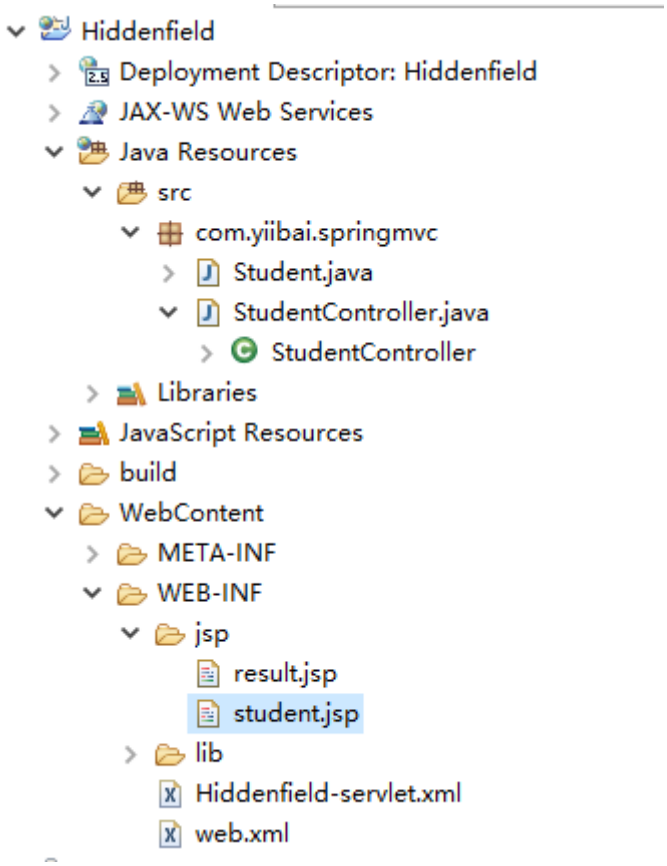
Spring MVC 隐藏字段域

以下示例显示如何在使用 Spring Web MVC 框架的表单中使用隐藏字段(Hidden)。首先使用 Eclipse IDE 来创建一个 WEB 工程，实现在隐藏字段中指定用户编号的功能。并按照以下步骤使用 Spring Web Framework 开发基于动态表单的 Web 应用程序：

1. 创建一个名称为 Hiddenfield 的动态 WEB 项目。
2. 在 com.yiibai.springmvc 包下创建两个 Java 类 Student，StudentController。
3. 在 jsp 子文件夹下创建两个视图文件：student.jsp，result.jsp。
4. 最后一步是创建所有源和配置文件的内容并运行应用程序，详细如下所述。

完整的项目文件目录结构如下所示 -



Student.java 的代码如下所示 -

```
package com.yiibai.springmvc;


public class Student {
    private Integer age;
    private String name;
    private Integer id;
```

```
    public void setAge(Integer age) {
        this.age = age;
    }

    public Integer getAge() {
        return age;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public Integer getId() {
        return id;
    }
}
Java
```

StudentController.java 的代码如下所示 -

```
package com.yiibai.springmvc;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.ui.ModelMap;

@Controller
public class StudentController {

    @RequestMapping(value = "/student", method = RequestMethod.GET)
    public ModelAndView student() {
        return new ModelAndView("student", "command", new Student());
    }
```

```
    @RequestMapping(value = "/addStudent", method = RequestMethod.POST)
    public String addStudent(@ModelAttribute("SpringWeb") Student student,
ModelMap model) {
        model.addAttribute("name", student.getName());
        model.addAttribute("age", student.getAge());
        model.addAttribute("id", student.getId());

        return "result";
    }
}
Java
```

这里的第一个服务方法 student()，我们已经在 ModelAndView 对象中传递了一个名称为“command”的空 Student 对象，因为如果在 JSP 文件中使用<form:form>标签，spring 框架需要一个名称为“command”的对象。 所以当调用 student()方法时，它返回 student.jsp 视图。

第二个服务方法 addStudent()将根据 URL => Hiddenfield/addStudent 上的 POST 方法请求时调用。根据提交的信息准备模型对象。 最后从服务方法返回“result”视图，这将呈现 result.jsp 视图。

student.jsp 的代码如下所示 -

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
    <title>Spring MVC 处理(隐藏字段)</title>
</head>
<body>

<h2>学生信息：</h2>
<form:form method="POST" action="/Hiddenfield/addStudent">
   <table>
    <tr>
        <td><form:label path="name">学生姓名：</form:label></td>
        <td><form:input path="name" /></td>
    </tr>
    <tr>
        <td><form:label path="age">年龄：</form:label></td>
        <td><form:input path="age" /></td>
    </tr>
    <tr>
        <td></td>
```

```
        <td><form:hidden path="id" value="1000" /></td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" value="提交"/>
        </td>
    </tr>
</table>
</form:form>
</body>
</html>
HTML
```

这里使用<form:hidden />标签来呈现 HTML 隐藏字段域。 例如 -

```
<form:hidden path="id" value="1000"/>
HTML
```

它将呈现以下 HTML 内容。

```
<input id="id" name="id" type="hidden" value="1000"/>
HTML
```

result.jsp 的代码如下所示 -

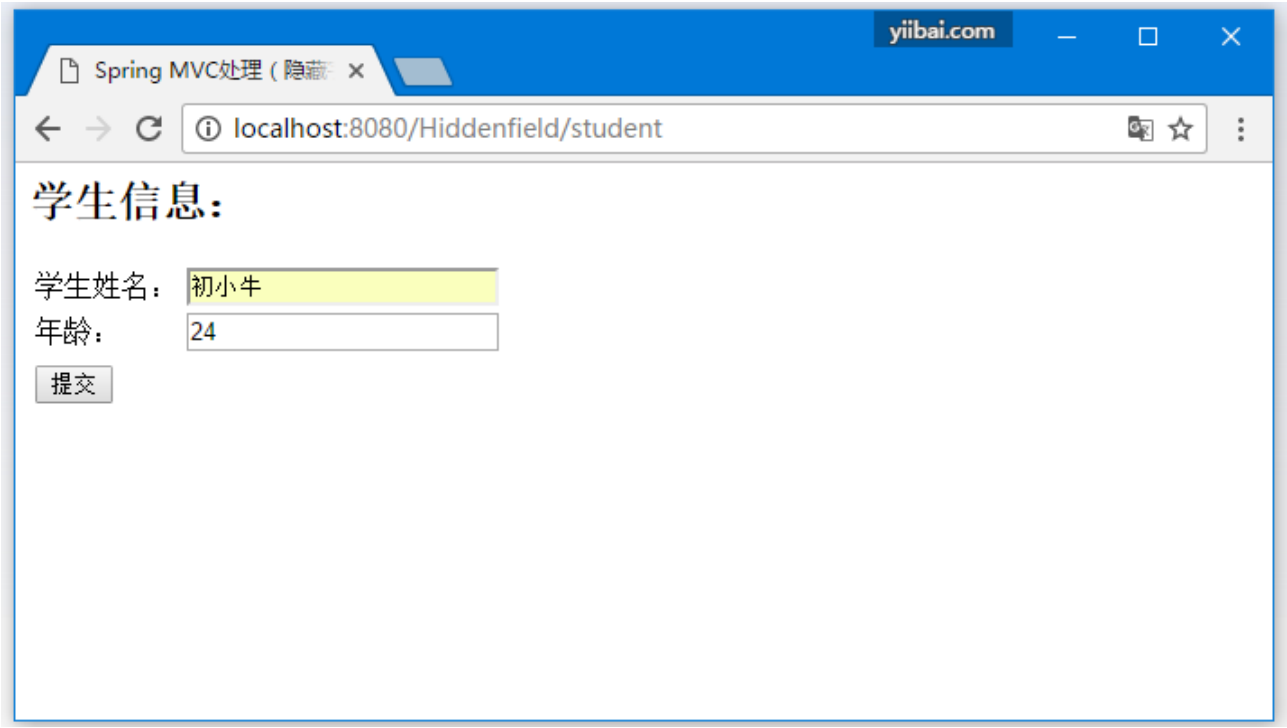
```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<title>Spring MVC 处理(隐藏字段)</title>
</head>
<body>

    <h2>提交的学生信息：</h2>
    <table>
        <tr>
            <td>学生姓名：</td>
            <td>${name}</td>
        </tr>
        <tr>
            <td>年龄：</td>
            <td>${age}</td>
        </tr>
        <tr>
            <td>编号：</td>
            <td>${id}</td>
```

```
        </tr>
    </table>
</body>
</html>
HTML
```

完成创建源和配置文件后，发布应用程序到 Tomcat 服务器。

现在启动 Tomcat 服务器，现在尝试访问 URL => http://localhost:8080/Hiddenfield/student ，如果 Spring Web 应用程序没有问题，应该看到以下结果：



提交所需信息后，点击提交按钮提交表单。 如果 Spring Web 应用程序没有问题，应该看到以下结果：