permissions	ADM	

R317: View Contacts Page

URL	/contacts
Description	Allows any user to view the contacts page
Method	GET
UI	UI05
Permissions	PUB, USR, OWN, ADM

R318: View FAQ Page

URL	/faq
Description	Allows Any User to View the FAQ Page
Method	GET
UI	UI04
Permissions	PUB, USR, OWN, ADM

R319: View About Page

URL	/about
Description	Allows any user to view the about page
Method	GET
UI	UI03
Permissions	PUB, USR, OWN, ADM

R320: View Error 404 Page

URL	/404
Description	Allows any user to view the error 404 page
Method	GET
UI	UI08
Permissions	PUB, USR, OWN, ADM

4. JSON/XML Types

JSON201: search question: { question }[]

```
{
  "matching questions": [
    {
      "id": 1,
      "title": "Is Java better than c++?",
      "description": "I'm new to programming languages and I would like
to know if java is better than c++ in...",
      "votes": -500,
      "poster": "programmingnewbie666"
    },
    {
      "id": 2,
      "title": "Why is java better than c++ having less bugs in code?",
      "description": "My teacher at university says that java is memory
leak proof, while c++ is really bad as it may have...",
      "votes": 100,
      "poster": "javalover8"
    }
  ]
}
```

JSON301: user admin lookup: { admin }[]

```
{
  "id": 15,
  "username": "ProgrammingGeek",
  "email": "programminggeek@gmail.com",
  "description": "Programmer at Microsoft. Favorite language is c++",
  "img_path": "15.png",
  "points": 250,
  "passwordToken": "fsadfasui24112hgdrclslo236121ksfhksd",
  "AuthType": 0,
  "State": "Active",
  "Bans": [
    {
      "StartDate": "15-Jan-2017",
      "EndDate": "15-Jan-2018",
      "IsPermanent": false,
      "Reason": "Harassment",
      "AdminID": 1694
    }
  ],
  "Posts": [
    {
      "status": "Removed by admin",
      "type": "Answer",
      "Content": "You sir are the dumbest piece of crap I have ever
seen please kill yourself",
      "points": -99999,
      "isVisible": false,
      "isCorrect": false
    }
  ]
}
```

JSON302: user admin search: { admin }[]

```
{
  "matchingUsers": [
    {
      "username": "fancyDogLover",
      "email": "fancydogs@gmail.com",
      "points": 50
    },
    {
      "username": "fancyCatLover",
      "email": "fancygator@gmail.com",
      "points": 5054
    }
  ]
}
```

JSON303: admin post report view: { admin }[]

```
{
  "id": 589,
  "reports": [
    {
      "id": 125,
      "reporterID": 51,
      "reporterUsername": "fancyCatLover",
      "Reason": "Contains Inappropriate language"
    },
    {
      "id": 126,
      "reporterID": 541,
      "reporterUsername": "fancyDogLover",
      "Reason": "Is Harassment"
    }
  ]
}
```

JSON304: admin contacts view: { admin }[]

```
{
  "contacts": [
    {
      "id":159,
      "userID":147,
      "username": javacompiler",
      "subject": "Bug",
      "message": "The Post Report Function Is Bugged",
      "date":"14-Jan-2018"
    },
    {
      "id":160,
      "userID":14796,
      "username": gccompiler",
      "subject": "Bug",
      "message": "The Post Report Function Is Not Working
Good",
      "date":"1-Jan-2018"
    }
  ]
}
```

Revision History

- Fixed incorrect redirect. In delete question, on success, the user is now redirected to Homepage logged in.
- Improved some User Administration module descriptions. Added relevant information about user permissions and server-side verifications.
- Improved some table layouts

Group

- Davide Henrique Fernandes da Costa, up201503995@fe.up.pt
- Dinis Filipe da Silva Trigo, up201504196@fe.up.pt
- Diogo Afonso Duarte Reis, up201505472@fe.up.pt
- Tiago José Sousa Magalhães, up201607931@fe.up.pt