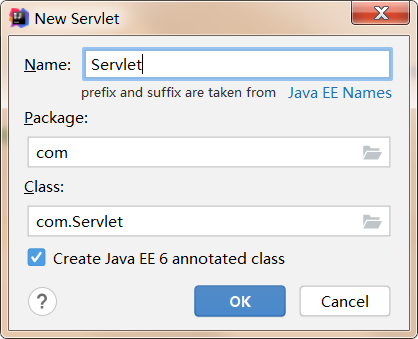
实习二 Servlet技术

1. 实习要求与目的
2. 掌握 Servlet 的工作机制及编写方法
3. 掌握 Filter工作机制及编写方法
4. 了解 Servlet 生命周期事件
5. 掌握数据库访问方法
6. 实习内容
7. Servlet
8. 第一个servlet

创建一个Java Web的maven工程。在工程结构窗格的“java”目录上点击右键，选择“new”->“Package”创建“com”包；然后在“com”上点击右键选择“new”->“new Create Servlet”打开如下对话框：



Name：输入Servlet类的名称；

Package：新建类所在包；

Class：新建类的完整类名；

Create Jave EE 6 annotated class：是否使用注解对Servlet进行配置；

点击“OK”按钮，完成Servlet的创建。修改新建Servlet的代码内容如下：

@WebServlet(name = "firstServlet",urlPatterns = "/serv")

public class FirstServlet extends HttpServlet {

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

doGet(request,response);

}

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

response.getWriter().println("欢迎来到天涯海角@@@");

}

}

启动服务器并访问servlet，查看结果是否正常；若不正常如何修改。

1. 表单处理

改造实习一中的2.1（表单处理）。新建一个servlet接收表单提交的内容，并对提交内容进行处理，然后交由result.jsp显示结果。

1. 过滤器的使用
2. 使用过滤器重写实习一中的2.3（登录验证）。
3. 使用Cookie为（1）中的登录验证实现自动登录功能。（在登录页面添加“自动登录”的复选框，若选择下次访问该应用时使用cookie存储的用户名和口令完成字典登录）。
4. Servlet生命周期事件

使用生命周期事件统计当前应用的在线人数、启动时间、请求数。

创建一个监听器完成如上功能，代码如下：

**public class** EventListener **implements** ServletContextListener,

ServletRequestListener, HttpSessionListener

{

@Override

**public void** sessionCreated(HttpSessionEvent arg0) {

// 当有一个会话创建，则给当前在线人数加 1

**int** cnt = 0;

ServletContext context = arg0.getSession().getServletContext();

Object sessionCnt = context.getAttribute("sessionCnt");

**if**(sessionCnt != **null**){

cnt = (Integer)sessionCnt;

}

cnt++;

context.setAttribute("sessionCnt", cnt);

}

@Override

**public void** sessionDestroyed(HttpSessionEvent arg0) {

// 当有一个会话关闭时，则给当前在线人数减 1

**int** cnt = 0;

ServletContext context = arg0.getSession().getServletContext();

Object sessionCnt = context.getAttribute("sessionCnt");

**if**(sessionCnt != **null**){

cnt = (Integer)sessionCnt;

cnt--;

}

context.setAttribute("sessionCnt", cnt);

}

@Override

**public void** requestDestroyed(ServletRequestEvent arg0) {}

@Override

**public void** requestInitialized(ServletRequestEvent arg0) {

// 每有一个请求，则计数

**int** cnt = 0;

Object reqcnt = arg0.getServletContext().getAttribute("reqcnt");

**if**(reqcnt != **null**){

cnt = (Integer)reqcnt;

}

cnt++;

arg0.getServletContext().setAttribute("reqcnt", cnt);

}

@Override

**public void** contextDestroyed(ServletContextEvent arg0) {}

@Override

**public void** contextInitialized(ServletContextEvent arg0) {

// 保存应用启动时间

arg0.getServletContext().setAttribute("StartDate", **new** Date());

}

}

创建一个 jsp 页面显示统计结果:

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE>

<html>

<head>

<title>生命周期事件</title>

</head>

<body>

<p>当前应用启动时间：<%=application.getAttribute("StartDate") %></p>

<p>当前在线人数：<%=application.getAttribute("sessionCnt") %></p>

<p>处理请求数：<%=application.getAttribute("reqcnt") %></p>

</body>

</html>

1. 总结

分别写出基于配置文件与注解的Servlet和Filter的配置，并分析优缺点！

**实习报告提交地址：www.zqldm.top:81**