Simple Demo:

Spring + Hibernate + JSF, Primefaces Intergration (Step by Step)

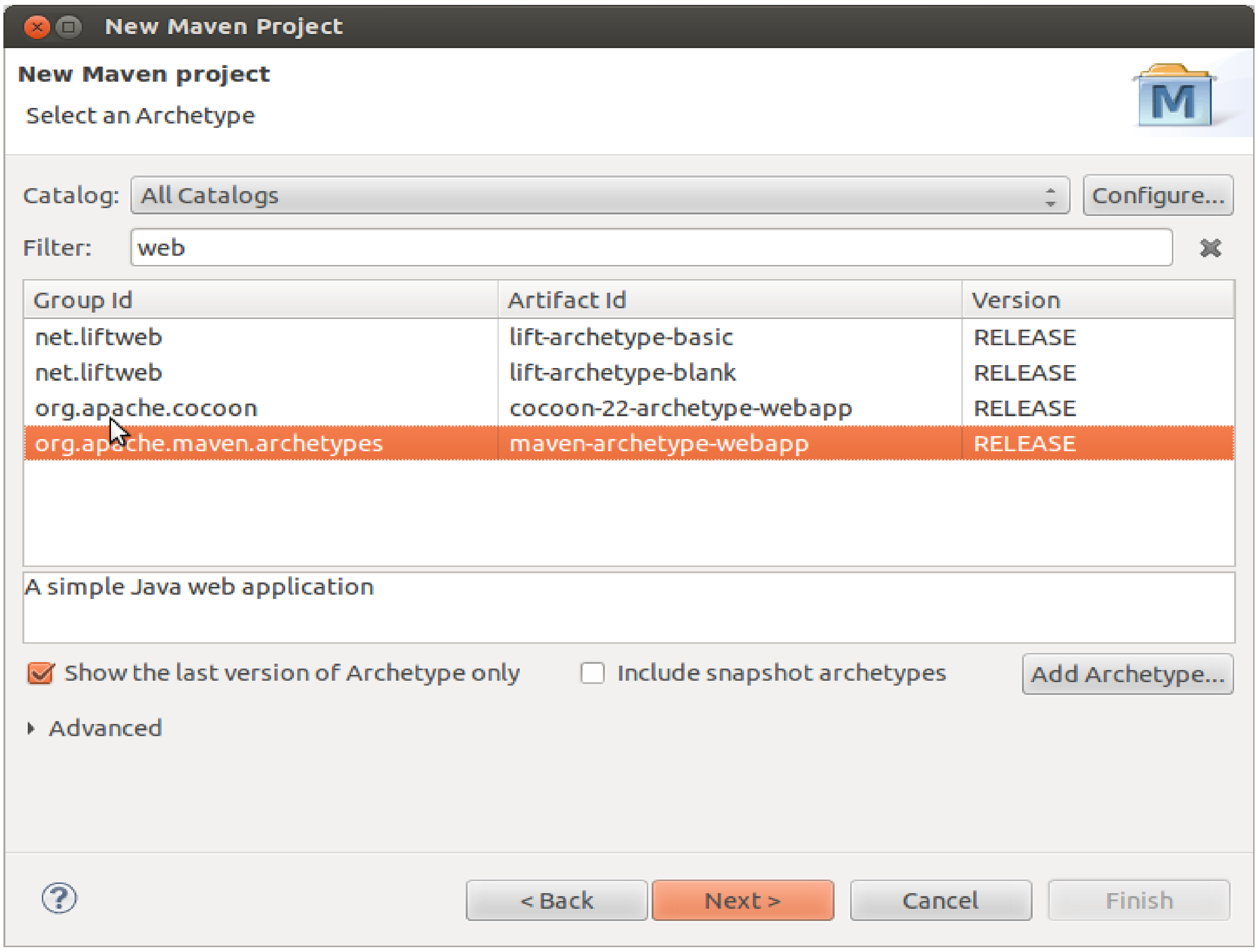
Tool:
- IDE: STS 3.3.0.RELEASE (or Eclipse Kepler with Maven and STS plug-in)
- Server: Apache Tomcat v7.0
- Database: PostgresQSL 9.1, pgAdmin 1.14.0

Used Technologies:
- Spring framework 3.2.3.RELEASE
- Hibernate 4.1.0.Final
- Myfaces 2.1.12 (JSF Implementation)
- Primefaces 3.5

Step 1. Maven

New → Maven Project → Next

Select an Archetype, on **Filter**: enter "**web**", chose "**maven-web-archetype**" for a simple Java web application.



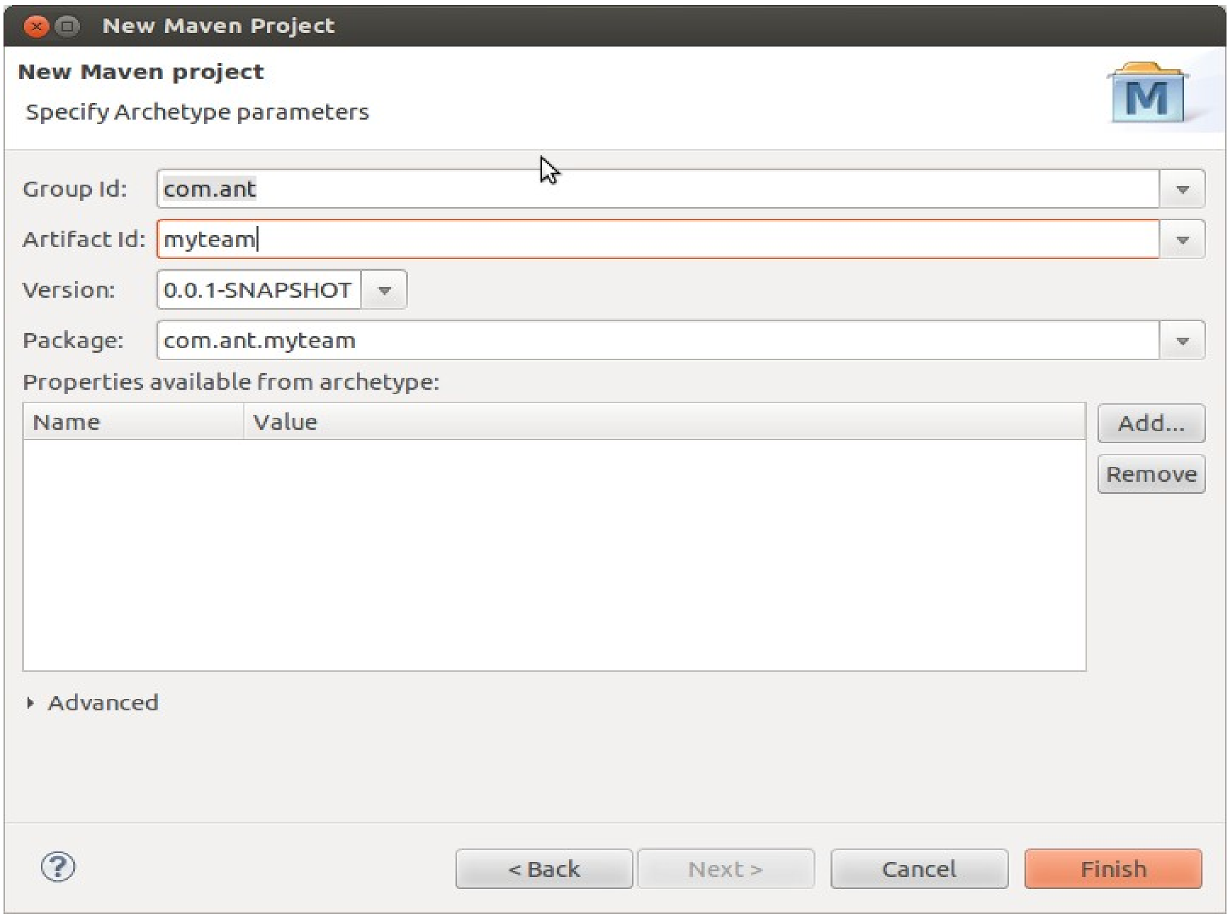
Catalog: All Catalogs
Filter: web

Group Id	Artifact Id	Version
net.liftweb	lift-archetype-basic	RELEASE
net.liftweb	lift-archetype-blank	RELEASE
org.apache.cocoon	cocoon-22-archetype-webapp	RELEASE
org.apache.maven.archetypes	maven-archetype-webapp	RELEASE

A simple Java web application

[✓] Show the last version of Archetype only
[ ] Include snapshot archetypes

Click **Next**


New Maven Project

New Maven project

Specify Archetype parameters

Group Id: com.ant

Artifact Id: myteam

Version: 0.0.1-SNAPSHOT

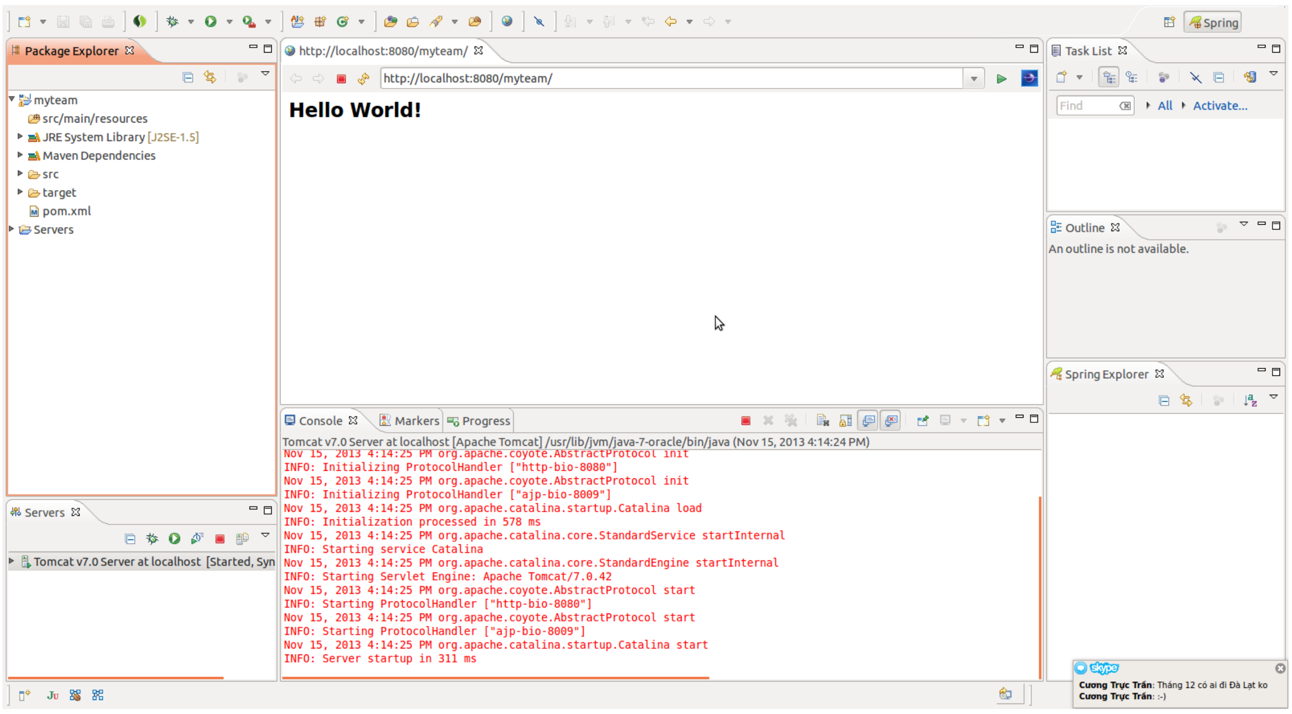
Package: com.ant.myteam

Properties available from archetype:

Name	Value

Click **Finish**

Test: Run the project!



http://localhost:8080/myteam/

Hello World!

```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 15, 2013 4:14:24 PM)
Nov 15, 2013 4:14:25 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
Nov 15, 2013 4:14:25 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["ajp-bio-8009"]
Nov 15, 2013 4:14:25 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 578 ms
Nov 15, 2013 4:14:25 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Catalina
Nov 15, 2013 4:14:25 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/7.0.42
Nov 15, 2013 4:14:25 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 15, 2013 4:14:25 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Nov 15, 2013 4:14:25 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 311 ms
```

Step 2. JSF

1. Add dependencies on **pom.xml** file

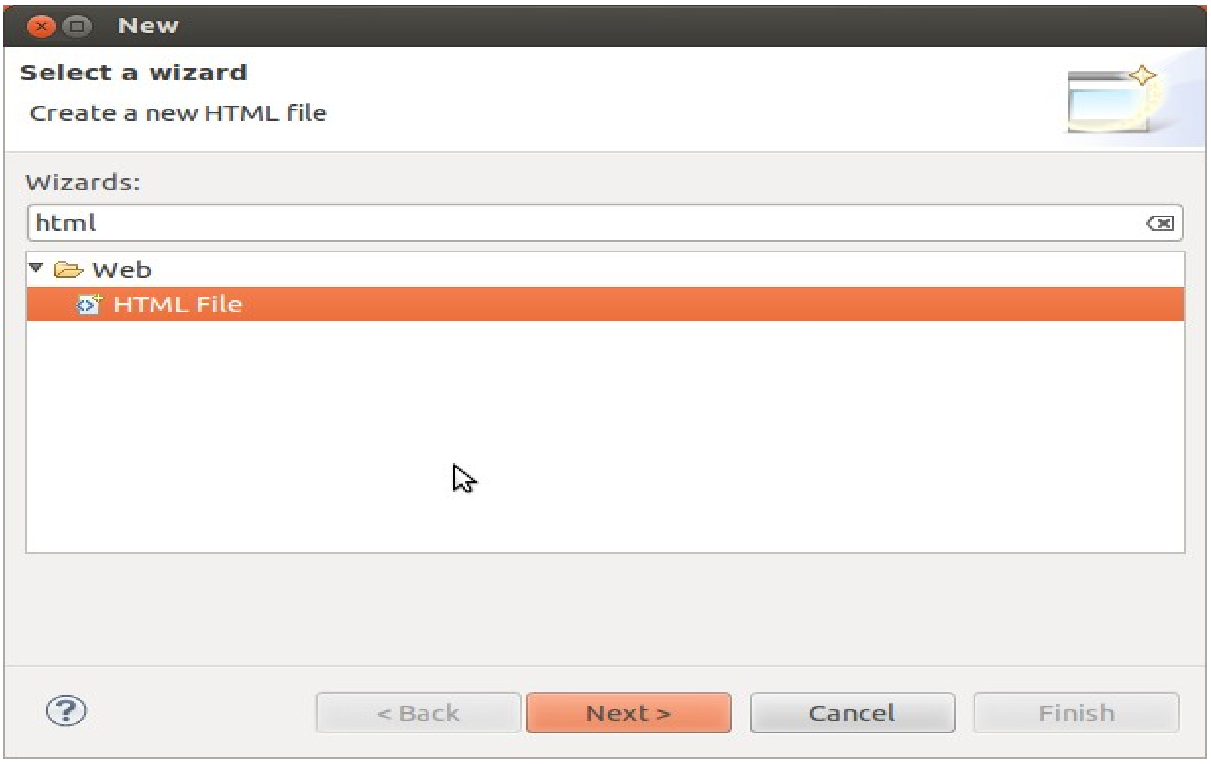
```
<properties>
      <myfaces-version>2.1.12</myfaces-version>
</properties>
…
<dependencies>
      <!-- MyFaces -->
      <dependency>
            <groupId>org.apache.myfaces.core</groupId>
            <artifactId>myfaces-api</artifactId>
            <version>${myfaces-version}</version>
      </dependency>
      <dependency>
            <groupId>org.apache.myfaces.core</groupId>
            <artifactId>myfaces-impl</artifactId>
            <version>${myfaces-version}</version>
      </dependency>
<dependencies>
```

2. Configure web configuration on **web.xml** file

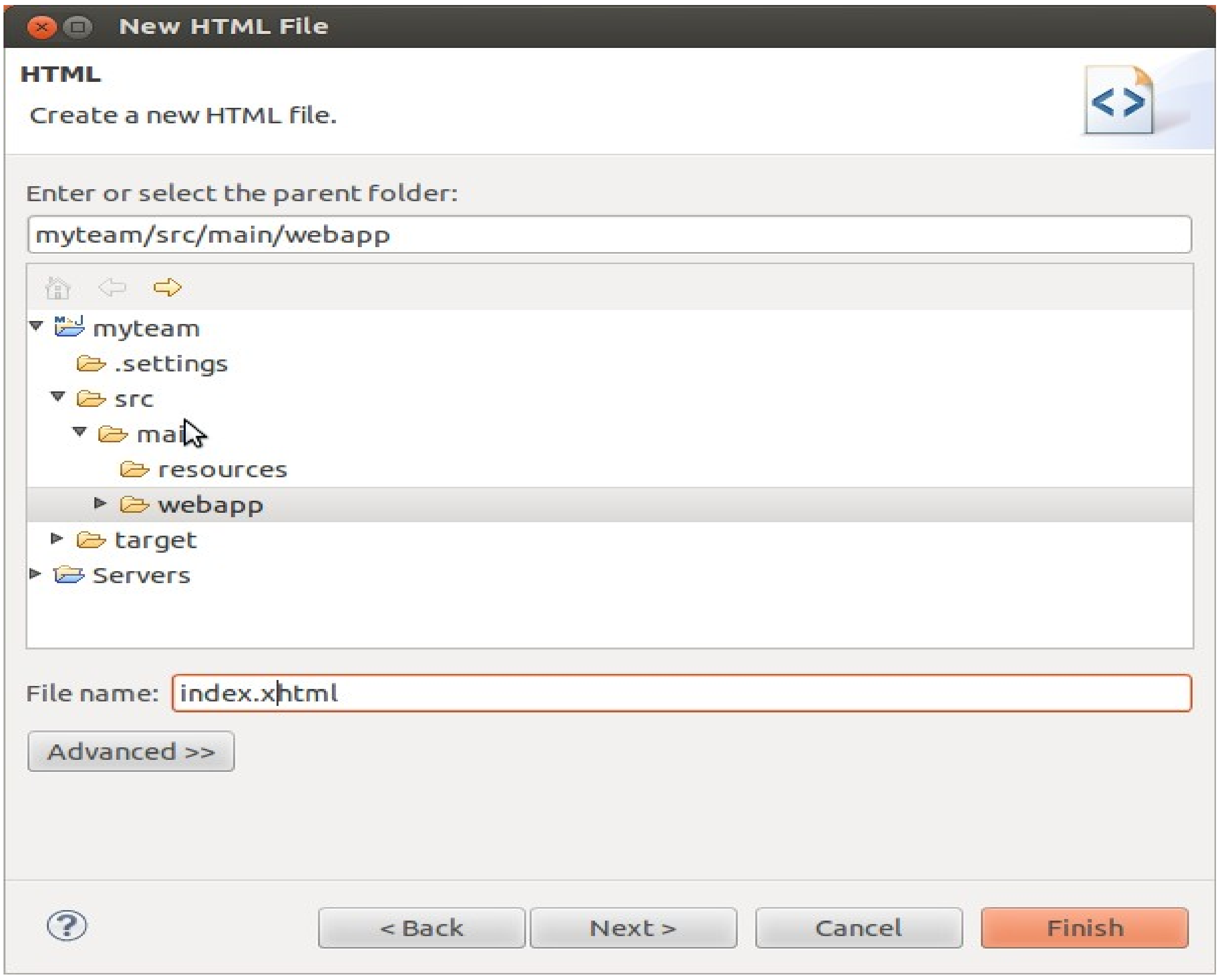
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://java.sun.com/xml/ns/javaee"
      xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
      http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
      version="2.5">

      <!-- JSF mapping -->
      <servlet>
            <servlet-name>Faces Servlet</servlet-name>
            <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
            <load-on-startup>1</load-on-startup>
      </servlet>
      <servlet-mapping>
            <servlet-name>Faces Servlet</servlet-name>
            <url-pattern>*.xhtml</url-pattern>
      </servlet-mapping>

       <!-- welcome page -->
       <welcome-file-list>
         <welcome-file>index.xhtml</welcome-file>
       </welcome-file-list>
</web-app>
```

3. Create welcome file **index.xhtml**

New → Other



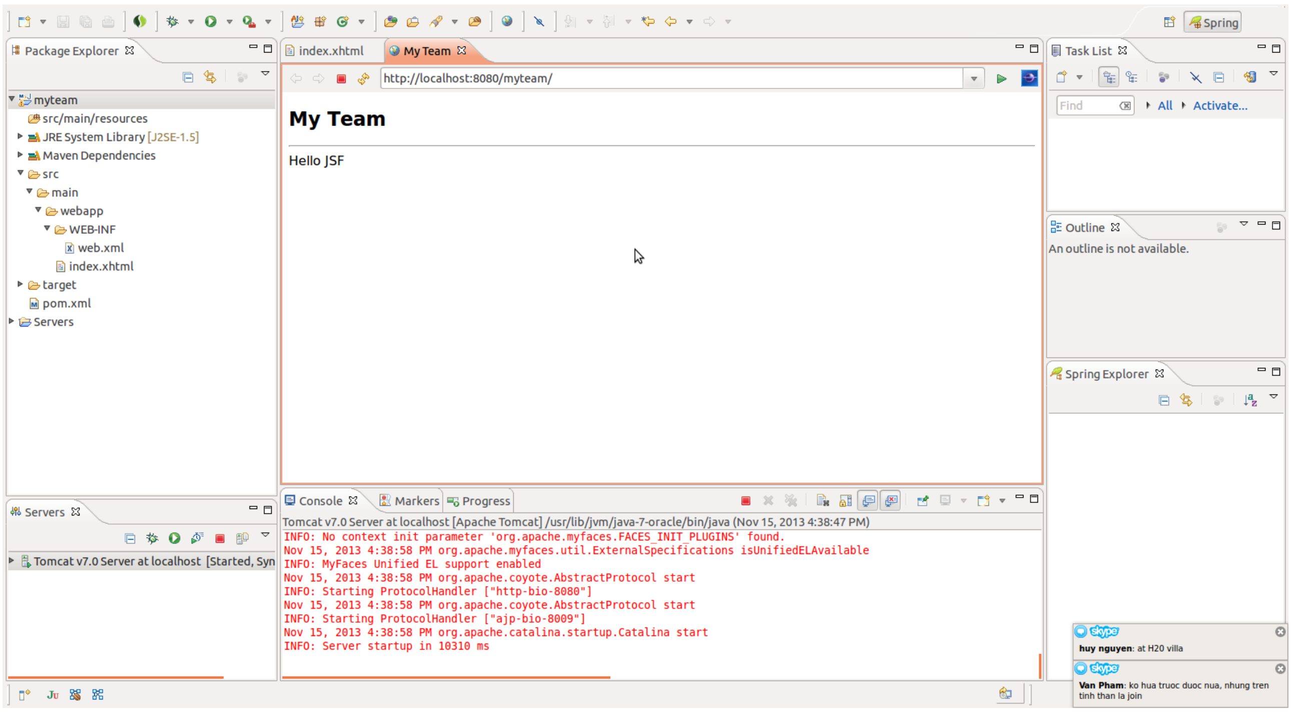
And, . . .



Delete the file **index.jsp** (redundant)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets" >
      <h:head>
            <title>My Team</title>
      </h:head>

      <h:body>
            <h2>My Team</h2>
            <hr/>
            <h:outputText value="Hello JSF"/>
      </h:body>
</html>
```

Test: Run the project!



```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 15, 2013 4:38:47 PM)
INFO: No context init parameter 'org.apache.myfaces.FACES_INIT_PLUGINS' found.
Nov 15, 2013 4:38:58 PM org.apache.myfaces.util.ExternalSpecifications isUnifiedELAvailable
INFO: MyFaces Unified EL support enabled
Nov 15, 2013 4:38:58 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 15, 2013 4:38:58 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Nov 15, 2013 4:38:58 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 10310 ms
```

Step 3. JSF and Primefaces

Add dependency on **pom.xml** file

```
<properties>
	...
	<primefaces-version>3.5</primefaces-version>
</properties>
<repositories>
	<repository>
		<id>prime-repo</id>
		<name>PrimeFaces Maven Repository</name>
		<url>http://repository.primefaces.org</url>
		<layout>default</layout>
	</repository>
</repositories>
…
<!-- Primefaces →
<dependencies>
	<dependency>
		<groupId>org.primefaces</groupId>
		<artifactId>primefaces</artifactId>
		<version>${primefaces-version}</version>
	</dependency>
</dependencies>
```

Update index.xhtml file

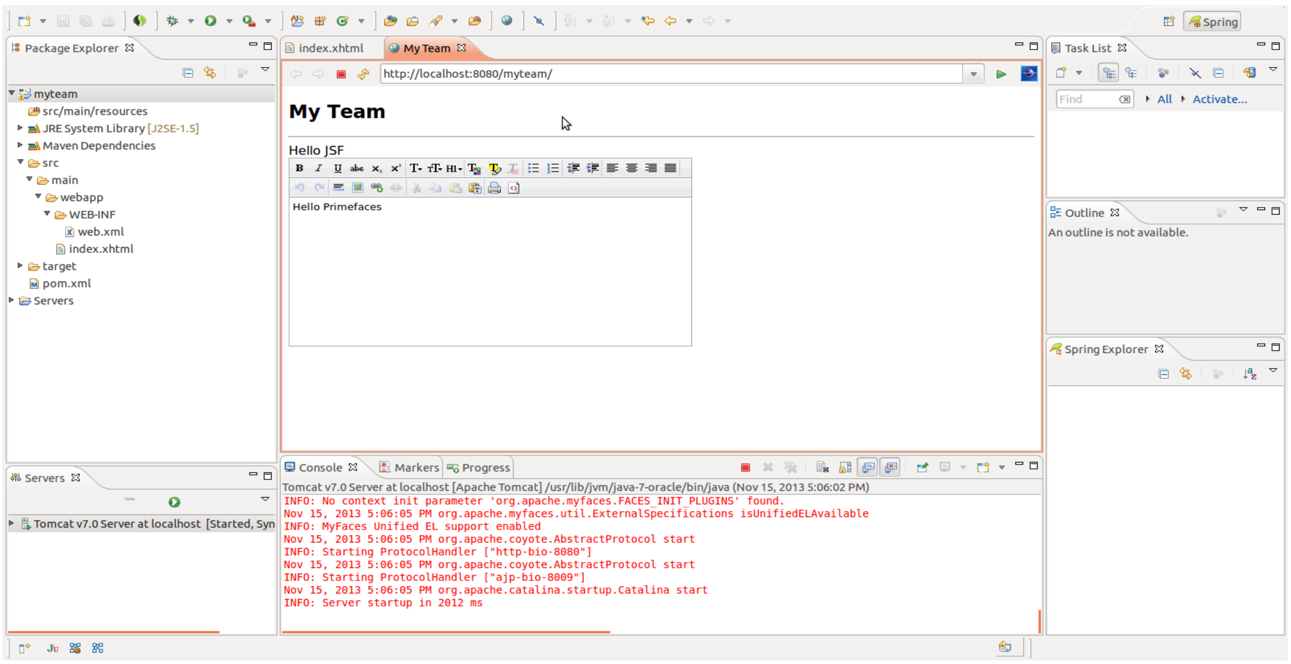
Add namespace

```
xmlns:p="http://primefaces.org/ui" as property of “html tag”
```

Use “p:editor tag” inside “h:body tag”

```
<p:editor value="Hello Primefaces"/>
```

Test: Run the project!



My Team

Hello JSF

Hello Primefaces

An outline is not available.

```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 15, 2013 5:06:02 PM)
INFO: No context init parameter 'org.apache.myfaces.FACES_INIT_PLUGINS' found.
Nov 15, 2013 5:06:05 PM org.apache.myfaces.util.ExternalSpecifications isUnifiedELAvailable
INFO: MyFaces Unified EL support enabled
Nov 15, 2013 5:06:05 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 15, 2013 5:06:05 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Nov 15, 2013 5:06:05 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 2012 ms
```

Right-click on project **New → Folder**, create folder "src/main/java"

Create three classes in the "src/main/java" folder, here:
package: com.ant.myteam.model

Employee.java

```java
package com.ant.myteam.model;

import java.io.Serializable;

public class Employee implements Serializable{
      private static final long serialVersionUID = 1L;

    private Long empId;
      private String firstName;
      private String lastName;
      private String gender;
      private String company;
      private String team;
      private String phone;
      private String job;
      private String imagePath;
      private String email;

      private  Department department;

      public Long getEmpId() {
            return empId;
      }
      public void setEmpId(Long empId) {
            this.empId = empId;
      }
      public String getFirstName() {
            return firstName;
      }
      public void setFirstName(String firstName) {
            this.firstName = firstName;
      }
      public String getLastName() {
            return lastName;
      }
      public void setLastName(String lastName) {
            this.lastName = lastName;
      }
      public String getGender() {
            return gender;
      }
      public void setGender(String gender) {
            this.gender = gender;
      }
      public String getCompany() {
            return company;
      }
      public void setCompany(String company) {
            this.company = company;
      }
      public String getTeam() {
            return team;
      }
      public void setTeam(String team) {
            this.team = team;
```

```java
	}
	public String getPhone() {
		return phone;
	}
	public void setPhone(String phone) {
		this.phone = phone;
	}
	public String getJob() {
		return job;
	}
	public void setJob(String job) {
		this.job = job;
	}
	public String getImagePath() {
		return imagePath;
	}
	public void setImagePath(String imagePath) {
		this.imagePath = imagePath;
	}
	public String getEmail() {
		return email;
	}
	public void setEmail(String email) {
		this.email = email;
	}
	public Department getDepartment() {
		return department;
	}
	public void setDepartment(Department department) {
		this.department = department;
	}

}
```

Deparment.java

```java
package com.ant.myteam.model;

import java.io.Serializable;
import java.util.List;

public class Department  implements Serializable{
	private static final long serialVersionUID = 1L;

	private Long deptId;
	private String depName;
	private List<Employee> employees;

	public Long getDeptId() {
		return deptId;
	}

	public void setDeptId(Long deptId) {
		this.deptId = deptId;
	}

	public String getDepName() {
		return depName;
	}

	public void setDepName(String depName) {
```

```
            this.depName = depName;
      }

      public List<Employee> getEmployees() {
            return employees;
      }

      public void setEmployees(List<Employee> employees) {
            this.employees = employees;
      }



}
```

package: com.ant.myteam.managedbean

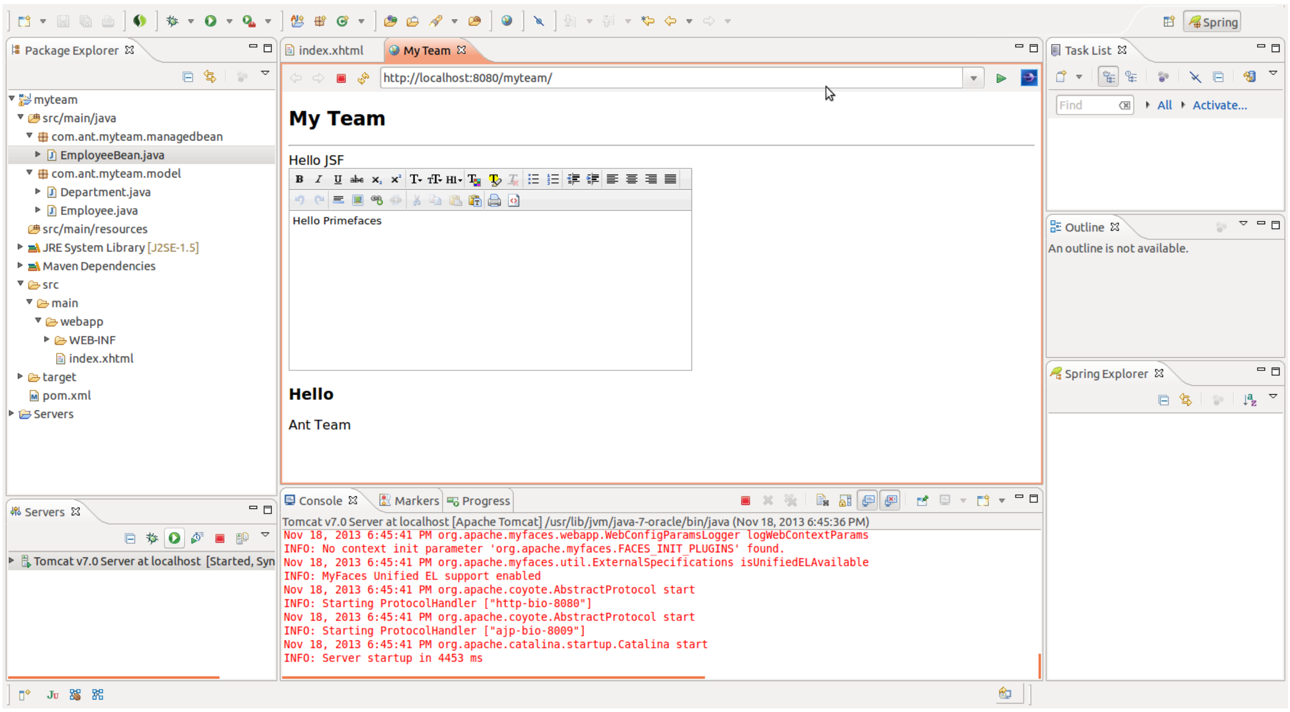
EmployeeBean.java

```
package com.ant.myteam.managedbean;

import java.io.Serializable;

import javax.faces.bean.ManagedBean;

import com.ant.myteam.model.Department;
import com.ant.myteam.model.Employee;

@ManagedBean(name ="empBean")
public class EmployeeBean implements Serializable{

      private static final long serialVersionUID = 1L;

      private Employee employee=new Employee();

      public Employee getEmployee() {
            employee.setEmpId(1L);
            employee.setDepartment(new Department());
            employee.setFirstName("Ant");
            employee.setLastName("Team");
            return employee;
      }
      public void setEmployee(Employee employee) {
            this.employee = employee;
      }
}
```

Update **index.html**

```
<h3>Hello</h3>
<h:outputText value="#{empBean.employee.firstName} #{empBean.employee.lastName}"/>
```

Test: Run the project!



My Team

Hello JSF

Hello Primefaces

Hello

Ant Team

```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 18, 2013 6:45:36 PM)
Nov 18, 2013 6:45:41 PM org.apache.myfaces.webapp.WebConfigParamsLogger logWebContextParams
INFO: No context init parameter 'org.apache.myfaces.FACES_INIT_PLUGINS' found.
Nov 18, 2013 6:45:41 PM org.apache.myfaces.util.ExternalSpecifications isUnifiedELAvailable
INFO: MyFaces Unified EL support enabled
Nov 18, 2013 6:45:41 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 18, 2013 6:45:41 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Nov 18, 2013 6:45:41 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 4453 ms
```

Step 4. JSF and Spring

Add Spring framework dependencies

```
<properties>
      ...
      <org.springframework-version>3.2.3.RELEASE</org.springframework-version>
</properties>

<dependencies>
      <!-- Spring Framework-->
      <!-- Support for JSF -->
      <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-web</artifactId>
            <version>${org.springframework-version}</version>
      </dependency>
</dependencies>
```

Spring configuration on **web.xml**

```
<!-- Add Support for Spring -->
 <listener>
   <listener-class>
           org.springframework.web.context.ContextLoaderListener
   </listener-class>
 </listener>
 <listener>
   <listener-class>
           org.springframework.web.context.request.RequestContextListener
   </listener-class>
</listener>
```

Create WEB_INF/**face-config.xml** file

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
```

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
        version="2.0">
      <!-- JSF and Spring are integrated -->
    <application>
      <el-resolver>
            org.springframework.web.jsf.el.SpringBeanFacesELResolver
        </el-resolver>
    </application>
</faces-config>
```

Create package “com.ant.myteam.service” and two files in this package

EmployeeService.java

```
package com.ant.myteam.service;

import com.ant.myteam.model.Employee;

public interface EmployeeService {

      public Employee findEmployeeById(long empId);

}
```

EmployeeServiceImp.java

```
package com.ant.myteam.service;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

import com.ant.myteam.model.Employee;

@Service
public class EmployeeServiceImpl implements EmployeeService,Serializable {

      private static final long serialVersionUID = 1L;

      private List<Employee> empList=new ArrayList<Employee>();

      public EmployeeServiceImpl(){
            Employee emp1 = new Employee();
            emp1.setEmpId(1L);
            emp1.setFirstName("Huong");
            emp1.setLastName("Nguyen");

            Employee emp2 = new Employee();
            emp2.setEmpId(2L);
            emp2.setFirstName("Khang");
            emp2.setLastName("Le");

            empList.add(emp1);
            empList.add(emp2);
      }

      public Employee findEmployeeById(long empId) {
            for(Employee emp: empList){
                  if(emp.getEmpId()==empId){
                        return emp;
                  }
            }
            return null;
      }

}
```

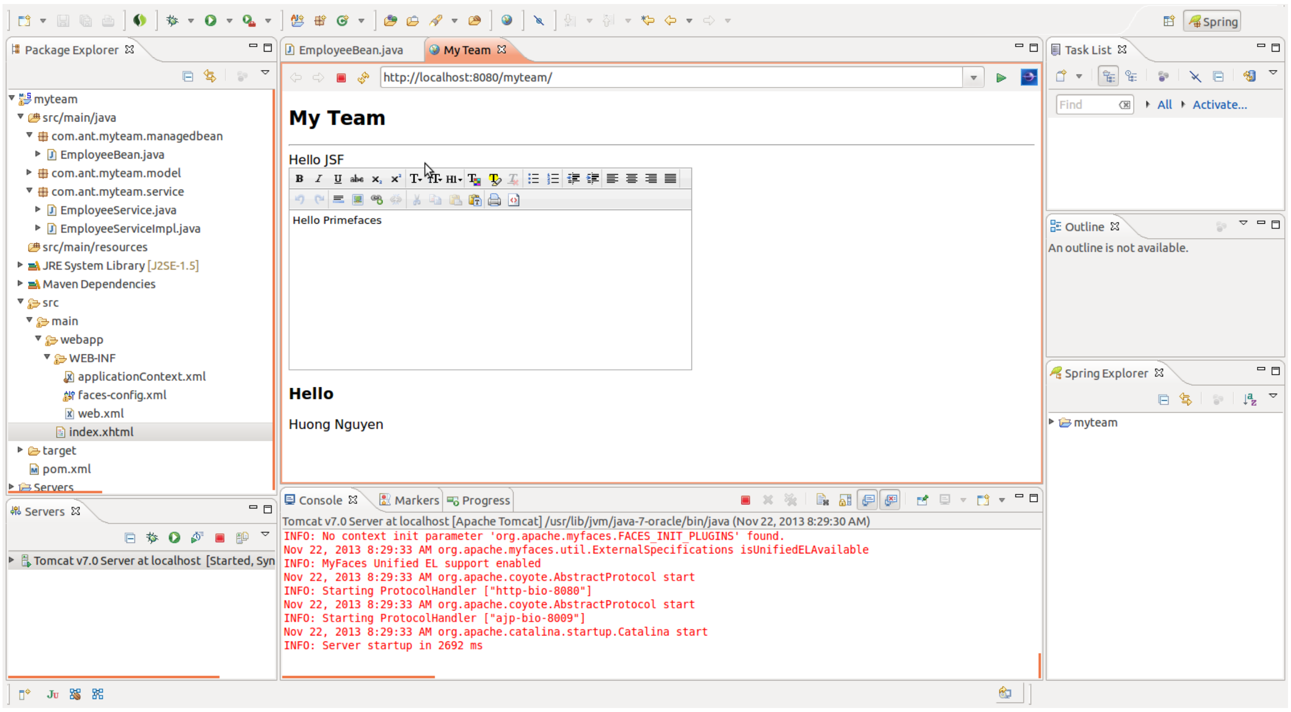
EmployeeBean.java

```
package com.ant.myteam.managedbean;

import java.io.Serializable;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

import com.ant.myteam.model.Employee;
import com.ant.myteam.service.EmployeeService;

@Component("empBean")
public class EmployeeBean implements Serializable{

      private static final long serialVersionUID = 1L;

      private Employee employee=new Employee();

      @Autowired
      private EmployeeService empService;

      public Employee getEmployee() {
            employee= empService.findEmployeeById(1L);
            return employee;
      }
      public void setEmployee(Employee employee) {
            this.employee = employee;
      }
}
```

Create WEB-INF/**applicationContext.xml** file

```
<?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:tx="http://www.springframework.org/schema/tx"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="http://www.springframework.org/schema/beans


      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd


      http://www.springframework.org/schema/tx


      http://www.springframework.org/schema/tx/spring-tx-3.0.xsd


      http://www.springframework.org/schema/context


      http://www.springframework.org/schema/context/spring-context-3.0.xsd">
      <!-- Enable autowire -->
      <context:annotation-config />

      <!-- Enable component scanning -->
      <context:component-scan base-package="com.ant.myteam" />

</beans>
```

Test: Run the project!



My Team

Hello JSF

Hello Primefaces

Hello

Huong Nguyen

```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 22, 2013 8:29:30 AM)
INFO: No context init parameter 'org.apache.myfaces.FACES_INIT_PLUGINS' found.
Nov 22, 2013 8:29:33 AM org.apache.myfaces.util.ExternalSpecifications isUnifiedELAvailable
INFO: MyFaces Unified EL support enabled
Nov 22, 2013 8:29:33 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Nov 22, 2013 8:29:33 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Nov 22, 2013 8:29:33 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 2692 ms
```

Step 5. JSF, Spring and Hibernate

Add dependencies

JDBCs and Hibernate API

```
<properties>
      ...
      <hibernate-version>4.1.0.Final</hibernate-version>
</propertise>
```

```
<!-- PostgreSQL JDBC Driver -->
<dependency>
      <groupId>postgresql</groupId>
      <artifactId>postgresql</artifactId>
      <version>9.1-901.jdbc4</version>
</dependency>
<!-- Apache DBCP Library (manage connection to data source) -->
<dependency>
      <groupId>commons-dbcp</groupId>
      <artifactId>commons-dbcp</artifactId>
      <version>1.4</version>
</dependency>
<!-- Hibernate -->
<dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>${hibernate-version}</version>
</dependency>
```

Spring ORM framework support for integrated with Hibernate

```
      <!-- Integration with Hibernate -->
      <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-orm</artifactId>
            <version>${org.springframework-version}</version>
      </dependency>
```

Configure in applicationContext.xml, add these lines

```
<!-- Data Source Declaration -->
<bean id="myDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
      <property name="driverClassName" value="org.postgresql.Driver"/>
      <property name="url" value="jdbc:postgresql://localhost:5432/myteam"/>
      <property name="username" value="postgres"/>
      <property name="password" value="postgres"/>
</bean>

  <!-- Session Factory Declaration -->
 <bean id="mySessionFactory"
class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
      <property name="dataSource" ref="myDataSource"/>
      <property name="packagesToScan" value="com.ant.myteam.model" />
       <property name="hibernateProperties">
            <props>
                  <prop
key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop>
                  <prop key="hibernate.show_sql">true</prop>
                  <prop key="hibernate.enable_lazy_load_no_trans">true</prop>
                  <prop key="hibernate.default_schema">myteam</prop>
                  <prop key="hibernate.hbm2ddl.auto">create</prop>
            </props>
       </property>
 </bean>

  <!-- Enable the configuration of transactional behavior based on annotations -->
<tx:annotation-driven transaction-manager="transactionManager"/>
<!-- Transaction Manager is defined -->
<bean id="transactionManager"
class="org.springframework.orm.hibernate4.HibernateTransactionManager">
      <property name="sessionFactory" ref="mySessionFactory"/>
</bean>
```

modify model classes:

Employee.java

```
package com.ant.myteam.model;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
```

```
@Entity
@Table(name = "Employee")
public class Employee implements Serializable{
      private static final long serialVersionUID = 1L;

      @Id
      @Column(name="id")
      @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private Long empId;
      @Column(nullable = false)
      private String firstName;
      @Column(nullable = false)
      private String lastName;
      private String gender;
      private String company;
      private String team;
      private String phone;
      private String job;
      private String imagePath;
      private String email;

      @ManyToOne
    @JoinColumn(name = "deptId")
      private  Department department;

      public Long getEmpId() {
            return empId;
      }
      public void setEmpId(Long empId) {
            this.empId = empId;
      }
      public String getFirstName() {
            return firstName;
      }
      public void setFirstName(String firstName) {
            this.firstName = firstName;
      }
      public String getLastName() {
            return lastName;
      }
      public void setLastName(String lastName) {
            this.lastName = lastName;
      }
      public String getGender() {
            return gender;
      }
      public void setGender(String gender) {
            this.gender = gender;
      }
      public String getCompany() {
            return company;
      }
      public void setCompany(String company) {
            this.company = company;
      }
      public String getTeam() {
            return team;
      }
      public void setTeam(String team) {
            this.team = team;
      }
      public String getPhone() {
            return phone;
      }
      public void setPhone(String phone) {
            this.phone = phone;
      }
      public String getJob() {
            return job;
      }
      public void setJob(String job) {
            this.job = job;
      }
      public String getImagePath() {
            return imagePath;
      }
      public void setImagePath(String imagePath) {
            this.imagePath = imagePath;
      }
      public String getEmail() {
            return email;
      }
      public void setEmail(String email) {
            this.email = email;
      }
      public Department getDepartment() {
            return department;
      }
      public void setDepartment(Department department) {
            this.department = department;
      }

}
```

Department.java

```
package com.ant.myteam.model;

import java.io.Serializable;
import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Department")
public class Department  implements Serializable{
      private static final long serialVersionUID = 1L;

      @Id
      @Column(name="id")
      @GeneratedValue(strategy = GenerationType.SEQUENCE)
      private Long deptId;

      @Column(nullable = false)
      private String depName;
      @OneToMany(cascade = CascadeType.ALL)
    @JoinColumn(name = "deptId")
      private List<Employee> employees;

      public Long getDeptId() {
            return deptId;
      }

      public void setDeptId(Long deptId) {
            this.deptId = deptId;
      }

      public String getDepName() {
            return depName;
      }

      public void setDepName(String depName) {
            this.depName = depName;
      }

}
```

create package: “com.ant.myteam.dao” with two files

EmployeeDao.java

```
package com.ant.myteam.dao;

import com.ant.myteam.model.Employee;

public interface EmployeeDao {

      public boolean addEmployee(Employee emp);

      public Employee findEmployeeById(long empId);

}
```

EmployeeDaoImpl.java

```
package com.ant.myteam.dao;

import java.io.Serializable;

import org.hibernate.SessionFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import com.ant.myteam.model.Employee;

@Repository
@Transactional
public class EmployeeDaoImpl implements EmployeeDao, Serializable{

      private static final long serialVersionUID = 1L;

      @Autowired
      private SessionFactory sessionFactory;
      public boolean addEmployee(Employee emp) {
            try {
                   sessionFactory.getCurrentSession().save(emp);
                   return true;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return false;
      }


      public Employee findEmployeeById(long empId) {
            Employee result = new Employee();
            try {
                  result=(Employee)
      sessionFactory.getCurrentSession().get(Employee.class, empId);
                   return result;
            } catch (Exception e) {
                  e.printStackTrace();
            }
            return result;
      }

}
```

Modify EmployeeServiceImpl.java

```
package com.ant.myteam.service;

import java.io.Serializable;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.ant.myteam.dao.EmployeeDao;
import com.ant.myteam.model.Employee;

@Service
public class EmployeeServiceImpl implements EmployeeService,Serializable {

      private static final long serialVersionUID = 1L;

      @Autowired
      private EmployeeDao empDao;

      public Employee findEmployeeById(long empId) {
            return empDao.findEmployeeById(empId);
      }

      public boolean addEmployee(Employee emp) {
            return empDao.addEmployee(emp);
      }

}
```

Modify EmployeeBean.java

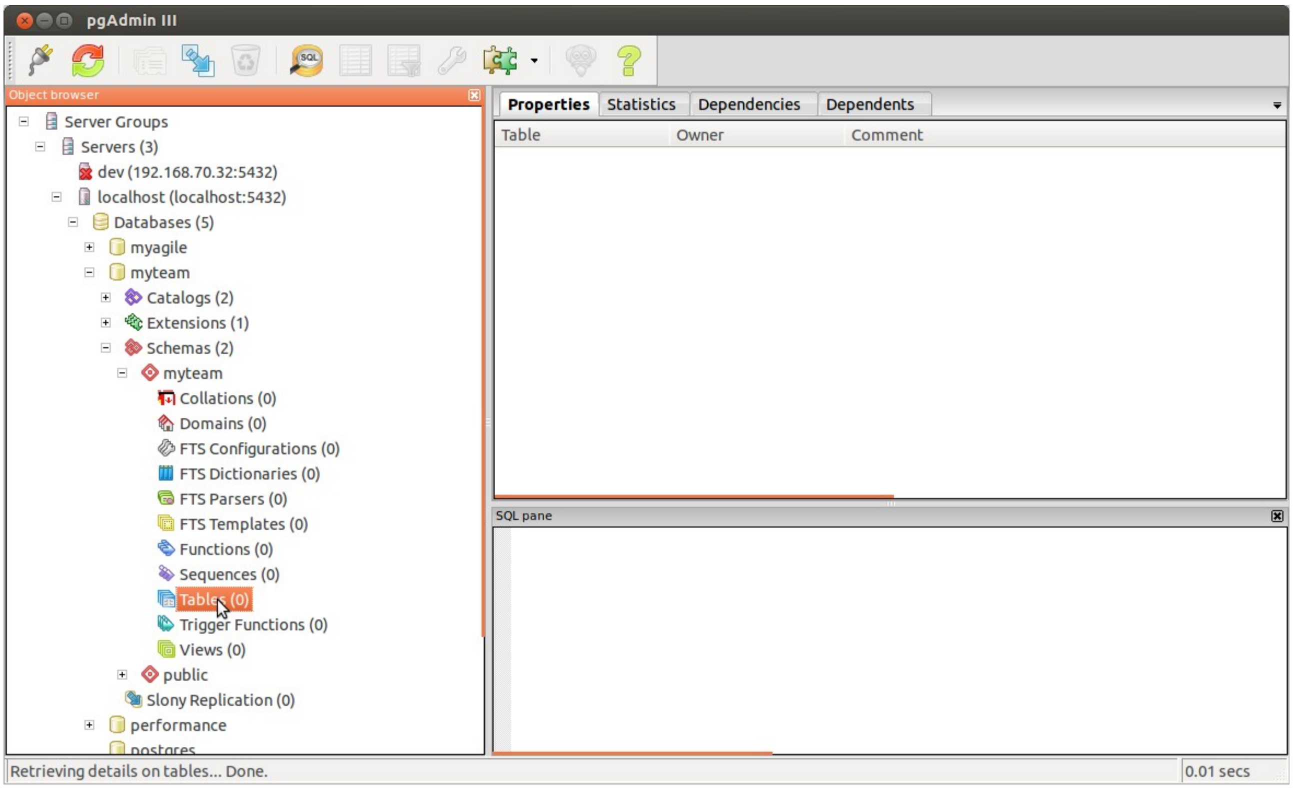
```
package com.ant.myteam.managedbean;

import java.io.Serializable;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

import com.ant.myteam.model.Employee;
import com.ant.myteam.service.EmployeeService;

@Component("empBean")
public class EmployeeBean implements Serializable{

	private static final long serialVersionUID = 1L;

	private Employee employee=new Employee();

	@Autowired
	private EmployeeService empService;

	private Employee emp1;
	private Employee emp2;

	public EmployeeBean(){
		emp1 = new Employee();
		emp1.setFirstName("Huong");
		emp1.setLastName("Nguyen");

		emp2 = new Employee();
		emp2.setFirstName("Khang");
		emp2.setLastName("Le");
	}

	public void addEmployee(){
		empService.addEmployee(emp1);
		empService.addEmployee(emp2);
		employee= empService.findEmployeeById(emp1.getEmpId());
	}
	public Employee getEmployee() {
		return employee;
	}
	public void setEmployee(Employee employee) {
		this.employee = employee;
	}
}
```

index.xhtml

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
	xmlns:f="http://java.sun.com/jsf/core"
	xmlns:h="http://java.sun.com/jsf/html"
	xmlns:ui="http://java.sun.com/jsf/facelets"
	xmlns:p="http://primefaces.org/ui">
	<h:head>
		<title>My Team</title>
	</h:head>

	<h:body>
		<h2>My Team</h2>
		<hr/>
		<h:outputText value="Hello JSF"/>
		<p:editor value="Hello Primefaces"/>
		<h:form id="empForm">
			<p:commandButton value="Add default"
action="#{empBean.addEmployee}" update="empForm"/>
			<h3>Hello</h3>
```

```
                  <h:outputText id="employeeId"
value="#{empBean.employee.firstName} #{empBean.employee.lastName}"/>
            </h:form>

      </h:body>
</html>
```

create database and schemas with name “myteam”



pgAdmin III

Object browser

```
Server Groups
  Servers (3)
    dev (192.168.70.32:5432)
    localhost (localhost:5432)
      Databases (5)
        myagile
        myteam
          Catalogs (2)
          Extensions (1)
          Schemas (2)
            myteam
              Collations (0)
              Domains (0)
              FTS Configurations (0)
              FTS Dictionaries (0)
              FTS Parsers (0)
              FTS Templates (0)
              Functions (0)
              Sequences (0)
              Tables (0)
              Trigger Functions (0)
              Views (0)
            public
          Slony Replication (0)
        performance
        postgres
```

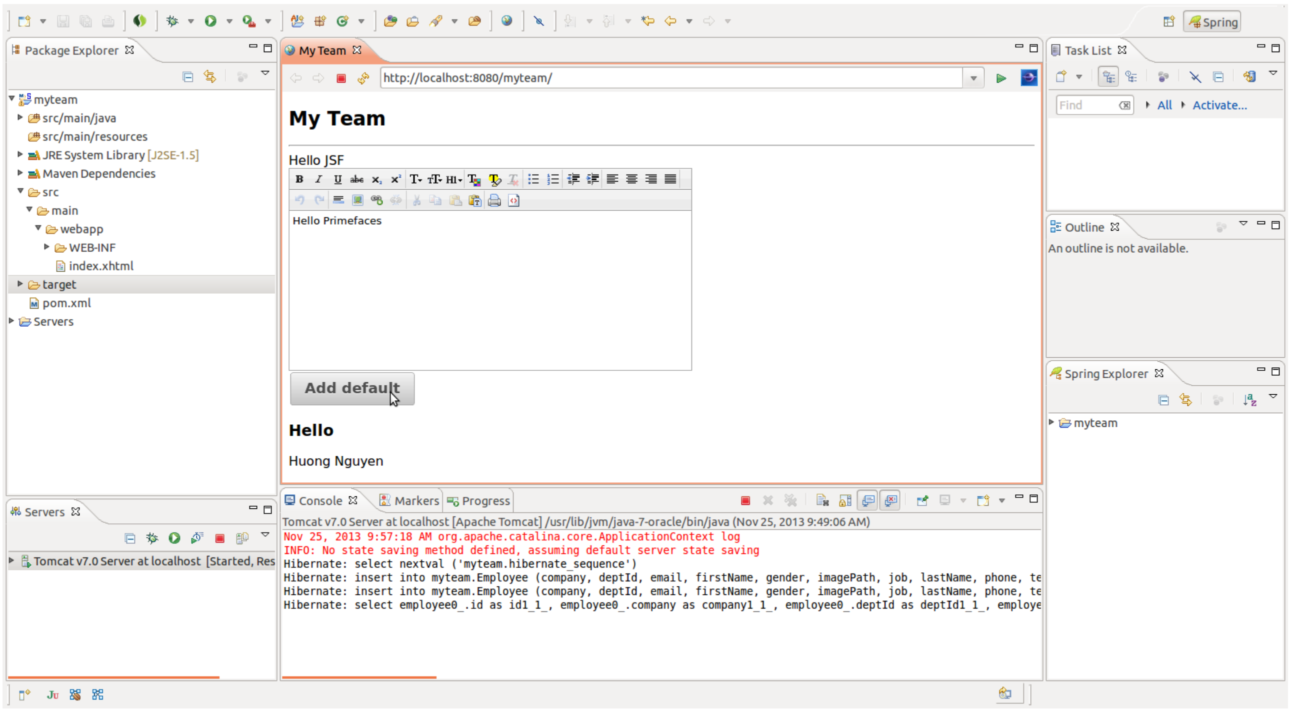
Properties | Statistics | Dependencies | Dependents

Table	Owner	Comment

SQL pane

Retrieving details on tables... Done. 0.01 secs

Test: Run the project!



Package Explorer

```
myteam
  src/main/java
  src/main/resources
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  src
    main
      webapp
        WEB-INF
        index.xhtml
  target
  pom.xml
Servers
```

Servers

Tomcat v7.0 Server at localhost [Started, Res

My Team — http://localhost:8080/myteam/

My Team

Hello JSF

Hello Primefaces

Add default

Hello

Huong Nguyen

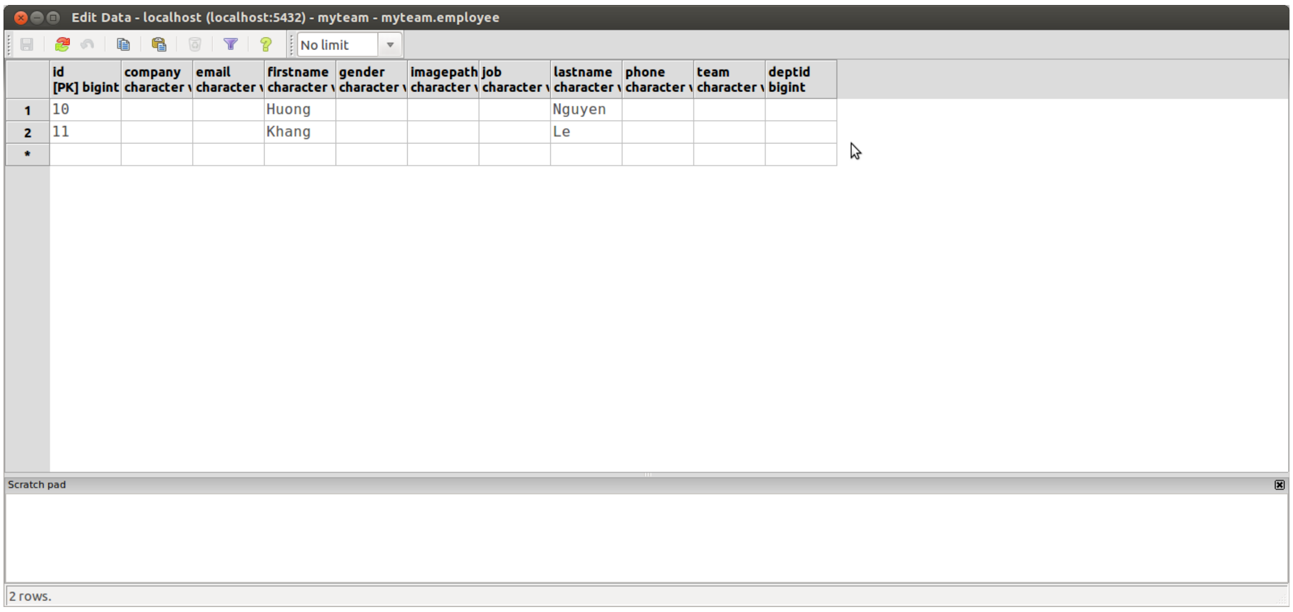
Task List — Find — All — Activate...

Outline — An outline is not available.

Spring Explorer — myteam

Console | Markers | Progress

```
Tomcat v7.0 Server at localhost [Apache Tomcat] /usr/lib/jvm/java-7-oracle/bin/java (Nov 25, 2013 9:49:06 AM)
Nov 25, 2013 9:57:18 AM org.apache.catalina.core.ApplicationContext log
INFO: No state saving method defined, assuming default server state saving
Hibernate: select nextval ('myteam.hibernate_sequence')
Hibernate: insert into myteam.Employee (company, deptId, email, firstName, gender, imagePath, job, lastName, phone, te
Hibernate: insert into myteam.Employee (company, deptId, email, firstName, gender, imagePath, job, lastName, phone, te
Hibernate: select employee0_.id as id1_1_, employee0_.company as company1_1_, employee0_.deptId as deptId1_1_, employe
```


Edit Data - localhost (localhost:5432) - myteam - myteam.employee

No limit

	id [PK] bigint	company character v	email character v	firstname character v	gender character v	imagepath character v	job character v	lastname character v	phone character v	team character v	deptid bigint
1	10			Huong				Nguyen			
2	11			Khang				Le			
*											

Scratch pad

2 rows.