



# DEPENDANCE INJECTION EMAIL & UPLOAD

ThS. Nguyễn Nghiê**̣**m  
0913.745.789 - NghiemN@fpt.edu.vn



- Dependence Injection
  - ✱ Định nghĩa và khai báo bean
  - ✱ Tiêm bean với @Autowire
  - ✱ Phân biệt bean với @Qualifier
- Sử dụng bean CommonsMultipartResolver để upload file
- Sử dụng bean JavaMailSender để gửi email
  - ✱ Gửi email có kèm file
  - ✱ Xây dựng bean gửi mail





# DEPENDANCE INJECTION

- DI là cách truyền đối tượng vào một thành phần khác để sử dụng
- Xét bean User gồm 2 thuộc tính id và password

## Định nghĩa bean

```
public class User {  
    private String id;  
    private String password;  
  
    getter/setter  
}
```

## Khai báo bean

```
<bean id="user" class="nhata nghe.model.User">  
    <property name="id" value="nghiemn"/>  
    <property name="password" value="iloveyou"/>  
</bean>
```

setPassword()



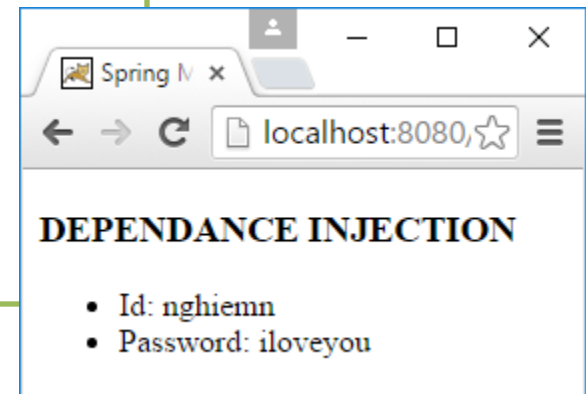
# TIÊM BEAN VÀO CONTROLLER

- Sử dụng **@Autowired** để tiêm bean vào Controller.

```
@Controller
public class DIController {
    @Autowired
    User user;

    @RequestMapping("user-info")
    public String showUserInfo(ModelMap model) {
        model.addAttribute("user", user);
        return "user-info";
    }
}
```

```
<ul>
    <li>Id: ${user.id}</li>
    <li>Password: ${user.password}</li>
</ul>
```





# DEPENDANCE INJECTION

- Sau khi controller được tạo ra thì bean sẽ được tiêm vào
- Có 3 cách tiêm bean vào controller
  - ✱ Tiêm trực tiếp vào field
  - ✱ Tiêm thông qua constructor
  - ✱ Tiêm qua phương thức setter

```
@Controller
public class DIController {

    @Autowired
    User user;

    ...
}
```

Tiêm vào field

```
@Controller
public class DIController {
    User user;

    @Autowired
    public DIController(User user){
        this.user = user;
    }

    ...
}
```

Tiêm vào constructor

```
@Controller
public class DIController {
    User user;

    @Autowired
    public void setUser(User user){
        this.user = user;
    }

    ...
}
```

Tiêm vào setter



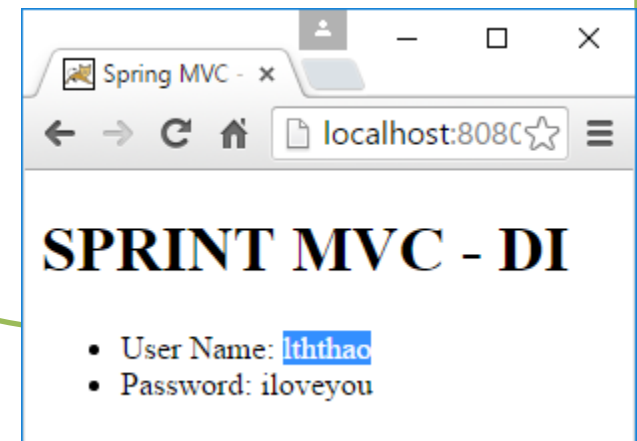
# DEPENDANCE INJECTION

- **@Autowired** sẽ nhận biết bean thông qua kiểu dữ liệu.
- Nếu có 2 bean cùng kiểu thì cần thêm **@Qualifier** để nhận biết qua id

```
<bean class="nhatnghe.model.User" id="nghiemn">
  <property name="id" value="nghiemn"/>
  <property name="password" value="iloveyou"/>
</bean>

<bean class="nhatnghe.model.User" id="lththao">
  <property name="id" value="lththao"/>
  <property name="password" value="iloveyou"/>
</bean>
```

```
@Controller
public class DIController {
    @Autowired @Qualifier("lththao")
    User user;
```





# @COMPONENT – BEAN TỰ KHAI BÁO

```
package nhatnghe.model;
```

```
import org.springframework.stereotype.Component;
```

```
@Component("user")
```

```
public class User {  
    private String id;  
    private String password;
```

```
    getters/setters
```

```
}
```

**@Component** cho phép một bean tự khai báo trong spring-config-mvc.xml. Tuy nhiên bạn phải khai báo package trong thuộc tính base-package

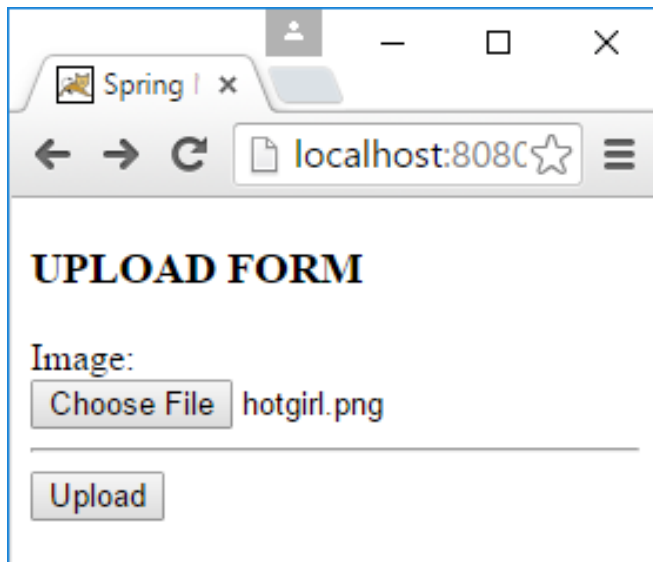
```
<!-- cấu hình controller -->
```

```
<context:component-scan
```

```
    base-package="nhatnghe.controller,nhatnghe.model.User"/>
```



# UPLOAD FILE CASE STUDY



A screenshot of a web browser window showing an "UPLOAD FORM". The browser's address bar displays "localhost:8080". The form contains a label "Image:" followed by a "Choose File" button and the text "hotgirl.png". Below this is an "Upload" button.



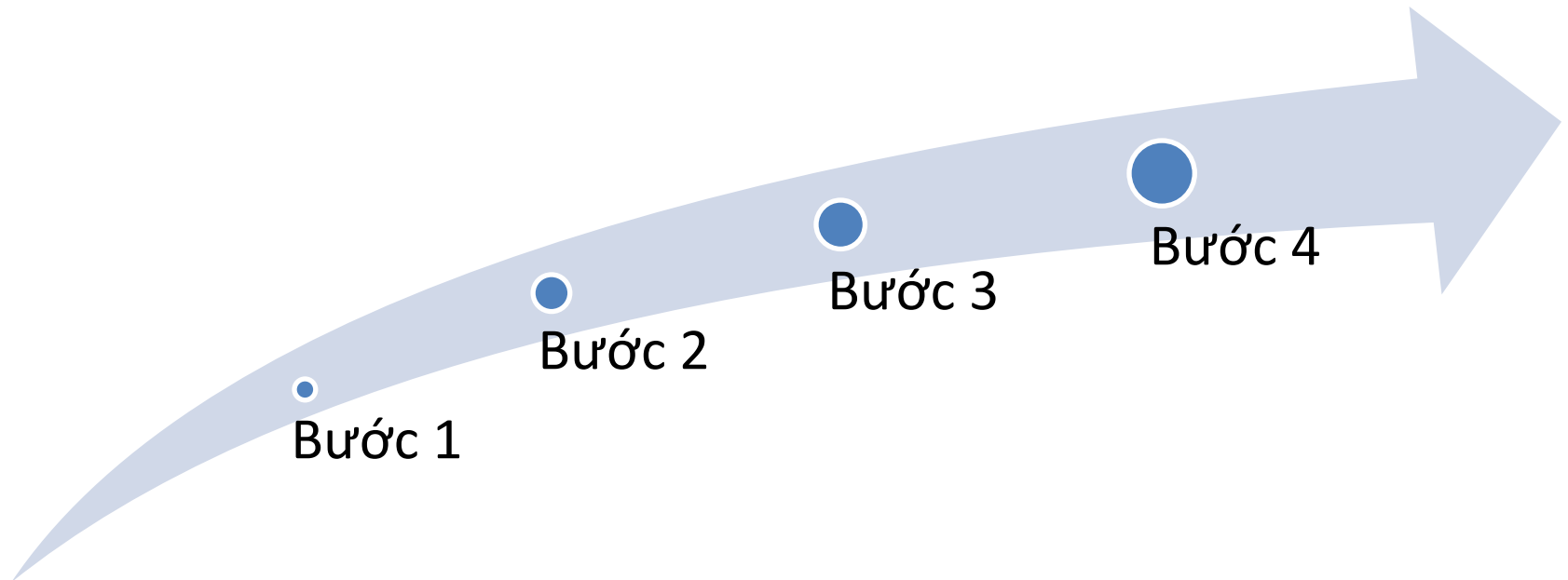
A screenshot of a web browser window showing the "UPLOAD SUCCESS" page. The browser's address bar displays "localhost:8080". The page features a cartoon illustration of a woman with blonde hair, wearing a blue top and black skirt, holding a red apple and a green book. Below the illustration, a list of file details is displayed:

- File Name: hotgirl.png
- File Size: 20776
- File Type: image/png





# CÁC BƯỚC THỰC HIỆN



- B1: Khai báo bean `CommonsMultipartResolver`
- B2: Xây dựng form upload
- B3: Xây dựng `UploadController`
- B4: Xây dựng view hiển thị kết quả



# KHAI BÁO BEAN

---

- Chép file spring-config-upload.xml vào WEB-INF
- File này chứa khai báo bean cho phép upload file như sau

```
<bean id="multipartResolver"  
      class="org.springframework.web.multipart.commons.CommonsMultipartResolver">  
    <!-- maxUploadSize=20MB -->  
    <property name="maxUploadSize" value="20485760"/>  
</bean>
```

- Thuộc tính **maxUploadSize** qui định kích thước tối đã được phép upload



# FORM UPLOAD FILE

```
<form action="upload.htm"  
  method="post" enctype="multipart/form-data">  
  <div>Image:</div>  
  <input type="file" name="image">  
  
  <hr>  
  <button>Upload</button>  
</form>
```

upload.jsp

Spring | x

localhost:8080

## UPLOAD FORM

Image:

Choose File hotgirl.png

Upload



# XÂY DỰNG UPLOADCONTROLLER

```
@Controller
@RequestMapping("upload")
public class UploadController {
    @Autowired
    ServletContext application;
    @RequestMapping(method=RequestMethod.GET)
    public String upload() {
        return "upload";
    }
    @RequestMapping(method=RequestMethod.POST)
    public String upload(ModelMap model,
        @RequestParam("image") MultipartFile image) {
        String filename = image.getOriginalFilename();
        String path = application.getRealPath("/images/" + filename);
        try {
            image.transferTo(new File(path));

            model.addAttribute("name", filename);
            model.addAttribute("size", image.getSize());
            model.addAttribute("type", image.getContentType());
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return "upload-success";
    }
}
```

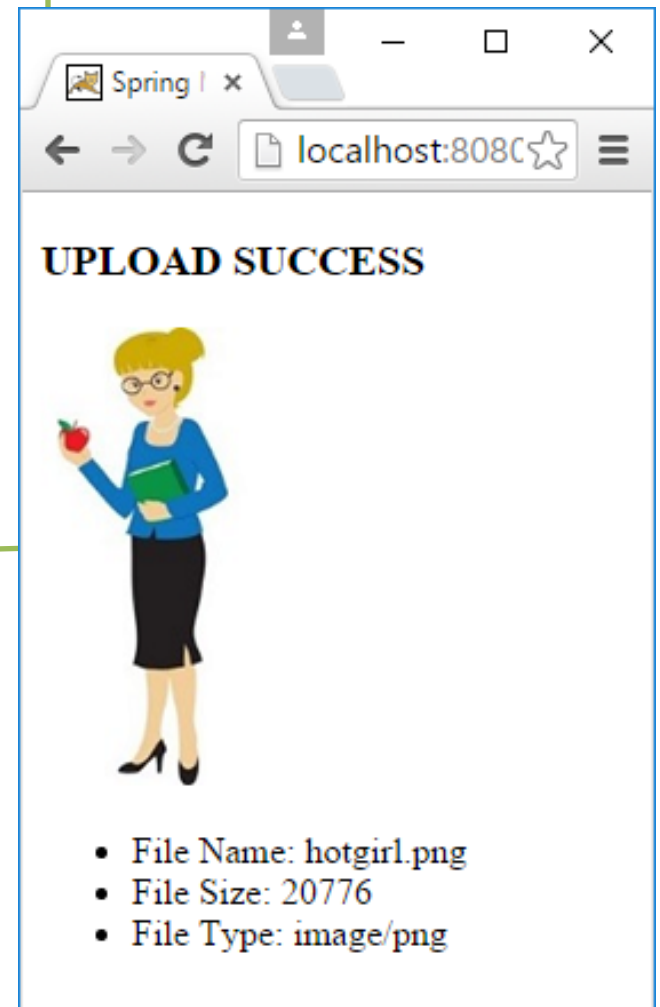


# XÂY DỰNG FORM UPLOAD VÀ HIỂN THỊ

```

<ul>
  <li>File Name: ${name}</li>
  <li>File Size: ${size}</li>
  <li>File Type: ${type}</li>
</ul>
```

upload-success.jsp





# SEND E-MAIL CASE STUDY

Spring MVC x

localhost:8080

## SEND MAIL

Gửi mail thành công !

From:  
songlong2k@gmail.com

To:  
nghiemn@fpt.edu.vn

Subject:  
Hello

Body:  
Ilove you

Send

Spring MVC - Send Mail x Inbox - nghiemn@fpt.edu x

https://mail.google.com/mail/u/0/#inbox

STARS TRƯỜNG ĐẠI HỌC FPT

Nguyễn

Mail 1-50 of 4,994

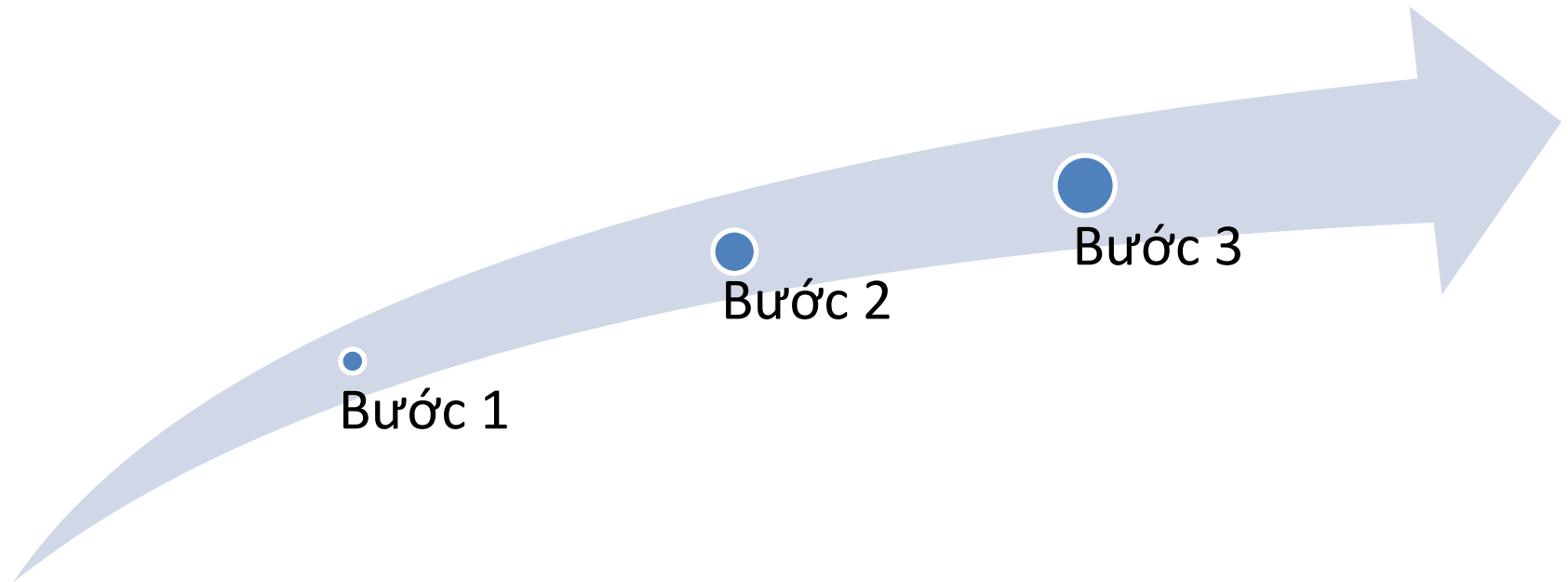
COMPOSE

Inbox

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	songlong2k@gmail.com	Hello - Ilove you
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Twoo	★ Trương Thị Mỹ Dung ở T
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Lê Vũ Hà Giảng viên	Cafe - Sáng mai Cafe nê



# CÁC BƯỚC THỰC HIỆN



- B1: Khai báo bean: JavaMailSenderImpl
- B2: Xây dựng bean và form send-mail
- B3: Xây dựng SendMailController



# KHAI BÁO MAIL BEAN

- Chép file spring-config-gmail.xml vào WEB-INF
- Chú ý khai báo bean gửi mail qua gmail

```
<bean id="mailSender"
      class="org.springframework.mail.javamail.JavaMailSenderImpl">
  <property name="host" value="smtp.gmail.com" />
  <property name="port" value="465" />
  <property name="username" value="kentphp2@gmail.com" />
  <property name="password" value="songlong" />

  <property name="defaultEncoding" value="UTF-8"/>
  <property name="javaMailProperties">
    <props>
      <prop key="mail.smtp.auth">true</prop>
      <prop key="mail.smtp.socketFactory.class">
        javax.net.ssl.SSLSocketFactory</prop>
      <prop key="mail.smtp.socketFactory.port">465</prop>
      <prop key="mail.debug">true</prop>
      <prop key="mail.smtp.starttls.enable">>false</prop>
    </props>
  </property>
</bean>
```

Tài khoản gmail cần được  
kích hoạt cho phép gửi  
mail qua ứng dụng

<https://www.google.com/settings/security/lesssecureapps>





# FORM VÀ MAILINFO

```
${message}
<form:form action="send-mail.htm" modelAttribute="mail">
    <div>From:</div>
    <form:input path="from"/>

    <div>To:</div>
    <form:input path="to"/>

    <div>Subject:</div>
    <form:input path="subject"/>

    <div>Body:</div>
    <form:textarea path="body" rows="3"/>

    <hr>
    <button>Send</button>
</form:form>
```

send-mail.jsp

```
public class MailInfo {
    String from;
    String to;
    String subject;
    String body;

    getters/setters
}
```

Spring M x

localhost:8080

### SEND MAIL

Gửi mail thành công !

From:

To:

Subject:

Body:



# SENDMAILCONTROLLER

```
@Controller
@RequestMapping("send-mail")
public class SendMailController {
    @Autowired
    JavaMailSender mailSender;
    @RequestMapping(method=RequestMethod.GET)
    public String send(ModelMap model) {
        model.addAttribute("mail", new MailInfo());
        return "send-mail";
    }
    @RequestMapping(method=RequestMethod.POST)
    public String send(ModelMap model, @ModelAttribute("mail") MailInfo mail) {
        try {
            MimeMessage message = mailSender.createMimeMessage();
            MimeMessageHelper helper = new MimeMessageHelper(message, true, "UTF-8");
            helper.setFrom(mail.getFrom(), mail.getFrom());
            helper.setReplyTo(mail.getFrom(), mail.getFrom());
            helper.setTo(mail.getTo());
            helper.setSubject(mail.getSubject());
            helper.setText(mail.getBody(), true);

            mailSender.send(message);
            model.addAttribute("message", "Gửi mail thành công !");
        }
        catch (Exception e) {
            model.addAttribute("message", "Gửi mail thất bại !");
            e.printStackTrace();
        }
        return "send-mail";
    }
}
```

Tiêm JavaMailSender vào để sử dụng



# SEND EMAIL CÓ ATTACH FILE

```
MimeMessage mail = mailSender.createMimeMessage();  
MimeMessageHelper helper = new MimeMessageHelper(mail, true, "utf-8");  
helper.setFrom(from);  
helper.setReplyTo(from);  
helper.setTo(to);  
helper.setSubject(subject);  
helper.setText(body, true);
```

```
File file = new File(path);  
helper.addAttachment(file.getName(), file);
```

```
mailSender.send(mail);
```

File được đính kèm

Tên file đính kèm được hiển thị trên email



# BEAN XMAILSENDER

@Component

```
public class XMailSender {
    @Autowired
    JavaMailSender mailSender;
    /**
     * Gửi email cho một người từ hệ thống
     */
    public void send(String to, String subject, String body) {
        String from = "Song Long <songlong2k@gmail.com>";
        this.send(from, to, subject, body);
    }
    /**
     * Gửi email cho một người
     */
    public void send(String from, String to, String subject, String body) {
        this.send(from, to, "", "", subject, body, "");
    }
    /**
     * Gửi email cho nhiều người có đính kèm file
     */
    public void send(String from, String to, String cc, String bcc, String subject, String body, String attachments) {
        ...
    }
}
```



- Dependence Injection
  - ✱ Định nghĩa và khai báo bean
  - ✱ Tiêm bean với @Autowire
  - ✱ Phân biệt bean với @Qualifier
- Sử dụng bean CommonsMultipartResolver để upload file
- Sử dụng bean JavaMailSender để gửi email
  - ✱ Gửi email có kèm file
  - ✱ Xây dựng bean gửi mail

