

EXPERIMENT NO. 05

Title: Email Validation using Struts Framework

Outcome: Students will be able to implement email Validation using Struts Framework

Theory:

Step-by-step guide to implement **Email Validation using Struts 2** in **Eclipse** with **Tomcat** and a **Dynamic Web Project** — perfect for learning or a small demo app.

Step 1: Setup Environment

Requirements:

- Eclipse IDE (Enterprise Edition)
- Apache Tomcat (9.x or 10.x)
- Struts 2 libraries (latest stable version, e.g., struts-2.5.x)

You can download Struts 2 from:
 <https://struts.apache.org/download.cgi>

Step 2: Create a New Dynamic Web Project

1. Open **Eclipse** → **File** → **New** → **Dynamic Web Project**
 2. Project Name: **EmailValidationDemo**
 3. Target Runtime: Select **Apache Tomcat**
 4. Click **Finish**
-

Step 3: Add Struts 2 Libraries

1. Download and extract Struts 2 zip file.
2. Copy the following **JARs** from:
3. struts-2.x.x/lib/

Add these to your project's **WEB-INF/lib** folder:

- struts2-core-x.x.x.jar
- xwork-core-x.x.x.jar
- commons-fileupload-x.x.jar
- commons-io-x.x.jar
- commons-lang3-x.x.jar
- commons-logging-x.x.jar
- ognl-x.x.x.jar
- freemarker-x.x.x.jar
- javassist-x.x.jar
- log4j-api-x.x.jar

- o log4j-core-x.x.jar
 - o commons-validator-x.x.jar
 - o commons-beanutils-x.x.jar
-

Step 4: Create the Directory Structure

Your project should look like this:

EmailValidationDemo/

```
├ src/
|   └ com/example/action/
|       └ EmailAction.java
└ WebContent/
    └ WEB-INF/
        └ lib/
        └ struts.xml
        └ web.xml
    └ index.jsp
    └ success.jsp
```

Step 5: Create the Action Class

File: src/com/example/action/EmailAction.java

```
package com.example.action;
```

```
import com.opensymphony.xwork2.ActionSupport;
```

```
public class EmailAction extends ActionSupport {
    private String email;
```

```
    public String getEmail() {
        return email;
    }
```

```
    public void setEmail(String email) {
        this.email = email;
    }
```

```
    @Override
    public String execute() {
        return SUCCESS;
    }
```

```

@Override
public void validate() {
    if (email == null || email.trim().isEmpty()) {
        addFieldError("email", "Email is required!");
    } else if (!email.matches("^[A-Za-z0-9+_.-]+@[.]+")) {
        addFieldError("email", "Invalid email format!");
    }
}
}

```

Step 6: Create struts.xml

File: WebContent/WEB-INF/struts.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
"http://struts.apache.org/dtds/struts-2.5.dtd">
```

```

<struts>
    <package name="default" extends="struts-default">
        <action name="validateEmail" class="com.example.action.EmailAction">
            <result name="success">/success.jsp</result>
            <result name="input">/index.jsp</result>
        </action>
    </package>
</struts>
```

Step 7: Configure web.xml

File: WebContent/WEB-INF/web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">
```

```
<display-name>Email Validation Demo</display-name>
```

```

<filter>
    <filter-name>struts2</filter-name>
```

```

<filter-
class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>

```

Step 8: Create the JSP Pages

index.jsp

```

<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
  <title>Email Validation</title>
</head>
<body>
  <h2>Email Validation Form</h2>
  <s:form action="validateEmail" method="post">
    <s:textfield name="email" label="Enter Email" />
    <s:submit value="Validate" />
  </s:form>

  <s:fielderror fieldName="email" />
</body>
</html>

```

success.jsp

```

<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
  <title>Success</title>
</head>
<body>
  <h2>Email Validation Successful ✓ </h2>
  <p>Your email: <s:property value="email" /></p>
</body>
</html>

```

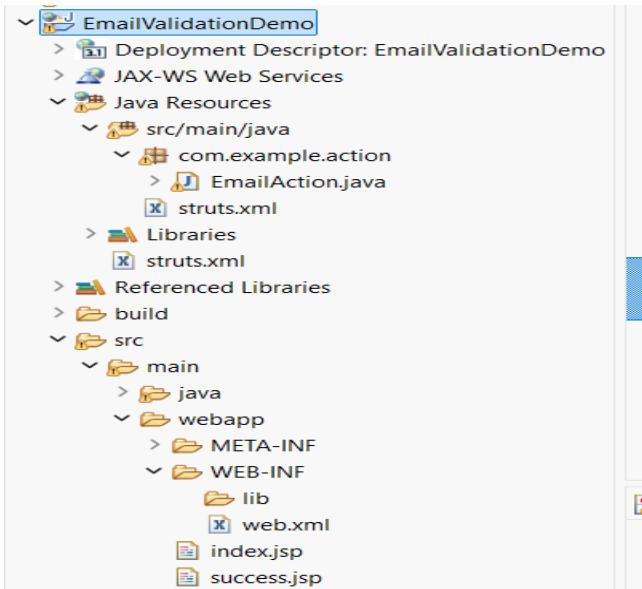
► Step 9: Run the Project

1. Right-click your project → **Run As** → **Run on Server**

2. Open your browser and go to:
👉 <http://localhost:8080/EmailValidationDemo/>
3. Try entering invalid and valid emails to test the validation.

✿ Output:

- If invalid email → Error message below the text field.
- If valid email → Redirects to success.jsp.



The screenshot shows the Eclipse IDE's Project Explorer view. The project 'EmailValidationDemo' is selected. The tree view displays the following structure:

- Deployment Descriptor: EmailValidationDemo
- JAX-WS Web Services
- Java Resources
 - src/main/java
 - com.example.action
 - EmailAction.java
 - struts.xml
 - Libraries
 - struts.xml
 - Referenced Libraries
 - build
 - src
 - main
 - java
 - webapp
 - META-INF
 - WEB-INF
 - lib
 - web.xml
 - index.jsp
 - success.jsp

localhost:8080/EmailValidationDemo/

Modernization and... GATE - 2021 :: Candi... Creating a U:

Email Validation Form

Enter Email:

localhost:8080/exp_05/validateEmail.action;jsessionid=B80B79FFD38A05D53DBE

Email Validation Form

Invalid email format!
Enter Email: devyanipmane@

- Invalid email format!
- Invalid email format!

localhost:8080/exp_05/validateEmail.action;jsessionid=B80B79FFD38A05D53DBEA7B400EB328D

Email Validation Successful 

Your email: devyanipmane@gmail.com

Conclusion : Thus, I have implemented email Validation using Struts Framework.