

f:validateRegex tag is used to validate a string value to a required format.

JSF Tag

```
<f:validateRegex pattern="((?=.*[a-z]).{6,})" />
```

Tag Attributes

S.N.	Attribute & Description
------	-------------------------

- | | |
|---|--------------------------------------|
| 1 | pattern
Formatting pattern |
|---|--------------------------------------|

Example Application

Let us create a test JSF application to test the above tag.

Step	Description
------	-------------

- | | |
|---|---|
| 1 | Create a project with a name <i>helloworld</i> under a package <i>com.tutorialspoint.test</i> as explained in the <i>JSF - First Application</i> chapter. |
| 2 | Modify <i>home.xhtml</i> as explained below. Keep rest of the files unchanged. |
| 3 | Create <i>result.xhtml</i> in the webapps directory as explained below. |
| 4 | Create <i>UserData.java</i> as a managed bean under package <i>com.tutorialspoint.test</i> as explained below. |
| 5 | Compile and run the application to make sure business logic is working as per the requirements. |
| 6 | Finally, build the application in the form of war file and deploy it in Apache Tomcat Webserver. |
| 7 | Launch your web application using appropriate URL as explained below in the last step. |

UserData.java

```
package com.tutorialspoint.test;

import java.io.Serializable;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

@ManagedBean(name = "userData", eager = true)
@SessionScoped
public class UserData implements Serializable {

    private static final long serialVersionUID = 1L;

    private String password;
    public String getPassword() {
```

```

        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}

```

home.xhtml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core">
    <h:head>
        <title>JSF tutorial</title>
    </h:head>
    <h:body>
        <h2>h:validateRegex Example</h2>
        <!-- password contains lower case letters only and.
        length of the password should be greater than 6. -->
        <h:form>
            <h:inputSecret
                label="password" >
                <f:validateRegex pattern="((?=.*[a-z]).{6,})" />
            </h:inputSecret>
            <h:commandButton value="submit" action="result"/>
            <h:message for="passwordInput" style="color:red" />
        </h:form>
    </h:body>
</html>

```

result.xhtml

```

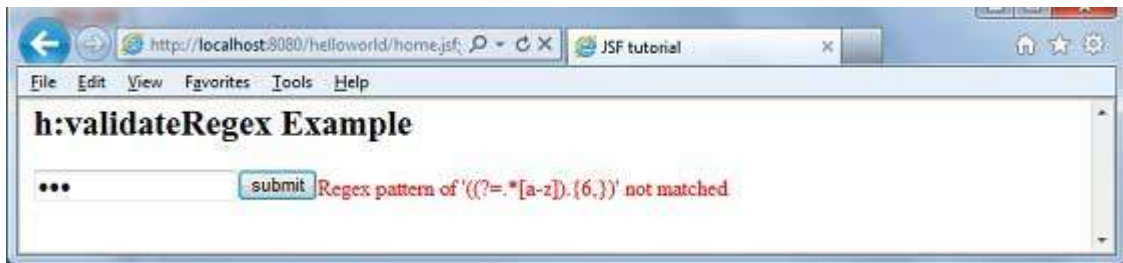
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html">
    <h:head>
        <title>JSF Tutorial!</title>
    </h:head>
    <h:body>
        <h2>Result</h2>
        <hr />
        Password: #{userData.password}
    </h:body>
</html>

```

Once you are ready with all the changes done, let us compile and run the application as we did in JSF - First Application chapter. If everything is fine with your application, this will produce following result:



Enter an invalid value



Enter a valid value

