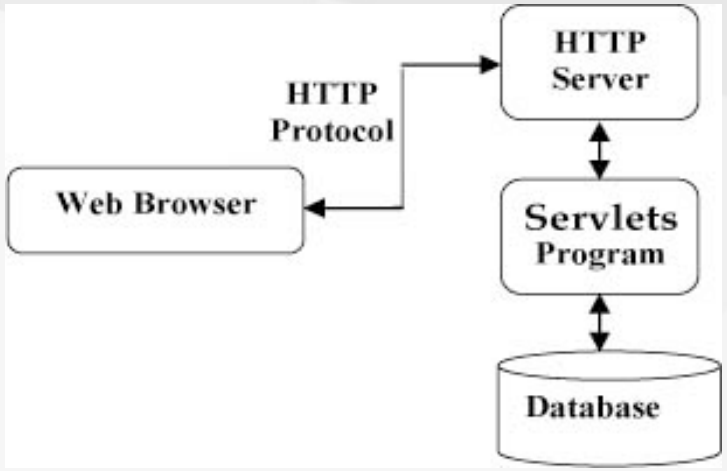
Servlet

浙江大学城市学院
彭 彬
pengb@zucc.edu.cn

Servlet是什么

Java Servlet 是运行在 Web 服务器或应用服务器上的程序，它是作为来自 Web 浏览器或其他 HTTP 客户端的请求和 HTTP 服务器上的数据库或应用程序之间的中间层。



Java Servlet 是运行在带有支持 Java Servlet 规范的 web 服务器上的 Java 类。当前主要的Java应用服务器有：**Tomcat**、**Jetty**、GlassFish、Jboss、WebLogic、WebSphere

http://www.runoob.com/servlet/servlet-intro.html

Servlet是什么

我们通过标准文档来理解一下Servlet是什么？

javax.servlet

Interface Servlet

All Known Subinterfaces:

`HttpJspPage`, `JspPage`

All Known Implementing Classes:

`FacesServlet`, `GenericServlet`, `HttpServlet`

`public interface Servlet`

Defines methods that all servlets must implement.

A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.

To implement this interface, you can write a generic servlet that extends `javax.servlet.GenericServlet` or an HTTP servlet that extends `javax.servlet.http.HttpServlet`.

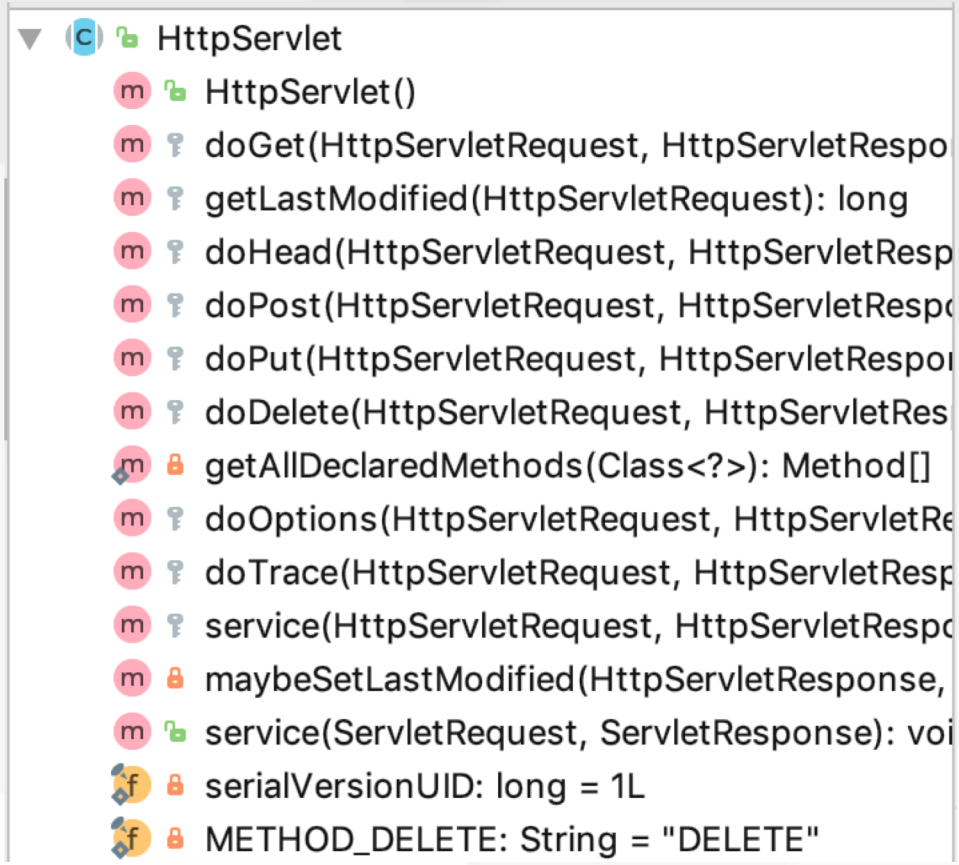
This interface defines methods to initialize a servlet, to service requests, and to remove a servlet from the server. These are known as life-cycle methods and are called in the following sequence:

1. The servlet is constructed, then initialized with the `init` method.
2. Any calls from clients to the `service` method are handled.
3. The servlet is taken out of service, then destroyed with the `destroy` method, then garbage collected and finalized.

https://docs.oracle.com/javaee/7/api/javax/servlet/Servlet.html

Servlet是什么

我们通过接口进一步理解HttpServlet是什么，基于Servlet机制实现了和HTTP协议方法对应的各种响应入口



而最基本的Servlet是一个接口，具有非常简洁的定义

HttpServlet是什么

一个HttpServlet的实现如下，响应GET方法，并且返回一个最简单的Html文档

```
@WebServlet(name = "Echo",urlPatterns = { "/echo"},loadOnStartup=1)
public class Echo extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        String browser = request.getHeader( s: "User-Agent");
        response.setStatus(HttpServletResponse.SC_OK);  // default
        response.setContentType("text/html");       // default
        PrintWriter out = response.getWriter();
        out.println("<HTML><HEAD><TITLE>Simple servlet");
        out.println("</TITLE></HEAD><BODY>");
        out.println ("Browser details: " + browser);
        out.println("</BODY></HTML>");
    }
}
```

一个最简单的Java Web应用发布

这里使用了标注（Annotation）进行配置

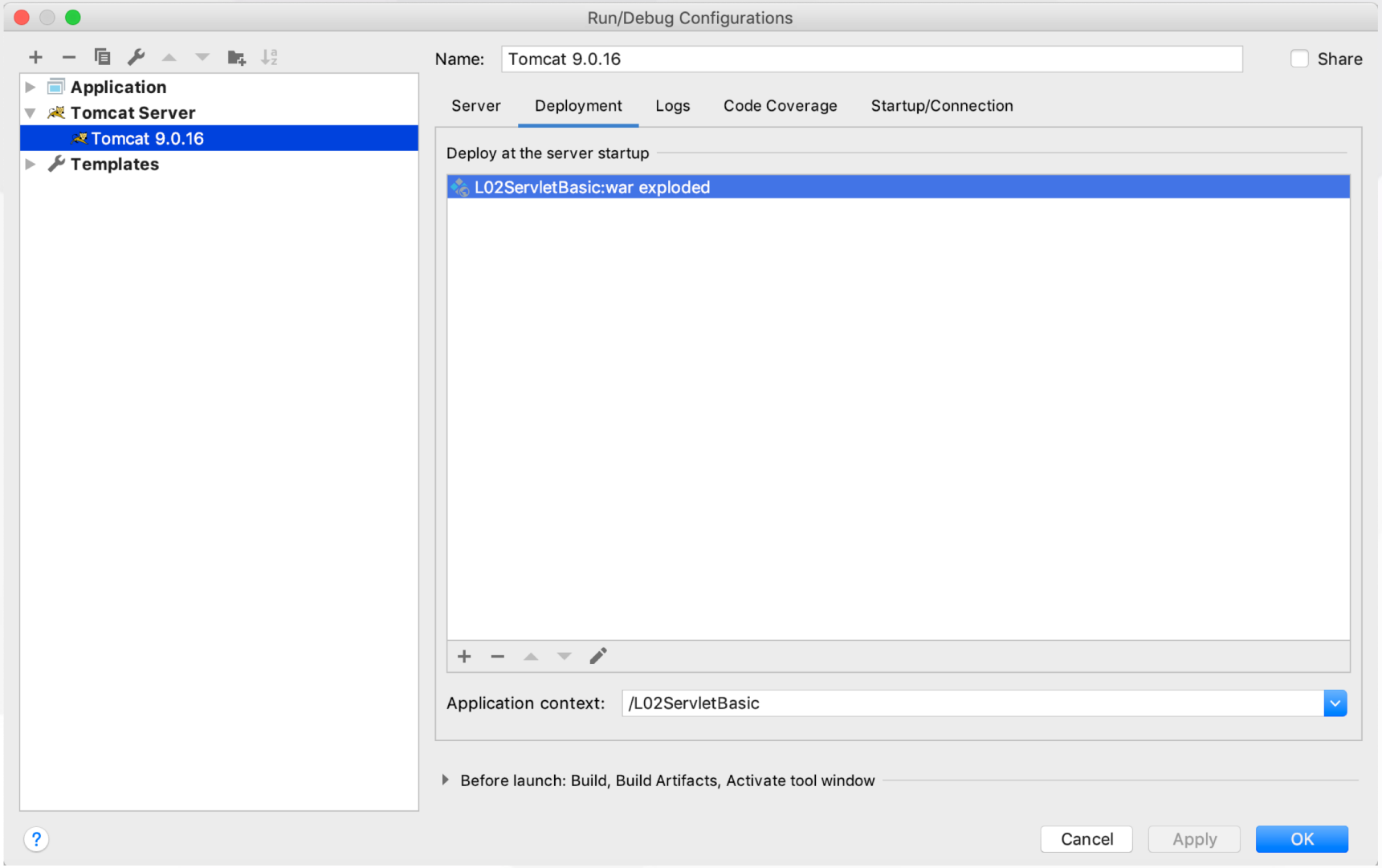
```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet(name = "Echo",urlPatterns = { "/echo"},loadOnStartup=1)
public class Echo extends HttpServlet {
```

这里使用了标注（Annotation）进行配置

An annotation is a form of metadata, that can be added to Java source code. Classes, methods, variables, parameters and packages may be annotated. Annotations have no direct effect on the operation of the code they annotate.

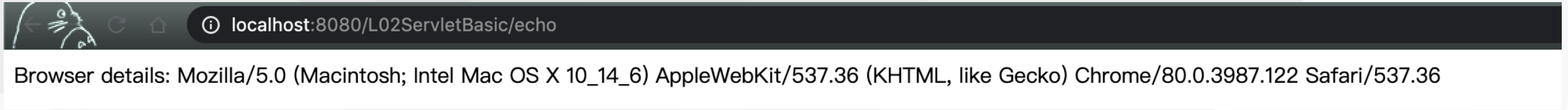
一个最简单的Java Web应用发布~Application Context

发布应用的根路径配置(以IDEA为例，配置运行环境中的Application Context进行指定



一个最简单的Java Web应用发布

注意发布路径,我这里的发布路径是”/L02ServiceBasic”，所以访问路径是/L02ServiceBasic/echo



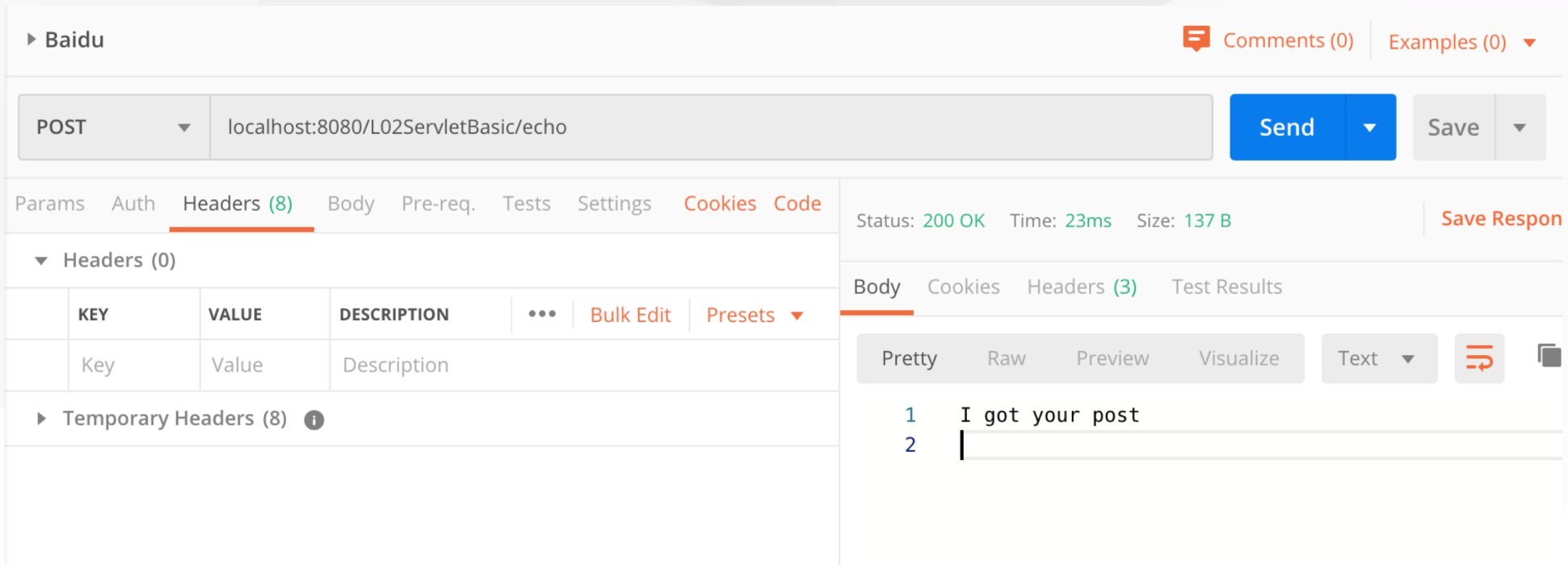
Servlet路径： echo

```
@WebServlet(name = "Echo",urlPatterns = { "/echo"},loadOnStartup=1)
public class Echo extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    }
```

向HttpServlet进行Post

为HttpServlet的实现增加Post方法的处理方法

```java
@WebServlet(name = "Echo",urlPatterns = { "/echo"},loadOnStartup=1)
public class Echo extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) thro
        response.setStatus(HttpServletResponse.SC_OK);
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("I got your post");
    }
```



需要容器的Servlet

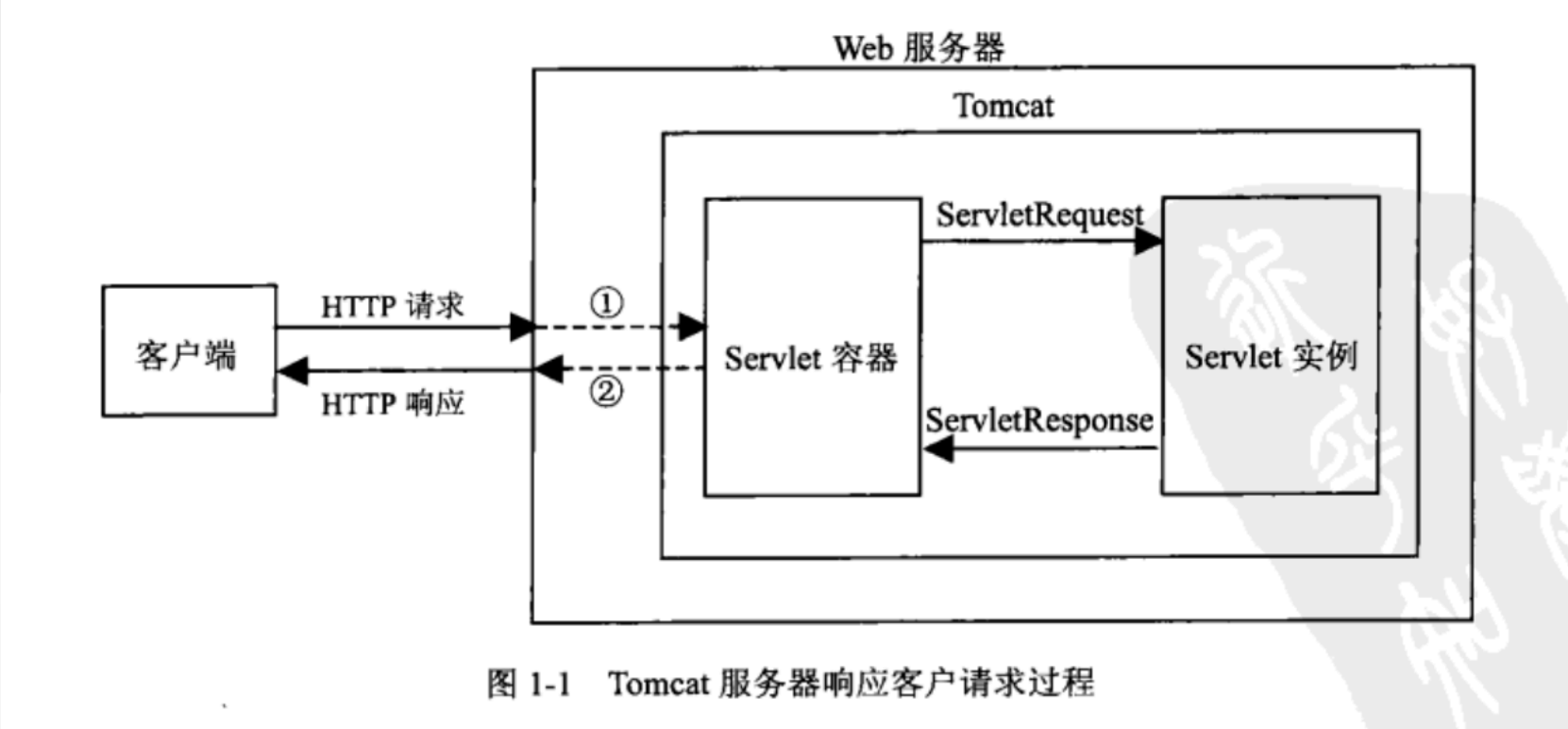


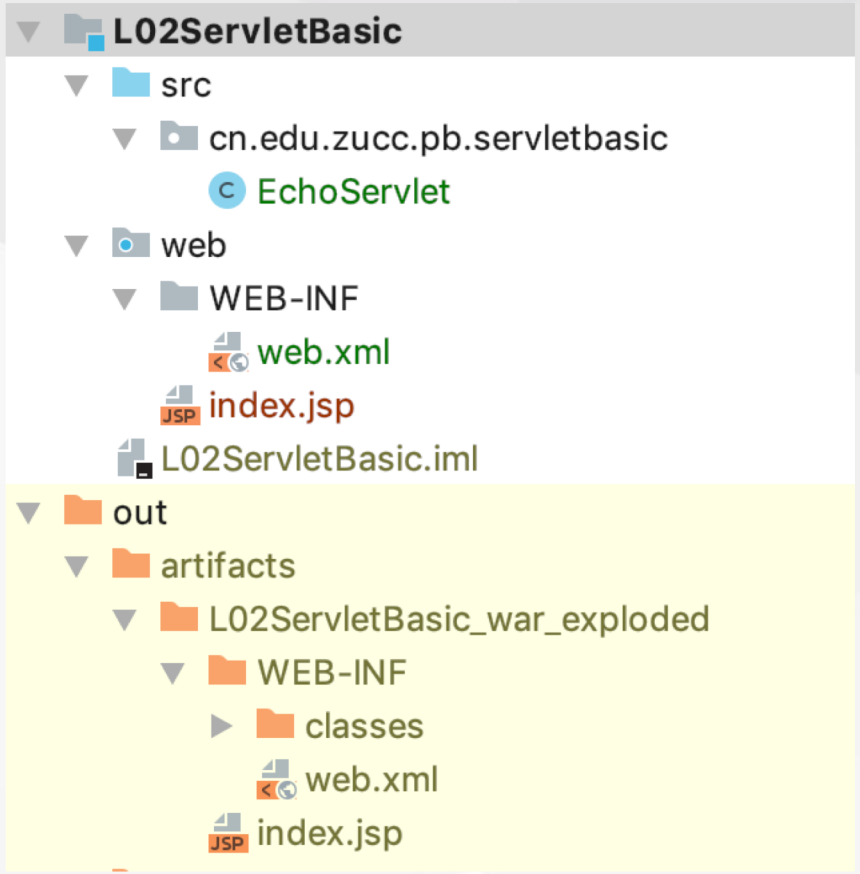
图 1-1　Tomcat 服务器响应客户请求过程

1）Tomcat将http请求文本接收并解析，然后封装成HttpServletRequest类型的request对象，所有的HTTP头数据读可以通过request对象调用对应的方法查询到。

2）Tomcat同时会要响应的信息封装为HttpServletResponse类型的response对象，通过设置response属性就可以控制要输出到浏览器的内容，然后将response交给tomcat，tomcat就会将其变成响应文本的格式发送给浏览器

一个最简单的Java Web应用的工程结构（idea为例）

工程中对应源代码，Web目录



Src:java源代码

Web：web根目录（开发用）

Web/WEB-INF: web配置目录（开发/调试/部署用）

Web/WEB-INF/web.xml:应用程序部署描述文件（开发/调试/部署用）

Web/WEB-INF/classes:应用程序类文件（部署用）

Web/WEB-INF/libs:应用程序依赖库文件（部署用）

Web/*/*.jsp:jsp页面文件

Web/ */ *.html等:其他静态资源文件

END

Pb&Lois