**实验七 Servlet应用**

## 一、实训目标

1. 掌握编写和配置Servlet、调用Servlet的方法；

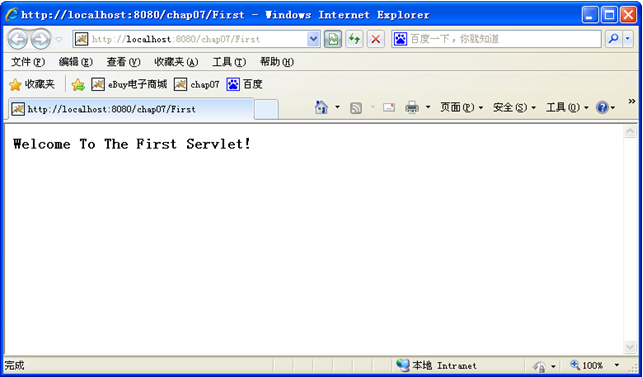
2. 掌握Servlet读取HTML表单数据的方法；

3. 掌握Servlet读取Cookie数据的方法；

4. 掌握Servlet读取Session数据的方法。

**二、实训内容**

1. 编写、配置和调用显示“Welcomet To Servlet！”的Servlet。运行效果如图1所示。



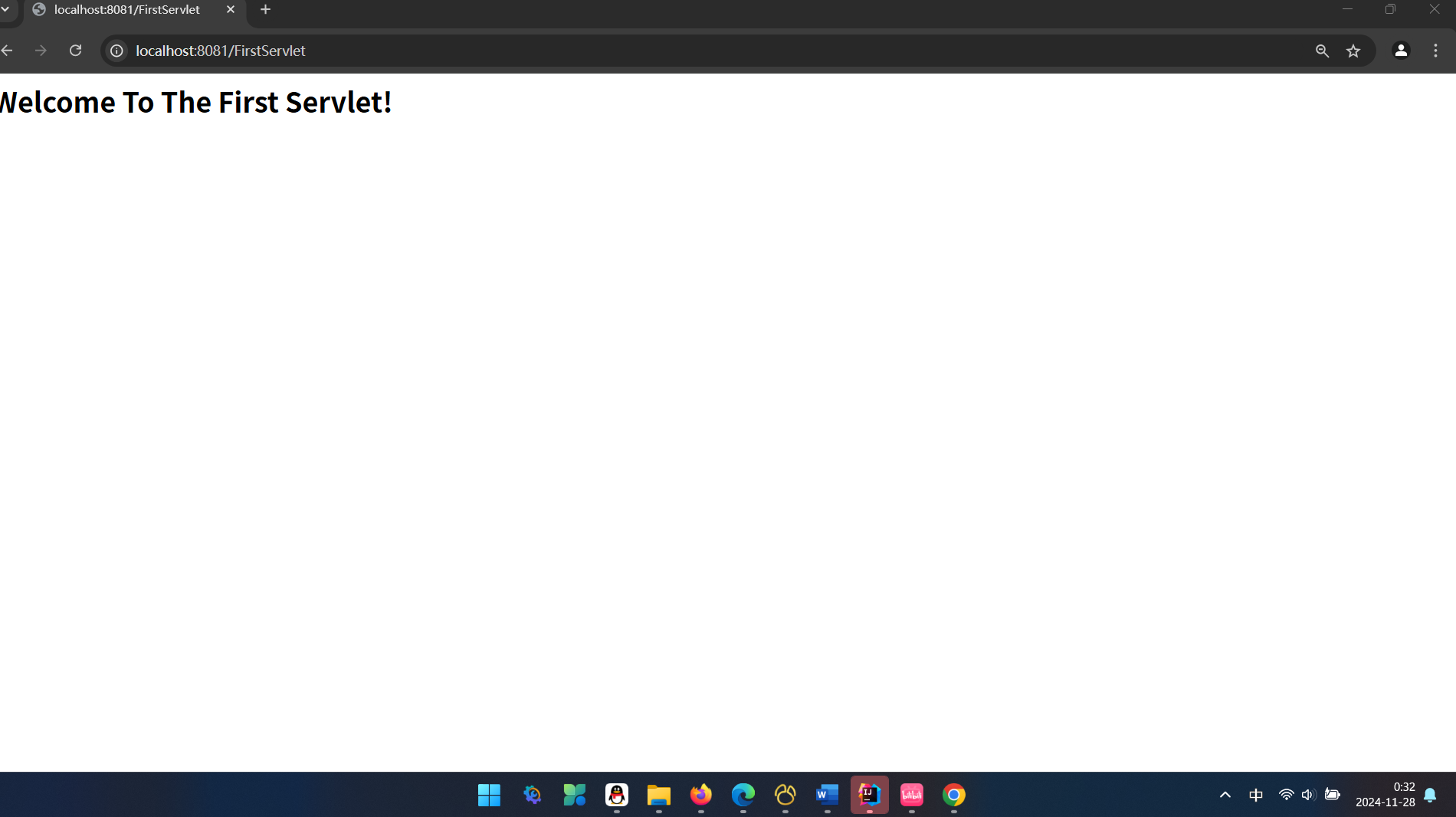
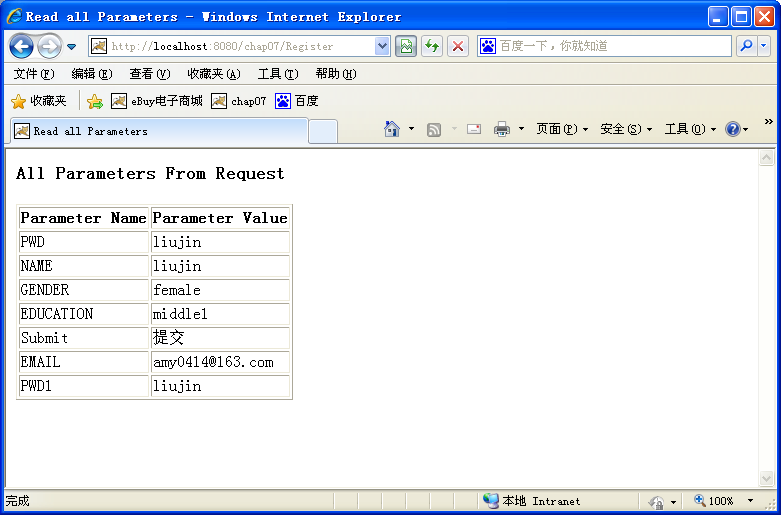
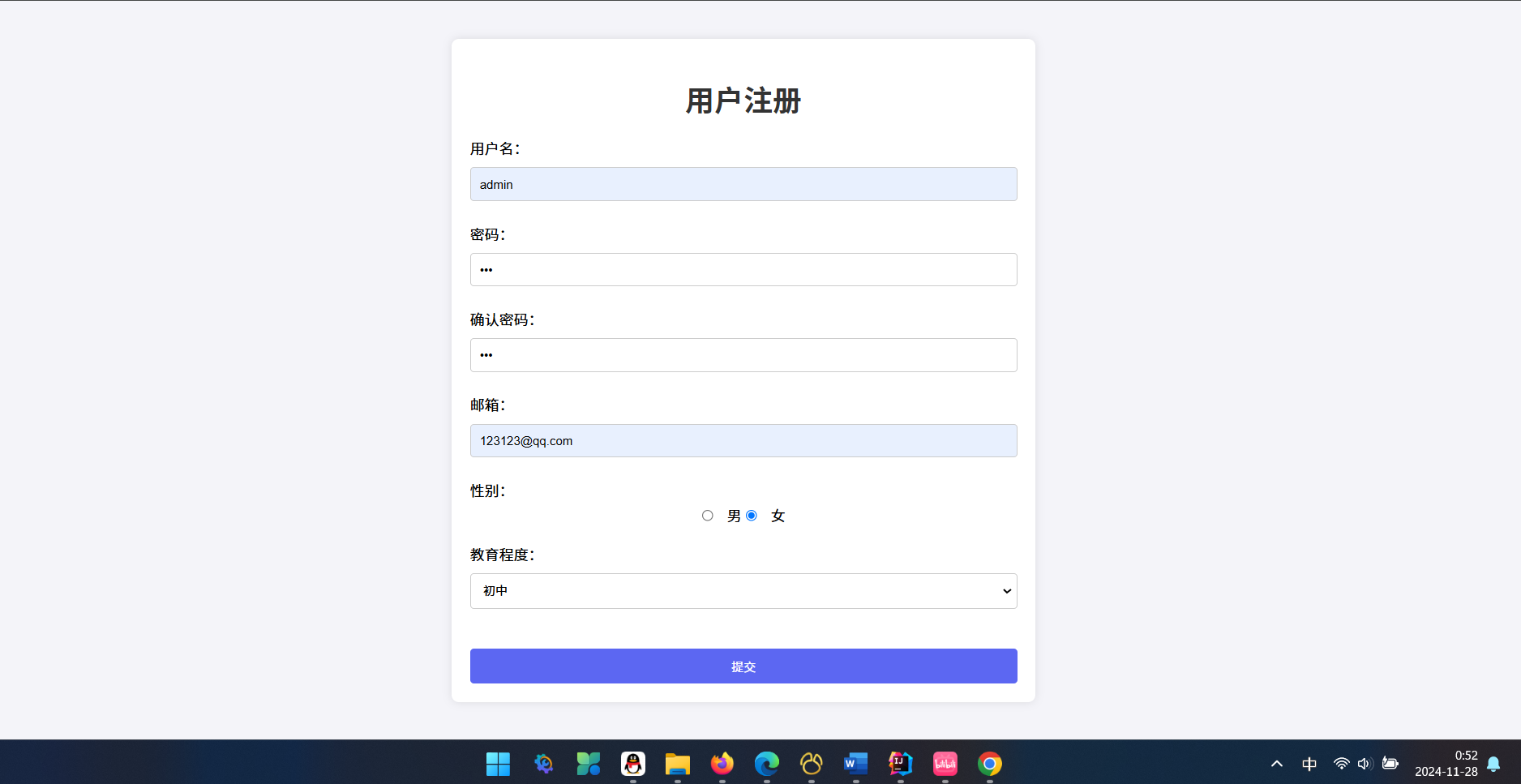


图1 FirstServlet运行结果

|  |
| --- |
| 此处插入FirstServlet.java文件的代码。  import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import java.io.IOException;  @WebServlet("/FirstServlet") public class FirstServlet extends HttpServlet {   @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  resp.setContentType("text/html"); // 设置响应类型为HTML  resp.getWriter().println("<h1>Welcome To The First Servlet!</h1>");  }   @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  super.doPost(req, resp);  } } |

2. 应用Servlet读取所有HTML表单数据。运行效果如图2所示。



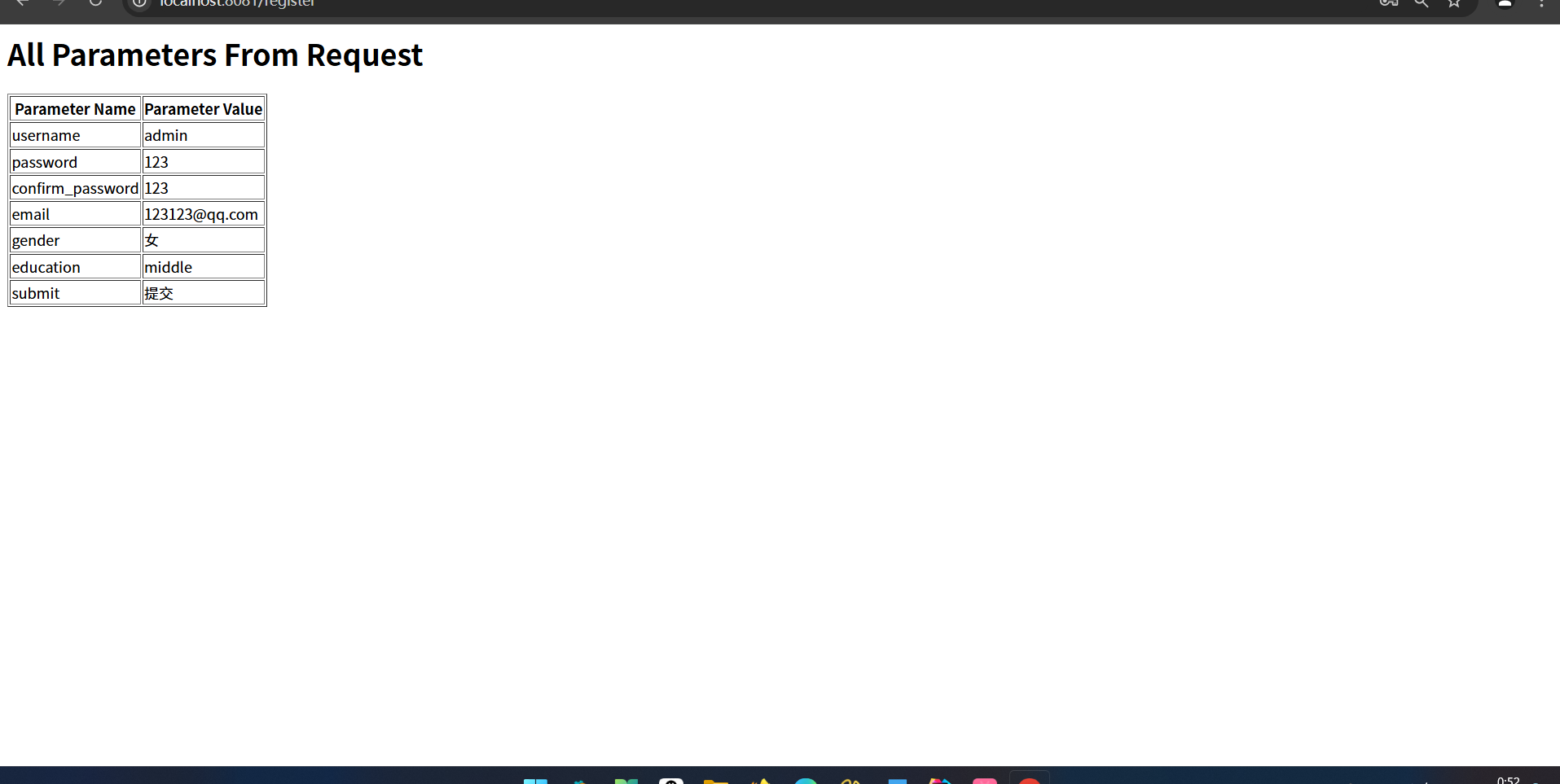


图2 运行结果

|  |
| --- |
| 此处插入RegisterServlet.java文件代码和register.html文件代码  package Servlet; import javax.servlet.ServletException; 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; import java.util.Enumeration;  @WebServlet("/register") public class RegisterServlet extends HttpServlet {   @Override  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  // 设置响应内容类型和字符编码  request.setCharacterEncoding("UTF-8");  response.setContentType("text/html;charset=UTF-8");  response.setCharacterEncoding("utf-8");   // 创建 PrintWriter 对象  PrintWriter pw = response.getWriter();   // 开始写入 HTML 内容  pw.println("<html><body>");   // 标题  pw.println("<h1>All Parameters From Request</h1>");   // 表头  pw.println("<table border=\"1\">");  pw.println("<tr><th>Parameter Name</th><th>Parameter Value</th></tr>");   // 遍历所有参数  Enumeration<String> parameterNames = request.getParameterNames();  while (parameterNames.hasMoreElements()) {  String paramName = parameterNames.nextElement();  String[] paramValues = request.getParameterValues(paramName);   for (String paramValue : paramValues) {  // 每个参数值一行  pw.println("<tr><td>" + paramName + "</td><td>" + paramValue + "</td></tr>");  }  }   // 结束表格和 HTML 文档  pw.println("</table>");  pw.println("</body></html>");   // 关闭 PrintWriter  pw.close();  } }  <%@ **page** contentType="text/html;charset=UTF-8" language="java" %> <!DOCTYPE html> <html lang="zh-CN"> <head>  <meta charset="UTF-8">  <title>用户注册 - Windows Internet Explorer</title>  <style>  body {  font-family: Arial, sans-serif;  background-color: #f4f4f9;  margin: 0;  padding: 0;  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  }   .container {  background: white;  padding: 20px;  border-radius: 8px;  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  width: 600px;  text-align: center;  }   h1 {  color: #333;  margin-bottom: 20px;  }   form {  display: flex;  flex-direction: column;  }   input[type="text"],  input[type="password"],  input[type="email"],  select {  padding: 10px;  margin: 5px 0;  border: 1px solid #ccc;  border-radius: 4px;  }   input[type="radio"] {  margin: 5px 10px 5px 0;  }   input[type="submit"] {  background: #5c67f2;  color: white;  padding: 10px;  border: none;  border-radius: 4px;  cursor: pointer;  margin-top: 20px;  }   input[type="submit"]:hover {  background: #4a54e1;  }   label {  margin-bottom: 5px;  text-align: left;  }  </style> </head> <body> <div class="container">  <h1>用户注册</h1>  <form action="/register" method="post">  <label for="username">用户名：</label>  <input type="text" id="username" name="username" required><br>   <label for="password">密码：</label>  <input type="password" id="password" name="password" required><br>   <label for="confirm\_password">确认密码：</label>  <input type="password" id="confirm\_password" name="confirm\_password" required><br>   <label for="email">邮箱：</label>  <input type="email" id="email" name="email" required><br>   <label for="gender">性别：</label>  <span>  <input type="radio" id="male" name="gender" value="男" required> 男  <input type="radio" id="female" name="gender" value="女" required> 女  </span>  <br>  <label for="education">教育程度：</label>  <select id="education" name="education" required>  <option value="">请选择</option>  <option value="primary">小学</option>  <option value="middle">初中</option>  <option value="high">高中</option>  <option value="college">大学</option>  </select><br>   <input type="submit" value="提交" name="submit">  </form> </div> </body> </html> |

3. 应用Servlet读取Cookie数据。运行结果如图3所示。

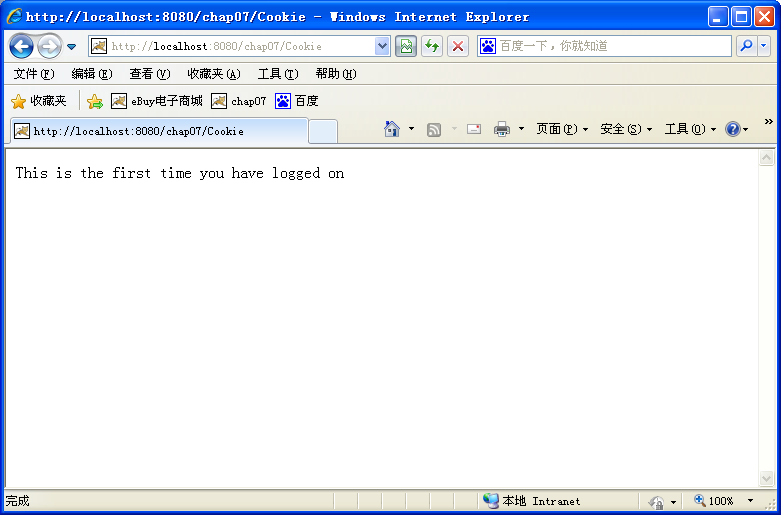
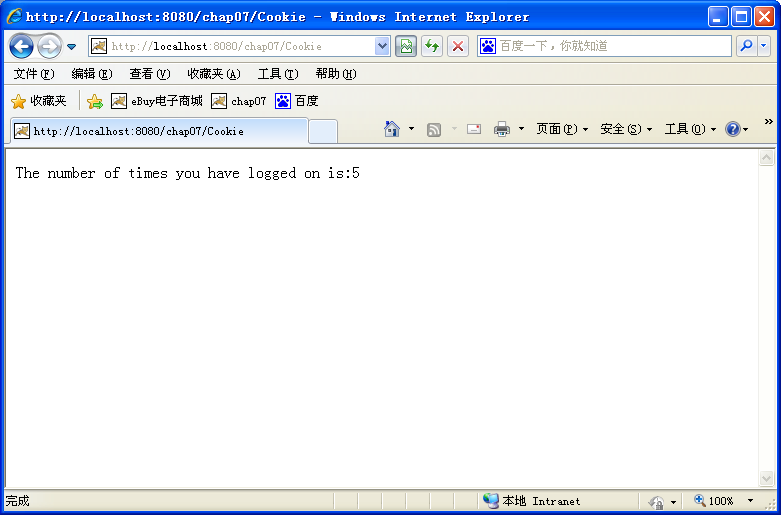
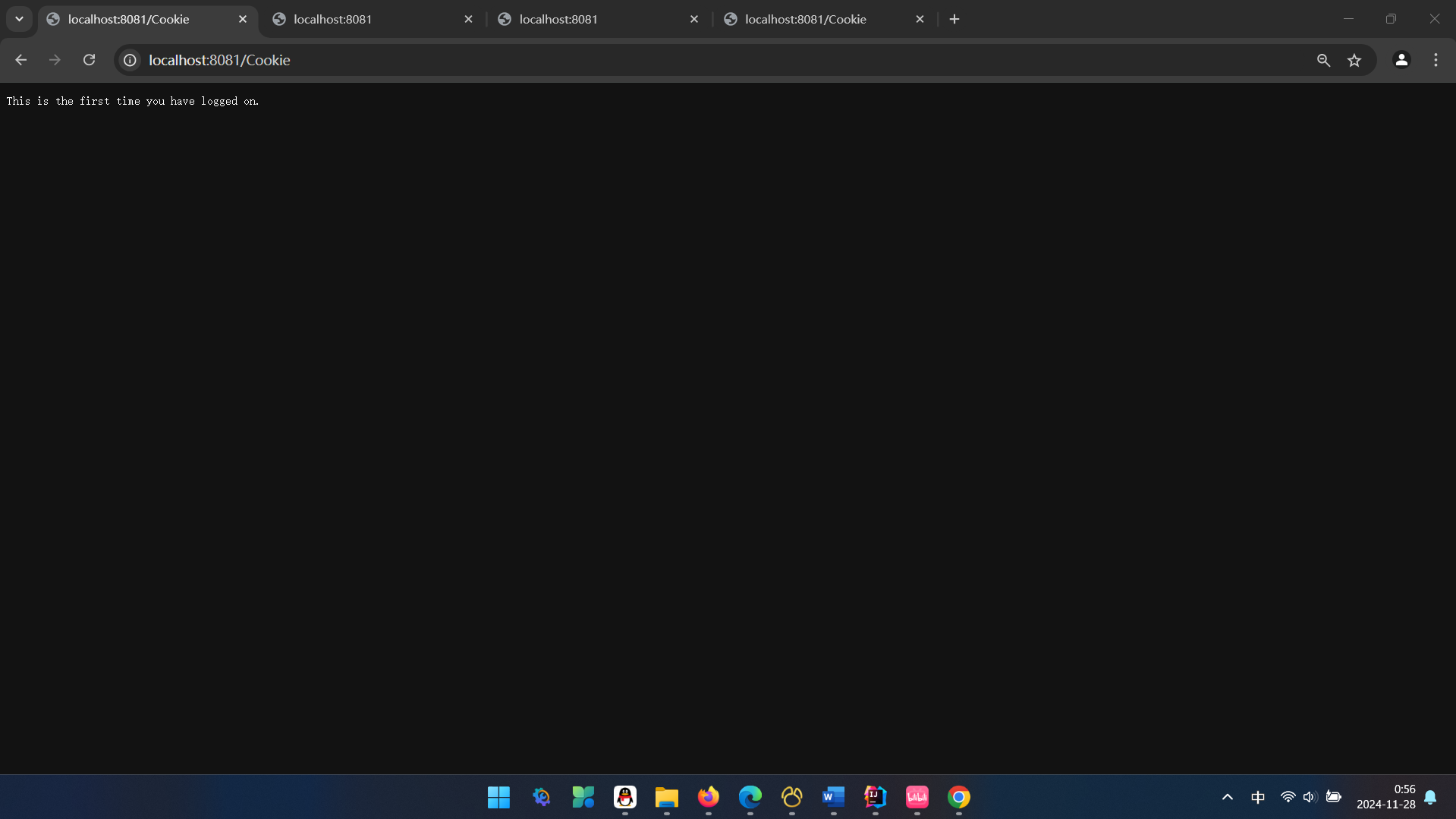
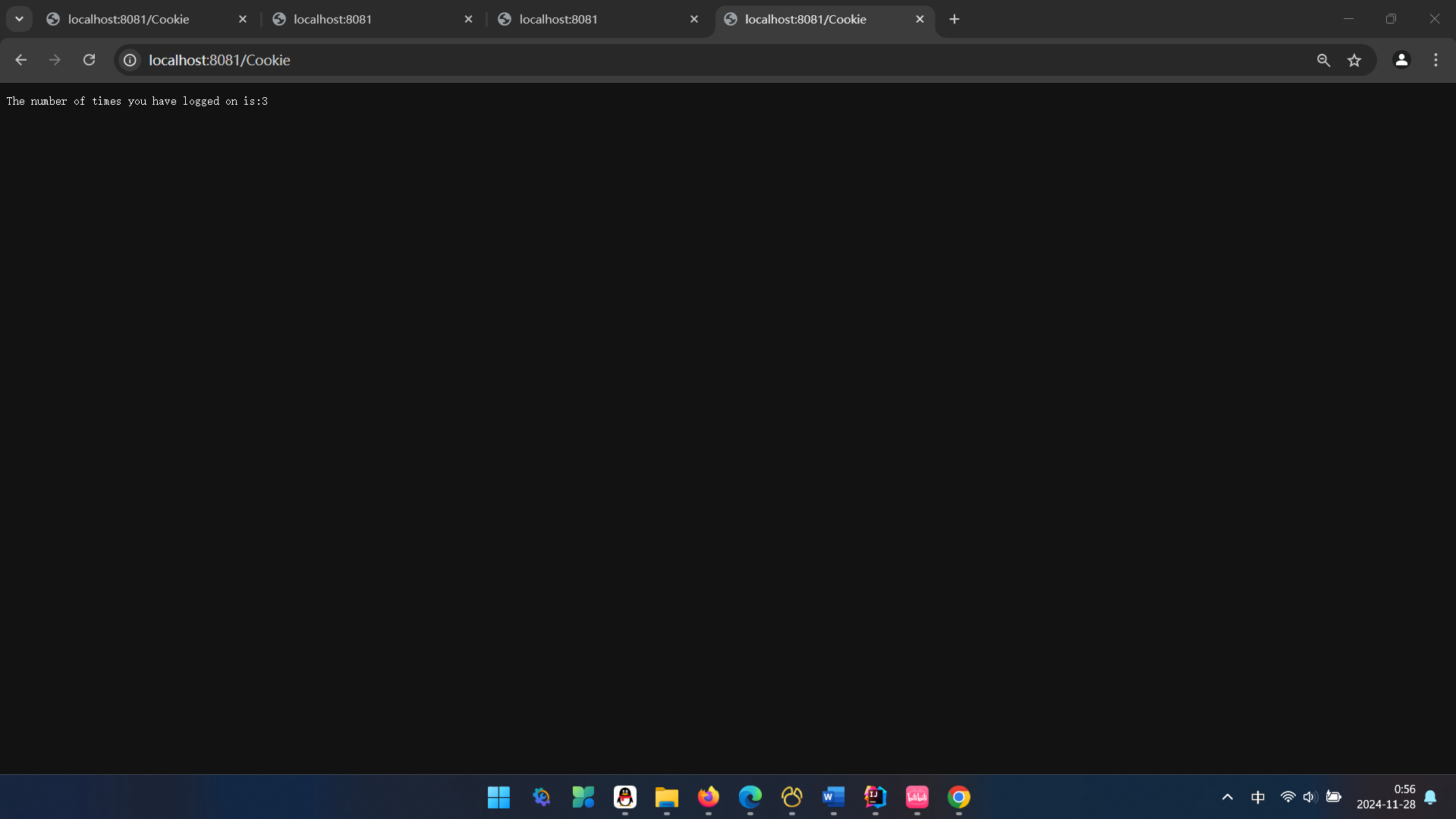
 

图3 运行结果





|  |
| --- |
| 此处插入CookieServlet.java文件代码  package Servlet; 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; import java.io.IOException;  @WebServlet("/Cookie") public class CookieServlet extends HttpServlet {   private static final long *serialVersionUID* = 1L;   @Override  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  int loginCount = getLoginCount(request);   if (loginCount == 0) {  response.getWriter().println("This is the first time you have logged on.");  } else {  response.getWriter().println("The number of times you have logged on is:" + loginCount);  }   setLoginCountCookie(response, ++loginCount);  }   private int getLoginCount(HttpServletRequest request) {  Cookie[] cookies = request.getCookies();  if (cookies != null) {  for (Cookie cookie : cookies) {  if ("loginCount".equals(cookie.getName())) {  return Integer.*parseInt*(cookie.getValue());  }  }  }  return 0; // First visit  }   private void setLoginCountCookie(HttpServletResponse response, int count) {  Cookie loginCountCookie = new Cookie("loginCount", String.*valueOf*(count));  loginCountCookie.setMaxAge(60 \* 60 \* 24); // One day in seconds  response.addCookie(loginCountCookie);  } } |
|  |

4. 应用Servlet读取Session数据。运行结果如图4所示。

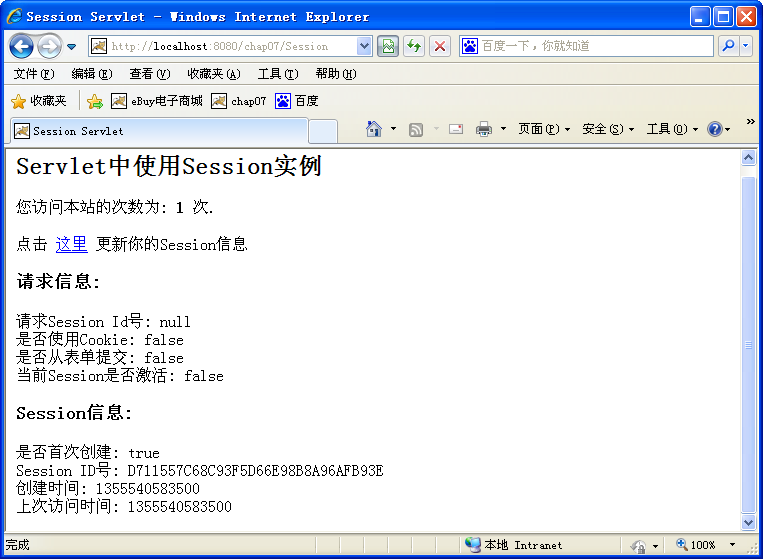
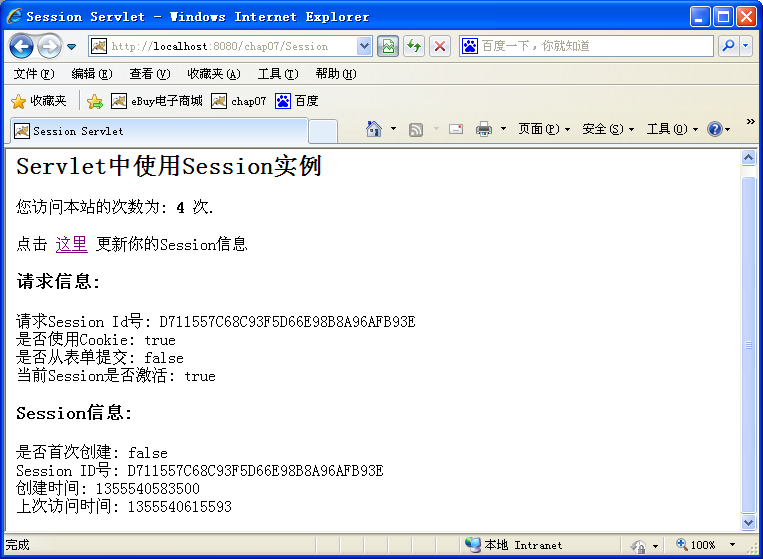
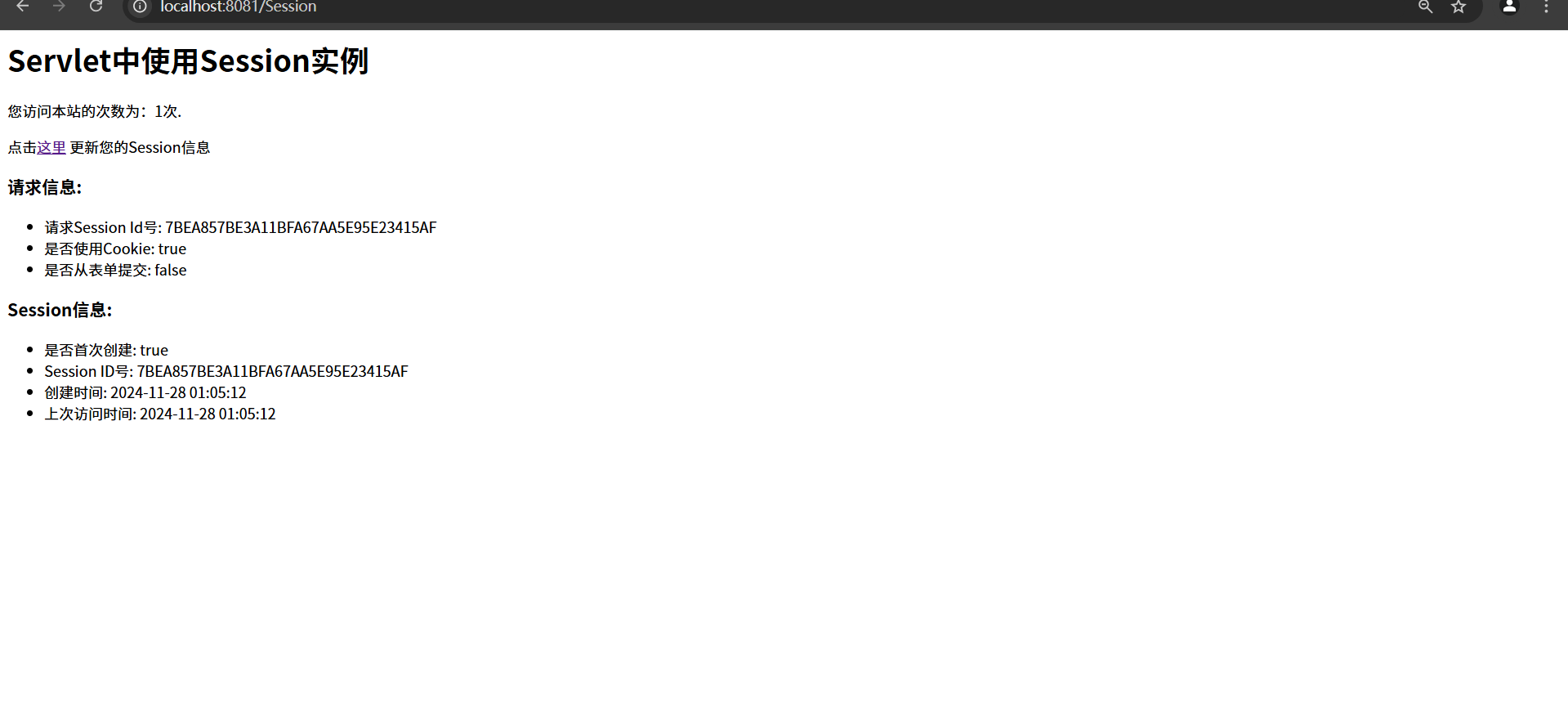
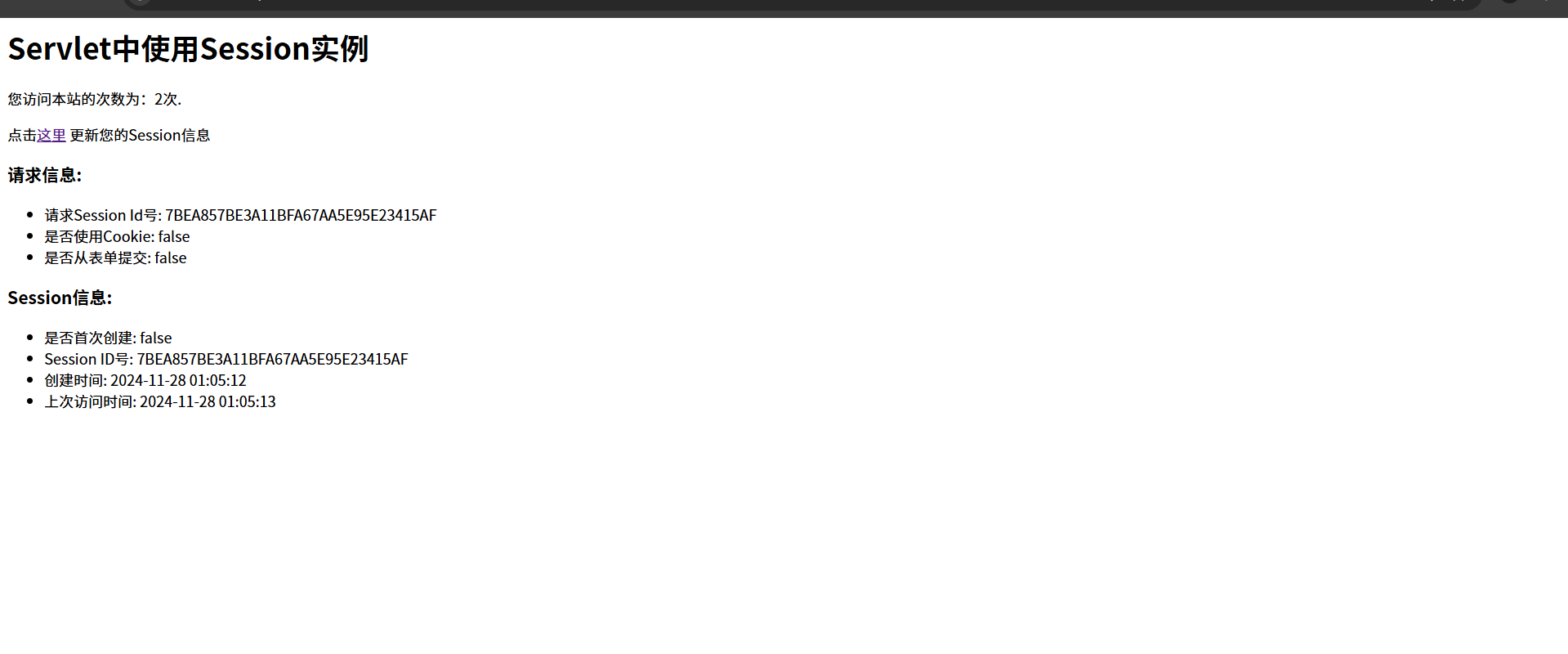
 

图4 运行结果





|  |
| --- |
| 此处插入SessionServlet.java文件代码  package Servlet;  import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import javax.servlet.http.HttpSession; import java.io.IOException; import java.text.SimpleDateFormat; import java.util.Date;  @WebServlet("/Session") public class SessionServlet extends HttpServlet {   private static final long *serialVersionUID* = 1L;   @Override  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  request.setCharacterEncoding("UTF-8");  response.setContentType("text/html;charset=UTF-8");  HttpSession session = request.getSession(true); // true to create if not exists   // Increment the visit counter  Integer visits = (Integer) session.getAttribute("visits");  if (visits == null) {  visits = 1;  } else {  visits++;  }  session.setAttribute("visits", visits);   // Get session information  boolean isNewlyCreated = session.isNew();  String sessionId = session.getId();  long creationTime = session.getCreationTime();  long lastAccessedTime = session.getLastAccessedTime();   SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");   // Display session information  response.setContentType("text/html");  response.getWriter().append("<h1>Servlet中使用Session实例</h1>")  .append("<p>您访问本站的次数为：" + visits + "次.</p>")  .append("<p>点击<a href='" + request.getContextPath() + "/Session'>这里</a> 更新您的Session信息</p>")  .append("<h3>请求信息:</h3>")  .append("<ul>")  .append("<li>请求Session Id号: ").append(sessionId).append("</li>")  .append("<li>是否使用Cookie: ").append(isNewlyCreated ? "true" : "false").append("</li>")  .append("<li>是否从表单提交: false</li>")  .append("</ul>")  .append("<h3>Session信息:</h3>")  .append("<ul>")  .append("<li>是否首次创建: ").append(isNewlyCreated ? "true" : "false").append("</li>")  .append("<li>Session ID号: ").append(sessionId).append("</li>")  .append("<li>创建时间: ").append(sdf.format(new Date(creationTime))).append("</li>")  .append("<li>上次访问时间: ").append(sdf.format(new Date(lastAccessedTime))).append("</li>")  .append("</ul>");  } } |