

System Documentation

URLShortener Challenge

January 20, 2019

Team Member(s)

Tarcisio Franco de Carvalho

Contents

Contents.....	2
1 Introduction.....	5
2 Application Server and REST ful framework.....	5
3 APIs.....	5
4 Web Client for Testing Purposes.....	8
5 Project Contact.....	8

Revision History

Versi on	Date	Name	Description
1	2019/01/20	Tarcisio Carvalho	Initial Document

1 Introduction

The development of the URLShortener back-end application is related to a VanHack challenge for Java developers ([challenge link](#)). This document will provide details about how this application works.

2 Data persistence

NoSQL MongoDB hosted on mlab was selected to persist the data of this application. There are two collections, the “urls” that stores all shortened and original URLs, and the “forward” that stores the forward URL requests. Below you can see the URI to the database connection

mongodb://tarcisiofcarvalho:mongo10mongo@ds231951.mlab.com:31951/url_shortener

3 Application Server and REST ful framework

Apache Tomcat Server 7.0 and Jersey 2.0 framework are being used to develop and test the APIs.

4 APIs

Were developed 3 APIs and 1 Web Client Application used to test the APIs. Below you can see the GitHub link for this solution.

<https://github.com/tarcisiofcarvalho/vanhack>

4.1 Shortened URL Generation API

The objective of this API is generate a short URL as **tfc.vh/Edt85** from original large URLs and save the data in NoSQL base. The method available for this API needs the parameters url (the original) and alias (a readable name just for reference). It will return a JSON message with short URL generated. Please see below more details about this API:

Project on Git

<https://github.com/tarcisiofcarvalho/vanhack/tree/master/url-generator-api/url-generator-api>

REST URI and Class

<http://localhost:8080/url-generator-api/rest/generator/data>

`com.urlshortener.api.URLGeneratorResource`

Method signature

@POST

@Path("data")

@Produces(MediaType.APPLICATION_JSON)

```
public Response generator(@FormParam(value = "url") String url,  
    @FormParam(value = "alias") String alias) {
```

JSON result

```
{'newURL' : 'tfc.vh/PzrQe'}
```

4.2 Forward URL API

This API gets the short URL, retrieves the original URL from the NoSQL and redirect the page to the original URL. The method available for this API needs the parameters shortened URL. It will redirect the request to the original URL. Please see below more details about this API:

Project on Git

<https://github.com/tarcisiofcarvalho/vanhack/tree/master/url-forward-api/url-forward-api>

REST URI

<http://localhost:8080/url-forward-api/rest/forward/data>

Method signature

```
@POST
@Path("/data")
@Produces(MediaType.APPLICATION_JSON)
public Response forward(@FormParam(value = "shortenedURL") String
shortenedURL) {
```

4.3 URL Generation and Forward Statistics API

There are two statistics, the number of shortened URL created by period, and the number of forward by period. Below you can see the details about both resources:

1. Get Number of Shortened URL generation by period

REST URI

<http://localhost:8080/url-statistics-api/rest/statistics/generation>

Project on Git

<https://github.com/tarcisiofcarvalho/vanhack/tree/master/url-statistics-api/url-statistics-api>

Method signature

```
@POST
@Path("/generation")
@Produces(MediaType.APPLICATION_JSON)
public Response generation(@FormParam(value = "urlShortened") String urlHash,
@FormParam(value = "fromDate") String fromDate,
@FormParam(value = "toDate") String toDate,
@FormParam(value = "alias") String alias) {
```

JSON result

```
{'urlGeneratedCount' : '2', 'fromDate' : '2019-01-01', 'toDate' : '2019-01-20',
'urlHash' : '', 'alias': ''}
```

2. Get Number of Forwards by period

REST URI

<http://localhost:8080/url-forward-api/rest/forward/data>

Method signature

```
@POST
@Path("forward")
@Produces(MediaType.APPLICATION_JSON)
public Response forward(@FormParam(value = "urlShortened") String urlHash,
    @FormParam(value = "fromDate") String fromDate, @FormParam(value = "toDate")
    String toDate)
```

JSON result

```
{'forwardCount' : '8', 'fromDate' : '2019-01-01', 'toDate' : '2019-01-20',
'urlHash' : ''}
```

5 Web Client for Testing Purposes

The simple Java Web Client below was developed in order to test the APIs. You can see in the test page a request for each API developed. The all them return a JSON with the answer each REST API service, just the Foward API that will redirect you request to the original URL. Below you can see more details:

Project on Git

<https://github.com/tarcisiofcarvalho/vanhack/tree/master/WebTestClient/WebTestClient>

6 Project Contact

Tarcisio Franco de Carvalho
tarcisiofcarvalho@yahoo.com.br
Mobile: +55 19 98118 2745