

# ***REST File Sharing***

***Peter Otruba***

## **Obsah:**

|                              |   |
|------------------------------|---|
| Popis aplikácie.....         | 1 |
| REST endpointy .....         | 2 |
| File-controller.....         | 2 |
| /api/file/upload .....       | 2 |
| /api/filejson.....           | 3 |
| /api/file/{id} .....         | 3 |
| /api/file/generatelink ..... | 4 |
| /api/deletefile .....        | 4 |
| User-controller.....         | 4 |
| api/register .....           | 5 |
| api/getuser/{username} ..... | 5 |
| UI – thymeleaf.....          | 6 |
| Zdroje .....                 | 6 |

## Popis aplikácie

Server je navrhnutý, aby fungoval v princípe ako populárne stránky pre file sharing, ako napríklad [uloz.to](http://uloz.to) alebo [mediafire](http://mediafire.com). Server je implementovaný pomocou REST servera, ktorý využíva Spring boot pre backend, Sqlite ako perzistentnú databázu a Thymeleaf ako jednoduché UI.

Server podporuje registráciu pomocou REST volania a prihlásenie pomocou REST alebo webového prihlásenia.

Server povoľuje anonymným a zaregistrovaným užívateľom uploadnúť súbory pomocou REST volaní. Zaregistrovaní užívatelia taktiež môžu vymazať súbor, ktorí vlastnia. Užívatelia, ktorí sú administrátori, môžu vymazať každý súbor a taktiež zistiť presné informácie o každom užívateľovi.

Po úspešnom uploade súboru sa vygeneruje unikátny link, ktorý umožňuje každému pomocou vygenerovaného linku stiahnuť súbor.

## REST endpointy - swagger

| file-controller |                         | ^ |
|-----------------|-------------------------|---|
| PUT             | /api/file/upload        | ▼ |
| GET             | /api/filejson           | ▼ |
| GET             | /api/file/{id}          | ▼ |
| GET             | /api/file/generatelink  | ▼ |
| DELETE          | /api/deletefile         | ▼ |
| user-controller |                         | ^ |
| POST            | /api/register           | ▼ |
| GET             | /api/getuser/{username} | ▼ |

### File-controller

Kontrolér pre endpointy, ktoré poskytujú funkcionality pre správu súborov.

#### /api/file/upload

|  |                    |              |
|--|--------------------|--------------|
| PUT  | /api/file/upload   | ^            |
| <b>Parameters</b>                          |                    |              |
| No parameters                              |                    |              |
| <b>Request body</b>                        |                    |              |
| multipart/form-data ▼                      |                    |              |
| <b>file</b> * required<br>string(\$binary) |                    |              |
| <b>Responses</b>                           |                    |              |
| <b>Code</b>                                | <b>Description</b> | <b>Links</b> |
| 200  | OK                 | No links     |
| Media type                                 |                    |              |
| */* ▼                                      |                    |              |
| Controls Accept header.                    |                    |              |
| Example Value   Schema                     |                    |              |
| string                                     |                    |              |

Endpoint konzumuje súbor – multipart/form-data. V tele je potrebné mať súbor v binárnej reprezentácii. Ak je upload úspešný, server odpovie statusom 200 – OK.

## /api/filejson

GET

/api/filejson

^

Parameters

Try it out

| Name  | Description                     |
|---|---------------------------------|
| <b>id</b> * required<br>integer(\$int64)<br>(query) | <input type="text" value="id"/> |

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | OK          | No links |

Media type

Controls Accept header.

Example Value | Schema

string

Endpoint vráti uploadnutý súbor v databáze vo formáte Json. Od užívateľa vyžaduje ID súboru, o ktorom chce informáciu vo formáte Json.

## /api/file/{id}

GET

/api/file/{id}

^

Parameters

Try it out

| Name   | Description                     |
|--|---------------------------------|
| <b>id</b> * required<br>integer(\$int64)<br>(path) | <input type="text" value="id"/> |

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | OK          | No links |

Media type

Controls Accept header.

Example Value | Schema

string

Endpoint vracia súbor v binárnej reprezentácii v tele, konzumentom endpointu taktiež poskytuje informáciu o type návratovej hodnoty pomocou definície typu médiá - multipart/form-data.

Od užívateľa vyžaduje ID súboru, ktoré sa snaží stiahnuť.

## /api/file/generatelink

**GET** /api/file/generatelink

Try it out

| Name  | Description                     |
|---|---------------------------------|
| <b>id</b> * required<br>integer(\$int64)<br>(query) | <input type="text" value="id"/> |

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | OK          | No links |

Media type

\*/\*

Controls Accept header:

Example Value | Schema

string

Endpoint vygeneruje unikátny link pre stiahnutie súboru. Od užívateľa potrebuje ID súboru, pre ktoré chce vytvoriť unikátny link.

## /api/deletefile

**DELETE** /api/deletefile

Try it out

| Name  | Description                     |
|---|---------------------------------|
| <b>id</b> * required<br>integer(\$int64)<br>(query) | <input type="text" value="id"/> |

Responses

| Code | Description | Links    |
|------|-------------|----------|
| 200  | OK          | No links |

Media type

\*/\*

Controls Accept header:

Example Value | Schema

{}

Endpoint poskytuje funkcionality pre vymazanie súboru. Od užívateľa vyžaduje ID súboru, ktoré chce vymazať. Pre úspešne vymazanie súboru je potrebné, aby užívateľ bol vlastníkom súboru alebo mal rolu administrátora. Ak je súbor vymazaný úspešne, server odpovie statusom 200 – OK.

## User-controller

## api/register

**POST** /api/register

Try it out

| Name  | Description                           |
|---|---------------------------------------|
| <b>name</b> * required<br>string<br>(query)     | <input type="text" value="name"/>     |
| <b>password</b> * required<br>string<br>(query) | <input type="text" value="password"/> |

Responses

| Code | Description   | Links    |
|------|---|----------|
| 200  | OK<br><br>Media type<br><input type="text" value="*/"/><br>Controls Accept header.<br>Example Value   Schema<br><pre>{}</pre> | No links |

Funkcionalita pre registráciu užívateľa vo forme REST endpointu. Vyžaduje od užívateľa meno a heslo, s ktorým sa chce zaregistrovať. Pre úspešnú registráciu je potrebné zadať unikátne meno.

## api/getuser/{username}

**GET** /api/getuser/{username}

Try it out

| Name   | Description                           |
|--|---------------------------------------|
| <b>username</b> * required<br>string<br>(path) | <input type="text" value="username"/> |

Responses

| Code | Description  | Links    |
|------|--|----------|
| 200  | OK<br><br>Media type<br><input type="text" value="*/"/><br>Controls Accept header.<br>Example Value   Schema<br><pre>{<br/>  "enabled": true,<br/>  "password": "string",<br/>  "authorities": [<br/>    {<br/>      "authority": "string"<br/>    }<br/>  ],<br/>  "username": "string",<br/>  "accountNonLocked": true,<br/>  "credentialsNonExpired": true,<br/>  "accountNonExpired": true<br/>}</pre> | No links |

Endpoint poskytuje detailné informácie o užívateľovi so špecifickým menom – meno, heslo a role pridelené ku danému užívateľovi. Pre prístup ku endpointu je potrebné, aby užívateľ, ktorý vykonáva request, mal rolu administrátora.

## UI – thymeleaf

Upload a file:  No file selected.

All files :

| Filename   | Link  | Owner     |
|------------|---|-----------|
| Testik.txt | <a href="http://localhost:13337/api/file/1">http://localhost:13337/api/file/1</a>   | Anonymous |
| Testik.txt | <a href="http://localhost:13337/api/file/2">http://localhost:13337/api/file/2</a>   | Anonymous |
| Testik.txt | <a href="http://localhost:13337/api/file/3">http://localhost:13337/api/file/3</a>   | Anonymous |
| Testik.txt | <a href="http://localhost:13337/api/file/4">http://localhost:13337/api/file/4</a>   | Anonymous |
| Testik.txt | <a href="http://localhost:13337/api/file/52">http://localhost:13337/api/file/52</a> | Anonymous |

UI je zobrazené pomocou [Thymeleaf](#). UI poskytuje tlačidlo pre výber súboru zo súborového systému a tlačidlo pre následný upload súboru. Taktiež zobrazuje všetky momentálne dostupné súbory, ktoré sú uložené v databáze, vrátane vlastníka súboru, mena súboru a unikátneho linku pre stiahnuté súboru.

## Zdroje

[Spring inicializr](#)

[Baeldung.com](#)

<https://docs.spring.io/spring-security/reference/servlet/configuration/java.html>

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#jpa.query-methods.query-creation>

<https://spring.io/guides/topicals/spring-security-architecture/>

<https://www.baeldung.com/spring-security-check-user-role>

<https://www.baeldung.com/spring-controllers>

<https://spring.io/guides/gs/uploading-files/>

<https://www.thymeleaf.org/doc/articles/standardurlsyntax.html>

[https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)

<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org.springframework.web.util.UriComponentsBuilder.html>

[Spring Boot With SQLite | Baeldung](#)

[Spring Security Custom Authentication Provider | Baeldung](#)

[Spring Security](#)

[Iteration in Thymeleaf | Baeldung](#)

[Defining Unique Constraints in JPA | Baeldung](#)

[Hashing a Password in Java | Baeldung](#)

<https://www.baeldung.com/java-db-storing-files>

