1.SPRING BOOT EXCEPTION HANDLING.

//exception handiling

@ControllerAdvice

public class GlobalExceptionHandler {

@ExceptionHandler(Exception.class)

public ResponseEntity<String> handleException(Exception ex) {

// Handle the exception and return an appropriate response

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body("An error occurred");

}

}

//Mycontroller.java

@RestController

public class MyController {

@ExceptionHandler(ResourceNotFoundException.class)

public ResponseEntity<String> handleResourceNotFoundException(ResourceNotFoundException ex) {

// Handle the exception and return an appropriate response

return ResponseEntity.status(HttpStatus.NOT\_FOUND).body(ex.getMessage());

}

// Other controller methods...

}

**//ResponseStatusException**

@GetMapping("/users/{id}")

public User getUser(@PathVariable Long id) {

User user = userRepository.findById(id)

.orElseThrow(() -> new ResponseStatusException(HttpStatus.NOT\_FOUND, "User not found"));

return user;

}

**//Default Error Handling**

@Controller

public class CustomErrorController implements ErrorController {

@RequestMapping("/error")

public ResponseEntity<String> handleError(HttpServletRequest request) {

// Extract error information from the request and handle the exception

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body("An error occurred");

}

@Override

public String getErrorPath() {

return "/error";

}

}

2.Create a project to Consume a RESTful web service

//xmlfile

<!-- Spring Boot Starter Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

//gradle

// Spring Boot Starter Web

implementation 'org.springframework.boot:spring-boot-starter-web'

//myrest.java

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.client.RestTemplate;

@RestController

public class MyRestClient {

@GetMapping("/consume")

public String consumeWebService() {

RestTemplate restTemplate = new RestTemplate();

ResponseEntity<String> response = restTemplate.getForEntity("https://api.example.com/data", String.class);

return response.getBody();

}

}

//Run the application

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MyApplication {

public static void main(String[] args) {

SpringApplication.run(MyApplication.class, args);

}

}

3. Create a project to upload and download a file in Spring Boot.

//xmlfile

<!-- Spring Boot Starter Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

//for gradle

// Spring Boot Starter Web

implementation 'org.springframework.boot:spring-boot-starter-web'

//app properties

# File upload properties

spring.servlet.multipart.enabled=true

spring.servlet.multipart.max-file-size=10MB

spring.servlet.multipart.max-request-size=10MB

spring.servlet.multipart.location=uploads

# File download properties

file.download-dir=downloads

//for gradle.yml

# File upload properties

spring:

servlet:

multipart:

enabled: true

max-file-size: 10MB

max-request-size: 10MB

location: uploads

# File download properties

file:

download-dir: downloads

//implements file and download

import org.springframework.core.io.Resource;

import org.springframework.core.io.UrlResource;

import org.springframework.http.HttpHeaders;

import org.springframework.http.MediaType;

import org.springframework.http.ResponseEntity;

import org.springframework.util.StringUtils;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;

import java.net.MalformedURLException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.StandardCopyOption;

@RestController

public class FileController {

private final Path fileStorageLocation;

public FileController() {

this.fileStorageLocation = Path.of("uploads").toAbsolutePath().normalize();

}

@PostMapping("/upload")

public String uploadFile(@RequestParam("file") MultipartFile file) throws IOException {

String fileName = StringUtils.cleanPath(file.getOriginalFilename());

Path targetLocation = fileStorageLocation.resolve(fileName);

Files.copy(file.getInputStream(), targetLocation, StandardCopyOption.REPLACE\_EXISTING);

return "File uploaded successfully";

}

@GetMapping("/download/{fileName:.+}")

public ResponseEntity<Resource> downloadFile(@PathVariable String fileName) throws MalformedURLException {

Path filePath = fileStorageLocation.resolve(fileName).normalize();

Resource resource = new UrlResource(filePath.toUri());

if (resource.exists()) {

return ResponseEntity.ok()

.contentType(MediaType.APPLICATION\_OCTET\_STREAM)

.header(HttpHeaders.CONTENT\_DISPOSITION, "attachment; filename=\"" + resource.getFilename() + "\"")

.body(resource);

} else {

return ResponseEntity.notFound().build();

}

}

}

// run the application @springboot

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MyApplication {

public static void main(String[] args) {

SpringApplication.run(MyApplication.class, args);

}

}

4. Create a project to enable HTTPS and display in browser.

//app properties

# HTTPS configuration

server.ssl.key-store=classpath:ssl/your\_certificate\_file.crt

server.ssl.key-store-password=your\_password

server.ssl.key-store-type=PKCS12

server.ssl.key-alias=your\_certificate\_alias

//application.yml

# HTTPS configuration

server:

ssl:

key-store: classpath:ssl/your\_certificate\_file.crt

key-store-password: your\_password

key-store-type: PKCS12

key-alias: your\_certificate\_alias

//controller.java

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ResponseBody;

@Controller

public class HomeController {

@GetMapping("/")

@ResponseBody

public String home() {

return "Hello, HTTPS World!";

}

}