

URL shortener specification

1 | Context

- The project is used for student's educational purposes
- Students should design and implement the project in teams of 3 to 5
- The project's goal is to emulate real-life software engineering workflow
- Teams will swap projects after Lab #3 is complete
- The project should be implemented in Java, as voted by the students
- The project doesn't include any UI

2 | Goal

The project is a REST API application that can be used to shorten your cumbersome links into more usable URLs that are easy to share and manage; briefly, a URL shortener. The main idea behind the project is the "The shorter the link, the better".

3 | Functional requirements

1. Register a new user
2. Authentication
3. A user can create new short links
4. A user can provide their own aliases (instead of a unique identifier generated by the service)
5. A user can enumerate all links that they have shortened
6. A user can delete the links that they have created
7. Anyone can use short links

4 | Scenarios

4.1 | Authorization flow:

1. The user creates an account
 - a. User's credentials (**email**, **password**) are sent via **POST /users/signup**
2. The account has been created
3. The user uses their **email** and **password** to sign in via **POST /users/signin**

4. The sign in response contains an authentication token, which should be placed in the Authorization header in future requests (e.g. **Authorization: Bearer simple-token-example-here**).
5. A signed in user can sign out (invalidate their token) via **POST /users/signout**

4.2 | General flow

1. *Precondition: A user is authenticated (a token is provided)*
2. A user's destination link (**url**) is passed for shortening via **POST /urls/shorten**
 - a. A user can pass optional param **alias** with desired URL
 - i. An error would occur if given **alias** already exists (among all aliases known to the service)
 - ii. An error would occur if given **alias** doesn't have alphanumeric format (*it should consist only of numbers, English letters and _ character*)
 - b. The response contains a constructed link URL (**shortened_url**), which is **public**, so that it can be used without authorization by following the resulting link or which equals to a **GET /r/<alias>** request (e.g. **GET /r/a1b2c3** - where *a1b2c3* is an alias)

4.3 | Listing user's URL aliases

1. *Precondition: A user is authenticated (a token is provided)*
2. All URLs shortened by the user can be listed by a **GET /urls** request

4.4 | Deletion of a URL alias

1. *Precondition: A user is authenticated (a token is provided)*
2. A user's URL alias can be deleted by its owner via **DELETE /urls/<alias>** request

5 | REST API endpoints

5.1 | Sign Up

POST /users/signup

Parameters:

email: string, required
password: string, required

Result:

OK/error

Possible errors:

- Malformed request
- Email is already used

5.2 | Sign In

POST /users/signin

Parameters:

- email:** string, required
- password:** string, required

Result:

- OK/error

Returns:

- token:** string

Possible errors:

- Malformed request
- User was not found

5.3 | Sign Out

POST /users/signout

Result:

- OK/error

Possible errors:

- User is not authorized*

5.4 | Create an URL alias

POST /urls/shorten

Parameters:

- url:** string, required - link which has to be shortened
- alias:** string, optional - desired alias (*not a full shortened link*)

for the link

Result:

- OK/error

Returns:

- shortened_url:** string

Possible errors:

- User is not authorized*
- Malformed request
- Provided **url** is not a valid URL
- Passed **alias** already exists (among all aliases known to the service)
- Passed **alias** is invalid (is not an alphanumeric string)

5.5 | List user's shortened links

GET /urls

Result:

OK/error

Returns:

urls: array

Possible errors:

User is not authorized

5.6 | Delete shortened link

DELETE /urls/<alias>

Result:

OK/error

Possible errors:

User is not authorized

alias was not found among created by the user

5.7 | Redirect by a shortened URL

GET /r/<alias>

Result:

Redirect/Error

Possible errors:

Given **alias** was not found