LẬP TRÌNH JAVA WEB SPRING MVC

Nguyễn Thái Sơn

FORM VALIDATE

SPRING FORM VALIDATION API

Message Properties

- Khai báo Bean để đọc file properties
- File này chứa các key/value sử dụng trong chương

trình của chúng ta

```
@Bean
public MessageSource messageSource() {
    ReloadableResourceBundleMessageSource
    bundleMessageSource = new
    ReloadableResourceBundleMessageSource();
    bundleMessageSource.setBasename("classpath:messages");
    bundleMessageSource.setDefaultEncoding("utf-8");
    return bundleMessageSource;
}
```

Dependency

- Sử dụng thư viện validation có sẵn

```
<!-- https://mvnrepository.com/artifact/javax.validation/validation-api -->
<dependency>
    <groupId>javax.validation</groupId>
    <artifactId>validation-api</artifactId>
    <version>2.0.1.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>6.1.5.Final</version>
</dependency>
```

Annotation

- Sử dụng thư viện validation có sẵn

```
@NotEmpty
private String name;

<p>Họ tên: </p><form:input path="name"/>
<p style="color:red"><form:errors path="name"></form:errors></p>

@RequestMapping(value = "/add-user", method = RequestMethod.POST)
public String addUser(HttpServletRequest request, @ModelAttribute("user")
@Valid User user, BindingResult bindingResult) {
    if (bindingResult.hasErrors()) {
        List<String> listFavourites = new ArrayList<String>();
        listFavourites.add("Xem phim");
        ...
        request.setAttribute("listFavourites", listFavourites);
        return "addUser";
    }
    request.setAttribute("user", user);
    return "viewUser";
}
```

FORM VALIDATOR

SPRING FORM VALIDATION VỚI VALIDATOR

Validator

- Tùy biến validate dữ liệu của form thông qua implement interface Validator
- Trong này chúng ta có thể validate theo cách chúng ta muốn cho từng thuộc tính của đối tượng

```
public class UserValidator implements Validator{
    @Override
    public boolean supports(Class<?> clazz) {
        // TODO Auto-generated method stub
        return false;
    }
    @Override
    public void validate(Object target, Errors errors) {
        // TODO Auto-generated method stub
    }
}
```

UPLOAD FILE

Upload File

Dependency

- Sử dụng thư viện common-fileupload để hộ trợ trong spring mvc cho việc upload file

```
<!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.4</version>
</dependency>
```

Tạo Bean

- Tạo Bean MutilpartResolver cho spring sử dụng trong việc upload file

```
@Bean(name = "multipartResolver")
public MultipartResolver multipartResolver() {
    CommonsMultipartResolver commonsMultipartResolver
    = new CommonsMultipartResolver();
    commonsMultipartResolver.setMaxUploadSize(-1);
    return commonsMultipartResolver;
}
```

Sử dụng

- Form client phải là form multipart/form-data
  - enctype=*"multipart/form-data"*
- Trong controller sử dụng đối tượng MultipartFile để map dữ liệu file từ client gửi lên

```
@RequestMapping(value = "upload", method = RequestMethod.POST)
public String upload(HttpServletRequest request, @RequestParam(name = "file")
List<MultipartFile> files) {
        for (MultipartFile file : files) {
        try {
                File saveFile = new File("E://"+file.getOriginalFilename());
                FileOutputStream fileOutputStream = new FileOutputStream(saveFile);
                fileOutputStream.write(file.getBytes());
                fileOutputStream.close();
        } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace();
        }
        }
        request.setAttribute("files", files);
        return "viewFile";
}
```

DOWNLOAD FILE

Download File

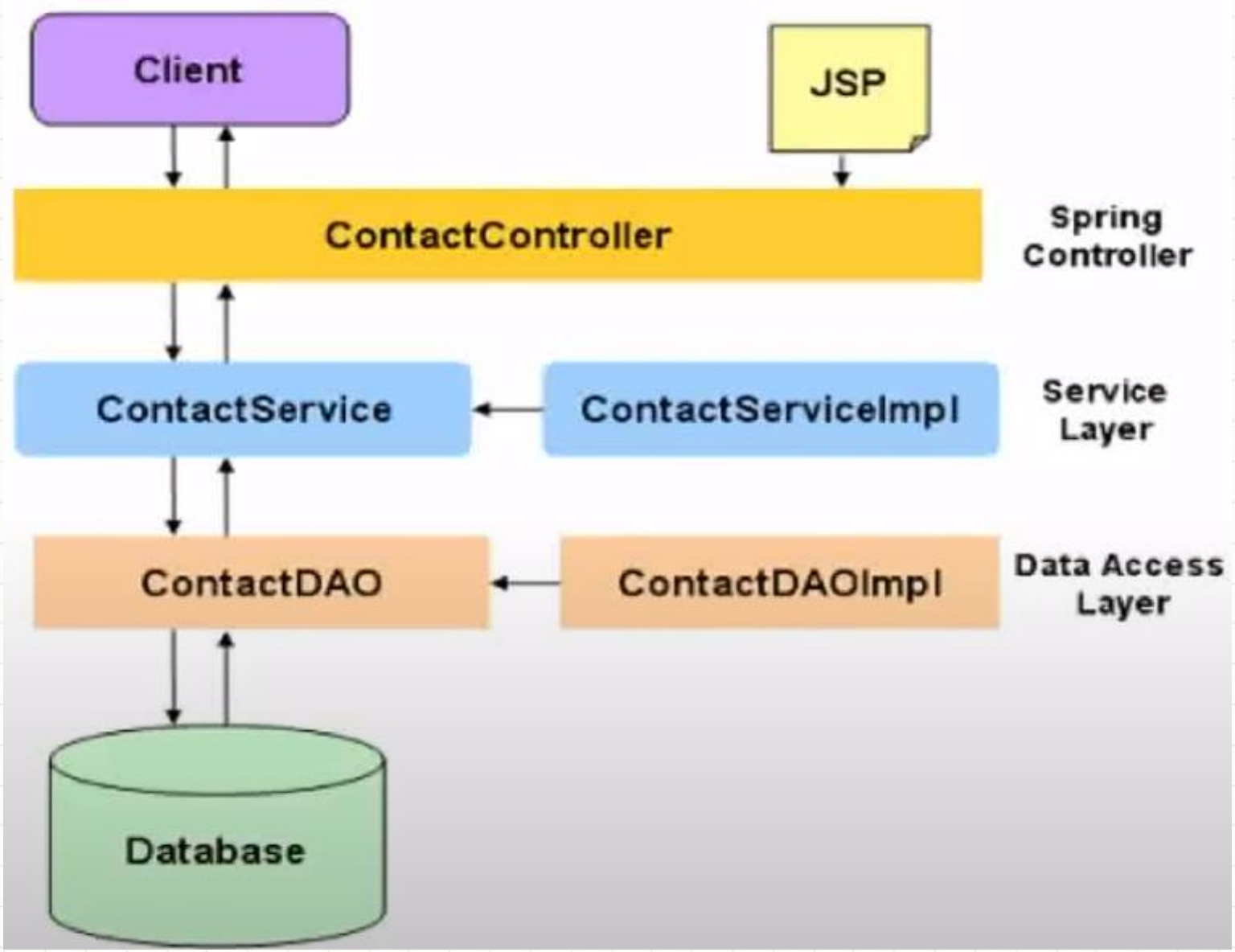
Sử dụng

```
@RequestMapping(value = "/download-file", method = RequestMethod.GET)
public void download(HttpServletRequest request, HttpServletResponse response)
{
    String dataDirectory = "E://";
    Path file = Paths.get(dataDirectory, "sinh nhat.jpg");
    if (Files.exists(file)) {
        response.setContentType("image/jpg");
        response.addHeader("Content-Disposition", "attachment;
        filename=anh.jpg");
        try {
            Files.copy(file, response.getOutputStream());
            response.getOutputStream().flush();
        } catch (Exception e) {
        }
    }
}
```

CLEAN STRUCTURE

CẤU TRÚC PHÂN CHIA PROJECT

Clean Structure



Spring Annotation

- @Controller: Tạo Bean và chỉ ra đây là lớp controller
- @Service: Tạo Bean và chỉ ra chức năng lớp service
- @Repository: Tạo Bean và chỉ ra chức năng lớp Dao
- @Component: Tạo Bean giành cho các bean khác

UTF-8 Encode

- Set maven để compile java file trong encode utf-8

  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

- Add filter để hộ trợ utf-8 trong project

  CharacterEncodingFilter characterEncodingFilter = **new CharacterEncodingFilter("UTF-8", true);**

  servletContext.addFilter("characterEncodingFilter", characterEncodingFilter).addMappingForUrlPatterns(**null, false, "/*");**

QUESTIONS?