

URL shortener

Author: Dragan Benčić (dragan.bencic@infobip.com)

Assignment: Write an HTTP service which is used to shorten URLs, with following functionalities:

- Registration of a web address (API)
- Redirect of clients call according to the shortened URL-u
- Statistic of API calls (API)

Description:

1. Basic Architecture

The service must have two parts: *configuration part* and *user part*.

1.1 Configuration part:

Configuration part is called over REST with JSON parameters and it has following functionalities:

- Account creation
- Registration of URL-s in the „Shortener“ service
- Statistics retrieval

1.1.1 Account creation

HTTP method	POST
URI	/account
Request type	application/json
Request Body	JSON with following parameters: <ul style="list-style-type: none"> • AccountId (String, mandatory) Example: { AccountId : 'myAccountId'}
Response Type	application/json
Response	Two types of response: successful and non-successful. Non-successful registration is when the account (AccountId) already exists. The parameters are: <ul style="list-style-type: none"> • success: true false • description: Status description • password: It is returned only for successfully created accounts. Automatically generated password is an 8 alphanumeric string Example {success: 'true', description: 'Your account is opened', password: 'xC345Fc0'}

1.1.2 URLs Registration

HTTP method	POST
URI	/register
Request type	application/json
Request Headers	Set Authorization header and perform user authentication
Request Body	JSON with following parameters: <ul style="list-style-type: none"> • URL (mandatory, URL that needs to be shortened) • redirectType : 301 302 (optional, default 302)

	Example: { URL: 'http://stackoverflow.com/questions/1567929/website-safe-data-access-architecture-question?rq=1', redirectType : 301 }
Response Type	application/json
Response	Response parameters in case of a successful registration: <ul style="list-style-type: none"> shortUrl (shortened URL) Example: { shortUrl: 'http://short.com/xYswIE' }

1.1.3 Statistics get

HTTP method	GET
URI	/statistic/{AccountId}
Request Headers	Set Authorization header and perform user authentication
Response Type	application/json
Response	The server responds with JSON object (key:value map) where the key is the registered URL, and the value is the number of calls to that URL. Example: <pre>{ 'http://myweb.com/something1/somethingelse1/: 10, 'http:// myweb.com/something2/somethingelse2/: 4, 'http:// myweb.com/something3/somethingelse3/: 91, }</pre>

1.2 Redirect

Redirect the client on the configured address with the configured HTTP status

2. General requirements

- Use Java programming language
- Make sure that the response HTTP statuses meet the REST standards (statuses <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>)
- The service needs to be written as an 'executable' or 'deployable' package.
- The application must not need any other dependency other than the ones in POM file (must run out-of-the-box).
- In accordance with the previous requirement, it is not allowed to use databases other than the embedded ones.
- Any framework can be used (Spring preferred)
- Make a help page (URI: /help) containing instructions of installation/run and usage
- Provide the source code with all the dependencies, preferably as a maven project.