

Java Web

Môn: Phát triển ứng dụng hệ thống thông tin
hiện đại

GV: Ths. Phạm Minh Tú



Nội dung

- **Giới thiệu**
- Servlet
- JSP
- Maven
- Hibernate
- Spring MVC



Giới thiệu

- Java hỗ trợ phát triển các ứng dụng web với các công nghệ:
 - Servlet
 - JSP
 - JSF
 - JSM
 - Java Mail
 -



Giới thiệu

- Các IDE hỗ trợ phát triển web dùng Java như:
 - Netbean
 - Eclipse
 - IntelliJ
 -



Servlet

- Là một chương trình chạy trên web server, chịu trách nhiệm xử lý các request của người dùng và phản hồi kết quả
- Servlet chạy trên nhiều phần mềm web server như:
 - Tomcat
 - Glassfish
 - Jboss
 -



Servlet

- Cài đặt servlet bằng cách cài đặt các chương trình web server.
- Ví dụ:
 - IDE: IntelliJ
 - Web Server: JBoss



Servlet

- Bước 1:

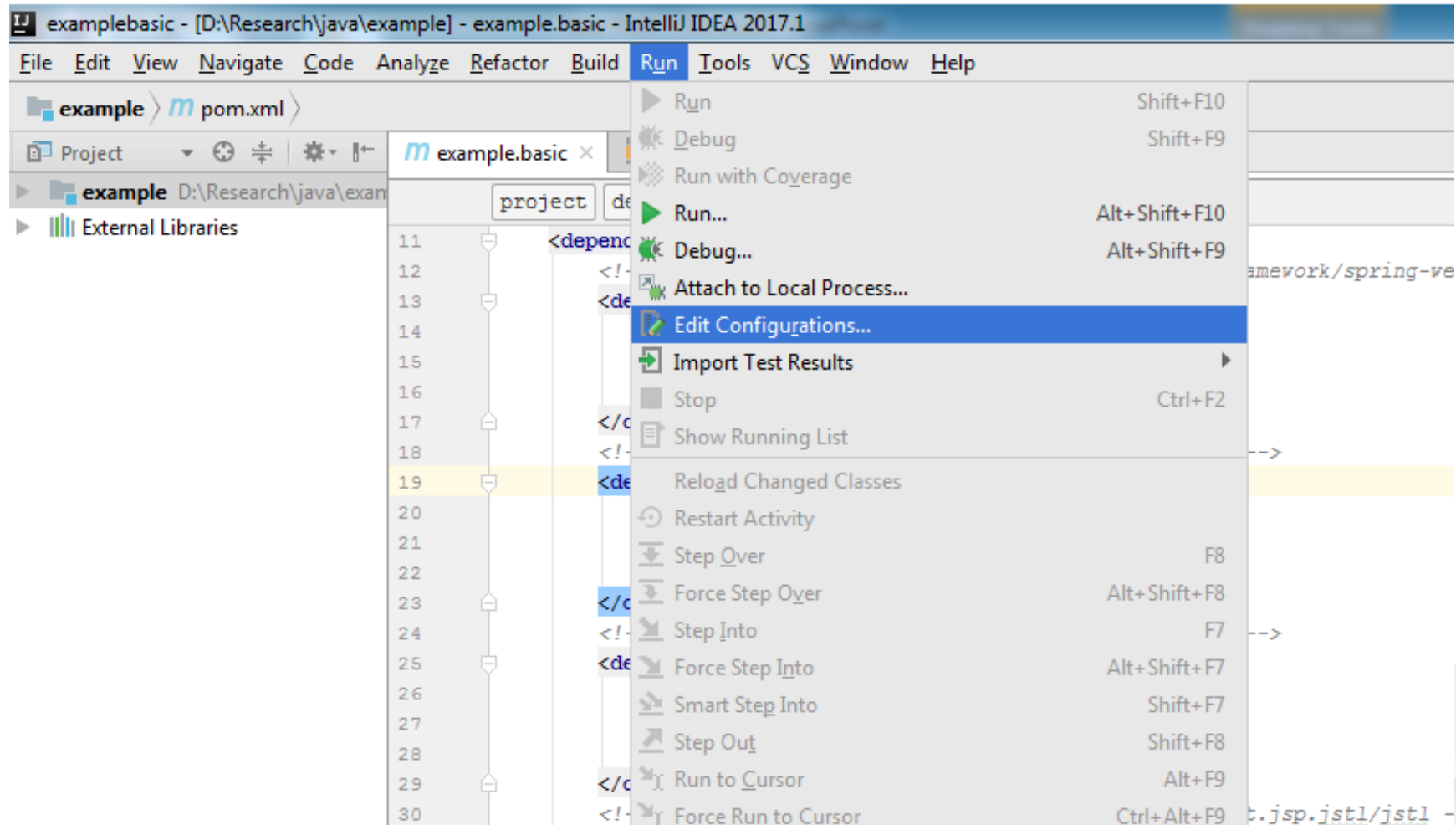
- Tải IDE và Web Server về máy.
- IDE IntelliJ cài đặt bình thường
- Jboss giải nén và đặt trong một thư mục

- Bước 2:

- Khởi động IDE IntelliJ
- Khai báo đường dẫn đến phần mềm Web server JBoss

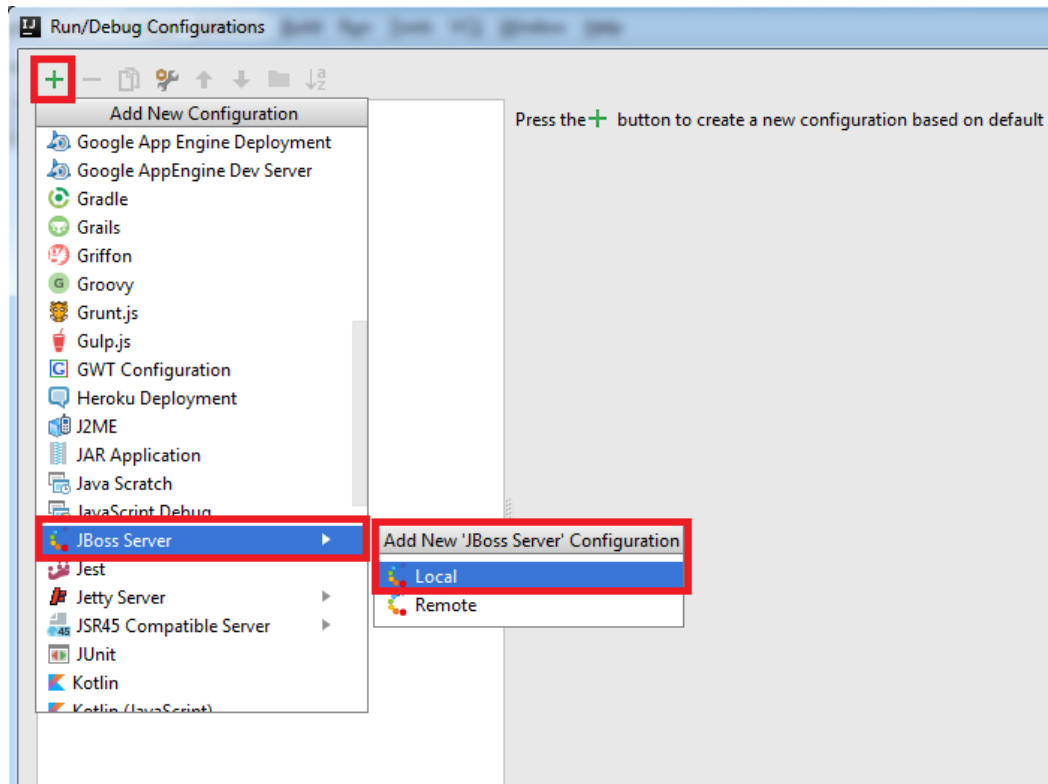


Servlet



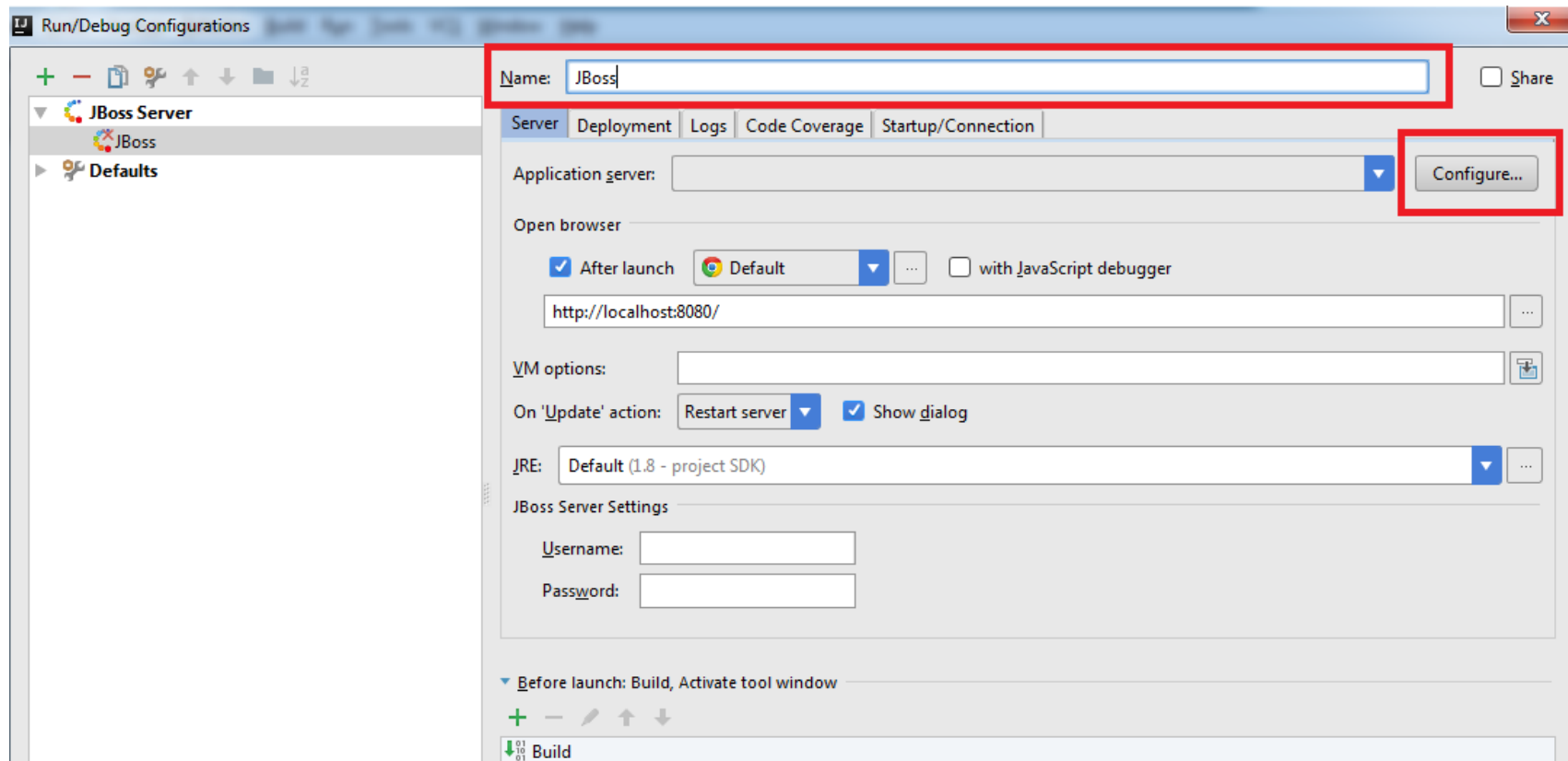
Servlet

- Thêm một web server Jboss



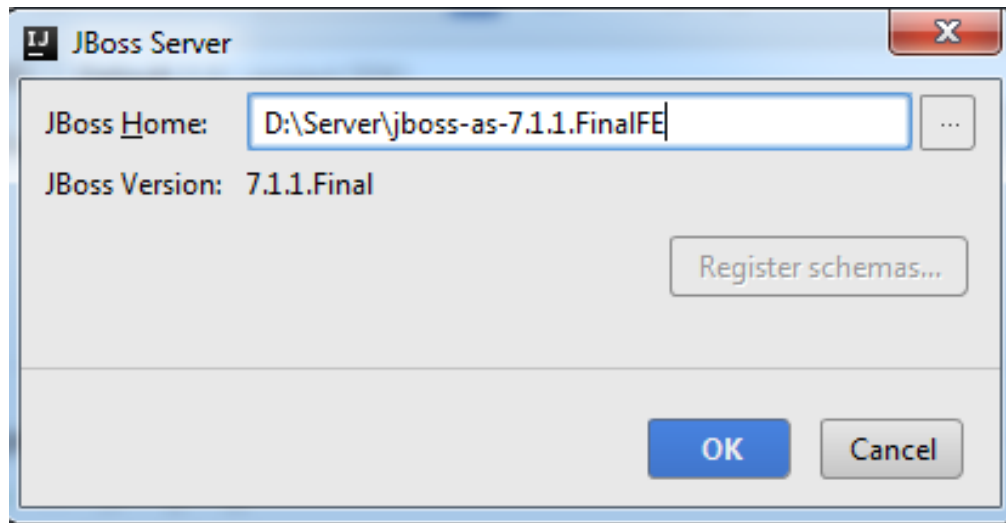
Servlet

- Đặt tên và chọn đường dẫn web server



Servlet

- Chọn thư mục Jboss



Servlet

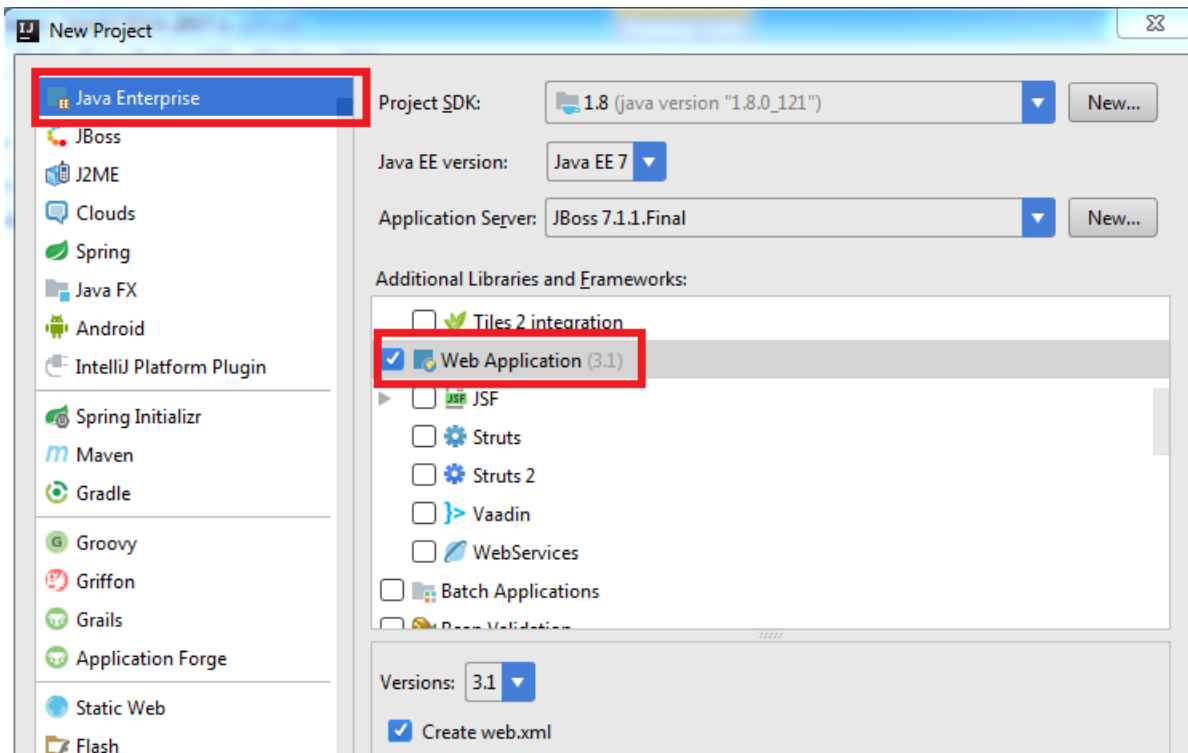
- Bước 4:

- Tạo một project đặt tên: HelloWorld
- Thêm một trang HTML hỗ trợ người dùng gửi request đến chương trình servlet
- Servlet nhận request và xử lý kết quả trả về người dùng



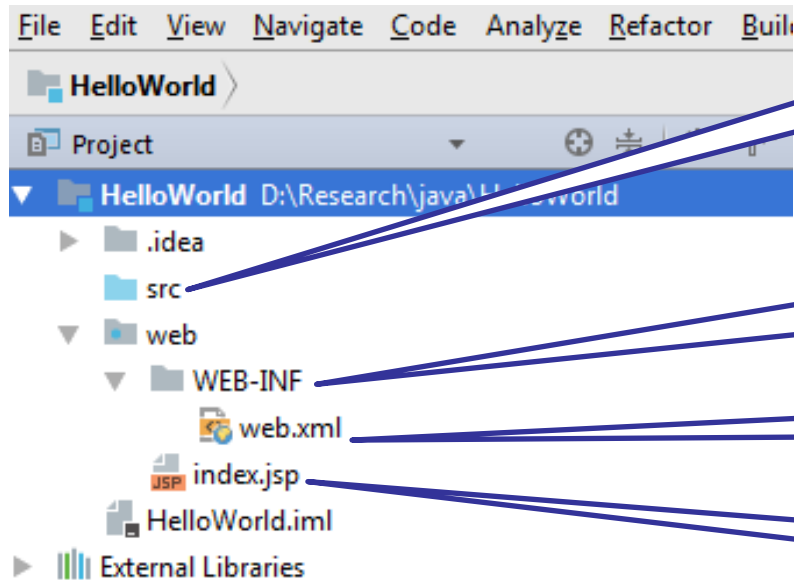
Servlet

- Từ giao diện chính IntelliJ, chọn File -> New Project
- Chọn ứng dụng web application -> Đặt tên project



Servlet

•Cấu trúc Project



Chứa các file java, bao gồm các file servlet

Chứa các file cấu hình, các tập tin web không cho người dùng truy cập trực tiếp

Tập tin cấu hình web

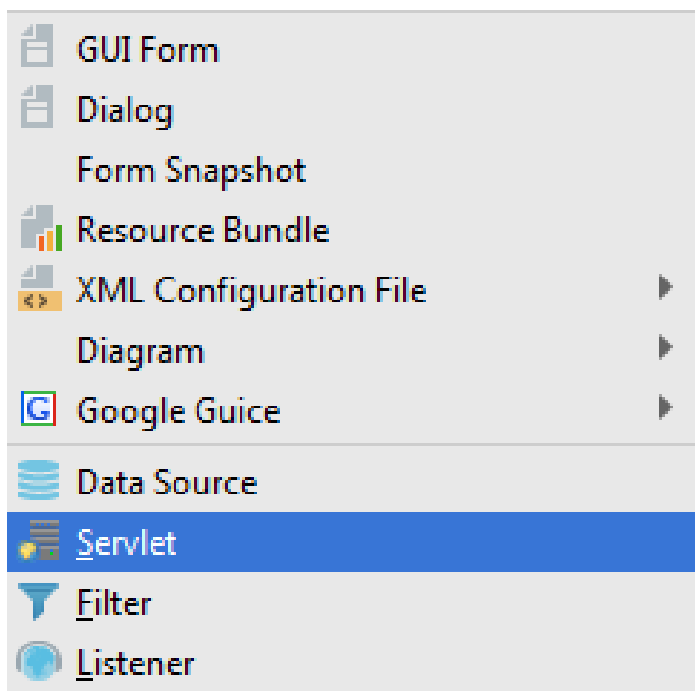
Tập tin web như: HTML, JSP,...



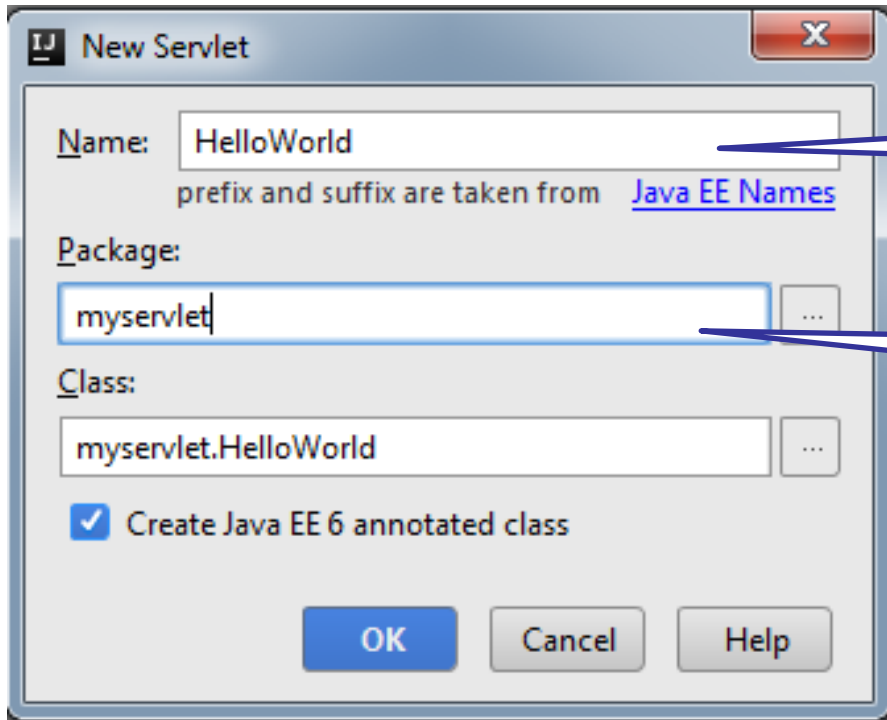
Servlet

- Bước 5: Tạo một servlet

- Click phải thư mục src -> Chọn New -> Servlet



Servlet



New Servlet

Name: HelloWorld
prefix and suffix are taken from [Java EE Names](#)

Package: myservlet

Class: myservlet.HelloWorld

☒ Create Java EE 6 annotated class

OK Cancel Help

Đặt tên servlet

Đặt tên package



Servlet

- Tập tin servlet được phát sinh 2 phương thức mặc định:

–**protected void** doPost(HttpServletRequest request,
HttpServletResponse response) **throws** ServletException,
IOException {

}

protected void doGet(HttpServletRequest request,
HttpServletResponse response) **throws** ServletException,
IOException {

}



Servlet

- Tập tin servlet được phát sinh 2 phương thức mặc định:
 - protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

}
=> Nhận xử lý request từ người dùng gửi bằng phương thức POST



Servlet

- Tập tin servlet được phát sinh 2 phương thức mặc định:

```
–protected void doGet(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException,  
    IOException {  
  
    }
```

=> Nhận xử lý request từ người dùng gửi bằng phương thức
GET



Servlet

- Lưu ý cả 2 phương thức này đều gọi phương thức:
public void service(ServletRequest req, ServletResponse res)
throws ServletException, IOException {
 super.service(req, res);
}
- Vì vậy có thể override phương thức này và nhận xử lý request từ người dùng gửi cả 2 phương thức: GET, POST



Servlet

- Nhấn tổ hợp phím Alt + insert
- Chọn Override Method

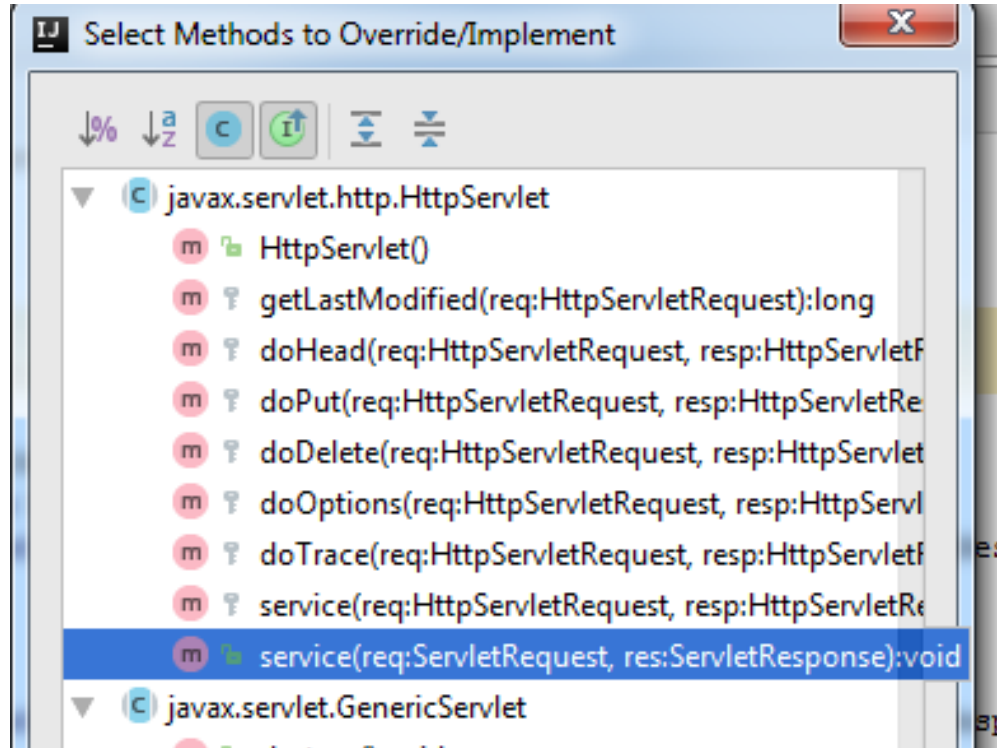
```
import ...  
  
/**  
 * Created by home on 4/9/2017.  
 */  
@WebServlet(name = "HelloWorld")  
public class HelloWorld extends HttpServlet {  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    }  
  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    }  
}  
  
}
```

Generate	
Constructor	
toString()	
Override Methods...	Ctrl+O
Delegate Methods...	
Copyright	



Servlet

- Chọn phương thức service



Servlet

- Viết lệnh xử lý request từ người dùng, và trả kết quả về chuỗi HTML

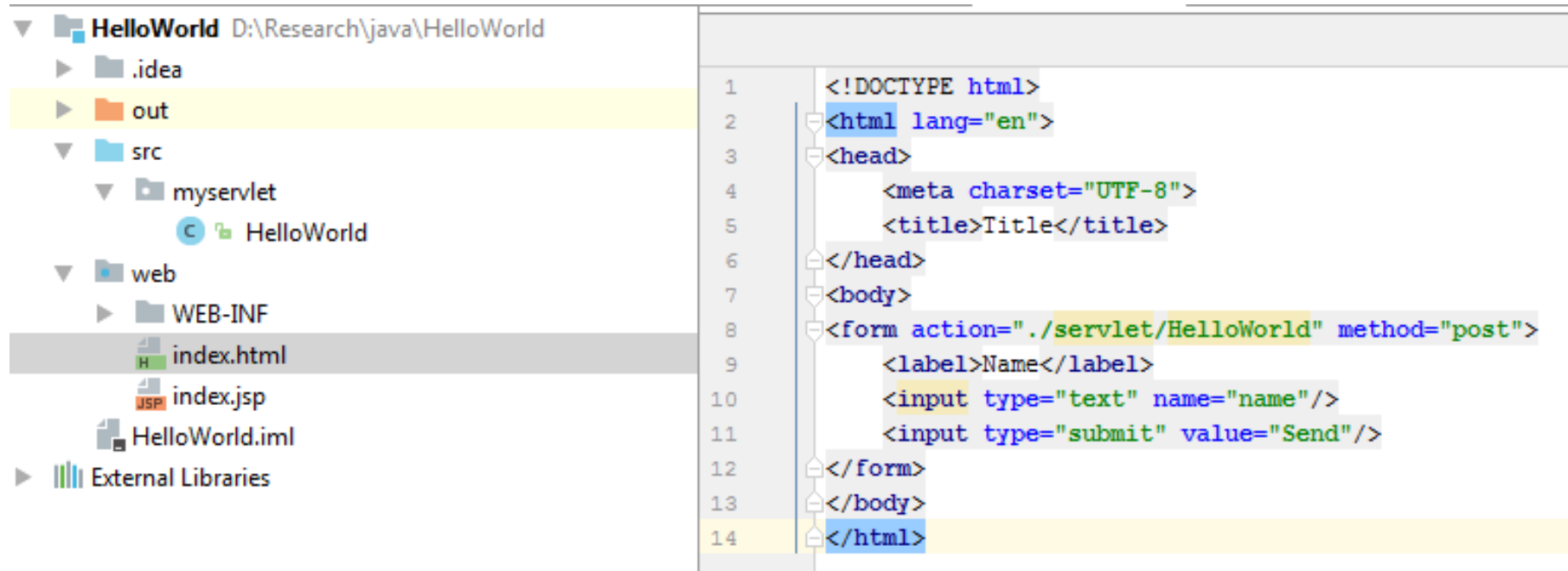
- @Override

```
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {  
    String param = req.getParameter("name");  
    PrintWriter w=res.getWriter();  
    StringBuilder str=new StringBuilder();  
    str.append("<html><head></head><body><h3>");  
    if (param != null){  
        str.append("Hello World " + param);  
    }  
    else{  
        str.append("Hello World");  
    }  
    str.append("</h3></body></html>");  
    w.print(str.toString());  
}
```



Servlet

- Tạo file HTML



The screenshot displays an IDE interface. On the left, a project tree for 'HelloWorld' is shown with the following structure:

- ▼ HelloWorld D:\Research\java\HelloWorld
 - ▶ .idea
 - ▶ out
 - ▼ src
 - ▼ myervlet
 - HelloWorld
 - ▼ web
 - ▶ WEB-INF
 - index.html
 - index.jsp
 - HelloWorld.iml
 - ▶ External Libraries

The 'index.html' file is selected and its content is displayed in the editor on the right:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8     <form action="./servlet/HelloWorld" method="post">
9         <label>Name</label>
10        <input type="text" name="name"/>
11        <input type="submit" value="Send"/>
12    </form>
13 </body>
14 </html>
```


Servlet

- Bước 5:

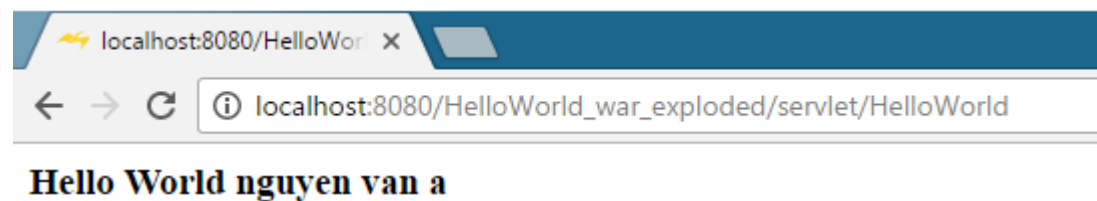
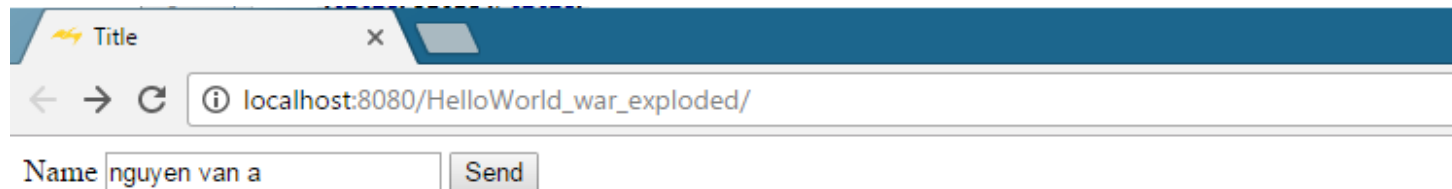
- Quan trọng nhất: Cấu hình servlet trong web.xml

```
<?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">
  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>myservlet.HelloWorld</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorld</servlet-name>
    <url-pattern>/servlet/*</url-pattern>
  </servlet-mapping>
</web-app>
```



Servlet

- Kết quả thực thi



Servlet



Servlet

- Khuyết điểm Servlet
 - Tính phụ thuộc cao
 - Khó bảo trì
 - Tăng độ phức tạp ứng dụng



JSP

- Khắc phục nhược điểm Servlet:
 - Tách biệt phần giao diện và code
 - Sử dụng lại các thành phần web bằng cách sử dụng các thẻ tiện ích
 - Dễ dàng bảo trì



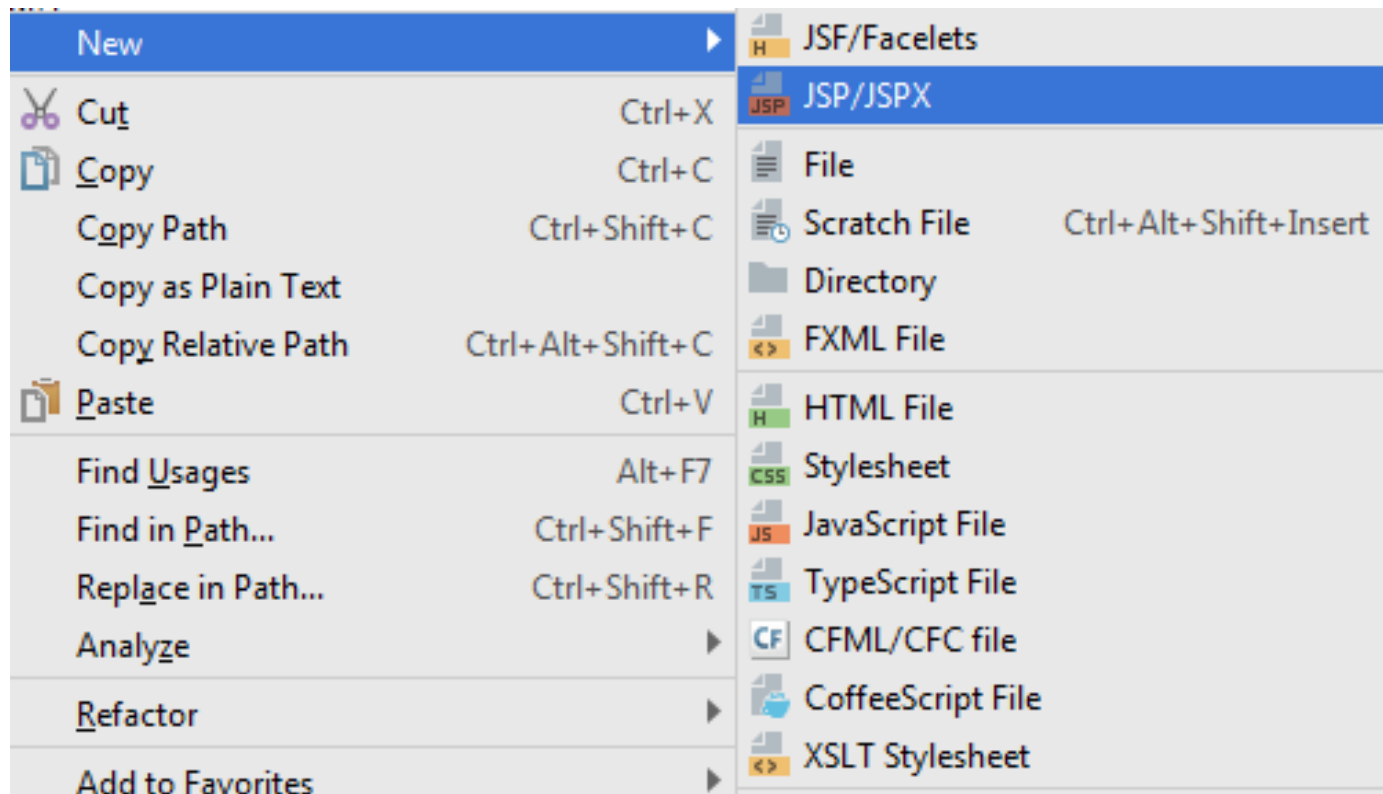
JSP

- Các đặc điểm JSP
 - Nhúng các script java
 - Sử dụng các thẻ chuẩn
 - Hỗ trợ biểu thức



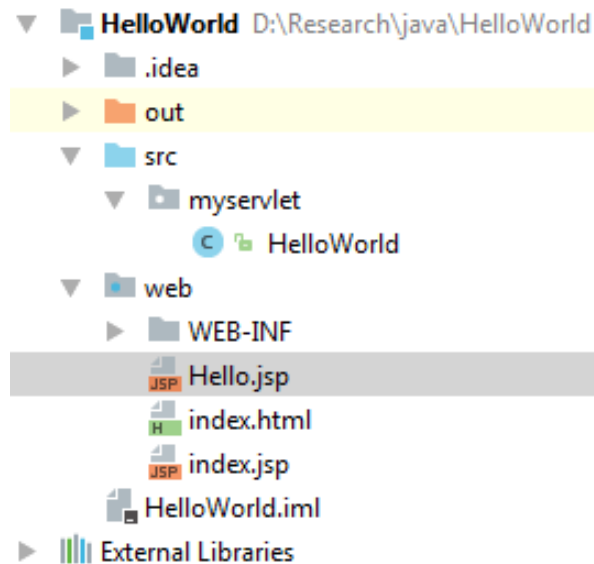
JSP

- Tạo trang JSP thay vì HTML



JSP

- Nhúng script java vào trang JSP

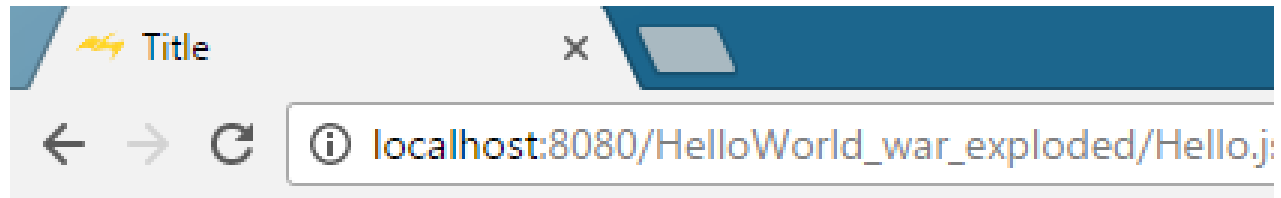


```
4      Date: 4/9/2017
5      Time: 9:01 AM
6      To change this template use Settings | File Templates.
7      <!--%>
8      <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9      <html>
10     <head>
11         <title>Title</title>
12     </head>
13     <body>
14     <%
15         out.print("<ul>");
16         for (int i = 1; i <= 10; i++)
17             out.print("<li> Loop " + i + "</li>");
18         out.print("</ul>");
19     %>
20     </body>
21 </html>
22
```

Đối tượng out tương tự như đối tượng
PrintWriter trong servlet

JSP

•Kết quả



- Loop 1
- Loop 2
- Loop 3
- Loop 4
- Loop 5
- Loop 6
- Loop 7
- Loop 8
- Loop 9
- Loop 10



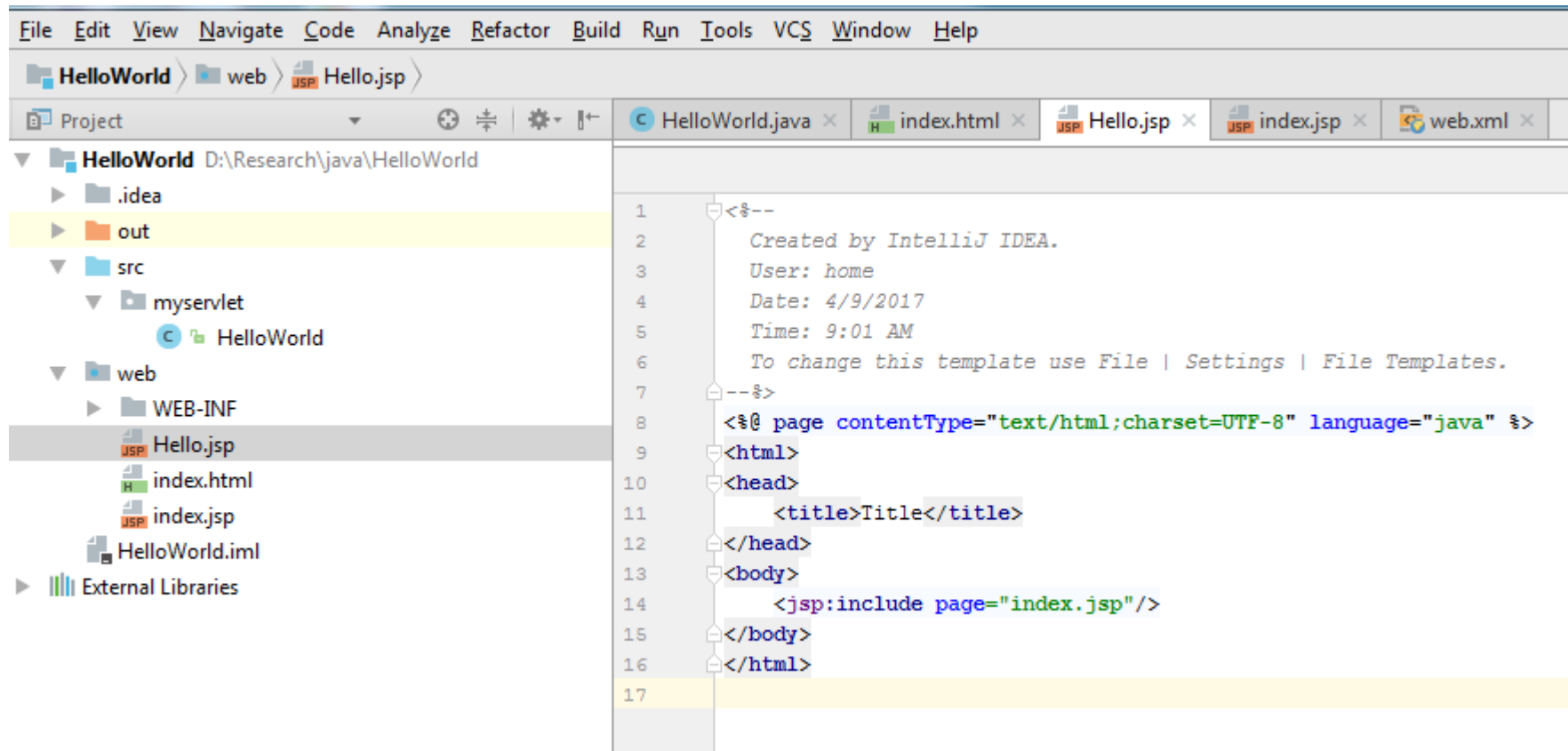
JSP

- Sử dụng các thẻ chuẩn



JSP

- Sử dụng các thẻ chuẩn



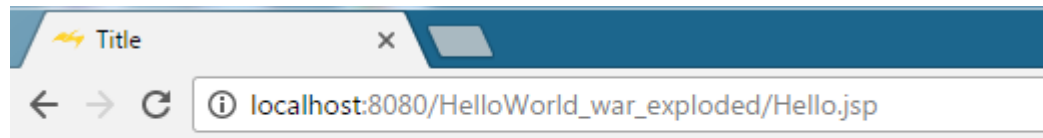
```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloWorld > web > Hello.jsp
Project
HelloWorld D:\Research\java\HelloWorld
  .idea
  out
  src
    myervlet
      HelloWorld
  web
    WEB-INF
    Hello.jsp
    index.html
    index.jsp
    HelloWorld.iml
  External Libraries

1  <%--
2    Created by IntelliJ IDEA.
3    User: home
4    Date: 4/9/2017
5    Time: 9:01 AM
6    To change this template use File | Settings | File Templates.
7  --%>
8  <%@ page contentType="text/html;charset=UTF-8" language="java" %>
9  <html>
10 <head>
11   <title>Title</title>
12 </head>
13 <body>
14   <jsp:include page="index.jsp"/>
15 </body>
16 </html>
17
```



JSP

- Kết quả



Index



JSP

- Khái niệm Java Bean
 - Class java
 - Constructor không có tham số
 - Thực thi interface Serializable
 - Getter và Setter

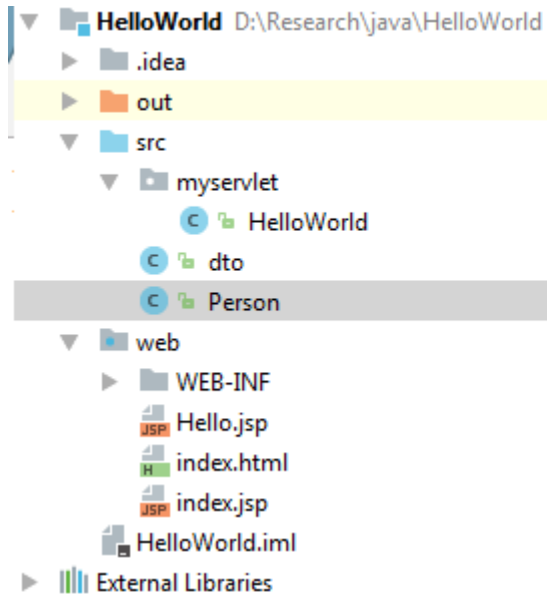


JSP

- Tạo Java Bean đơn giản là tạo Java Class
- Click phải thư mục src -> new -> Java Class



JSP



```
1  import java.io.Serializable;
2
3  /**
4   * Created by home on 4/9/2017.
5   */
6  public class Person implements Serializable {
7
8      private String name;
9      private String address;
10
11     public Person() {
12     }
13
14     public String getName() {
15         return name;
16     }
17
18     public void setName(String name) {
19         this.name = name;
20     }
21
22     public String getAddress() {
23         return address;
24     }
25
26     public void setAddress(String address) {
27         this.address = address;
28     }
29 }
```



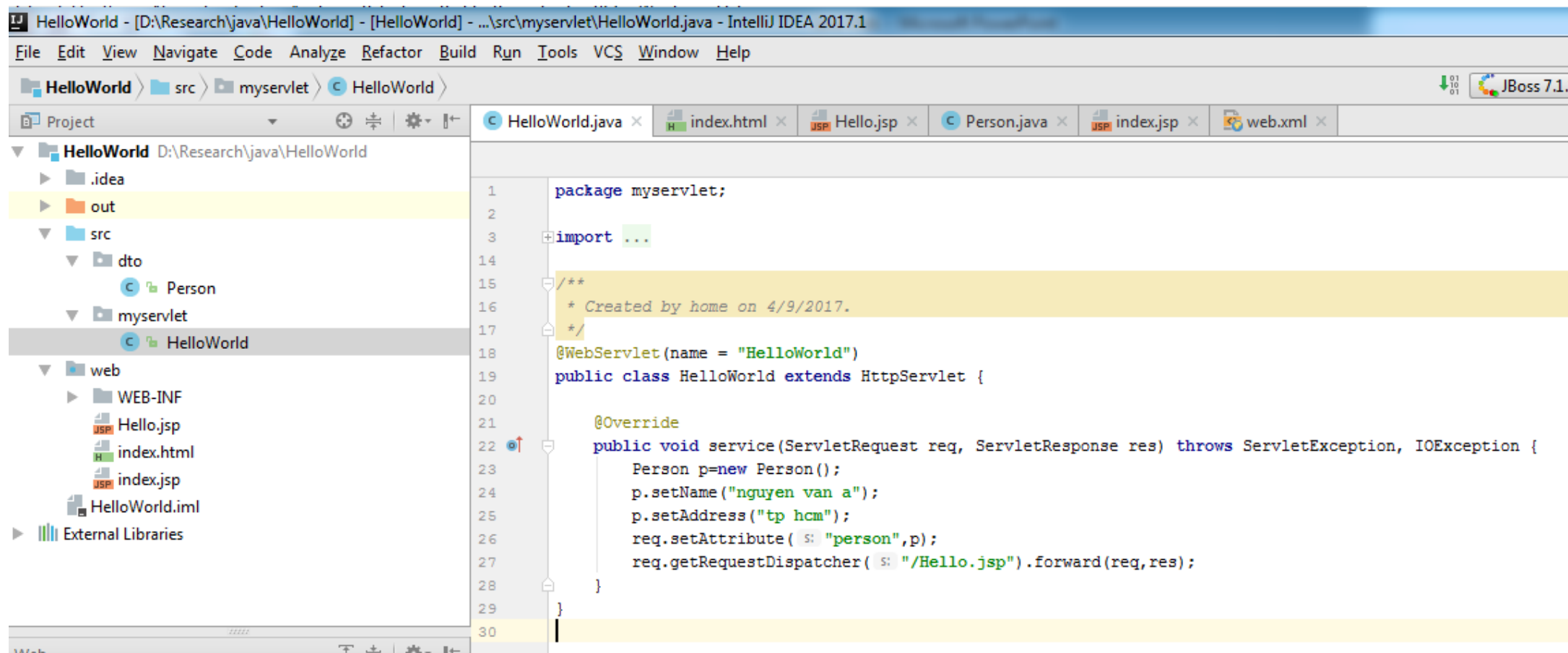
JSP

- Java bean: Dùng đóng gói thông tin và được gửi đến JSP để hiển thị thông tin. Các giá trị java bean có thể được khởi tạo bởi servlet hoặc truy xuất từ data base
- Servlet chuyển đối tượng java bean đến trang jsp



JSP

- Chỉnh sửa lại servlet như sau:



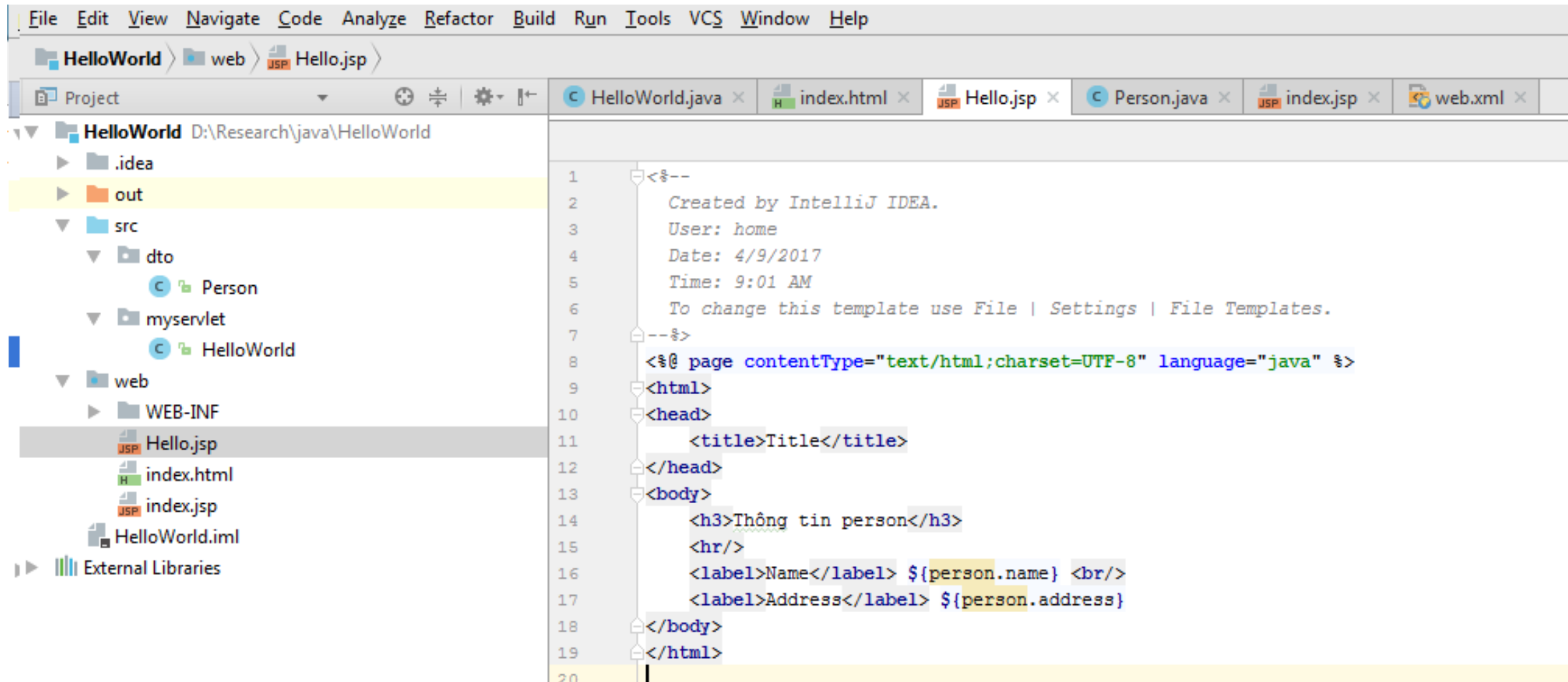
```
1 package myervlet;
2
3 import ...
4
5 /**
6  * Created by home on 4/9/2017.
7  */
8 @WebServlet(name = "HelloWorld")
9 public class HelloWorld extends HttpServlet {
10
11     @Override
12     public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {
13         Person p=new Person();
14         p.setName("nguyen van a");
15         p.setAddress("tp hcm");
16         req.setAttribute("person",p);
17         req.getRequestDispatcher("/Hello.jsp").forward(req,res);
18     }
19 }
```

JSP

- Để truy cập java bean tại JSP, dùng biểu thức
\$ { biểu thức }



JSP

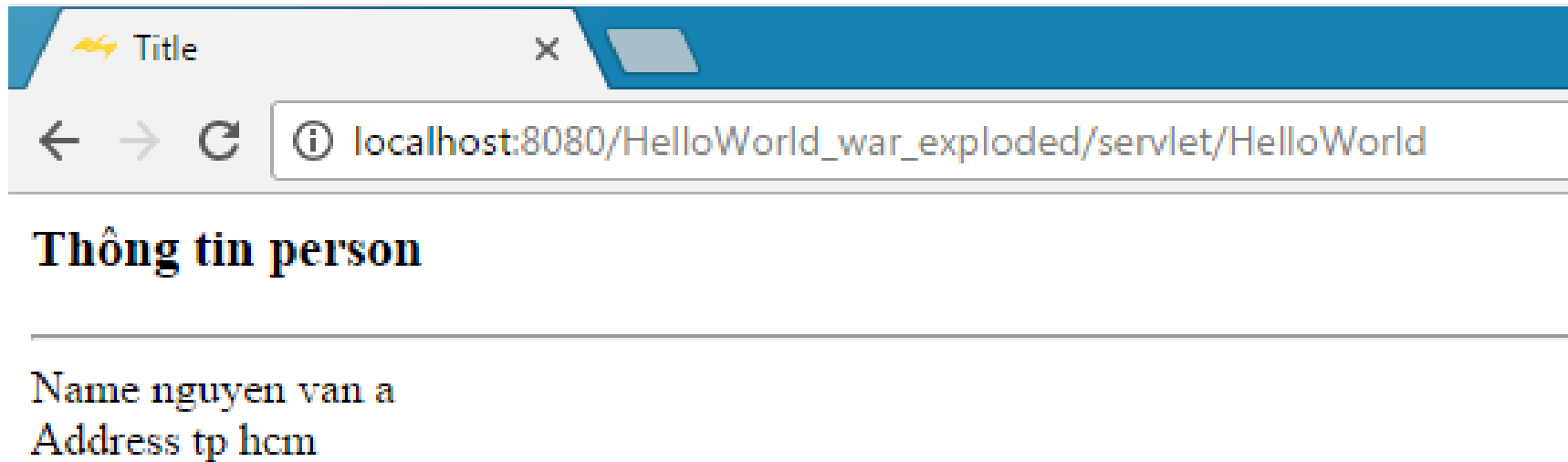


```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloWorld > web > JSP Hello.jsp
Project
HelloWorld D:\Research\java\HelloWorld
  .idea
  out
  src
    dto
      Person
    myservlet
      HelloWorld
  web
    WEB-INF
    Hello.jsp
    index.html
    index.jsp
    HelloWorld.iml
  External Libraries

1  <!--
2    Created by IntelliJ IDEA.
3    User: home
4    Date: 4/9/2017
5    Time: 9:01 AM
6    To change this template use File | Settings | File Templates.
7  -->
8  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9  <html>
10 <head>
11   <title>Title</title>
12 </head>
13 <body>
14   <h3>Thông tin person</h3>
15   <hr/>
16   <label>Name</label> ${person.name} <br/>
17   <label>Address</label> ${person.address}
18 </body>
19 </html>
20
```

JSP

- Kết quả



JSP



Maven

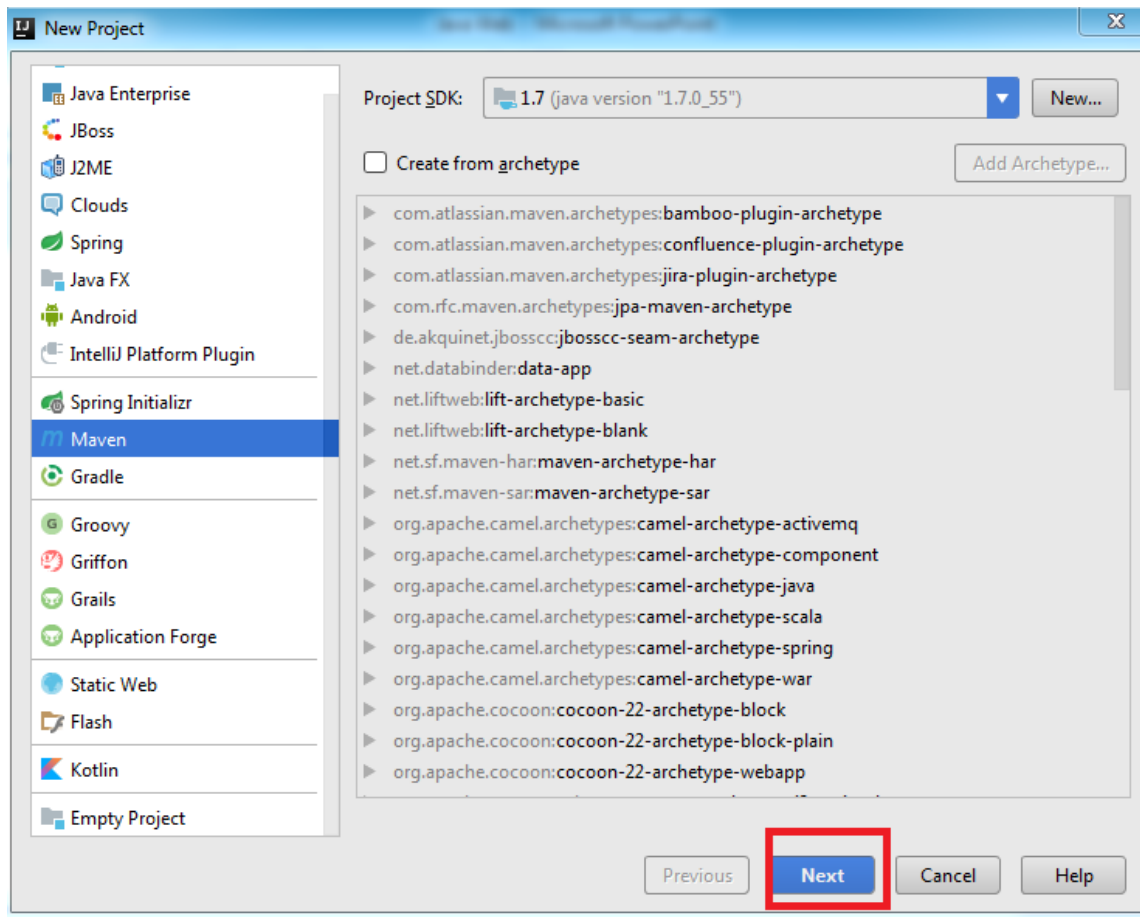
- Giới thiệu

- Apache maven là một chương trình quản lý dự án cho phép các developers có thể quản lý về version, các dependencies (các thư viện sử dụng trong dự án) , **quản lý build**, tự động download javadoc & source
- **Repository** : Là nơi để chương trình maven download các dependency



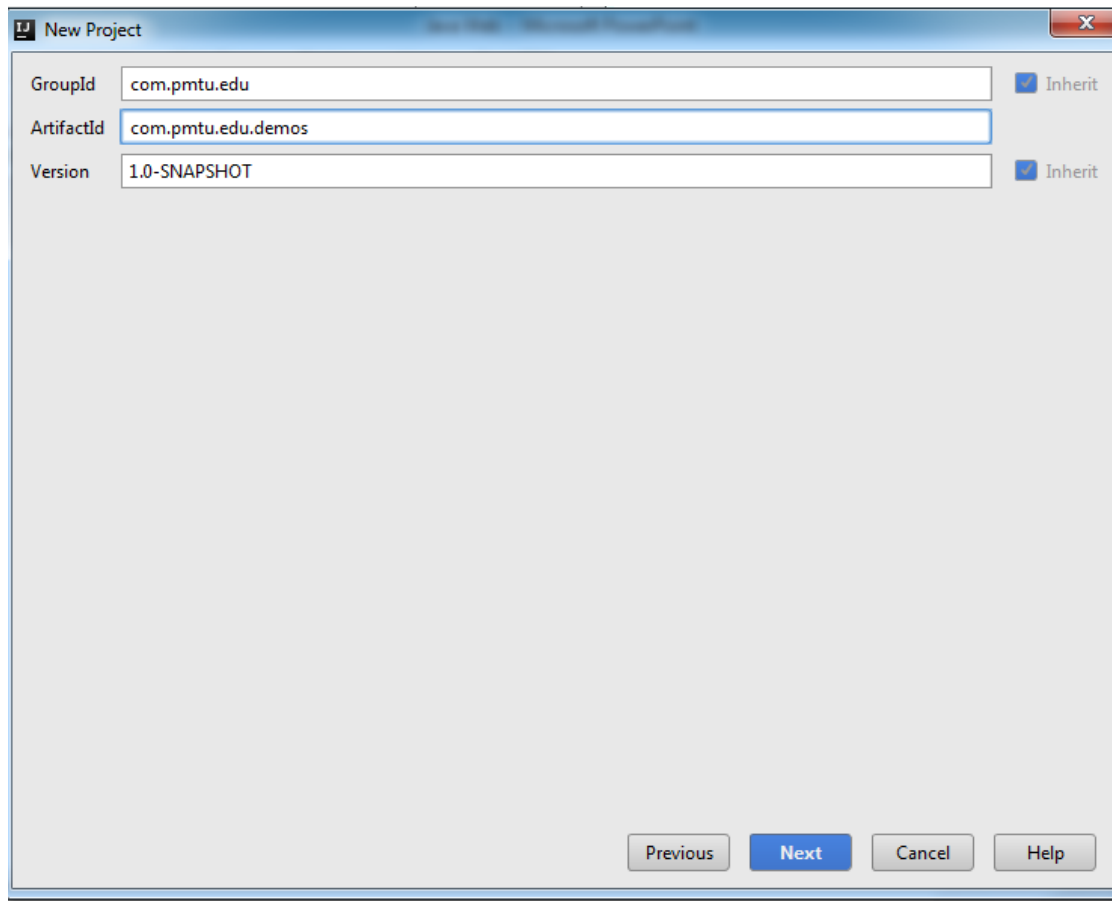
Maven

- Ví dụ: Tạo một project có sử dụng thư viện của Hibernate



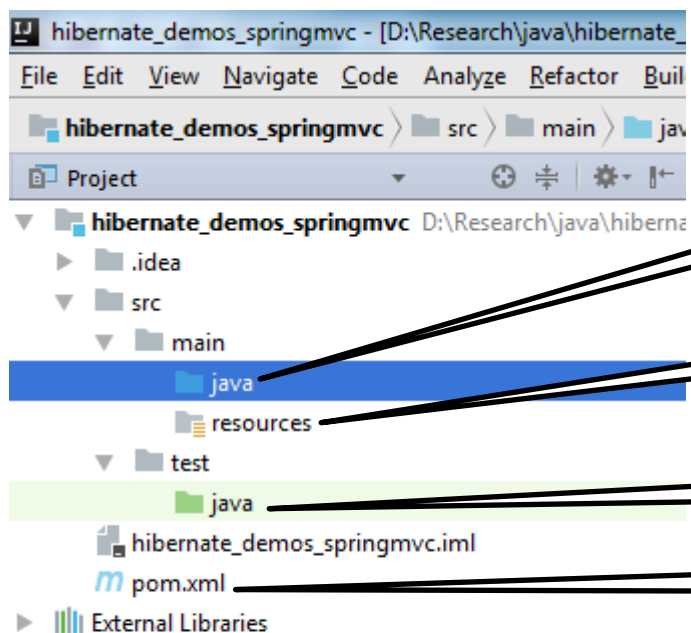
Maven

- Đặt tên project, quản lý theo maven



Maven

•Cấu trúc thư mục trong project



Các file java

Chứa các file tài nguyên

Test các chức năng

Quản lý thư viện, biên dịch và triển khai



Maven

- File pom
 - Chứa kiểu ứng dụng muốn biên dịch
 - Chứa các thư viện muốn sử dụng
 -



Maven

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.pmtu.edu</groupId>
  <artifactId>com.pmtu.edu.demos</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- MySQL database driver -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.9</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate</artifactId>
      <version>3.2.3.ga</version>
    </dependency>

    <!-- Hibernate library dependency start -->
```



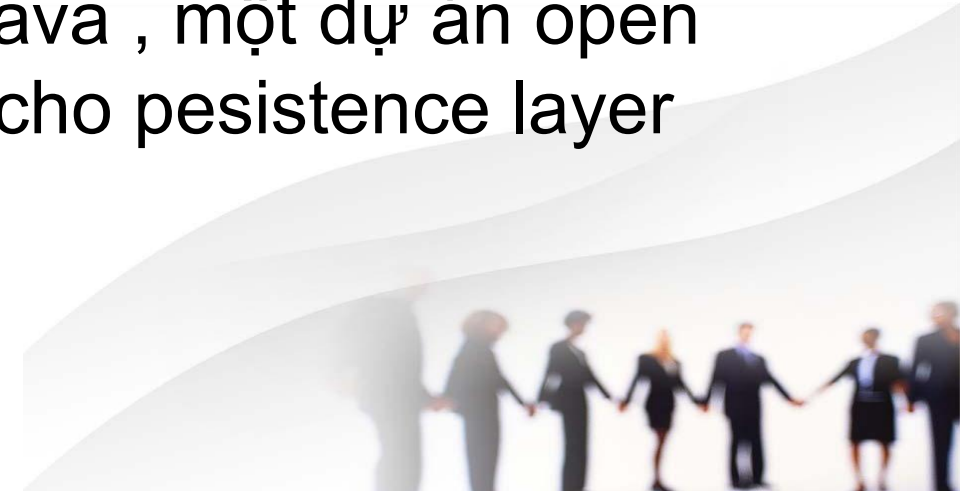
Hibernate

- Giới thiệu

- ORM

- ORM viết tắt của Object Relation Mapping, được hiểu như là kỹ thuật ánh xạ từ mô hình đối tượng với các table trong cơ sở dữ liệu quan hệ. Trong đó các đối tượng ánh xạ với các bảng và quan hệ của table trong database sẽ được ánh xạ với sự ràng buộc liên quan trong đối tượng

- Hibernate là một giải pháp ORM cho Java , một dự án open source chuyên nghiệp, một framework cho pesistence layer



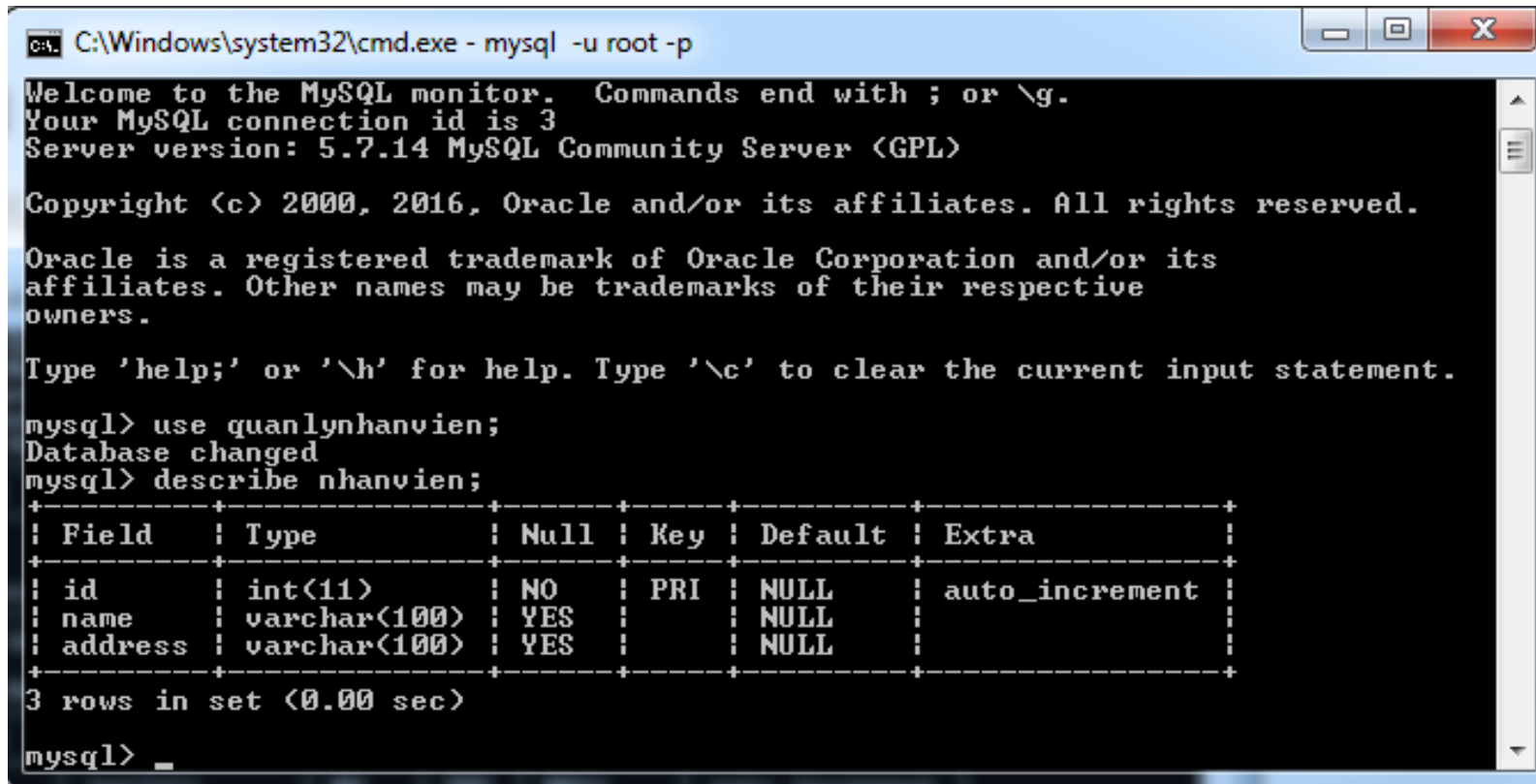
Hibernate

- Cấu hình Hibernate
 - Annotation (Tìm hiểu thêm)
 - XML



Hibernate

- Ví dụ
 - Chuẩn bị: database, table trong MYSQL



```
C:\Windows\system32\cmd.exe - mysql -u root -p

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

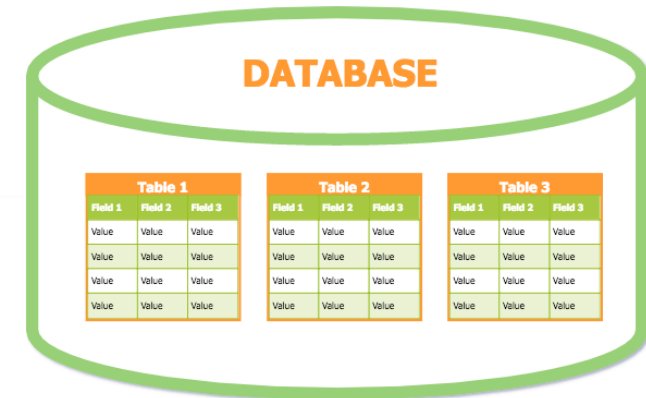
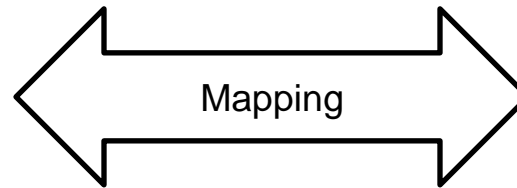
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use quanlynhanvien;
Database changed
mysql> describe nhanvien;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int(11)       | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100)  | YES  |     | NULL    |                |
| address | varchar(100)  | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

Hibernate

- Ví dụ
 - Tạo file XML cấu hình mapping giữa class – table (**nhanvien.hbm.xml**)



Hibernate

```
1  <?xml version="1.0"?>
2  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3      "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
4  <hibernate-mapping>
5      <class name="entities.NhanVien" table="nhanvien" catalog="quanlynhanvien">
6          <id name="id" type="java.lang.Integer">
7              <column name="id"/>
8              <generator class="identity"/>
9          </id>
10         <property name="name" type="java.lang.String">
11             <column name="name" length="100" not-null="true"/>
12         </property>
13         <property name="address" type="java.lang.String">
14             <column name="address" length="100" not-null="true"/>
15         </property>
16     </class>
17 </hibernate-mapping>
```


Hibernate

- Cấu hình kết nối database (hibernate.cfg.xml)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
5 <hibernate-configuration>
6     <session-factory>
7         <property name="hibernate.bytecode.use_reflection_optimizer">false</property>
8         <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
9         <property name="hibernate.connection.password"></property>
10        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/QuanLyNhanVien</property>
11        <property name="hibernate.connection.username">root</property>
12        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
13        <property name="hibernate.show_sql">true</property>
14        <mapping resource="common/nhanvien.hbm.xml"></mapping>
15    </session-factory>
16 </hibernate-configuration>
```

Hibernate

- Cấu hình kết nối database (hibernate.cfg.xml)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE hibernate-configuration PUBLIC
3      "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4      "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
5  <hibernate-configuration>
6      <session-factory>
7          <property name="hibernate.bytecode.use_reflection_optimizer">false</property>
8          <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
9          <property name="hibernate.connection.password"></property>
10         <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/QuanLyNhanVien</property>
11         <property name="hibernate.connection.username">root</property>
12         <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
13         <property name="show_sql">>true</property>
14         <mapping resource="nhanvien.hbm.xml"></mapping>
15     </session-factory>
16 </hibernate-configuration>
```



Hibernate

- Tạo đối tượng quản lý session

```
11 public class HibernateUtil {
12     private static final SessionFactory sessionFactory = buildSessionFactory();
13
14     private static SessionFactory buildSessionFactory() {
15         try {
16             return new Configuration().configure().buildSessionFactory();
17         } catch (Throwable e) {
18             e.printStackTrace();
19             throw new ExceptionInInitializerError(e);
20         }
21     }
22     @ public static SessionFactory getSessionFactory() {
23         return sessionFactory;
24     }
25     public static void shutdown() {
26         getSessionFactory().close();
27     }
28
29 }
```



Hibernate

•Sử dụng API thao tác database

```
13 public class MyTest {
14
15     @Test
16     public void testgetall()
17     {
18         Session session = HibernateUtil.getSessionFactory().openSession();
19         session.beginTransaction();
20         List<NhanVien> lst=session.createCriteria(NhanVien.class).list();
21         Assert.assertNotNull(lst);
22         Assert.assertNotSame( expected: 0,lst.size());
23         session.getTransaction().commit();
24     }
25     @Test
26     public void insert()
27     {
28         NhanVien nv=new NhanVien();
29         nv.setName("nguyen thanh tan");
30         nv.setAddress("long an");
31         Session session = HibernateUtil.getSessionFactory().openSession();
32         session.beginTransaction();
33         session.save(nv);
34         Assert.assertNotNull(nv);
35         Assert.assertNotSame( expected: 0,nv.getId());
36         session.getTransaction().commit();
37     }
38 }
```



Spring MVC

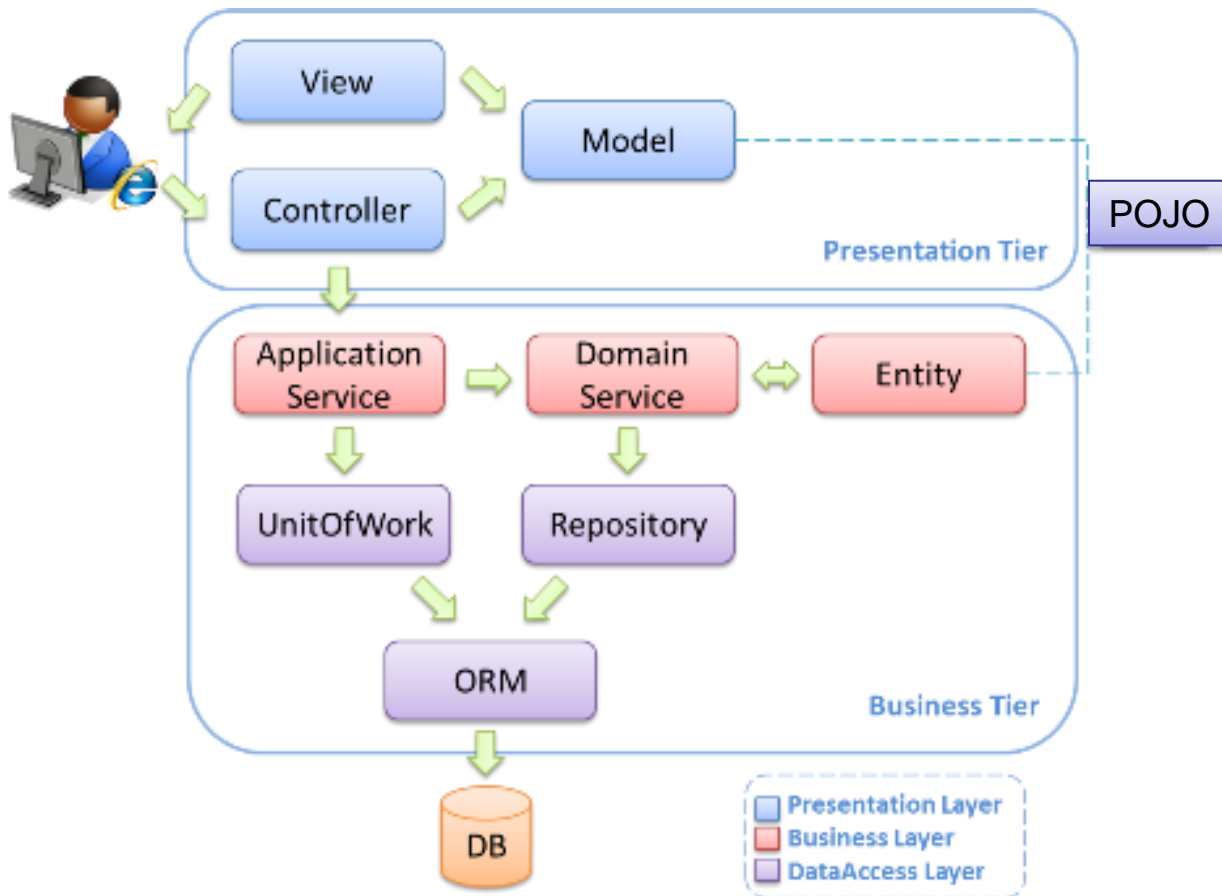
- Giới thiệu MVC

- MVC - **Model**, **View** và **Controller**.
- MVC là một mô hình ứng dụng ,các thành phần trong ứng dụng được phân tách ra thành các lớp riêng biệt với các nhiệm vụ đặc trưng.
- View sẽ là lớp cho các thành phần có chức năng hiển thị, giao tiếp trực tiếp với người dùng.
- Model là các thành phần có khả năng lưu trữ và vận chuyển thông tin.
- Controller: Nhận request từ người dùng và tạo dữ liệu lưu vào Model



Spring MVC

- MVC trong kiến trúc ứng dụng doanh nghiệp



Spring MVC

- Giới thiệu MVC

- Spring MVC là một framework phát triển trên giải pháp MVC

- Cài đặt

- Đơn giản nhất dùng Maven tải các thư viện

- Cấu hình

- Dựa trên 2 cách

- Code

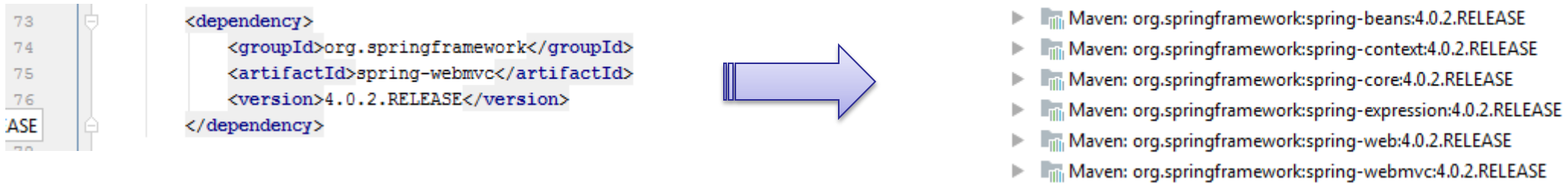
- XML



Spring MVC

- Ví dụ

- Khai báo thư viện dùng Spring MVC



Spring MVC

- Ví dụ

- Cấu hình servlet để dùng Spring MVC

```
1 <web-app id="WebApp_ID" version="2.4"
2     xmlns="http://java.sun.com/xml/ns/j2ee"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5     http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
6
7     <display-name>Spring Web MVC Application</display-name>
8
9     <servlet>
10         <servlet-name>mvc-dispatcher</servlet-name>
11         <servlet-class>
12             org.springframework.web.servlet.DispatcherServlet
13         </servlet-class>
14         <load-on-startup>1</load-on-startup>
15     </servlet>
16
17     <servlet-mapping>
18         <servlet-name>mvc-dispatcher</servlet-name>
19         <url-pattern>/</url-pattern>
20     </servlet-mapping>
21
22 </web-app>
```



Spring MVC

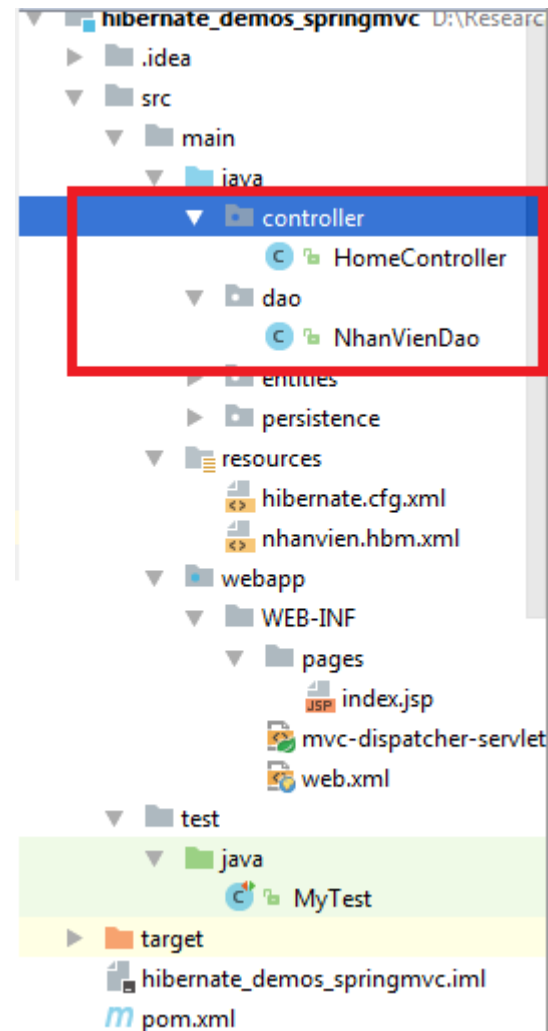
- Ví dụ
 - Cấu hình Spring MVC

```
1 <beans xmlns = "http://www.springframework.org/schema/beans"
2     xmlns:context = "http://www.springframework.org/schema/context"
3     xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation = "http://www.springframework.org/schema/beans
5     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
6     http://www.springframework.org/schema/context
7     http://www.springframework.org/schema/context/spring-context-3.0.xsd">
8
9     <context:component-scan base-package = "controller" />
10
11     <bean class = "org.springframework.web.servlet.view.InternalResourceViewResolver">
12         <property name = "prefix" value = "/WEB-INF/pages/" />
13         <property name = "suffix" value = ".jsp" />
14     </bean>
15
16 </beans>
```



Spring MVC

- Ví dụ
 - Định nghĩa dao và controller



Spring MVC

- Ví dụ
–DAO

```
/**
 * Created by home on 16-04-17.
 */
public class NhanVienDao {
    public List<NhanVien> getall()
    {
        Session session = HibernateUtil.getSessionFactory().openSession();
        session.beginTransaction();
        List<NhanVien> lst=session.createCriteria(NhanVien.class).list();
        session.getTransaction().commit();
        return lst;
    }
}
```



Spring MVC

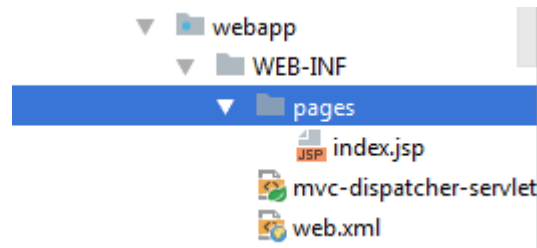
- Ví dụ
 - Controller

```
15 public class HomeController {  
16  
17     @RequestMapping(value = "/home/nhanvien/list.html")  
18     public ModelAndView getall() {  
19         ModelAndView mav=new ModelAndView( viewName: "index");  
20         List<NhanVien> lst=new NhanVienDao().getall();  
21         mav.addObject( attributeName: "listNhanVien",lst);  
22         return mav;  
23     }  
24 }  
25
```



Spring MVC

- Ví dụ
 - Định nghĩa View



```
8      <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9      <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
10     <html>
11     <head>
12         <title>Title</title>
13     </head>
14     <body>
15         <table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse">
16             <tr>
17                 <th>ID</th>
18                 <th>Name</th>
19                 <th>Address</th>
20             </tr>
21
22             <c:forEach items="${listNhanVien}" var="nv">
23                 <tr>
24                     <td>${nv.id}</td>
25                     <td>${nv.name}</td>
26                     <td>${nv.address}</td>
27                 </tr>
28             </c:forEach>
29         </table>
30     </body>
31 </html>
```

Spring MVC

- Ví dụ
 - Kết quả

← → ↻ ⓘ localhost:8080/com.pmtu.edu.demos-1.0-SNAPSHOT/home/nhanvien/list.html

ID	Name	Address
2	Nguyen Van B	Long AN
3	Nguyen Van c	Ben Tre
6	Nguyen Trong Tin	TP HCM
10	nguyen thanh tan	long an



Câu hỏi

