Wicket
Introduction

DBMS 2013-2014 Fall

Prepared by: Nagehan İlhan

Required Softwares

- Download Eclipse for Java EE developers:

  http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junosr1

- Download Apache Tomcat 7.0:

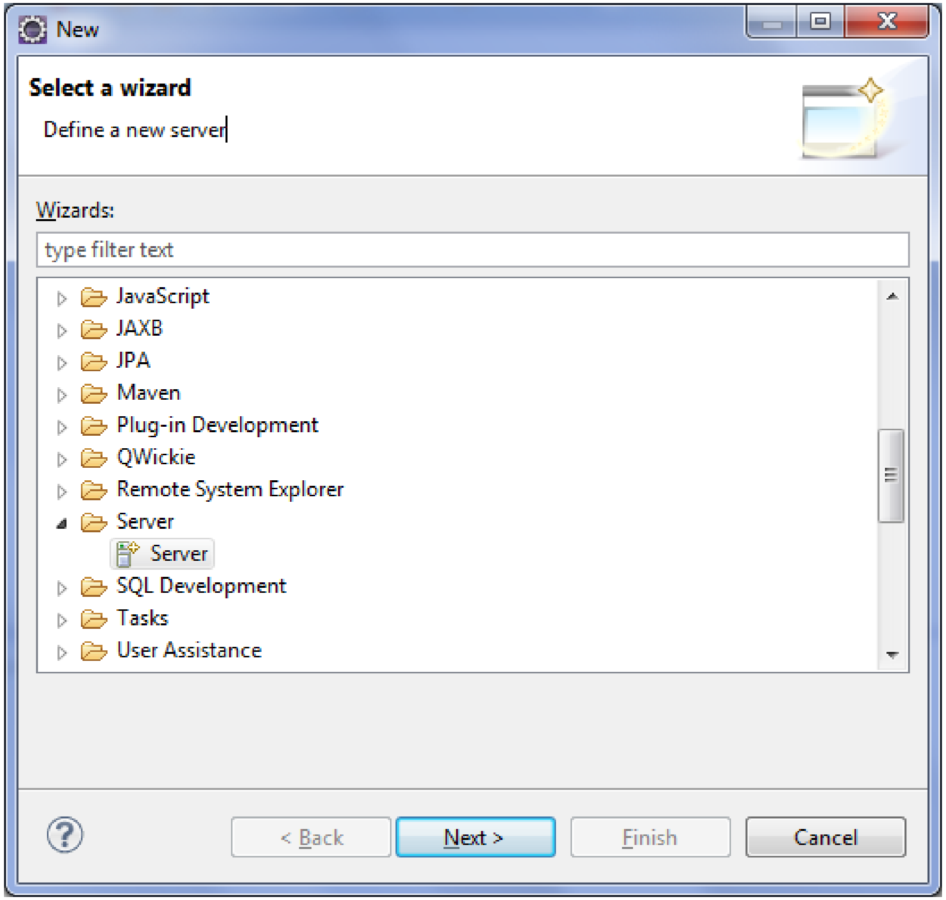
  http://apache.bilkent.edu.tr/tomcat/tomcat-7/v7.0.42/bin/apache-tomcat-7.0.42.exe

- Be sure to have last version of Java Runtime Environment:

  http://www.oracle.com/technetwork/java/javase/downloads/java-se-jre-7-download-432155.html

Define Apache Tomcat Server

- Select File->New->Other->Server->Server



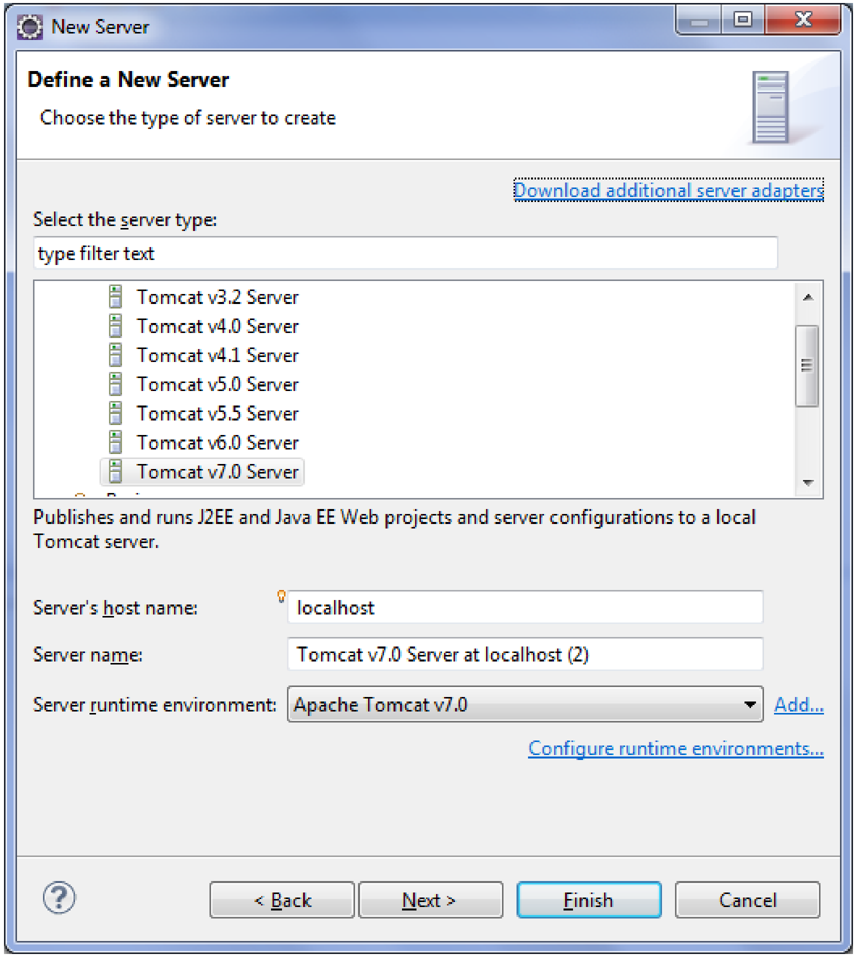
Select a wizard

Define a new server

Wizards:

type filter text

Define Apache Tomcat Server



New Server

Define a New Server

Choose the type of server to create

Download additional server adapters

Select the server type:

type filter text

- Tomcat v3.2 Server
- Tomcat v4.0 Server
- Tomcat v4.1 Server
- Tomcat v5.0 Server
- Tomcat v5.5 Server
- Tomcat v6.0 Server
- Tomcat v7.0 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost

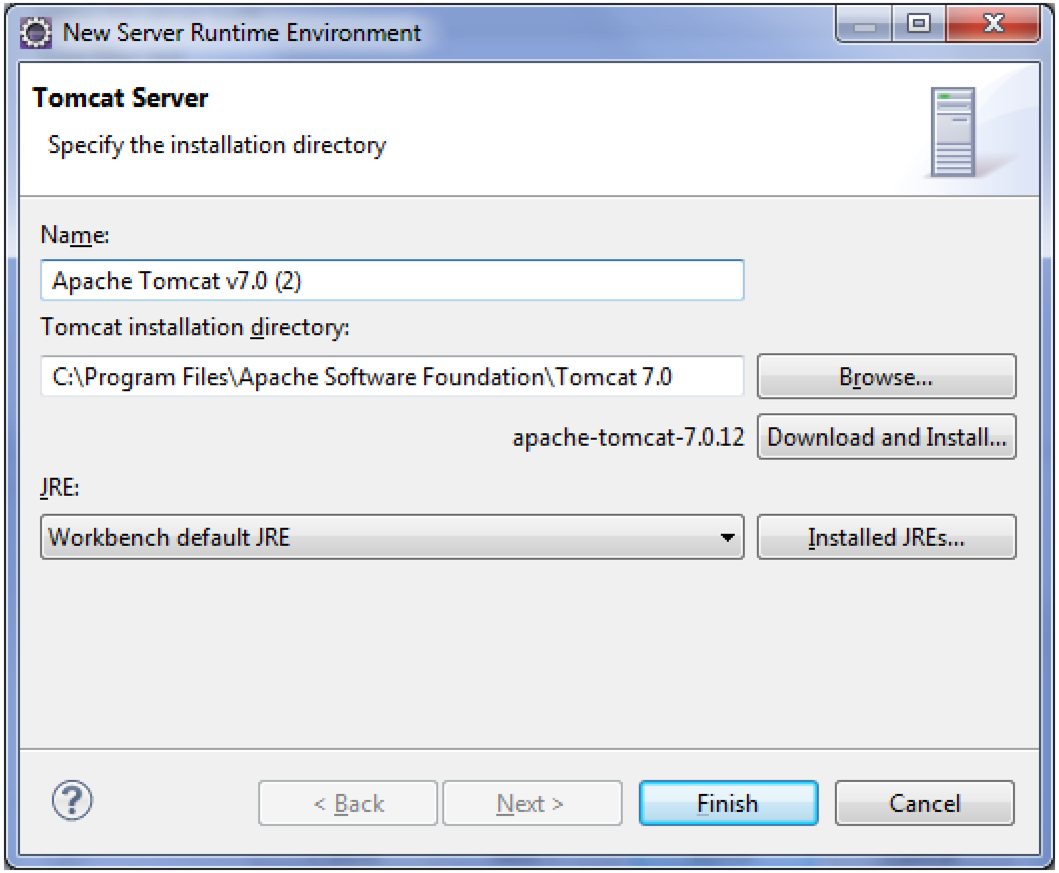
Server name: Tomcat v7.0 Server at localhost (2)

Server runtime environment: Apache Tomcat v7.0

- Click Add

Define Apache Tomcat Server

- Show the location of Tomcat in your computer.



New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name: Apache Tomcat v7.0 (2)

Tomcat installation directory: C:\Program Files\Apache Software Foundation\Tomcat 7.0

apache-tomcat-7.0.12

JRE: Workbench default JRE

Creating a Wicket Project

- Read `Wicket Application Development Tutorial` from Turgut Uyar's page:

  http://web.itu.edu.tr/uyar/wicket-application-development/

- Download the project files:

https://dl.dropboxusercontent.com/u/61579540/quickstart.zip

https://dl.dropboxusercontent.com/u/61579540/wicket.zip

https://dl.dropboxusercontent.com/u/61579540/slf4j.zip

https://dl.dropboxusercontent.com/u/61579540/jetty.zip

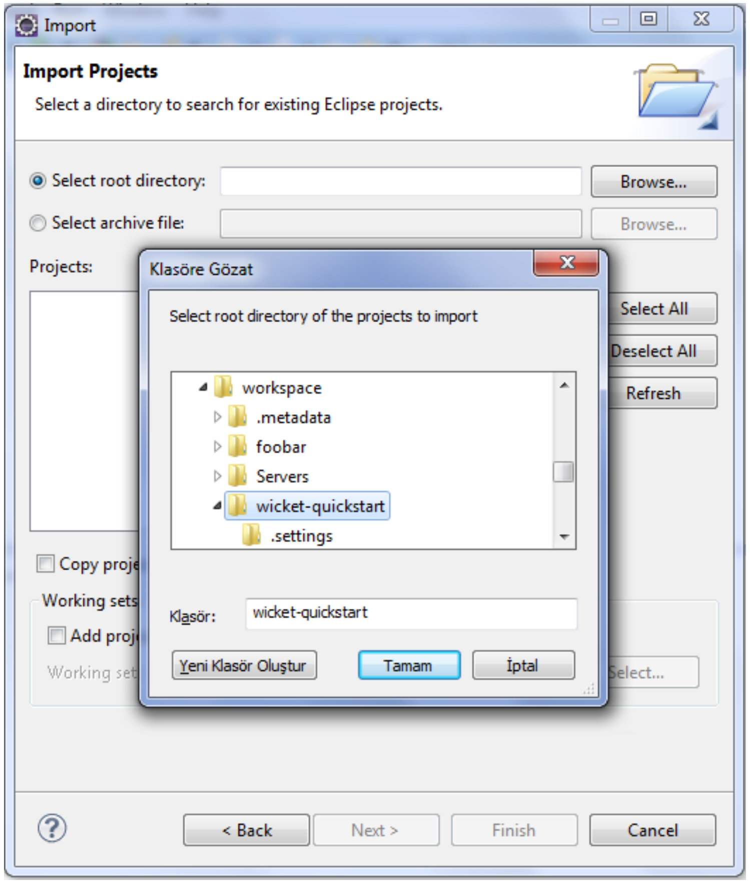
Creating a Wicket Project

- Extract the files and organize them by following the given hierarchy:

```
wicket-quickstart
|- src
|  |- resources
|  |- webapp
|  |- wicket
|- lib
|  |- wicket-*.jar
|  |- slf4j-*.jar
|  |- jetty-*.jar
```

Creating a Wicket Project

- Copy wicket-quickstart folder into your workspace
- Select File->Import->General->Existing Projects into Workspace and select wicket-quickstart



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

□ Copy proje

Working sets

□ Add proj

Working set

Klasöre Gözat

Select root directory of the projects to import

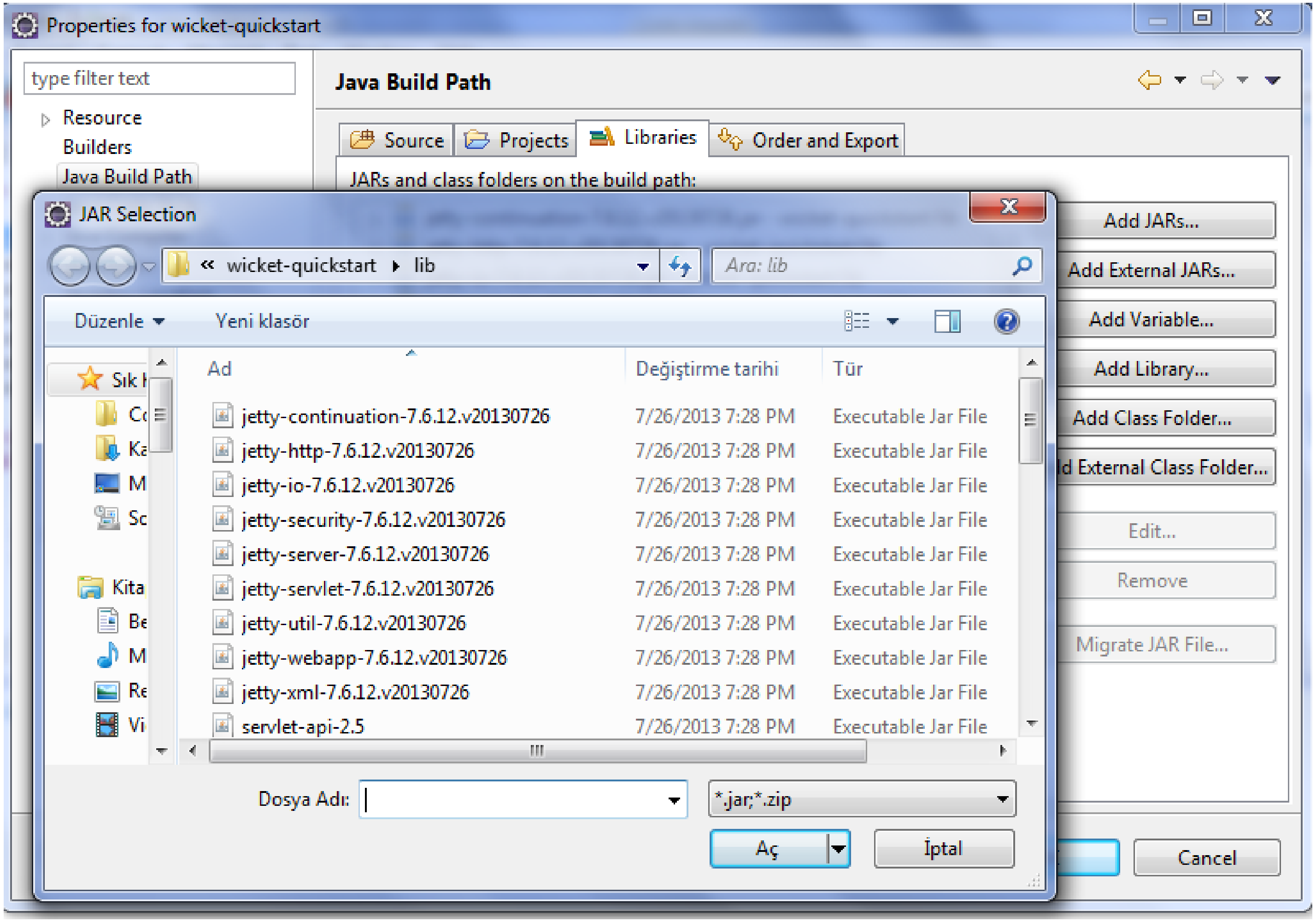
- workspace
  - .metadata
  - foobar
  - Servers
  - wicket-quickstart
    - .settings

Klasör: wicket-quickstart

Creating a Wicket Project

- Right click on wicket-quickstart project and click Properties->Java Build Path->Libraries and click Add External Jars.
- Select .jar files located in wicket-quickstart\lib folder.
- Than Run the WicketApplication.java file.

Creating a Wicket Project



Java Build Path

JARs and class folders on the build path:

wicket-quickstart ▸ lib

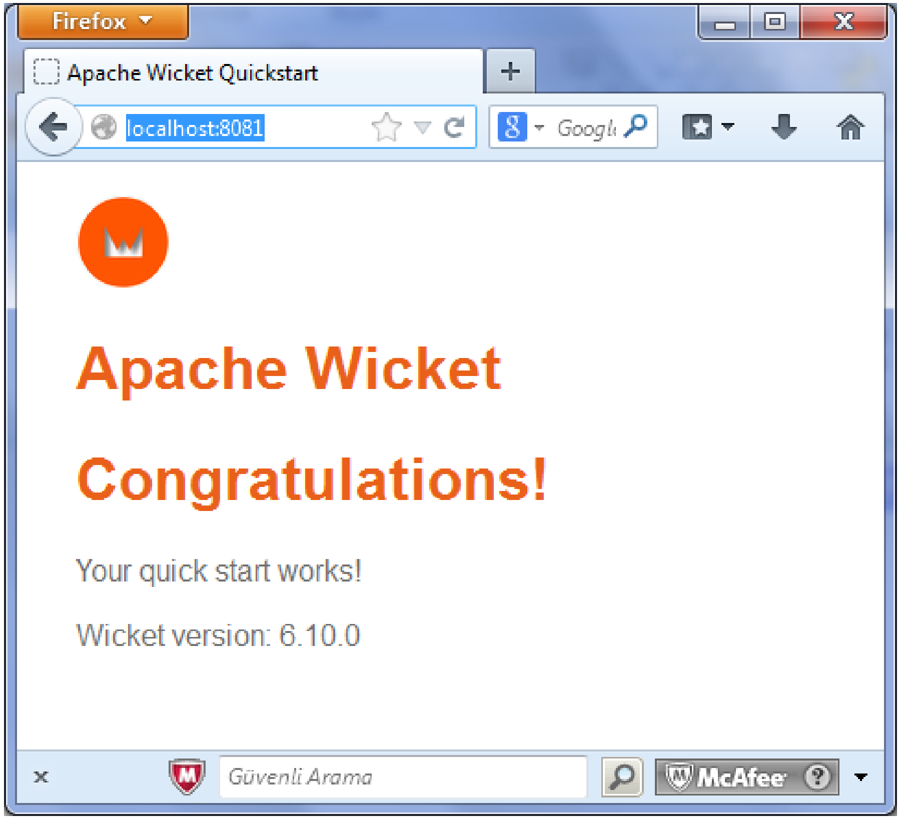
Ad	Değiştirme tarihi	Tür
jetty-continuation-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-http-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-io-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-security-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-server-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-servlet-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-util-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-webapp-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
jetty-xml-7.6.12.v20130726	7/26/2013 7:28 PM	Executable Jar File
servlet-api-2.5	7/26/2013 7:28 PM	Executable Jar File

Dosya Adı:

.jar;.zip

Creating a Wicket Project

- Open explorer and http://localhost:8081/



Apache Wicket

Congratulations!

Your quick start works!

Wicket version: 6.10.0

Creating a simple home page containing only a page title

- Modify the contents of the file ***HomePage.html***, and the file ***HomePage.java***.
- Modify ***HomePage.html*** as below.

```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
  <head>
    <meta charset="utf-8" />
    <title>MovieDB - Home</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>MovieDB Homepage</h1>
  </body>
</html>
```

Creating a simple home page containing only a page title

- Modify ***HomePage.java*** as below.

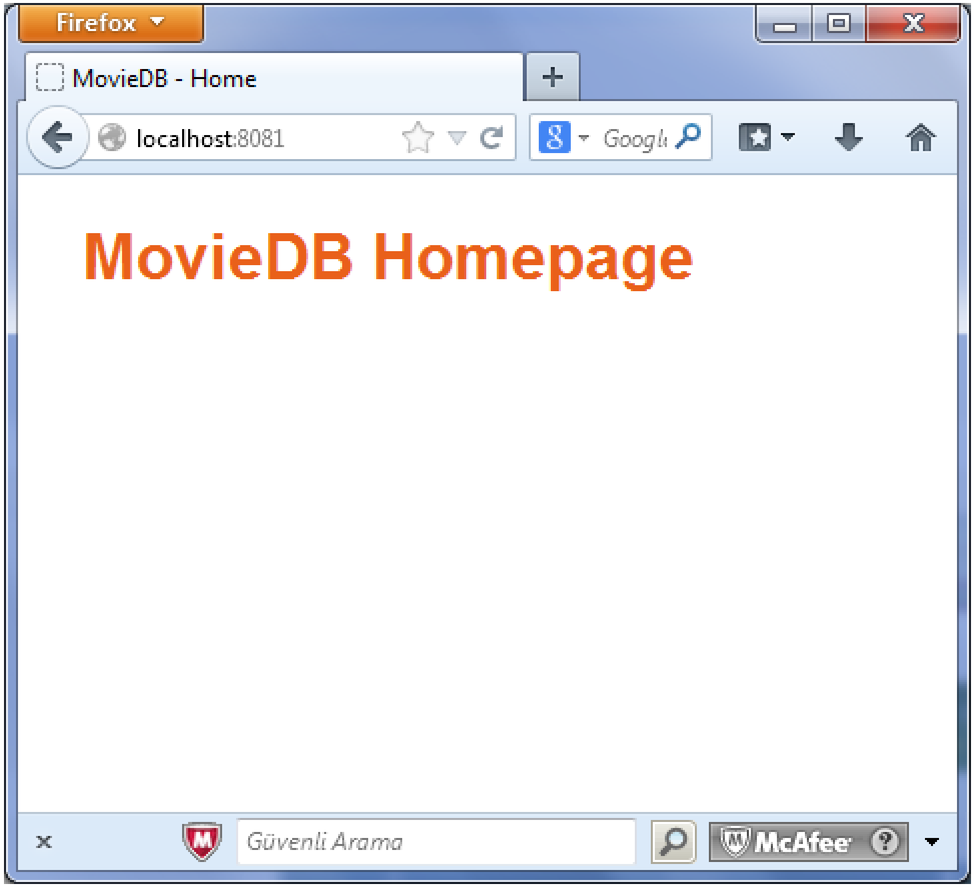
```
package wicket.quickstart;

import org.apache.wicket.markup.html.WebPage;

public class HomePage extends WebPage {
    public HomePage() {
    }
}
```

Creating a simple home page containing only a page title

- Modify **src/webapp/style.css** as below.

```
body {
  background-color: white;
  color: #6F6F6F;
  font-family: 'Droid Sans', 'Helvetica', 'sans';
  font-size: 12pt;
  margin: 2ex 2em;
}
h1, h2, h3, h4 {
  color: #E9601A;
}
a {
  color: #6F6F6F;
  text-decoration: underline;
}
```

Creating a simple home page containing only a page title



MovieDB Homepage

Creating a simple home page containing dynamic content

- Add the current date and time to the home page.
- Modify ***HomePage.html*** as below.
- The ***wicket:id*** attribute indicates that the contents of this **div element** are dynamic.

```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
  <head>
    <meta charset="utf-8" />
    <title>MovieDB - Home</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>MovieDB Homepage</h1>

    <footer>
      <div id="datetime" wicket:id="datetime">
         Wed Sep 4 15:32:40 EEST 2013
      </div>
    </footer>
  </body>
</html>
```

Creating a simple home page containing dynamic content

- Modify ***HomePage.java*** as below.

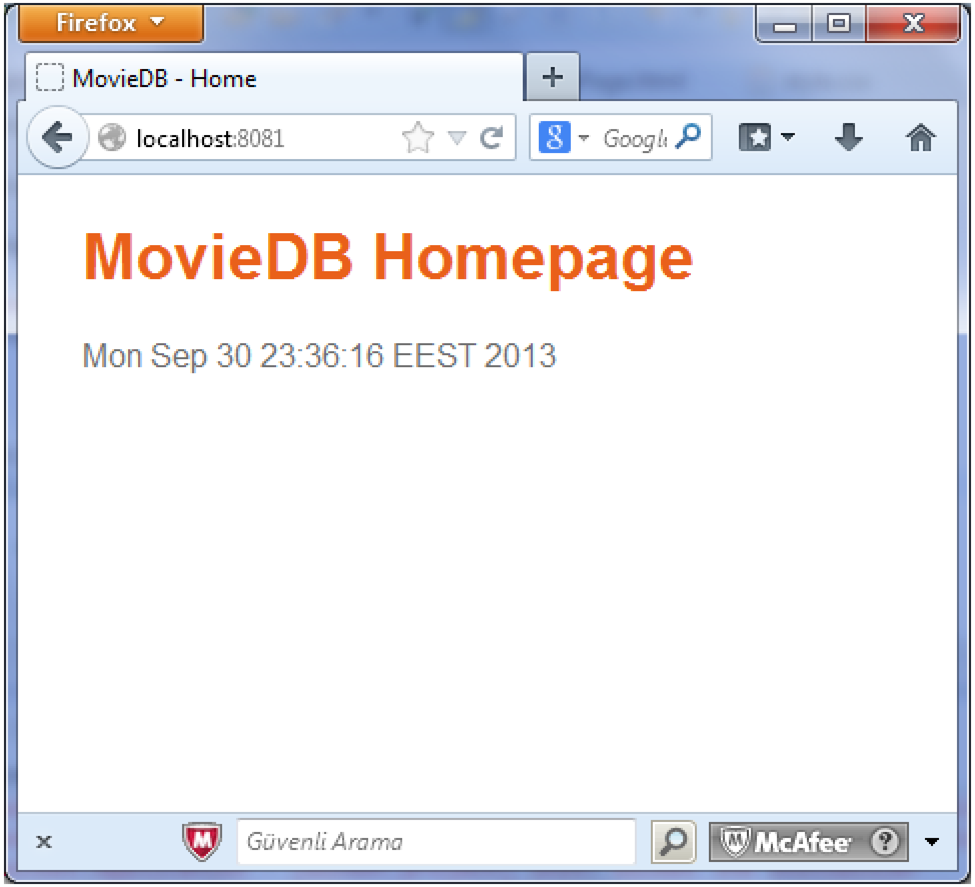
```
package wicket.quickstart;

import java.util.Date;

import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.markup.html.basic.Label;

public class HomePage extends WebPage {
    public HomePage() {
        Date now = new Date();
        Label dateTimeLabel = new Label("datetime", now.toString());
        this.add(dateTimeLabel);
    }
}
```

Creating a simple home page containing dynamic content

- Modify **src/webapp/style.css** as below.

```
div#datetime {
  font-size: 80%;
  text-align: right;
}
```

Creating a simple home page containing dynamic content



MovieDB Homepage

Mon Sep 30 23:36:16 EEST 2013

Element Hierarchy

- Adding a component to another component creates a hierarchy between the two.
- The HTML template also has a hierarchy due to the nesting of the HTML tags.
- Wicket matches the hierarchy of the HTML template to the hierarchy of the Java components.

Element Hierarchy

- The template below where a span element is nested inside a div element.

```
<body>
    <div wicket:id="x"> some text
        <span wicket:id="y">
                  some other text
        </span>
    </div>
</body>
```

Adding Links

- Create a second page which will be responsible for listing the movies in the collection.
- To create this page, right click on the package in the “Projects” pane,
- Select ***New* ▸ *Wicket Page*** from the context menu and enter “**MovieListPage**” as the name.
- ***MovieListPage.html*** and ***MovieListPage.java*** files will be created.

Adding Links

- Modify ***MovieListPage.java*** as below.

```
package wicket.quickstart;

import org.apache.wicket.markup.html.WebPage;

public class MovieListPage extends WebPage {
   public MovieListPage() {
   }
}
```

Adding Links

- Modify ***MovieListPage.html*** as below.

```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
  <head>
    <meta charset="utf-8" />
    <title>MovieDB - Movie List</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h2>Movie List</h2>
  </body>
</html>
```

Adding Links

- For the new class, implement a constructor and override the onClick() method that specifies what will happen when the link is clicked.
- To create this class, right click on the package name, select ***New* ‣ *File*** and give it the name ***"MovieListPageLink.java"***.

Adding Links

- Modify ***MovieListPageLink.java*** as below.

```
package wicket.quickstart;

import org.apache.wicket.markup.html.link.Link;

public class MovieListPageLink extends Link {
    public MovieListPageLink(String id) {
        super(id);
    }

    @Override
    public void onClick() {
        MovieListPage targetPage = new MovieListPage();
        this.setResponsePage(targetPage);
    }
}
```

Adding Links

- Modify ***HomePage.html*** as below.

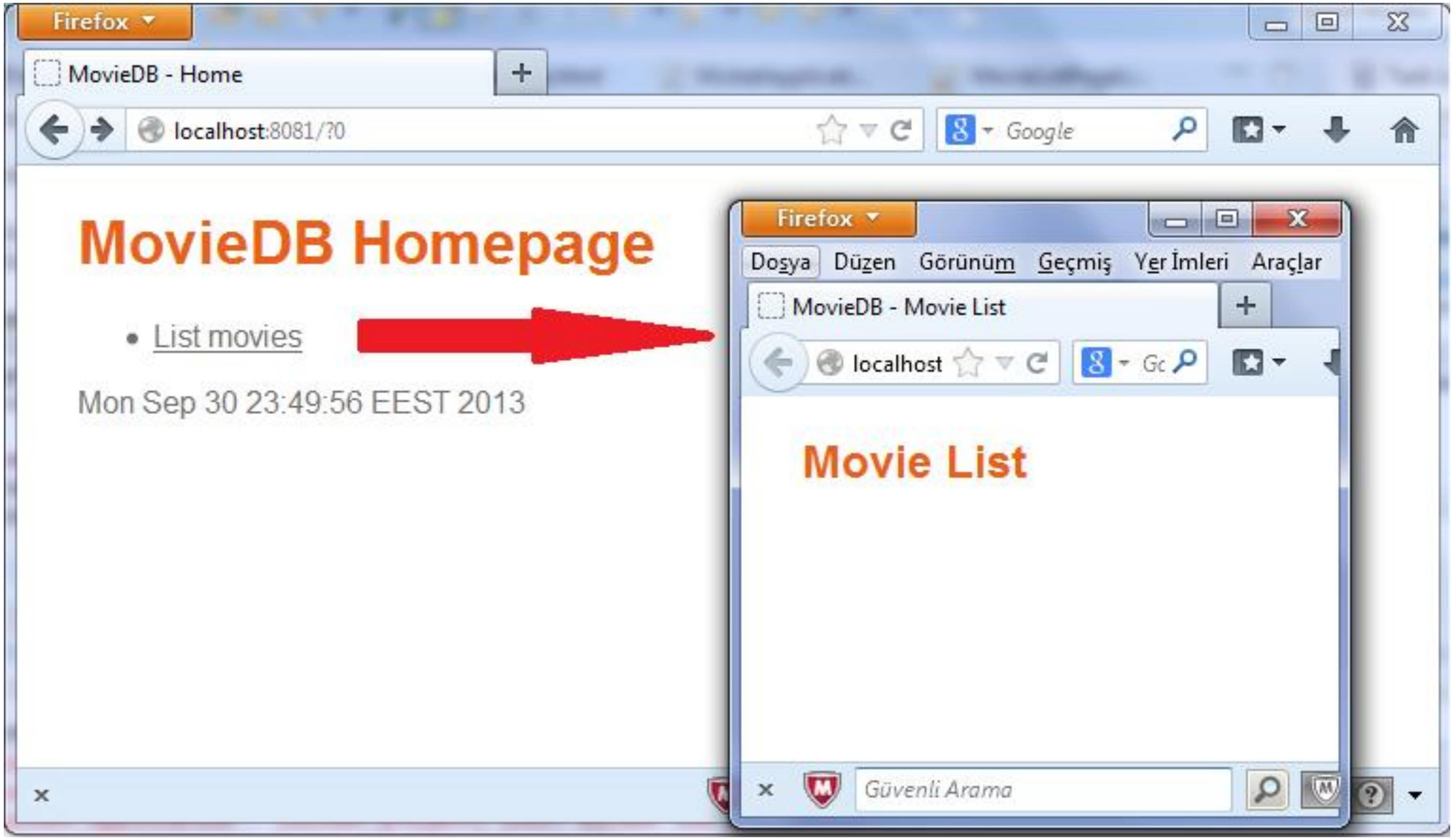
```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
 <head>
  <meta charset="utf-8" />
  <title>MovieDB - Home</title>
  <link rel="stylesheet" href="style.css" />
 </head>
 <body>
  <h1>MovieDB Homepage</h1>

  <ul>
   <li><a href="#" wicket:id="list_movies">List movies</a></li>
  </ul>

  <footer>
   <div id="datetime" wicket:id="datetime">
    Wed Sep 4 15:32:40 EEST 2013
   </div>
  </footer>
 </body>
</html>
```

Adding Links

- Modify ***HomePage.java*** as below.

```
public HomePage() {
   MovieListPageLink movieListPageLink = new MovieListPageLink("list_movies");
   this.add(movieListPageLink);

   Date now = new Date();
   Label dateTimeLabel = new Label("datetime", now.toString());
   this.add(dateTimeLabel);
}
```

Adding Links



MovieDB Homepage

- List movies

Mon Sep 30 23:49:56 EEST 2013

Movie List

Adding a link from the movie list page to the home page

- Modify ***MovieListPage.html*** as below.

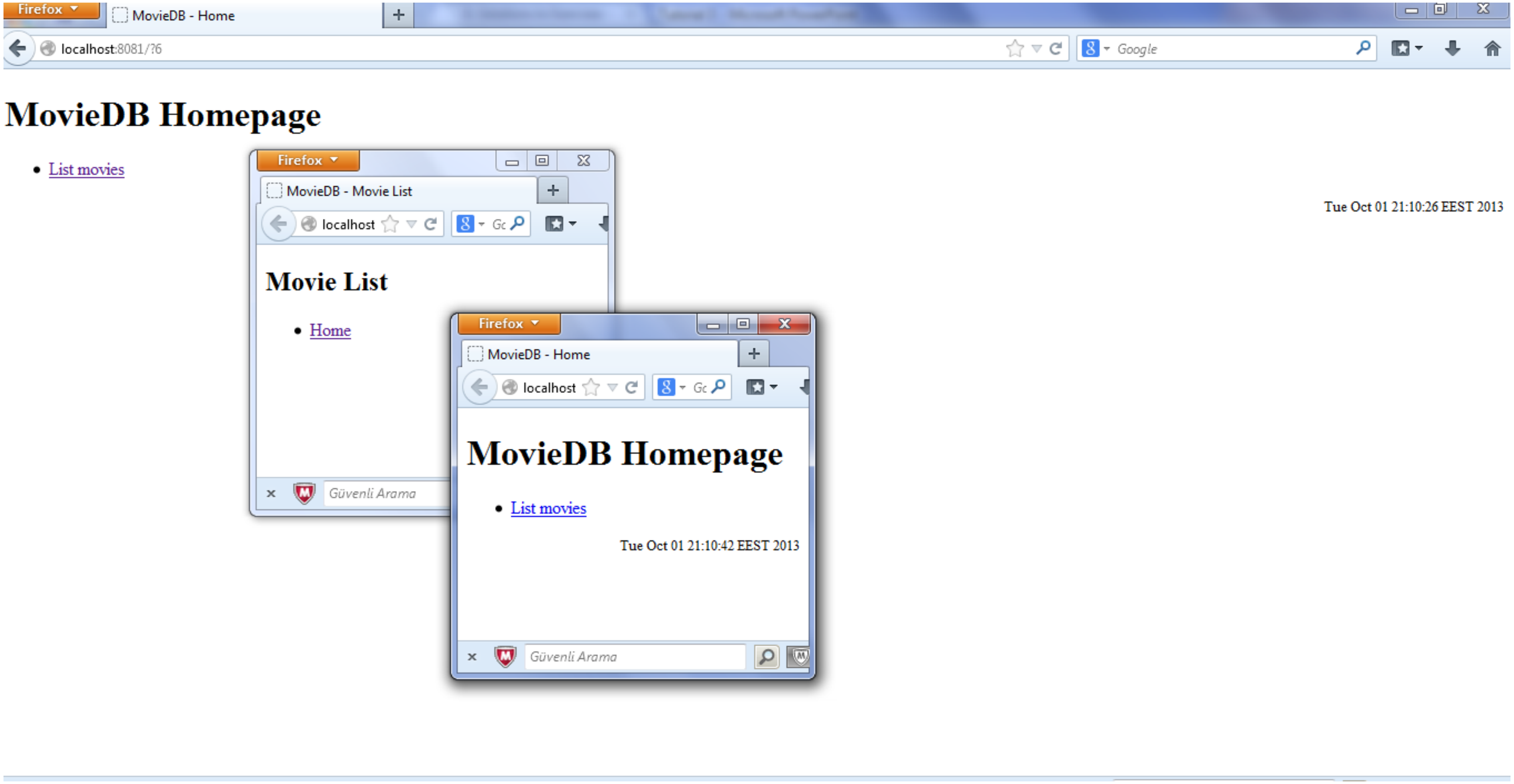
```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
  <head>
    <meta charset="utf-8" />
    <title>MovieDB - Movie List</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h2>Movie List</h2>

    <ul>
      <li><a href="#" wicket:id="home">Home</a></li>
    </ul>
  </body>
</html>
```

Adding a link from the movie list page to the home page

- Modify ***MovieListPage.java*** as below.

```
package wicket.quickstart;

import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.markup.html.link.Link;

public class MovieListPage extends WebPage {
    public MovieListPage() {
        Link homePageLink = new Link("home") {
            @Override
            public void onClick() {
                this.setResponsePage(new HomePage());
            }
        };
        this.add(homePageLink);
    }
}
```

Adding a link from the movie list page to the home page



MovieDB Homepage

- List movies

Tue Oct 01 21:10:26 EEST 2013

Movie List

- Home

MovieDB Homepage

- List movies

Tue Oct 01 21:10:42 EEST 2013