A

Project Report On

FILE UPLOAD APPLICATION

For the course in

PROFESTIONAL PRACTICE IN IT

Submitted by

Student name : **Muhammad Adil**

Gnumber : **g00376679**

For degree in

BSC Software Development

Acknowledgment

I like to share our sincere gratitude to all those who help us in completion of this project. During the work we faced many challenges due to our lack of knowledge and experience but these people help us to get over from all the difficulties and in final compilation of our idea to a shaped sculpture.

I would like to thank Sir Joseph Corr for his governance and guidance,

because of which our whole team was able to learn the minute aspects of a project work.

We would also like to show our gratitude to our supervisors for this module :

Dominic Carr

Joe Corr

Damien Costello

Daniel Cregg

John French

Martin Hynes

I are also thankful to our whole class and most of all to our parents who have inspired us to face all the challenges and win all the hurdles in life.

Thank you All.

Abstract

The aim of the project work was to build a minimum working model for a file upload app, it is a web based application, which allows user to upload and download the resources , it runs on any web browser which support java-script, the front-end is based on angular and back-end is based on java , developed using spring boot framework which has an embedded tomcat server. This report contains a comprehensive view of the design, development approaches involved and implemented.

Prerequisites

These are the technologies that we are using in our project.

Server-side technologies

- Spring Boot - 2.0.5. RELEASE
- JDK - 1.8 or later
- Tomcat Server
- Hibernate - 5.2.17. Final
- Spring Data JPA - 2+

Front end technologies

- Angular 8/9
- Bootstrap 3/4
- npm- 6.9.0

Tools

- Maven - 3.2+
- IDE – Eclipse – Used for Spring boot API development
- Visual Studio 2017 – Used for Angular App development
- Angular CLI

Explanation Of Each Technology

Back-end technologies

Java 8 – It provides some best feature that makes the code re-usable, and makes code more readable, also avoid boiler plate code

Some of the important Java 8 features are;

- forEach() method in Iterable interface.
- default and static methods in Interfaces.
- Functional Interfaces and Lambda Expressions.
- Java Stream API for Bulk Data Operations on Collections.
- Java Time API.
- Collection API improvements.
- Concurrency API improvements.
- Java IO improvements.

Spring Boot Framework-

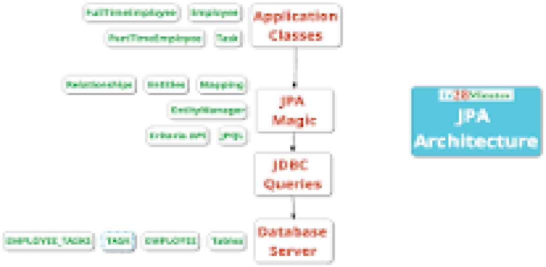
Java Spring Boot (Spring Boot) is **a tool that makes developing web application and microservices with Spring Framework faster and easier** through three core capabilities: c. An opinionated approach to configuration. The ability to create standalone applications

Embedded Tomcat Server

Spring boot provided embedded tomcat server to run the spring boot application efficiently, by default is used 8080 port that can we configured easily using application. Properties file

Hibernate/JPA-

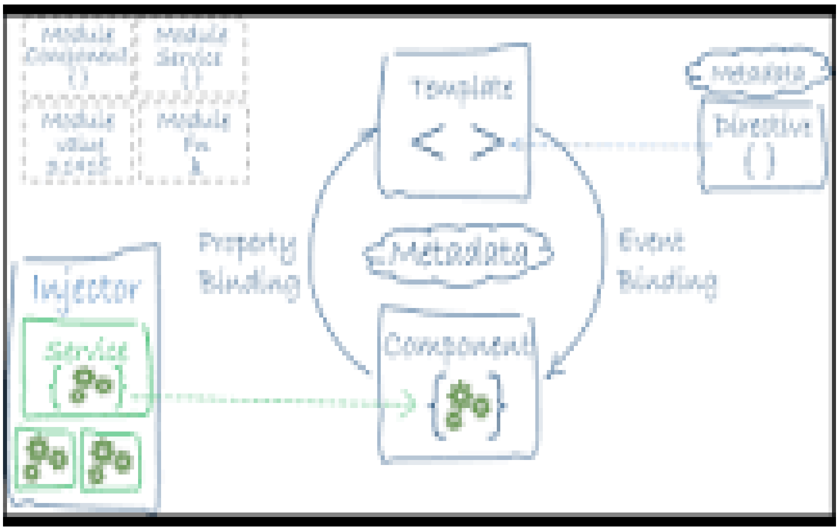
Why we use Hibernate in spring boot?



Hibernate understands the mappings that we add between objects and tables. **It ensures that data is stored/retrieved from the database based on the mappings**. Hibernate also provides additional features on top of JPA. But depending on them would mean a lock in to Hibernate

Font-end technologies

Angular 8



Angular is a platform and framework for **building single-page client applications using HTML and TypeScript**. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

Bootstrap

Bootstrap is a potent front-end framework used to **create modern websites and web apps**. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions.

NPM

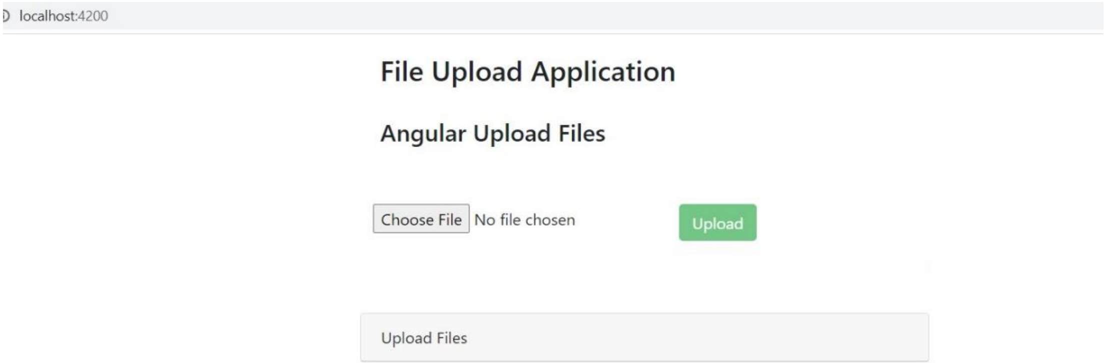
npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to **publish, discover, install, and develop node programs**.

Angular CLI

The Angular CLI is a command-line interface tool that you use to **initialize, develop, scaffold, and maintain Angular applications directly from a command shell**.

<u>Some Snapshot of UI along with its functionality</u>

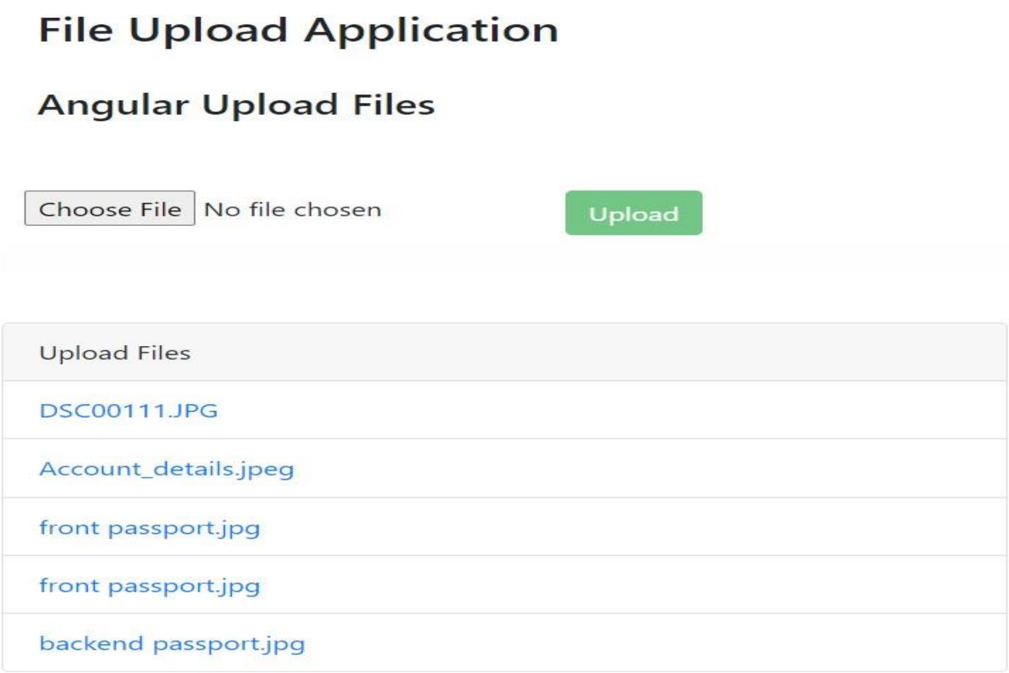
Angular 8 App for upload File



File Upload Application

Angular Upload Files

Upload Files


File Upload Application

Angular Upload Files

Upload Files

DSC00111.JPG

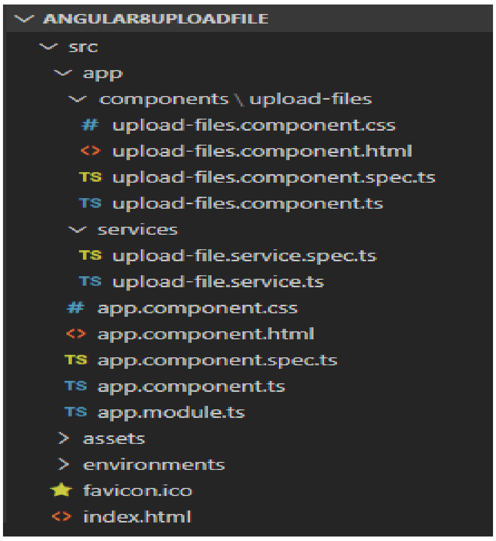
Account_details.jpeg

front passport.jpg

front passport.jpg

backend passport.jpg

Angular Project Package Structure



```
ANGULAR8UPLOADFILE
  src
    app
      components \ upload-files
        upload-files.component.css
        upload-files.component.html
        upload-files.component.spec.ts
        upload-files.component.ts
      services
        upload-file.service.spec.ts
        upload-file.service.ts
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
      app.module.ts
    assets
    environments
    favicon.ico
    index.html
```

Let me explain it briefly.

– We import necessary library, components in *app.module.ts*.
– *upload-file.service* provides methods to save File and get Files from Spring Boot Server.
– *upload-files.component* contains upload form, progress bar, display of list files.
– *app.component* is the container that we embed all components.
– *index.html* for importing the Bootstrap.

Add Bootstrap to the project

Open *index.html* and add following line into `<head>` tag:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <link type="text/css" rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" />
  </head>
  ...
</html>
```

Spring Boot Rest APIs for uploading Files to Database

Our Spring Boot Application will provide APIs for:

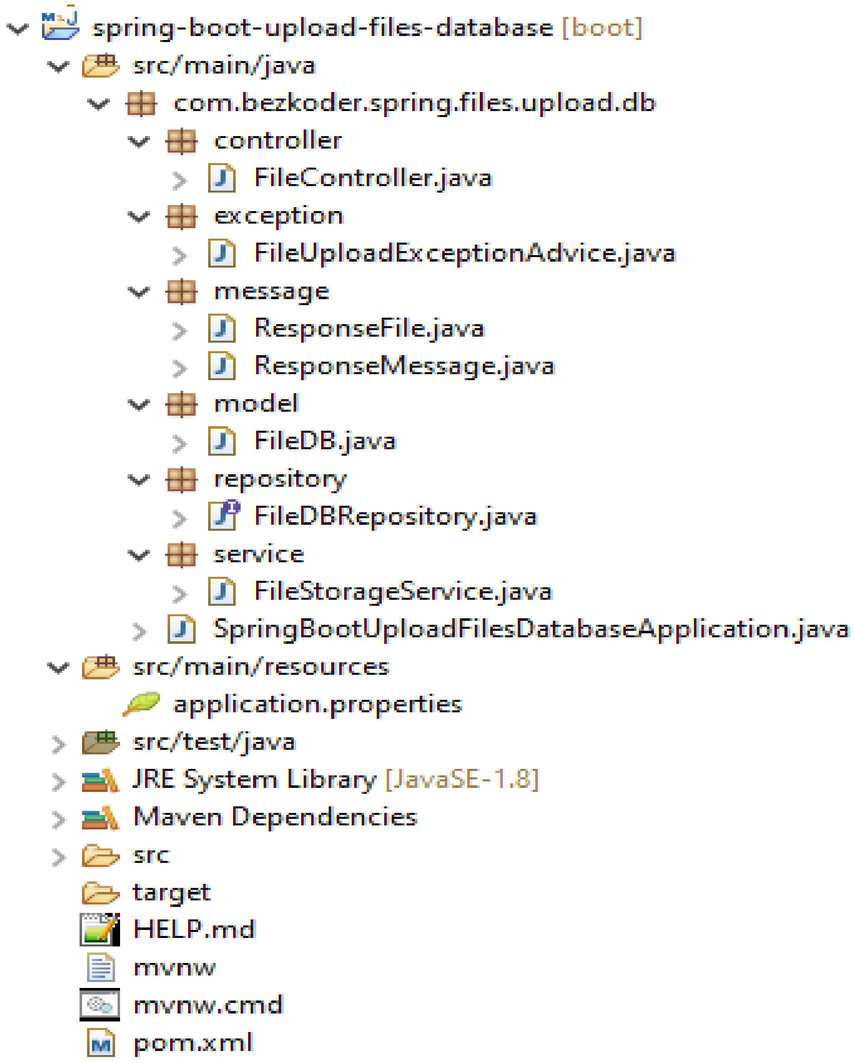
- uploading File to PostgreSQL/MySQL database
- downloading File database with the link
- getting list of Files' information (file name, url, type, size)

These are APIs to be exported:

Methods	Urls	Actions
POST	/upload	upload a File
GET	/files	get List of Files (name, url, type, size)
GET	/files/[fileId]	download a File

The uploaded files will be stored in MySQL Database **files** table with these fields: `id`, `name`, `type` and `data` as **BLOB** type (Binary Large Object is for storing binary data like file, image, audio, or video).

Project Structure



```
spring-boot-upload-files-database [boot]
  src/main/java
    com.bezkoder.spring.files.upload.db
      controller
        FileController.java
      exception
        FileUploadExceptionAdvice.java
      message
        ResponseFile.java
        ResponseMessage.java
      model
        FileDB.java
      repository
        FileDBRepository.java
      service
        FileStorageService.java
      SpringBootUploadFilesDatabaseApplication.java
  src/main/resources
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Let me explain it briefly.

– `FileDB` is the data model corresponding to **files** table in database.
– `FileDBRepository` extends Spring Data `JpaRepository` which has methods to store and retrieve files.
– `FilesStorageService` uses `FileDBRepository` to provide methods for saving new file, get file by id, get list of Files.
– `FilesController` uses `FilesStorageService` to export Rest APIs: POST a file, GET all files' information, download a File.
– `FileUploadExceptionAdvice` handles exception when the controller processes file upload.
– `ResponseFile` contains information of the file (name, url, type, size) for HTTP response payload.
– *application.properties* contains configuration for Servlet Multipart and PostgreSQL/MySQL database connection.
– *pom.xml* for Spring Boot, Spring Data JPA and PostgreSQL/MySQL connector dependency.

Configure Spring Datasource, JPA, Hibernate

Under src/main/resources folder, open application.properties and write these lines.

– For MySQL:

```
spring.datasource.url= jdbc:mysql://localhost:3306/testdb?useSSL=false

spring.datasource.username= root

spring.datasource.password= root

spring.jpa.properties.hibernate.dialect=
org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)

spring.jpa.hibernate.ddl-auto= update
```

Let’s define the maximum file size that can be uploaded in *application.properties* as following:

```
spring.servlet.multipart.max-file-size=2MB

spring.servlet.multipart.max-request-size=2MB
```

– `spring.servlet.multipart.max-file-size`: max file size for each request.
– `spring.servlet.multipart.max-request-size`: max request size for a multipart/form-data.

Run the App

Run Spring Boot Server with command: `mvn spring-boot:run`.
Refresh the project directory and you will see *uploads* folder inside it.

Backend will be started on port number `http://localhost:8080`

Because we configure CORS for origin: `http://localhost:4200`, so you need to run Angular 8 Client with command:
`ng serve --port 4200`

Open Browser with url `http://localhost:4200/` and check the result.

References

https://angular.io/

https://spring.io/projects/spring-boot

https://www.w3schools.com/bootstrap/

https://www.baeldung.com/spring-boot-hibernate

https://www.geeksforgeeks.org/spring-boot-integrating-hibernate-and-jpa/