在Servlet中，当任务过多的时候，Servlet也会变多，不方便管理。

解决方法：

将servlet进行归类，对于同一个实体的操作方法，写到一个servlet中。例如BrandServlet，UsersServlet.

第一步，将访问路径改为通配符 \*

@WebServlet("/User/\*")

public class UserServlet {

public void selectAll(){}

public void add(){}

public void update(){}

public void delect(){}

}

获取最后一段路径，就是获取方法的名称

通过反射获取方法对象，执行方法。

因为HttpServlet会分析请求的方式来执行相对应的doGet和doPost方法，

而我们不使用这两个方法，我们将使用selectAll(){}， add(){}等方法

则我们需要自定义Servlet，使用请求路径来进行方法分发，替换HttpServlet的根据请求方式来进行方法分发。

自定义servlet：BaseServlet

/\*\*

\* 替换HttpServlet，根据请求方法的路径来进行方法的分发

\* \*/

public class BaseServlet extends HttpServlet {

//根据请求方法的路径来进行方法的分发

@Override

protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

//1.获取请求的路径

String uri = req.getRequestURI();//uri=/ajaxTest\_war\_exploded/User/selectAll

//1.2获取最后一段路径，为方法名

int index = uri.lastIndexOf("/");

String methodname =uri.substring(index+1);

//2.执行方法，

//2.1.获取UserServlet的字节码对象（反射）

Class<? extends BaseServlet> cls = this.getClass();

//2.2.获取方法的method的对象

try {

Method method = cls.getMethod(methodname,HttpServletRequest.class,HttpServletResponse.class);

//2.3.执行方法

method.invoke(this,req,resp);

} catch (NoSuchMethodException | IllegalAccessException | InvocationTargetException e) {

e.printStackTrace();

}

}

}