

## 监听器Listener

### 监听器的使用：

- 实现接口，实现方法完成业务逻辑。
- web.xml中进行配置。

### 监听器的分类：

- 监听域对象的创建和销毁（request, session, application）
  - ServletRequestListener用来监听request
  - HttpSessionListener用来监听session
  - ServletContextListener用来监听application
- 监听域对象中属性的新增，删除，修改事件。

新增：setAttribute(String key,Object value)

删除：removeAttribute(String key)

修改：setAttribute(String key,Object value)

- 监听request的属性变化实现ServletRequestAttributeListener。

当客户端访问web资源（Servlet, JSP）都会默认修改一次request对象的属性。

```
package com.southwind.listener;

import javax.servlet.ServletRequestAttributeEvent;
import javax.servlet.ServletRequestAttributeListener;

public class RequestListener implements
ServletRequestAttributeListener {
    @Override
    public void attributeAdded(ServletRequestAttributeEvent srae)
    {
        System.out.println("-----新增-----");
        System.out.println("request对象新增了一个属性");
        System.out.println(srae.getName());
        System.out.println(srae.getValue());
        System.out.println("-----新增-----");
    }

    @Override
    public void attributeRemoved(ServletRequestAttributeEvent
srae) {
        System.out.println("-----删除-----");
        System.out.println("request对象删除了一个属性");
        System.out.println(srae.getName());
        System.out.println(srae.getValue());
        System.out.println("-----删除-----");
    }
}
```

```

@Override
public void attributeReplaced(ServletRequestAttributeEvent
srae) {
    System.out.println("-----修改-----");
    System.out.println("request对象修改了一个属性");
    System.out.println(srae.getName());
    System.out.println(srae.getValue());
    System.out.println("-----修改-----");
}
}

```

- 监听session的属性变化实现HttpSessionAttributeListener。

当客户端访问web资源（Servlet, JSP）不会默认修改session对象的属性。

```

package com.southwind.listener;

import javax.servlet.http.HttpSessionAttributeListener;
import javax.servlet.http.HttpSessionBindingEvent;

public class SessionListener implements
HttpSessionAttributeListener {
    @Override
    public void attributeAdded(HttpSessionBindingEvent se) {
        System.out.println("-----新增-----");
        System.out.println("session新增了一个属性");
        System.out.println(se.getName());
        System.out.println(se.getValue());
        System.out.println("-----新增-----");
    }

    @Override
    public void attributeRemoved(HttpSessionBindingEvent se) {
        System.out.println("-----删除-----");
        System.out.println("session删除了一个属性");
        System.out.println(se.getName());
        System.out.println(se.getValue());
        System.out.println("-----删除-----");
    }

    @Override
    public void attributeReplaced(HttpSessionBindingEvent se) {
        System.out.println("-----修改-----");
        System.out.println("session修改了一个属性");
        System.out.println(se.getName());
        System.out.println(se.getValue());
        System.out.println("-----修改-----");
    }
}

```

```
}  
}
```

- 监听application的属性变化实现ServletContextAttributeListener。

tomcat启动之后，当客户端第一次访问任意一个JSP资源时，会默认向application中添加两个属性：

JspApplicationContextImpl和ELInterpreter。

第二次。。。第N次访问任何JSP资源时，不再添加，即它只会在tomcat启动时添加一次。

如果客户端访问的是Servlet资源，则不添加。

```
package com.southwind.listener;  
  
import javax.servlet.ServletContextAttributeEvent;  
import javax.servlet.ServletContextAttributeListener;  
  
public class ApplicationListener implements  
ServletContextAttributeListener {  
    @Override  
    public void attributeAdded(ServletContextAttributeEvent scae)  
    {  
        System.out.println("-----新增-----");  
        System.out.println("application新增了一个属性");  
        System.out.println(scae.getName());  
        System.out.println(scae.getValue());  
        System.out.println("-----新增-----");  
    }  
  
    @Override  
    public void attributeRemoved(ServletContextAttributeEvent  
scae) {  
        System.out.println("-----删除-----");  
        System.out.println("application删除了一个属性");  
        System.out.println(scae.getName());  
        System.out.println(scae.getValue());  
        System.out.println("-----删除-----");  
    }  
  
    @Override  
    public void attributeReplaced(ServletContextAttributeEvent  
scae) {  
        System.out.println("-----修改-----");  
        System.out.println("application修改了一个属性");  
        System.out.println(scae.getName());  
        System.out.println(scae.getValue());  
        System.out.println("-----修改-----");  
    }  
}
```

```
}
```

- 监听一个被绑定到HttpSession域中的普通对象，通过让实体类实现HttpSessionBindingListener接口来完成，不需要在web.xml中进行配置。

当服务端向session中添加被监听的对象时，会触发valueBound方法。

当服务端从session中移除被监听的对象时，会触发valueUnBound方法。

当服务器关闭tomcat时，会触发valueUnBound方法。

当服务器端销毁session时，会触发valueUnBound方法。

```
package com.southwind.entity;

import javax.servlet.http.HttpSessionBindingEvent;
import javax.servlet.http.HttpSessionBindingListener;

public class User implements HttpSessionBindingListener {
    private int id;
    private String name;
    private int age;

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    public int getAge() {
        return age;
    }

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

    @Override
    public void valueBound(HttpSessionBindingEvent event) {
        System.out.println(event.getName());
        System.out.println(event.getValue());
    }
}
```

```

        System.out.println("User对象被绑定到了session中");
    }

    @Override
    public void valueUnbound(HttpSessionBindingEvent event) {
        System.out.println(event.getName());
        System.out.println(event.getValue());
        System.out.println("User对象从session中解除了绑定");
    }
}

```

## 文件上传下载

上传：客户端将本地的资源通过网络传输给服务端，然后保存到服务端本地。

HTTP网络协议+IO流

JSP:

- input标签的type设置为file
- form表单的method设置为post，如果设置为get，那么传给服务端的是文件名，是个字符串，而非文件本身。
- form表单的enctype设置为multipart/form-data，enctype默认是传输文本类型的数据，修改之后传输的是二进制数据。

```

<%--
    Created by IntelliJ IDEA.
    User: southwind
    Date: 2018/11/29
    Time: 9:34 PM
    To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <form action="upload.do" method="post" enctype="multipart/form-data">
        请选择上传文件: <input type="file" name="file" /><br/>
        <input type="submit" value="上传" />
    </form>
</body>
</html>

```

```
package com.southwind.controller;
```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.*;

public class UploadServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        this.doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        InputStream inputStream = req.getInputStream();
        Reader reader = new InputStreamReader(inputStream);
        BufferedReader bufferedReader = new BufferedReader(reader);
        String path = req.getServletContext().getRealPath("file");
        OutputStream outputStream = new FileOutputStream(path+"/copy.txt");
        Writer writer = new OutputStreamWriter(outputStream);
        BufferedWriter bufferedWriter = new BufferedWriter(writer);
        String data = null;
        while((data = bufferedReader.readLine())!=null){
            bufferedWriter.write(data);
        }
        bufferedWriter.close();
        writer.close();
        outputStream.close();
        bufferedReader.close();
        reader.close();
        inputStream.close();
    }
}

```