Web应用开发实验（三）

姓名：叶倩琳

学号：201706061330

完成日期：2019/4/30

# 实验目的

1. 掌握web应用程序的目录结构，servlet注解和web.xml文件的结构。
2. 理解HTTP协议是一个无状态的协议。
3. 掌握会话概念及其管理机制和HttpSession接口的使用方法。
4. 掌握Cookie的使用方法。

# 实验内容

1. 在Tomcat服务器上配置一个名称为zjut的Web应用。
2. 运行测试书本程序3.3（P66），然后仔细阅读其源代码。
3. 修改程序3.3，实现如下功能：
   1. 在向客户端响应的页面上输出**需要猜测的数**；
   2. 在向客户端响应的页面上输出**用户猜测的数**；
   3. 在向客户端响应的页面上输出会话ID以及是否为新会话（**参考书本程序3.2**）。
   4. 设置猜测时间最长为20秒，如果超时，则提示用户“Time out, you lost”；
4. 源代码

/\*

1)在向客户端响应的页面上输出需要猜测的数；

2)在向客户端响应的页面上输出用户猜测的数；

3)在向客户端响应的页面上输出会话ID以及是否为新会话（参考书本程序3.2）。

4)设置猜测时间最长为20秒，如果超时，则提示用户“Time out, you lost”；

\*/

package com.example.zjut03;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.WebServlet;

@WebServlet("/GuessNumberServlet")

public class GuessNumberServlet extends HttpServlet{

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

int magic = (int)(Math.*random*()\*101);

HttpSession session = request.getSession();

session.setMaxInactiveInterval(20); //设置session生命周期为20s

session.setAttribute("num",new ~~Integer~~(magic));

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("我想出一个0到100之间的数，请你猜！");

out.println("<form action='/zjut03/GuessNumberServlet' method='post'>");

out.println("<input type='text' name='guess' />");

out.println("<input type='submit' value='确定'/>");

out.println("</form>");

out.println("该会话是否为新会话？"+(session.isNew() ? "YES" : "NO")+"<br>");

out.println("</body></html>");

}

public void doPost(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

int guess = Integer.*parseInt*(request.getParameter("guess")); //要进行强制类型转换

HttpSession session = request.getSession(true);

int magic = (Integer)session.getAttribute("num");

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

out.println("<html><body>");

//判断session是否超时

if(request.getSession(false)==null) {

out.println("Sorry!Time out, you lost!"+"<br>");

}else {

out.println("Session is active!"+"<br>");

}

out.println("会话ID为："+(String)session.getId()+"<br>");

out.println("该会话是否为新会话？"+(session.isNew() ? "YES" : "NO")+"<br>");

out.println("-------------------------------------"+"<br>");

if(guess==magic){

session.invalidate(); // 销毁会话对象

out.println("祝贺你，答对了!");

out.println("<a href='/zjut03/GuessNumberServlet'>再猜一次.</a>");

}else if(guess>magic){

out.println("太大了! 请重猜!");

}else{

out.println("太小了! 请重猜!");

}

out.println("<form action='/zjut03/GuessNumberServlet'method='post'>");

out.println("<input type='text' name='guess' />");

out.println("<input type='submit' value='确定'/>");

out.println("</form>");

out.println("-------------------------------------"+"<br>");

out.println("需要猜测的数为："+magic+"<br>");

out.println("用户猜测的数为："+guess+"<br>");

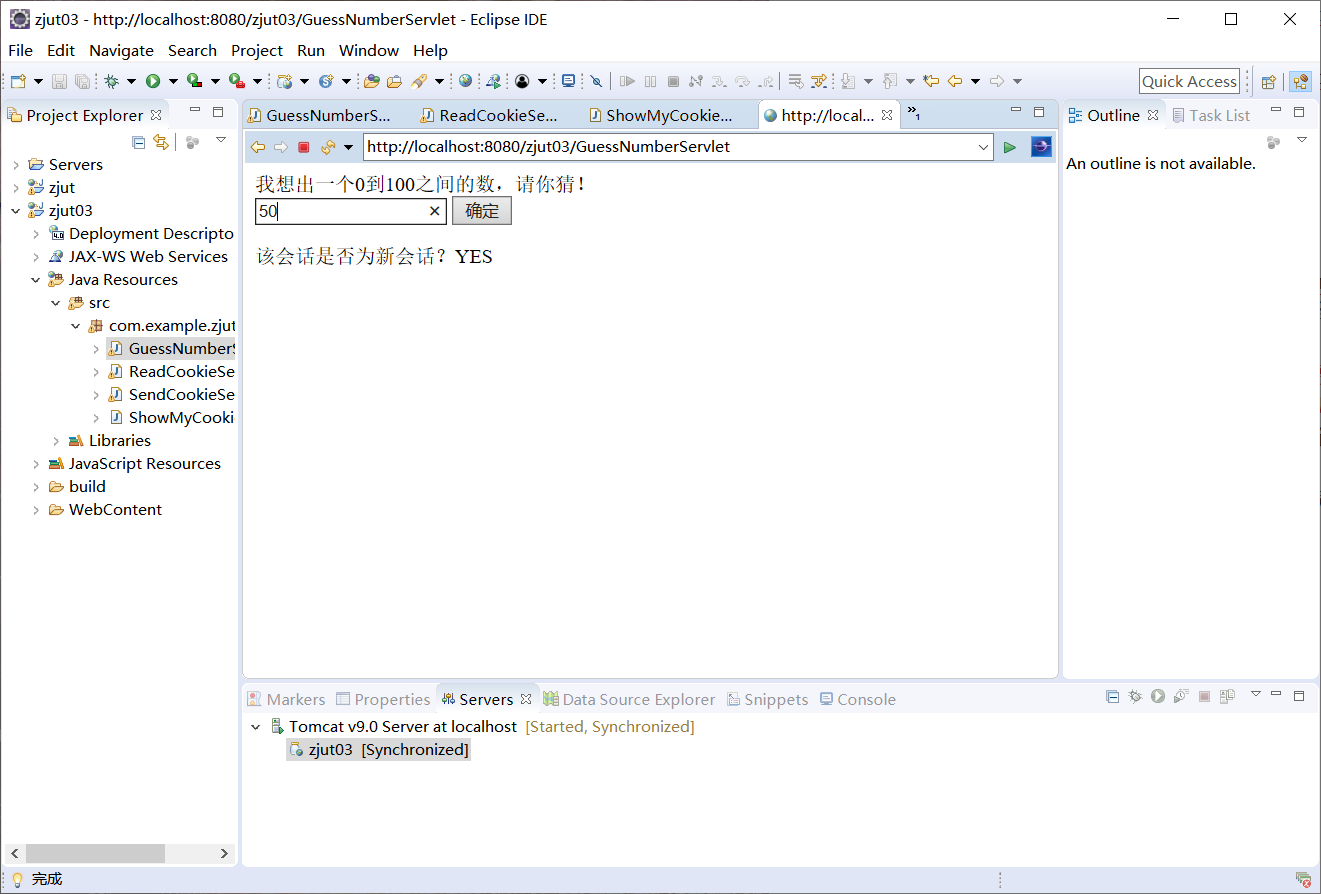
out.println("</body></html>");

}

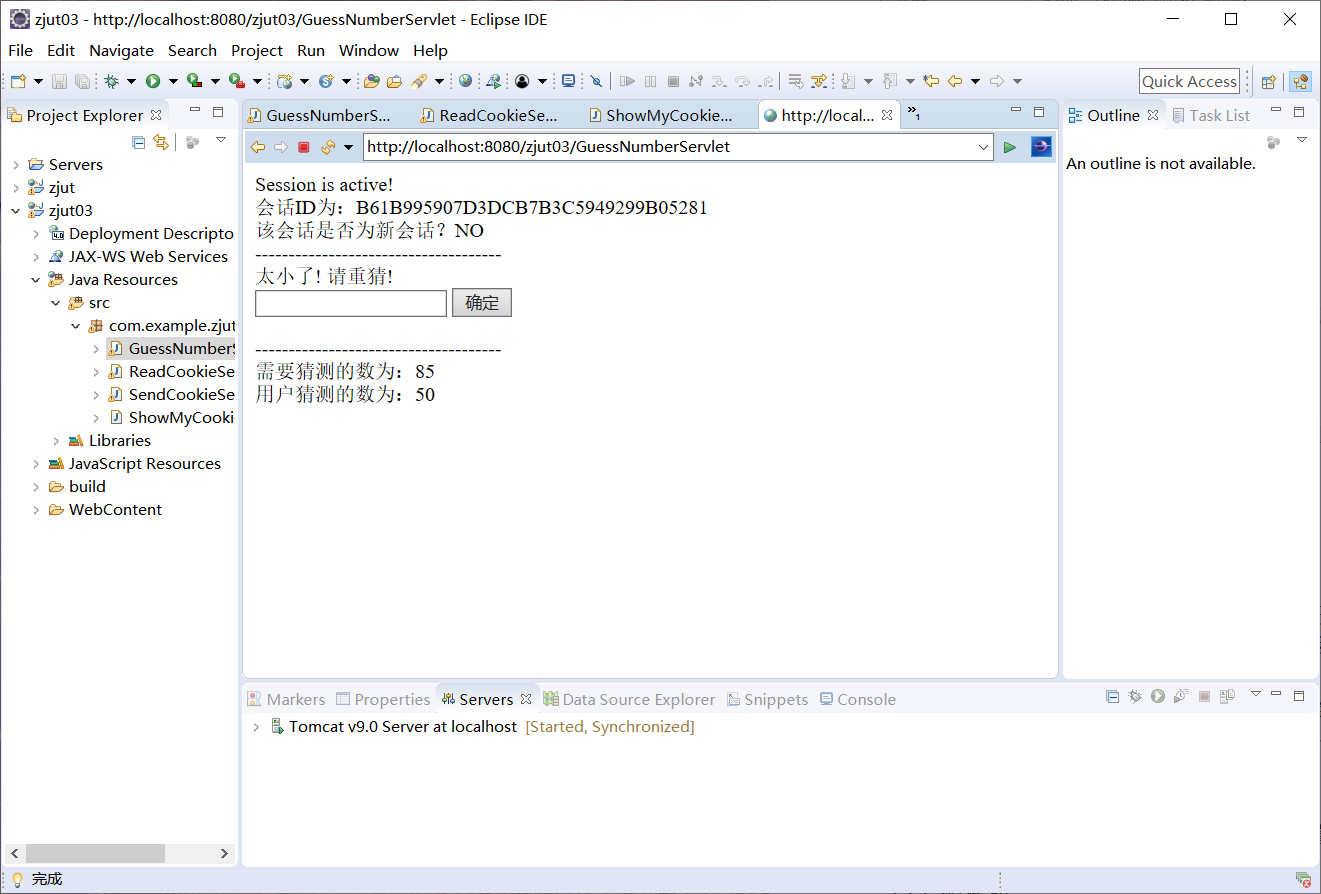
}

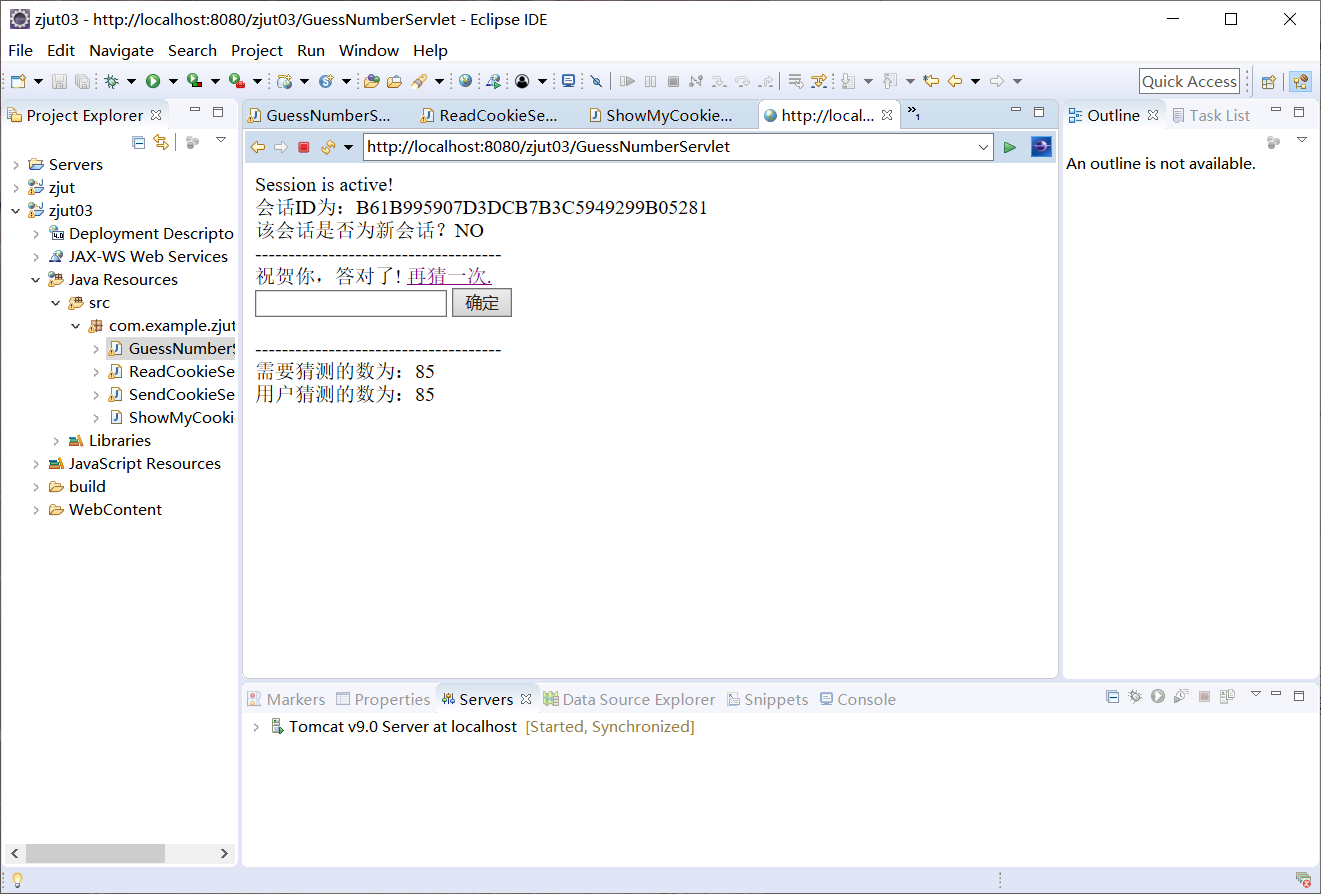
(2)实验截屏

新创建会话为新会话，即session.isNew()==true

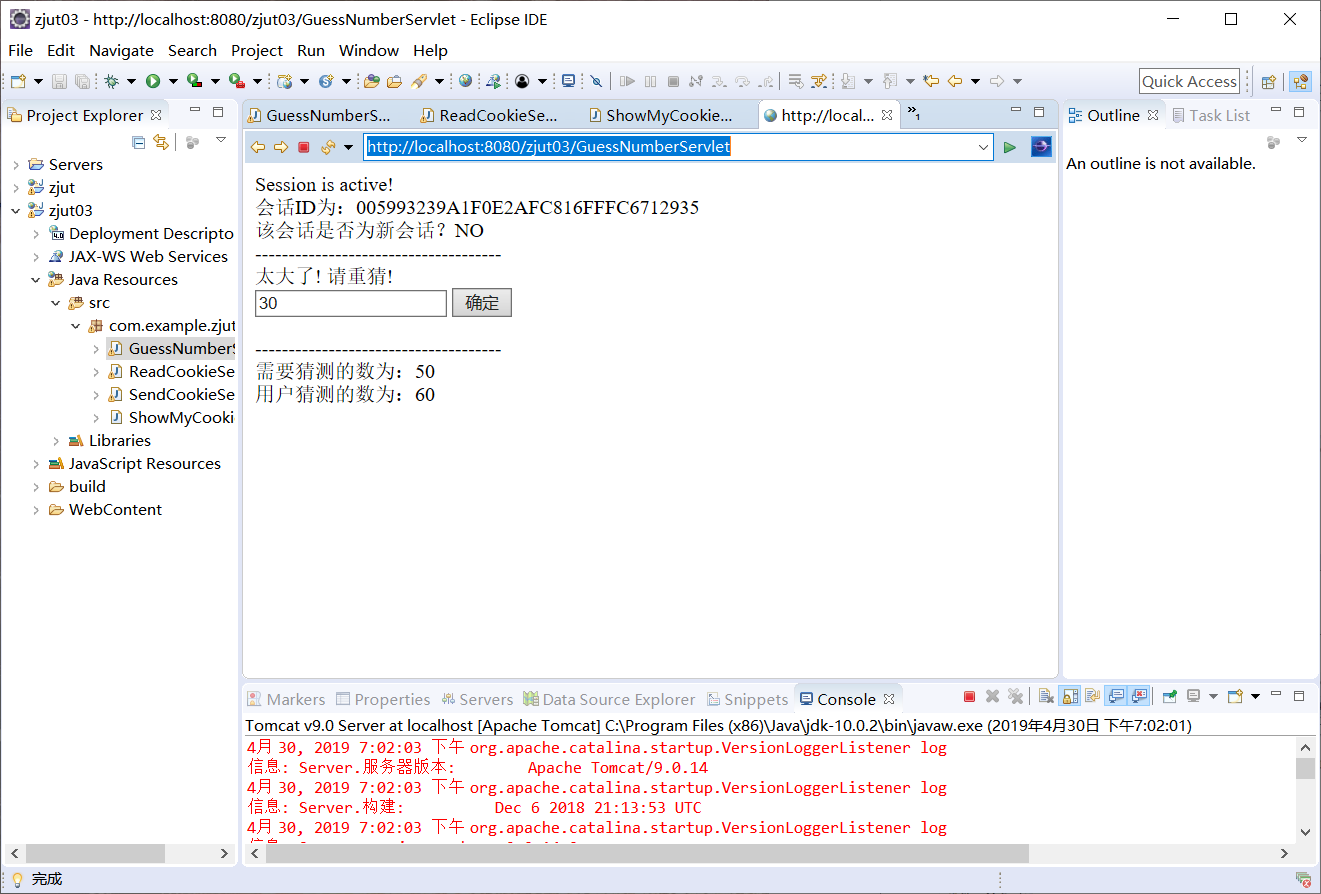


同一个session中，会话ID不变。





“再猜一次”后，新建会话，会话ID不同。



1. 运行书本程序3.4和3.5（P68），然后仔细阅读其源代码。
2. 参考程序3.4和3.5，编写Servlet（名称为ShowMyCookies.java），具体功能描述如下：
   1. 首先读取客户端的Cookie，然后判断是否存在一个名称为“myCookie”的Cookie，如果存在则输出其值；如果不存在则设置一个名称为“myCookie”，值为“Fedor”的Cookie，并提示用户“cookie has been set”。
   2. 调整IE浏览器的Cookie安全设置，然后运行调试本程序。
3. 源代码

package com.example.zjut03;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.WebServlet;

@WebServlet("/SendCookie")

public class SendCookieServlet extends HttpServlet {

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

Cookie userCookie = new Cookie("username", "hacker");

userCookie.setMaxAge(60 \* 60 \* 24 \* 7);

response.addCookie(userCookie);

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

out.println("<html><title>发送Cookie</title>");

out.println("<body><h3>已向浏览器发送一个Cookie。</h3></body>");

out.println("</html>");

}

}

/\* 1)首先读取客户端的Cookie，然后判断是否存在一个名称为“myCookie”的Cookie，如果存在则输出其值；

如果不存在则设置一个名称为“myCookie”，值为“Fedor”的Cookie，并提示用户“cookie has been set”。

2)调整IE浏览器的Cookie安全设置，然后运行调试本程序。\*/

package com.example.zjut03;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet("/ShowMyCookies")

public class ShowMyCookies extends HttpServlet {

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

String cookieName = "myCookie";

String cookieValue = null;

// 先用数组获取所有的Cookies

Cookie[] cookies = request.getCookies();

if (cookies != null) {

for (int i = 0; i < cookies.length; i++) {

Cookie cookie = cookies[i];

// 判断是否存在一个名称为“myCookie”的Cookie

if (cookie.getName().equals(cookieName))

cookieValue = cookie.getValue();

}

}

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

out.println("<html><title>读取Cookie</title>");

if (cookieValue != null) {

out.println("<body><h3>从浏览器读回一个名为“myCookie”的Cookie</h3>");

out.println("Cookie名:" + cookieName + "<br>");

out.println("Cookie值:" + cookieValue + "<br>");

}

//如果不存在则设置一个名称为“myCookie”，值为“Fedor”的Cookie，并提示用户“cookie has been set”

else {

out.println("<body><h3>浏览器中不存在一个名为“myCookie”的Cookie</h3>");

Cookie userCookie = new Cookie("myCookie", "Fedor");

userCookie.setMaxAge(60 \* 60 \* 24 \* 7);

response.addCookie(userCookie);

out.println("Cookie has been set!"+"<br>");

}

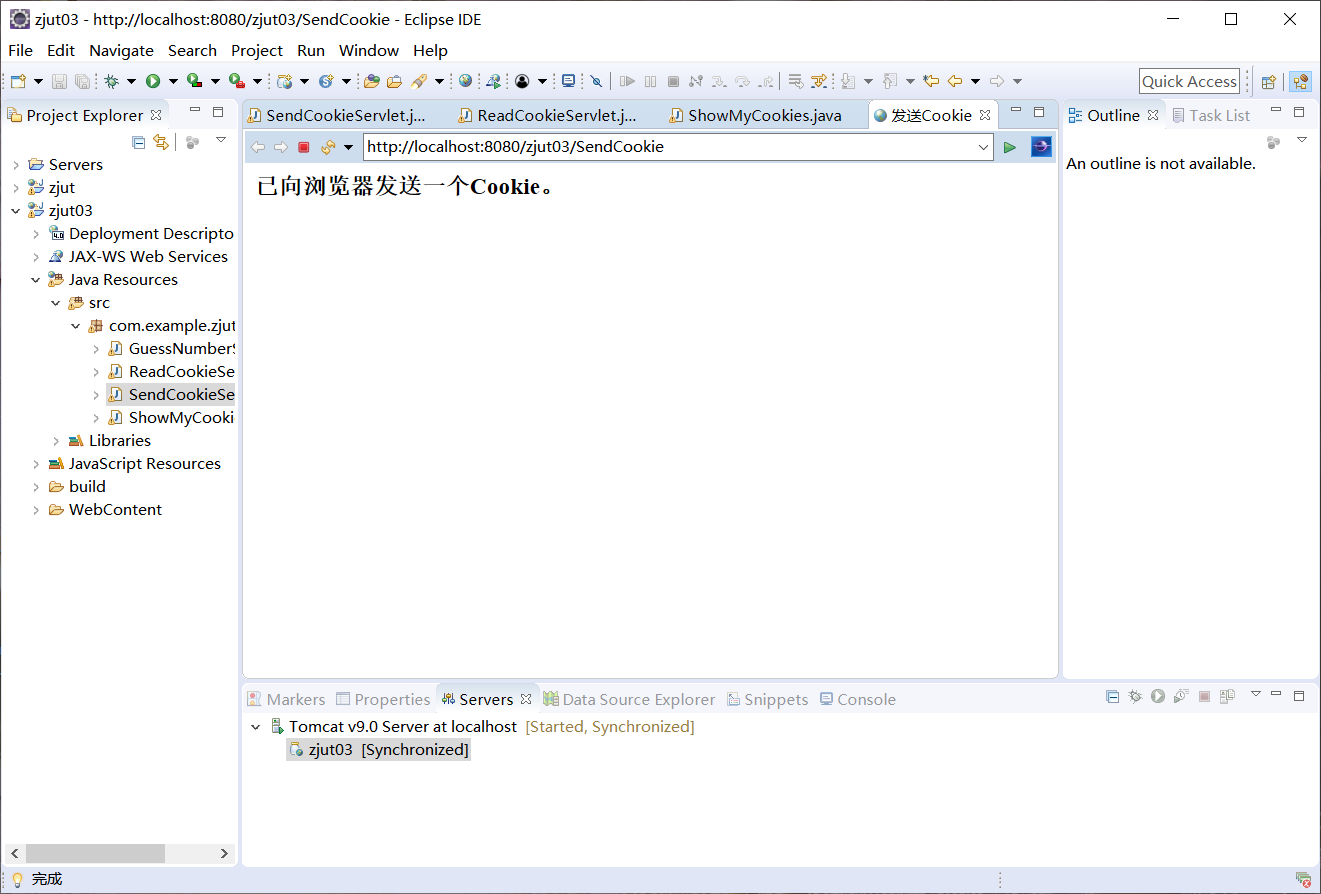
out.println("</body></html>");

}

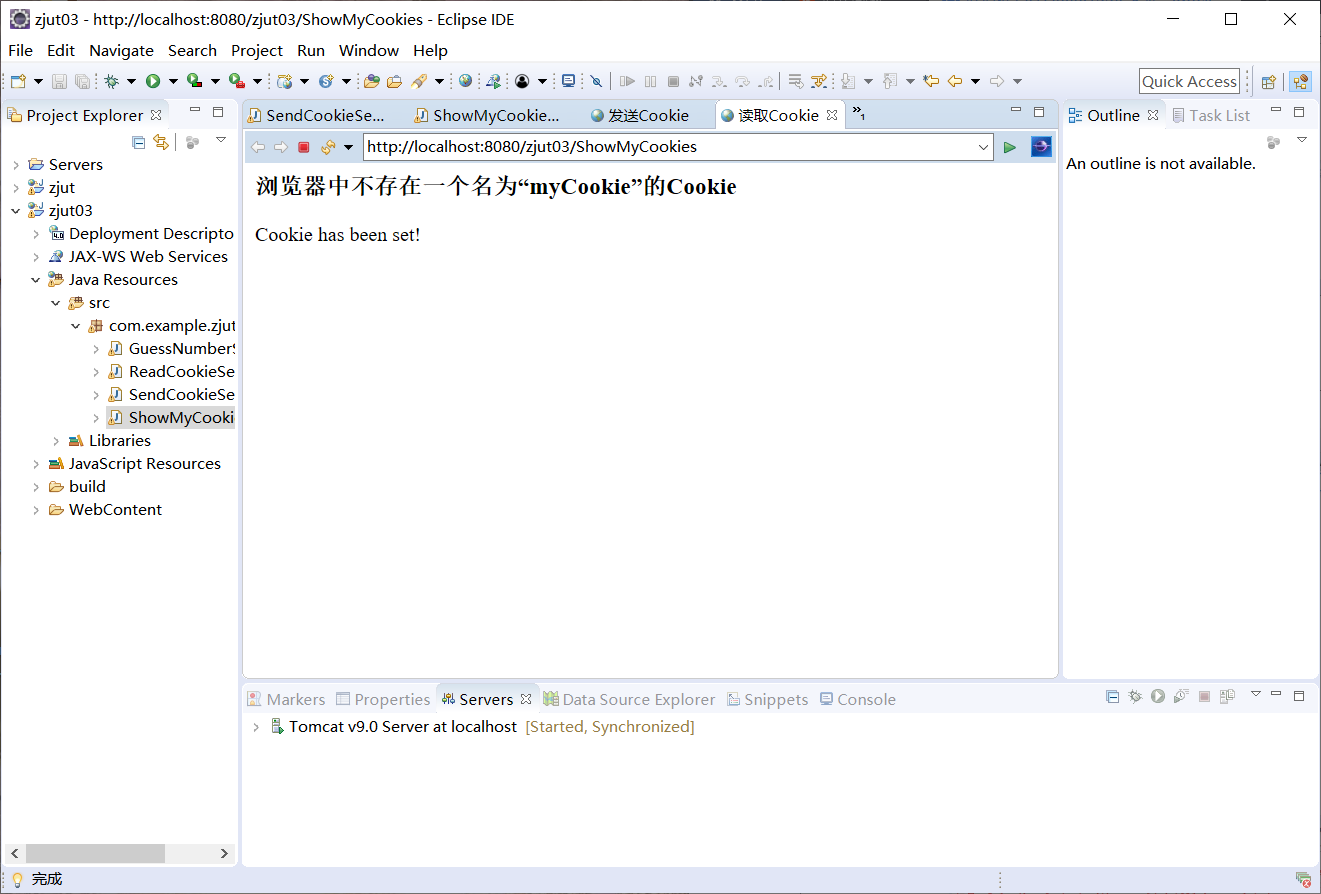
}

1. 实验截屏

SendCookieServlet向浏览器发送一个名为”username”的cookie。



第一次访问ShowMyCookies，不存在一个名为“myCookie”的Cookie，执行写入操作



第二次访问ShowMyCookies，因为第一次访问时写入了”myCookie“，因此存在一个名为“myCookie”的Cookie，返回其Cookie值

