



R\$ 263,45 R\$ 122,27 R\$ 499,90 R\$ 89,90 R\$ 178,74 R\$ 65,-

ANDROID CORE JAVA DESKTOP JAVA ENTERPRISE JAVA JAVA BASICS JVM LANGUAGES SOFTWARE DEVELOPMENT DEVOPS

[Home](#) » Enterprise Java » jsf » JSF Navigation Example

ABOUT VEERAMANI KALYANASUNDARAM



Veera is a Software Architect working in telecom domain with rich experience in Java Middleware Technologies. He is a OOAD practitioner and interested in Performance Engineering.



JSF Navigation Example

Posted by: Veeramani Kalyanasundaram in jsf February 19th, 2015

Google

Descubra o que o Google pode fazer pelo seu negócio

Juntos dá pra fazer mais

[CLIQUE E DESCUBRA](#)

In this example of JSF Navigation, we will show you how to use various navigation techniques available in Java Server Faces. A set of rules for choosing the next page based on an action, for example a click of button are called navigation rules. In Java Server Faces technology, navigation consists of one or more navigation rules for an application.

We will discuss about implicit navigation and user defined navigation with examples. Let's begin with setting up a JSF project and do all the necessary configuration to run the application.

Want to be a JSF Ninja?

Subscribe to our newsletter and download the JSF 2.0 Programming Cookbook [right now!](#)

In order to get you prepared for your JSF development needs, we have compiled numerous recipes to help you kick-start your projects. Besides reading them online you may download the eBook in PDF format!

Email address:

Your email address

[Sign up](#)

Our preferred environment is Eclipse. We are using Eclipse Luna SR1 with Maven Integration Plugin, JDK 8u25 (1.8.0_25) and Tomcat 8 application server. Having said that, we have tested the code against JDK 1.7 and Tomcat 7 as well.

Tip

You may skip project creation and jump directly to the [beginning of the example](#) below.

1. Create a new Maven Project

Go to File -> New->Other-> Maven Project

volkswagen#vale
Conheça todas as ofertas.

[COMPRE AGORA](#)

Golf.
Com teto solar pan

CONHEÇA

toptal [Find Top Developers](#)
The search is on to find developers who match you with

NEWSLETTER

169,851 insiders are already subscribed to our weekly updates and complimentary whitepapers!

Join them now to gain [access](#) to the latest news in Java as well as insights about Andr Groovy and other related technologies.

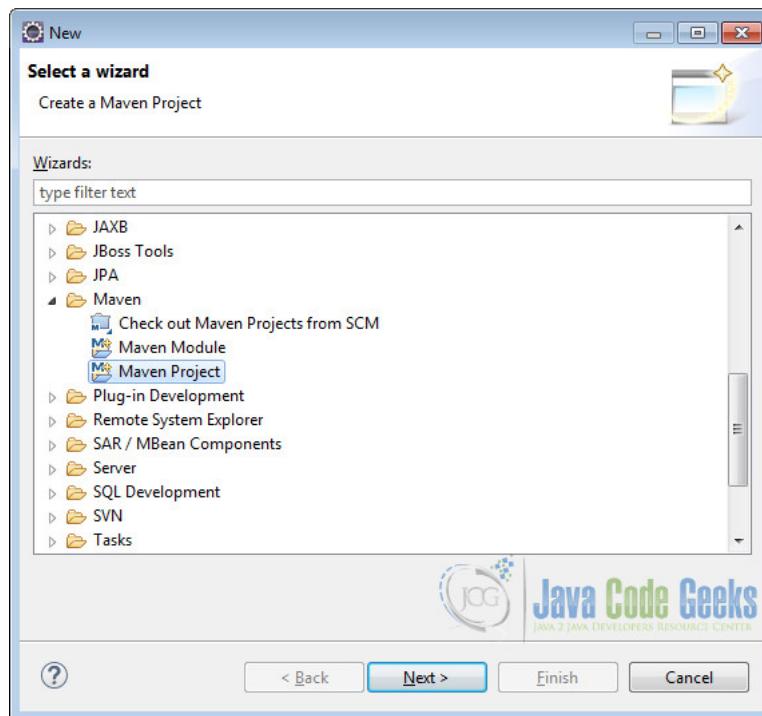
Email address:

Your email address

[Sign up](#)

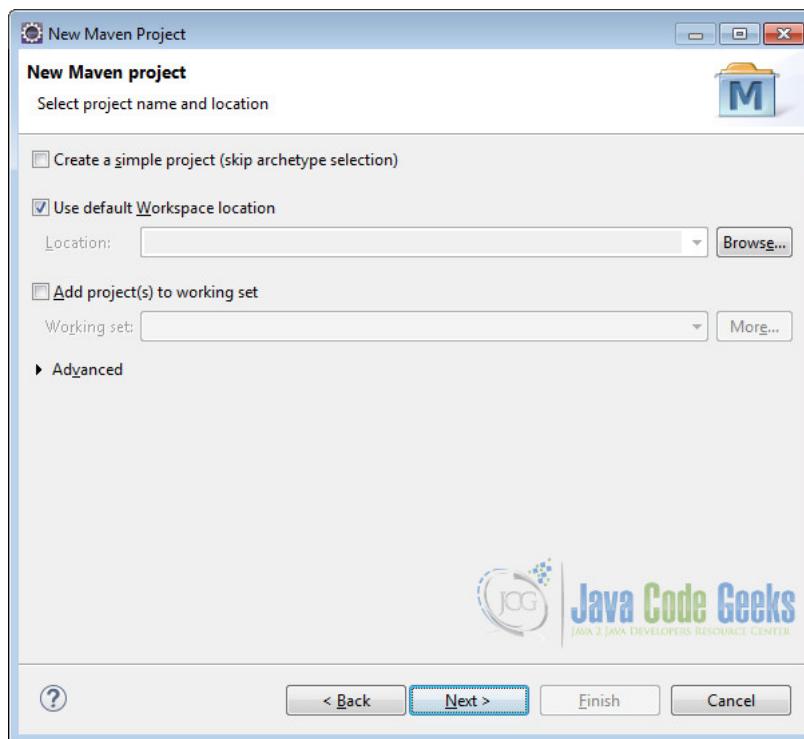
JOIN US

With **1,240,600** monthly unique **500** authors we are placed among the top 50 most popular related sites around. Constantly looking for partners; we encourage you to join us! So if you have a blog with unique content check out our program [guest writing](#) skills!



Maven setup – step 1

In the "Select project name and location" page of the wizard, make sure that "Create a simple project (skip archetype selection)" option is **unchecked**, hit "Next" to continue with default values.



Maven setup – step 2

Here choose "maven-archetype-webapp" and click on Next.



R\$ 75,21 R\$ 89,90

R\$ 11,93 R\$ 263,45

R\$ 254,90 R\$ 279,90

R\$ 65,79 R\$ 65,79

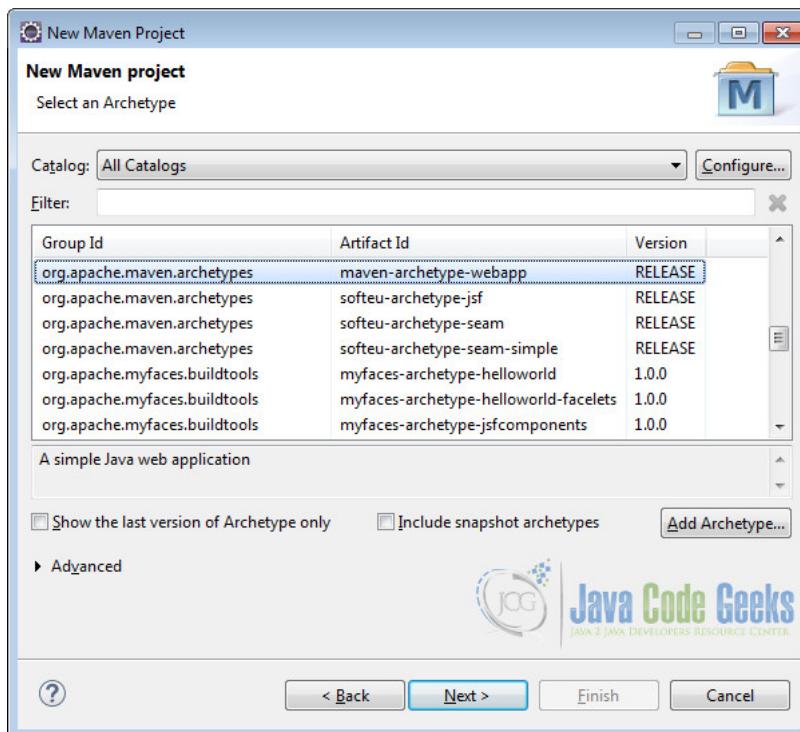
R\$ 195,42 R\$ 122,27

CAREER OPPORTUNITIES

- Jr. Software Developer **Techpor**
Richmond, VA
Oct, 13
- Jr. Software Developer **Techpor**
Duluth, GA
Oct, 05
- Java Developer **Transamerica**
Plano, TX
Oct, 05
- Sr. Java Developer - Struts/Hibernate
Trenton, NJ
Oct, 10
- Java Software Engineer - Entry Level
REQUIRED **LexisNexis**
King of Prussia, PA
Oct, 08
- Java Software Engineer - Entry Level
REQUIRED **Reed Elsevier**
King of Prussia, PA
Oct, 08
- Web Developer - Entry Level **Verizon**
Alpharetta, GA
Sep, 26
- Java Developer **Blue Cross Blue Shield NM, OK & TX**
Chicago, IL
Oct, 04
- Web Developer - Entry Level **Verizon**
Temple Terrace, FL
Sep, 26
- JAVA Developer **Advanced Technical Inc**
Greenwood Village, CO
Oct, 06

1 2 ... 5906 >>

Keyword ...
Location ...
Country ...



Maven setup – step 3

In the "Enter an artifact id" page of the wizard, you can define the name and main package of your project. Set the "Group Id" variable to "com.javacodegeeks.snippets.enterprise"

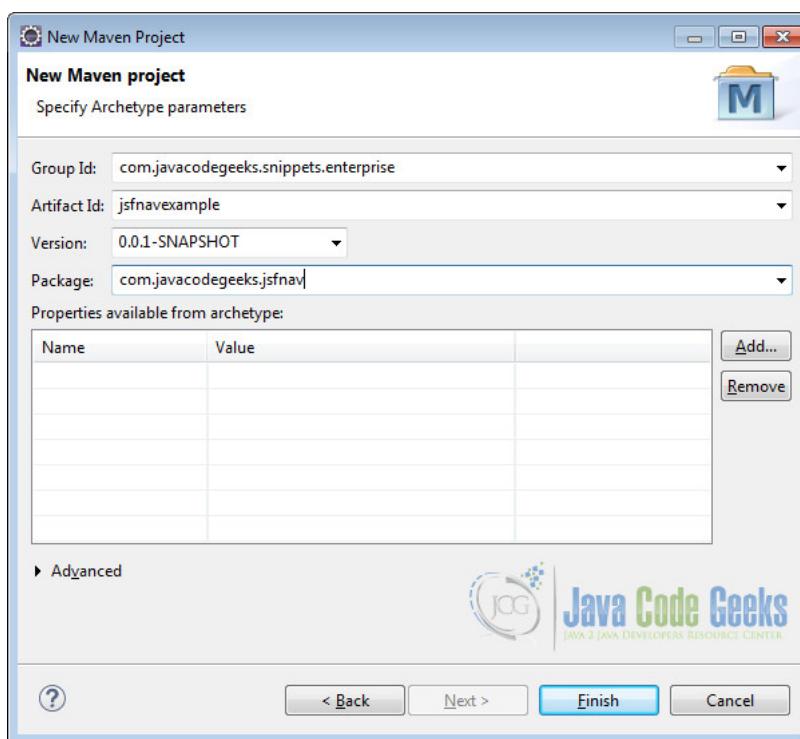
and the "Artifact Id" variable to

"jsfnavexample"

. For package enter

"com.javacodegeeks.jsfnav"

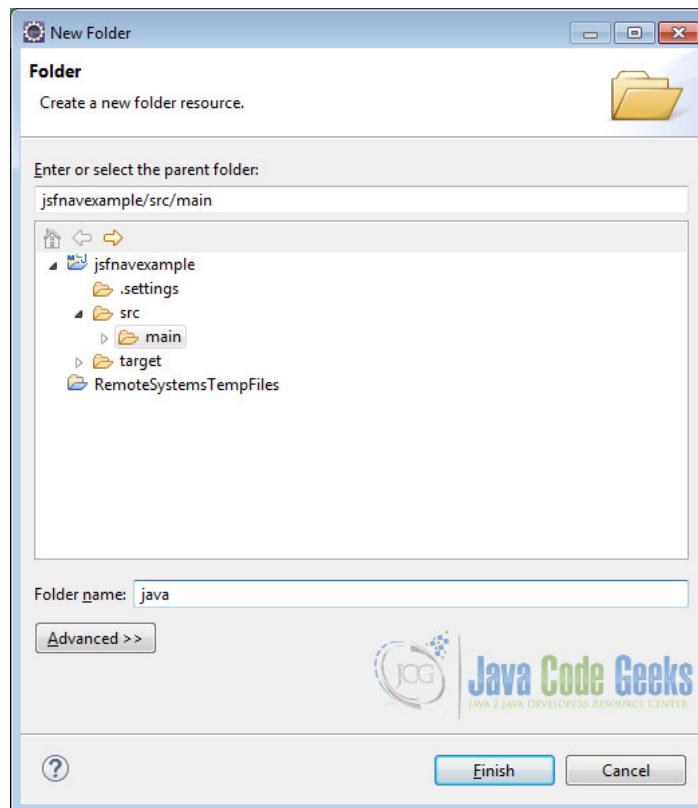
and hit "Finish" to exit the wizard and to create your project.



Maven setup – step 4

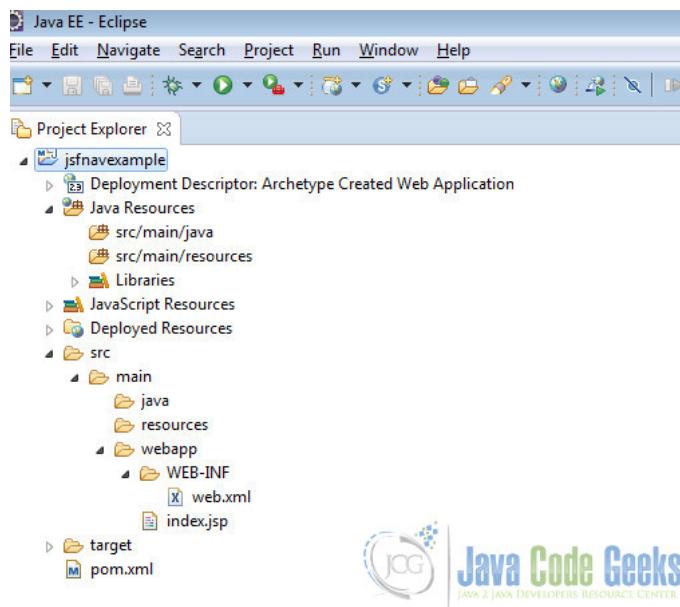
Now create a folder called java under

src.



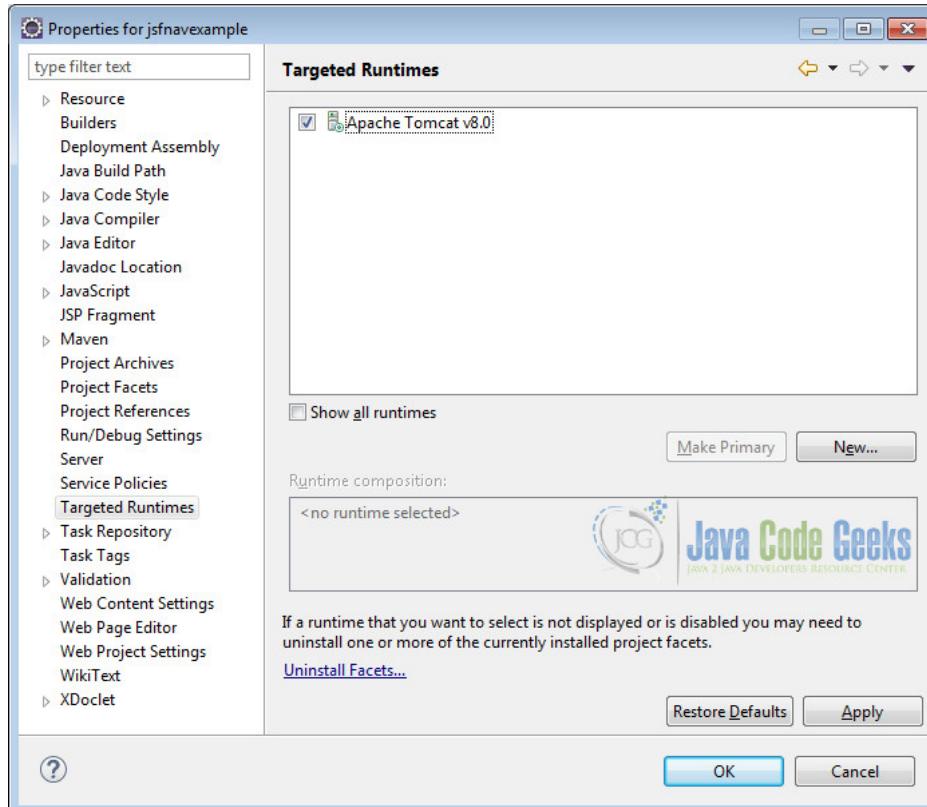
Maven setup – step 5

Refresh the project. Finally, the project structure will look like the below.



Maven setup – step 6

If you see any errors in the index.jsp , set target runtime for the project.



Maven setup – step 7

2. Modify POM to include JSF dependency

Add the dependencies in Maven's

pom.xml

file, by editing it at the "Pom.xml" page of the POM editor.

```

01 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
02 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
03   <modelVersion>4.0.0</modelVersion>
04   <groupId>com.javacodegeeks.snippets.enterprise</groupId>
05   <artifactId>jsfnavexample</artifactId>
06   <packaging>war</packaging>
07   <version>0.0.1-SNAPSHOT</version>
08   <name>jsfnavexample Maven Webapp</name>
09   <url>http://maven.apache.org</url>
10   <dependencies>
11
12     <dependency>
13       <groupId>junit</groupId>
14       <artifactId>junit</artifactId>
15       <version>3.8.1</version>
16       <scope>test</scope>
17     </dependency>
18     <dependency>
19       <groupId>com.sun.faces</groupId>
20       <artifactId>jsf-api</artifactId>
21       <version>2.2.9</version>
22     </dependency>
23     <dependency>
24       <groupId>com.sun.faces</groupId>
25       <artifactId>jsf-impl</artifactId>
26       <version>2.2.9</version>
27     </dependency>
28
29   </dependencies>
30
31   <build>
32     <finalName>jsfnavexample</finalName>
33   </build>
34
35 </project>
```

3. Add Faces Servlet in web.xml

The

web.xml

file has to be modified to include the faces servlet configuration as below.

```

01 <web-app>
02   <display-name>Archetype Created Web Application</display-name>
03 
```

```

04 <servlet>
05   <servlet-name>Faces Servlet</servlet-name>
06   <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
07   <load-on-startup>1</load-on-startup>
08 </servlet>
09
10 <servlet-mapping>
11   <servlet-name>Faces Servlet</servlet-name>
12   <url-pattern>*.xhtml</url-pattern>
13 </servlet-mapping>
14
15 </web-app>

```

4. Implicit Navigation

JSF provides default navigation using convention. Create a view

helloNav.xhtml

under

/WEB-INF/

folder.

helloNav.xhtml

```

01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml"
04   xmlns:h="http://java.sun.com/jsf/html">
05 <head>
06 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
07 <title>JSF Navigation</title>
08 </head>
09 <body>
10 <h1>JSF Navigation</h1>
11 <h:form>
12 <div>
13 Click here to try Implicit Navigation
14 <h:commandButton value="Implicit" action="implicitNav"></h:commandButton>
15 </div>
16 </h:form>
17 </body>
18 </html>

```

Now create another view with the exact name as given in the

action

tag.

implicitNav.xhtml

```

01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml">
04 <head>
05 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
06 <title>JSF Navigation</title>
07 </head>
08 <body>
09 <h2>Implicit Navigation using Convention</h2>
10 <h4>The file name implicitNav.xhtml should match with action
11 mentioned in helloNav.xhtml</h4>
12 </body>
13 </html>

```

Now we can create the deployment package using Run as → Maven clean and then Run as → Maven install. This will create a war file in the target folder. The

war

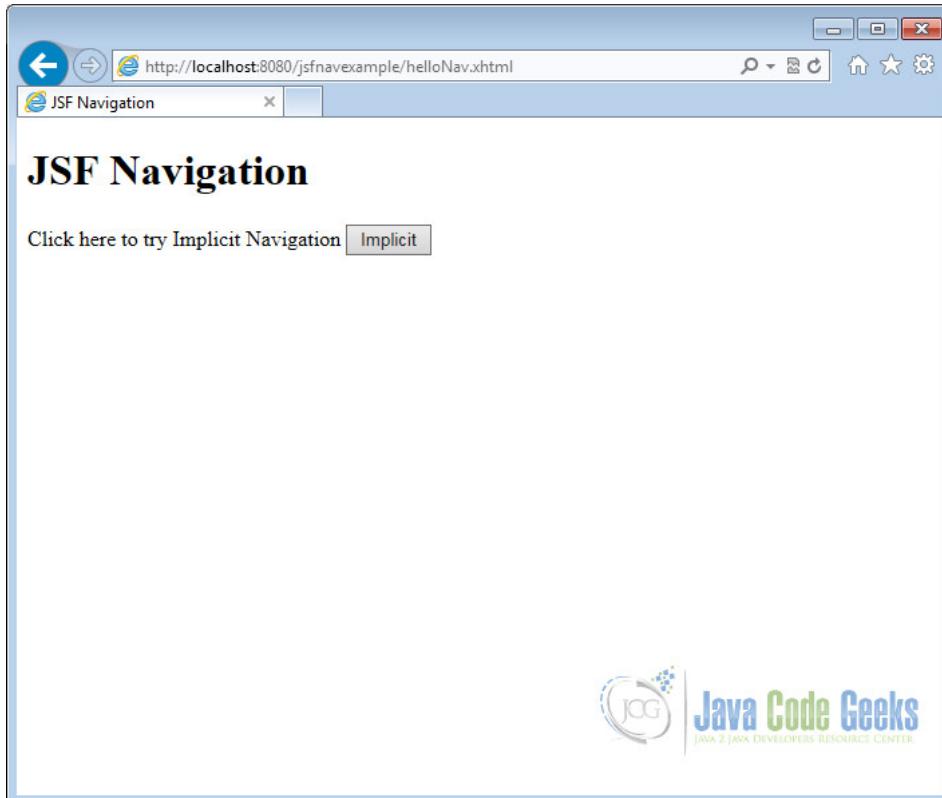
file produced must be placed in

webapps

folder of tomcat. Now we can start the server.

Open the following URL in the browser

http://localhost:8080/jsfnavexample/helloNav.xhtml



Implicit Navigation – output 1

On clicking the **implicit** button, the default navigation handler will match the suitable page and navigate to it.



Implicit Navigation – output 2

5. User defined Navigation

We can also define our own navigation rules instead of convention by using configuration file

```
faces-config.xml
```

5.1 Action Tag based

Create a file called

```
faces-config.xml
```

under

```
/WEB-INF/
```

folder.

faces-config.xml

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <faces-config version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
03   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04   xsi:schemalocation="http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">
05 <navigation-rule>
06   <from-view-id>/helloNav.xhtml</from-view-id>
07   <navigation-case>
08     <from-outcome>userDefined</from-outcome>
09     <to-view-id>/myNav.xhtml</to-view-id>
10   </navigation-case>
11 </navigation-rule>
12 </faces-config>
13 </>
```

Now create a file called

```
myNav.xhtml
```

under

```
/webapp/
```

folder.

myNav.xhtml

```
01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml">
04 <head>
05 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
06 <title>My Nav</title>
07 </head>
08 <body>
09   <h2>Welcome to user defined Navigation!!</h2>
10 </body>
11 </html>
```

Modify the

```
helloNav.xhtml
```

to include "user defined Navigation".

helloNav.xhtml

```
01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml"
04   xmlns:h="http://java.sun.com/jsf/html">
05 <head>
06 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
07 <title>JSF Navigation</title>
08 </head>
09 <body>
10   <h1>JSF Navigation</h1>
11   <h:form>
12     <div>
13       Click here to try Implicit Navigation
14       <h:commandButton value="Implicit" action="implicitNav"></h:commandButton>
15     </div>
16     <div>&nbsp;</div>
17   </div>
18   Click for User defined Navigation
19   <h:commandButton value="User Defined" action="userDefined"></h:commandButton>
20 </div>
21
22 </h:form>
23 </body>
24 </html>
```

Now again package using maven and copy the

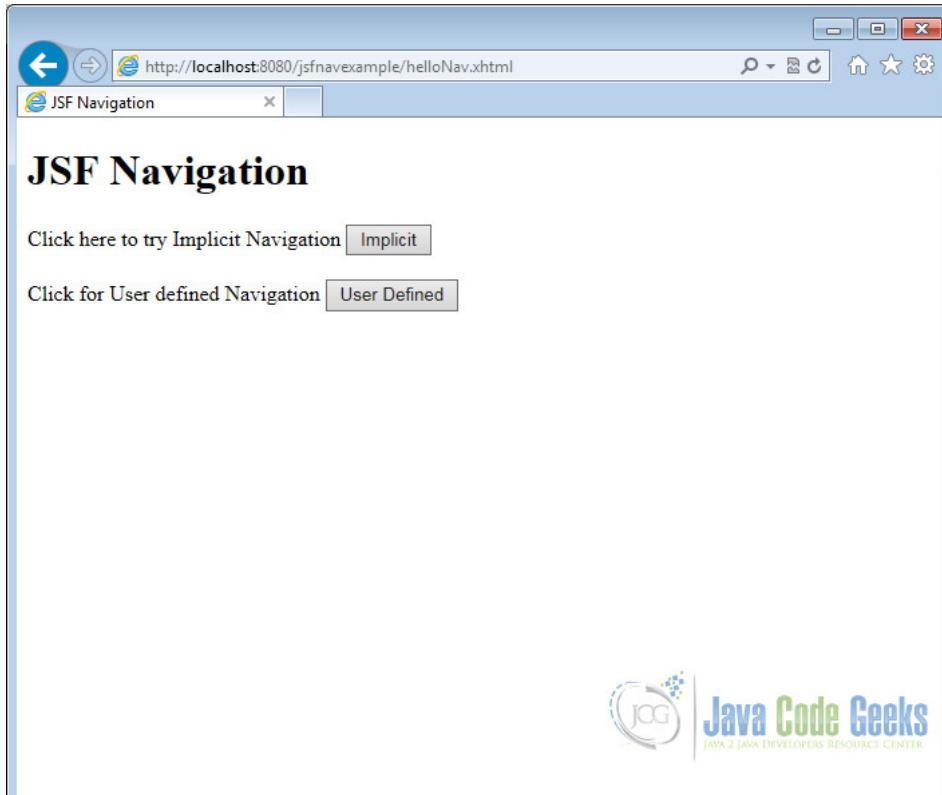
```
war
```

to the apache tomcat

```
webapps
```

folder. Open the following URL in the browser

```
http://localhost:8080/jsfnavexample/helloNav.xhtml
```



Action tag – output 1

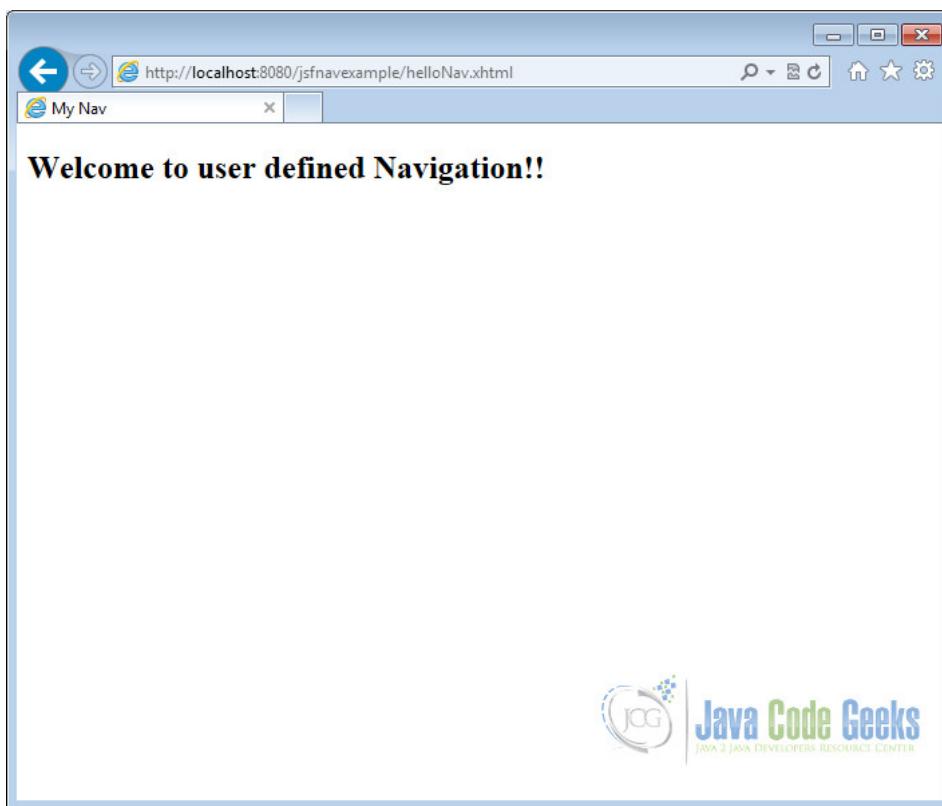
On clicking the user "User Defined" button, the JSF

NavigationHandler

will match the action outcome and navigate to the user defined view

myNav.xhtml

in this case,



Action tag – output 2

5.2 Action Method based

For some **complex scenarios** we need to use the outcome of the **action method** to determine the navigation.

Now we create a package called com.javacodegeeks.jsfnav under Java resources

```
src/main/java
```

. We need to create a managed bean called

```
UserNav.java
```

to handle the action.

UserNav.java

```
01 package com.javacodegeeks.jsfnav;
02
03 import javax.faces.bean.ManagedBean;
04
05 @ManagedBean(name = "userNav", eager = true)
06
07 public class UserNav {
08
09     public String submit(){
10         return "success";
11     }
12 }
13 }
```

Add a new navigation rule to the

```
faces-config.xml
```

```
01 <?xml version='1.0' encoding='UTF-8'?>
02 <faces-config version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
03   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">
05   <navigation-rule>
06     <from-view-id>/helloNav.xhtml</from-view-id>
07     <navigation-case>
08       <from-outcome>userDefined</from-outcome>
09       <to-view-id>/myNav.xhtml</to-view-id>
10     </navigation-case>
11   </navigation-rule>
12   <navigation-rule>
13     <from-view-id>/helloNav.xhtml</from-view-id>
14     <navigation-case>
15       <from-action>#{userNav.submit}</from-action>
16       <from-outcome>success</from-outcome>
17       <to-view-id>/actionMethod.xhtml</to-view-id>
18     </navigation-case>
19   </navigation-rule>
20 </faces-config>
```

Now we create another view called

```
actionMethod.xhtml
```

under

```
/webapp/ folder.
```

actionMethod.xhtml

```
01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml">
04
05   <head>
06     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
07     <title>JSF Navigation</title>
08   </head>
09   <body>
10     <h1>JSF Navigation</h1>
11     <h2>Action method outcome success!!</h2>
12   </body>
13 </html>
```

Modify the

```
helloNav.xhtml
```

to include **Action Method** based Navigation.

helloNav.xhtml

```
01 <?xml version="1.0" encoding="ISO-8859-1" ?>
02 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml"
04   xmlns:h="http://java.sun.com/jsf/html">
05   <head>
06     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
07     <title>JSF Navigation</title>
08   </head>
09   <body>
10     <h1>JSF Navigation</h1>
11     <h2>Action method outcome success!!</h2>
12     <div>
13       Click here to try Implicit Navigation
14       <h:commandButton value="Implicit" action="implicitNav"></h:commandButton>
15     </div>
16   </body>
17 </html>
```

```

15  </div>
16  <div>&nbsp;</div>
17  <div>
18  Click for User defined Navigation
19  <h:commandButton value="User Defined" action="#{userDefined}"></h:commandButton>
20  </div>
21  <div>&nbsp;</div>
22  <div>
23  Click for User defined action method Navigation
24  <h:commandButton value="Action Method" action="#{userNav.submit}"></h:commandButton>
25  </div>
26
27  </h:form>
28  </body>
29  </html>

```

Now again do maven package and copy the

war

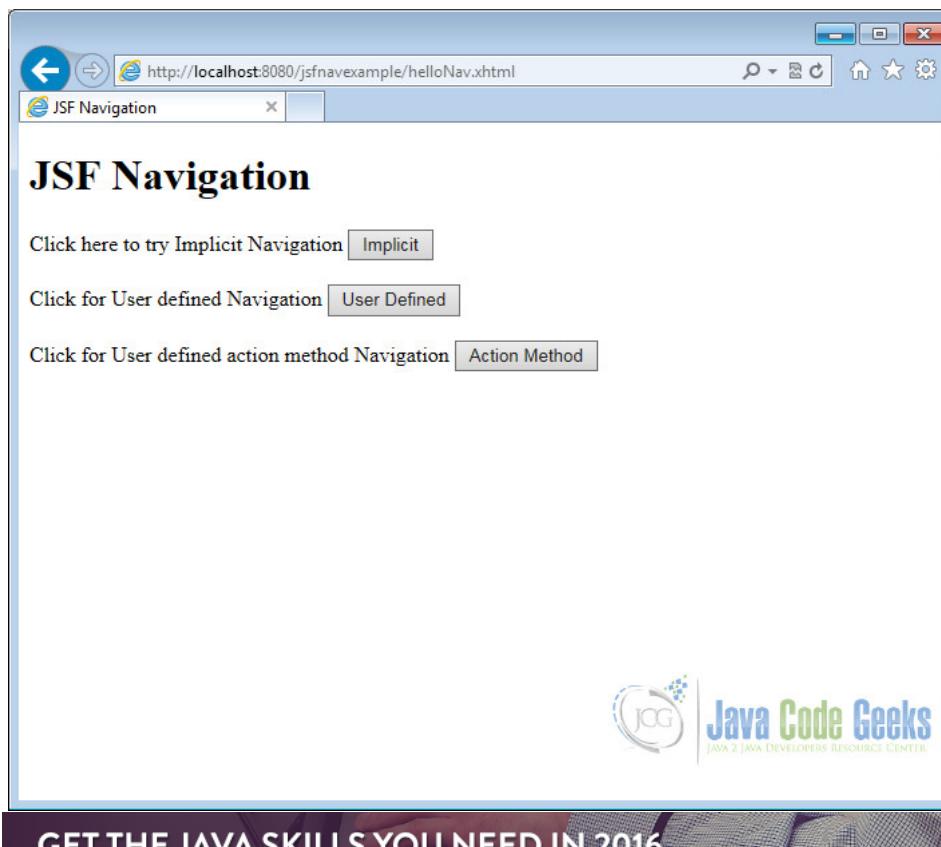
to the apache tomcat

webapps

folder.

Open the following URL in the browser

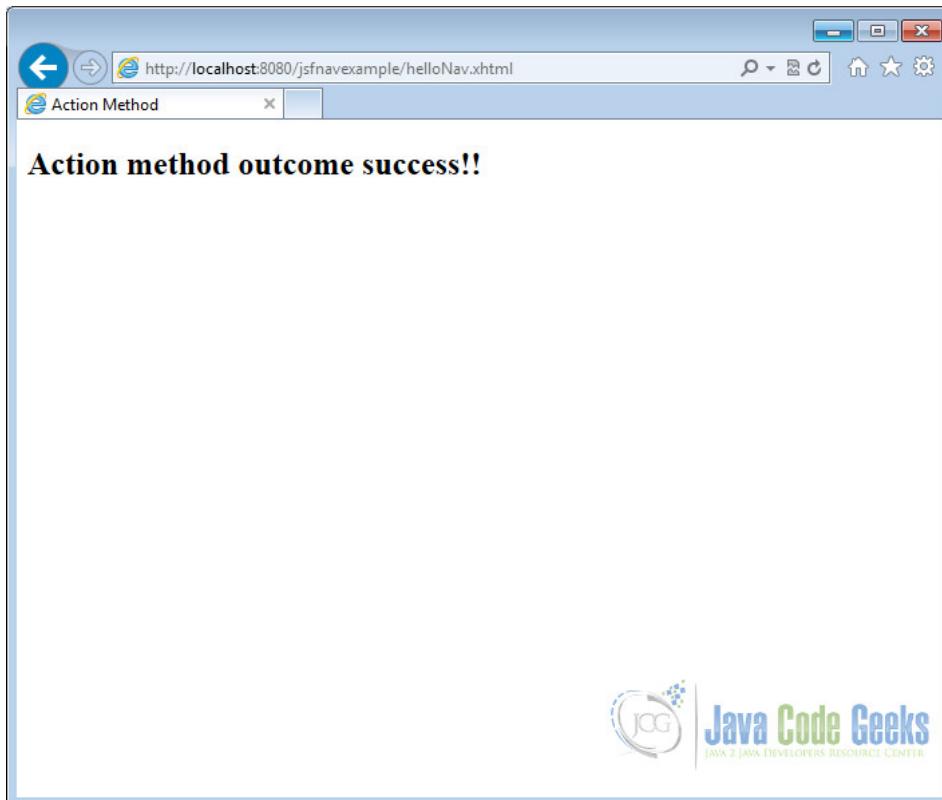
<http://localhost:8080/jsfnavexample/helloNav.xhtml>



On clicking the "Action Method" button, the

NavigationHandler

will match the action outcome and navigate to the user defined view.



Action method – output 2

6. Download the Eclipse Project

This was an example of various approaches in Java Server Faces Navigation.

Download

You can download the full source code of this example here : [JSF Navigation Example](#).

Do you want to know how to develop your skillset to become a **Java Rockstar?**

Subscribe to our newsletter to start Rocking [right now!](#)
To get you started we give you our best selling eBooks for **FREE!**

1. JPA Mini Book
2. JVM Troubleshooting Guide
3. JUnit Tutorial for Unit Testing
4. Java Annotations Tutorial
5. Java Interview Questions
6. Spring Interview Questions
7. Android UI Design

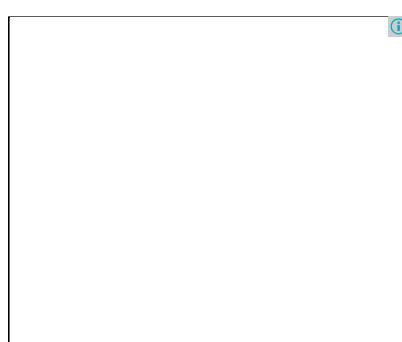
GET THE JAVA SKILLS YOU NEED IN 2016.

[Start Learn](#)

Email address:

Your email address

Sign up



KNOWLEDGE BASE

Courses
News
Resources
Tutorials
Whitepapers

THE CODE GEEKS NETWORK

.NET Code Geeks
Java Code Geeks
System Code Geeks
Web Code Geeks

HALL OF FAME

Android Alert Dialog Example
Android OnClickListener Example
How to convert Character to String and a String to Character Array in Java
Java Inheritance example
Java write to File Example
java.io.FileNotFoundException – How to solve File Not Found Exception
java.lang.ArrayIndexOutOfBoundsException – How to handle Array Index Out Of Bounds Exception
java.lang.NoClassDefFoundError – How to solve No Class Def Found Error
JSON Example With Jersey + Jackson
Spring JdbcTemplate Example

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on ultimate Java to Java developers resource center; targeted at the technical team lead (senior developer), project manager and junior dev JCGs serve the Java, SOA, Agile and Telecom communities with daily news, domain experts, articles, tutorials, reviews, announcements, code snippets and source projects.

DISCLAIMER

All trademarks and registered trademarks appearing on Java Code Geeks are the property of their respective owners. Java is a trademark or registered trademark of Oracle Corporation in the United States and other countries. Examples Java Code Geeks is not connected to Oracle Corporation and is not sponsored by Oracle Corporation.

Examples Java Code Geeks and all content copyright © 2010-2016, Exelixis Media P.C. | Terms of Use | Privacy Policy | Contact



GET THE JAVA SKILLS YOU NEED IN 2016.

Start Learning