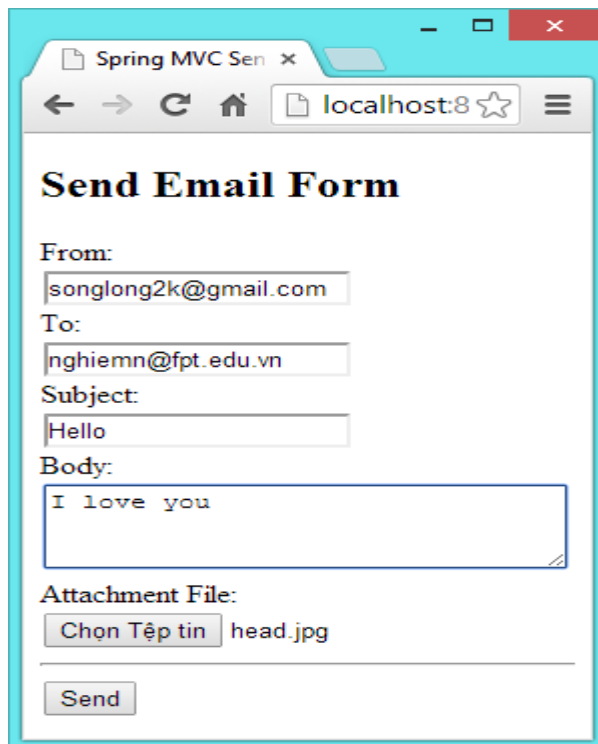


## MỤC TIÊU

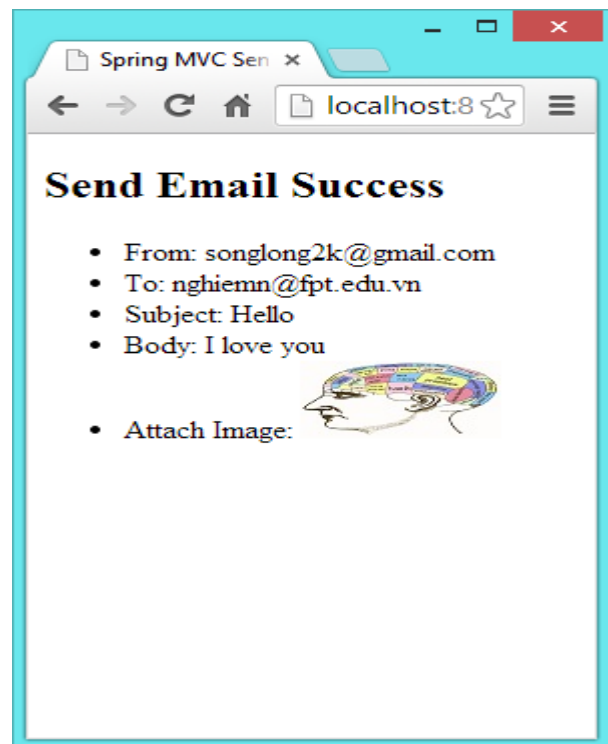
Nắm vững 2 kỹ thuật lập trình web cơ bản là

- ✓ Upload file lên server
- ✓ Gửi email thông qua tài khoản gmail có đính kèm file

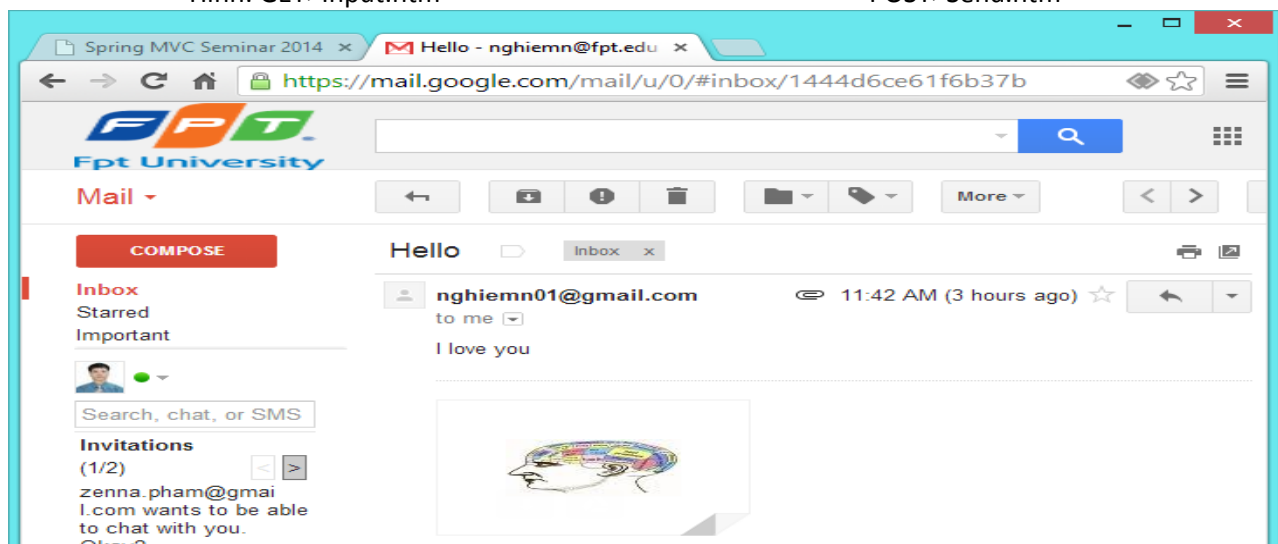
## MÔ TẢ



Hình: GET>Input.htm

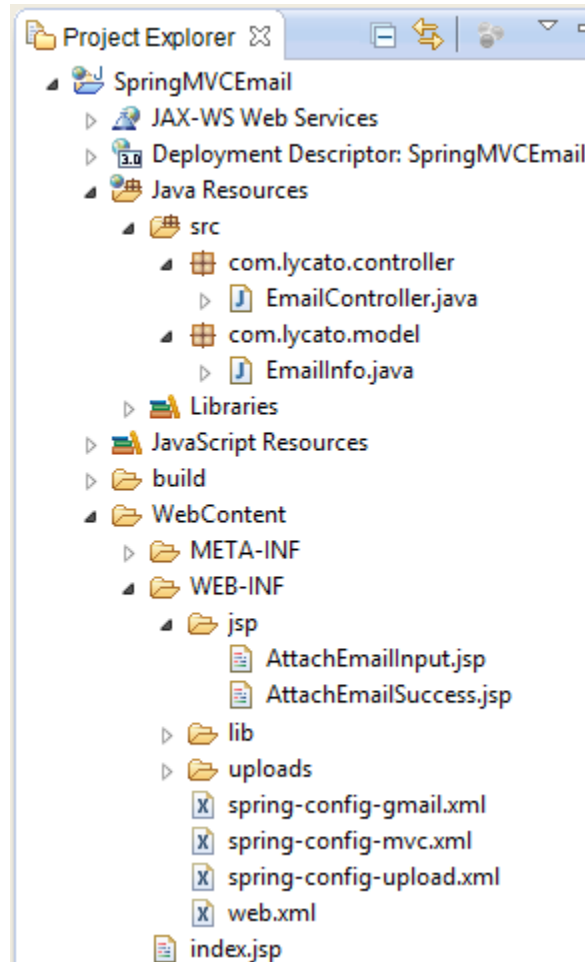


POST>Send.htm



## THỰC HIỆN

Xây dựng một project có cấu trúc các thành phần như sau



- ✓ Bước 1: Thư viện
- ✓ Bước 2: Các file cấu hình
- ✓ Bước 3: Form gửi email AttachEmailInput.jsp
- ✓ Bước 4: EmailController.java và EmailInfo.java
- ✓ Bước 5: Hiển thị kết quả AttachEmailSuccess.jsp
- ✓ Bước 6: Chạy

### Bước 1: Thư viện

Ngoài các thư viện cần thiết để Spring MVC hoạt động, bạn cần các thư viện sau cho việc gửi email và upload file.

- ✓ Gửi email:
  - activation.jar

- mail.jar
- commons-email-1.0.jar
- ✓ Upload file
  - commons-fileupload-1.2.2.jar
  - commons-io-1.3.2.jar

## Bước 2: Các file cấu hình

### ❖ Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
    <display-name>SpringMVCEmail</display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>
            org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/spring-config-*.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>dispatcher</servlet-name>
        <url-pattern>*.htm</url-pattern>
    </servlet-mapping>
</web-app>
```

Cấu hình để Spring MVC nạp nhiều file cấu hình: spring-config-\*.xml. Dấu \* sẽ đại diện cho nhóm ký tự bất kỳ. Cụ thể ở bài này là mvc, gmail và upload

### ❖ spring-config-mvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

    <!-- Declare a view resolver -->
```

```
<bean id="viewResolver" p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"
class="org.springframework.web.servlet.view.InternalResourceViewResolver"/>

<!-- Spring MVC Annotation -->
<mvc:annotation-driven />
<context:annotation-config />

<!-- Where to find component -->
<context:component-scan base-package="com.lycato" />

</beans>
```

- ✓ Khai báo bean InternalResourceViewResolver để xử lý view
- ✓ Chỉ rõ package tìm kiếm các component là com.lycato
- ✓ Chỉ rõ ứng dụng Spring này được phép sử dụng annotation

### ❖ spring-config-gmail.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.2.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

<!-- Spring Mail Sender -->
<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="user@gmail.com" />
<property name="password" value="*****" />
<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">true</prop>
</props>
</property>
</bean>
</beans>
```

Để gửi email thông qua tài khoản gmail trong Spring bạn cần cấu hình bean JavaMailSenderImpl với các thông số cần thiết như trên.

### ❖ spring-config-upload.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.2.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

  <bean id="multipartResolver"

    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
      <!-- maxUploadSize=10MB -->
      <property name="maxUploadSize" value="10485760"/>
    </bean>

</beans>
```

Chú ý:

- ✓ Để upload file được bạn phải khai báo bean CommonsMultipartResolver
- ✓ Để hạn chế kích thước file bạn cần chỉ rõ thuộc tính maxUploadSize

### Bước 3: Form gửi email AttachEmailInput.jsp

```
<%@page pageEncoding="utf-8" contentType="text/html; charset=utf-8" %>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form" %>

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Spring MVC Seminar 2014</title>
</head>
<body>
  <h2>Send Email Form</h2>
  ${error}
  <form:form action="send.htm"
    method="post" enctype="multipart/form-data" 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" cols="30"/>
    <div>Attachment File:</div>
    <input name="attachment" type="file">
    <hr>
    <input type="submit" value="Send">
  </form:form>
</body>
</html>
```

Form upload file phải thỏa mãn:

- ✓ Có trường <input type="file">
- ✓ Form phải thiết lập các thuộc tính
  - Method="POST"
  - enctype="multipart/form-data"

## Bước 4: EmailController.java và EmailInfo.java

### EmailController.java

```
package com.lycato.controller;

import java.io.File;

import javax.mail.internet.MimeMessage;
import javax.servlet.ServletContext;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;

import com.lycato.model.EmailInfo;

@Controller
public class EmailController {

    @Autowired
    ServletContext context;

    @Autowired
    JavaMailSender mailSender;

    @RequestMapping(value="input", method=RequestMethod.GET)
    public String showForm(ModelMap model) {
        model.addAttribute("mail", new EmailInfo());
        return "AttachEmailInput";
    }

    @RequestMapping(value="send", method=RequestMethod.POST)
    public String sendWithAttach(ModelMap model,
        @ModelAttribute("mail") EmailInfo mailInfo,
        @RequestParam("attachment") MultipartFile file) {
        try{
            MimeMessage message = mailSender.createMimeMessage();

            MimeMessageHelper helper = new MimeMessageHelper(message, true);
            helper.setFrom(mailInfo.getFrom());
            helper.setTo(mailInfo.getTo());
            helper.setReplyTo(mailInfo.getFrom());
            helper.setSubject(mailInfo.getSubject());
            helper.setText(mailInfo.getBody(), true);
```

```
        if(!file.isEmpty()){
            String imageUrl = "uploads/" + file.getOriginalFilename();
            String absolutePath = context.getRealPath(imageUrl);
            File uploadFile = new File(absolutePath);
            file.transferTo(uploadFile);

            helper.addAttachment(uploadFile.getName(), uploadFile);

            model.addAttribute("imageUrl", imageUrl);
        }

        mailSender.send(message);
    }
    catch (Exception ex){
        model.addAttribute("error", ex.getMessage());
        return "AttachEmailInput";
    }
    return "AttachEmailSuccess";
}
}
```

- ✓ Action send.htm sẽ nhận thông tin email vào đối tượng EmailInfo và file upload có tên attachment dưới dạng MultipartFile để xử lý file upload.
- ✓ JavaMailSender (được khai báo trong file cấu hình spring-config-gmail) được tiêm vào để gửi email. Phương thức createMimeMessage() và send() là các phương thức được sử dụng để tạo ra MimeMessage và gửi đi.
- ✓ MimeMessageHelper cung cấp các phương thức để thiết lập các thông tin mail như setFrom(), setSubject() hay addAttachment().

### EmailInfo.java

```
package com.lycato.model;

public class EmailInfo {
    private String from, to, subject, body;

    public String getFrom() {
        return from;
    }
    public void setFrom(String from) {
        this.from = from;
    }

    public String getTo() {
        return to;
    }
    public void setTo(String to) {
        this.to = to;
    }

    public String getSubject() {
        return subject;
    }
    public void setSubject(String subject) {
        this.subject = subject;
    }

    public String getBody() {
        return body;
    }
    public void setBody(String body) {
```

```
}  
    this.body = body;  
}
```

### Bước 5: Hiển thị kết quả AttachEmailSuccess.jsp

```
<%@page pageEncoding="utf-8" contentType="text/html; charset=utf-8" %>  
  
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>Spring MVC Seminar 2014</title>  
</head>  
<body>  
    <h2>Send Email Success</h2>  
    <ul>  
        <li>From: ${mail.from}</li>  
        <li>To: ${mail.to}</li>  
        <li>Subject: ${mail.subject}</li>  
        <li>Body: ${mail.body}</li>  
        <li>Attach Image:  
              
        </li>  
    </ul>  
</body>  
</html>
```

Trong model có chứa 2 thuộc tính là mail và imageUrl do EmailController thiết lập trong action send.html với các dòng mã lệnh sau:

- ✓ @ModelAttribute("mail")
- ✓ model.addAttribute("imageUrl", imageUrl);

Trong đó mail là đối tượng của EmailInfo gồm các thuộc tính from, to, subject, body. Biểu thức EL \${mail.from} sẽ truy xuất thuộc tính from của đối tượng mail còn \${imageUrl} là thuộc tính model.

### Bước 6: Chạy

<http://localhost:8080/SpringMVCEmail/input.htm>