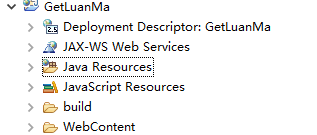
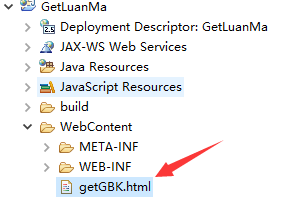
一.创建一个GBK编码的html文件, 发送get请求

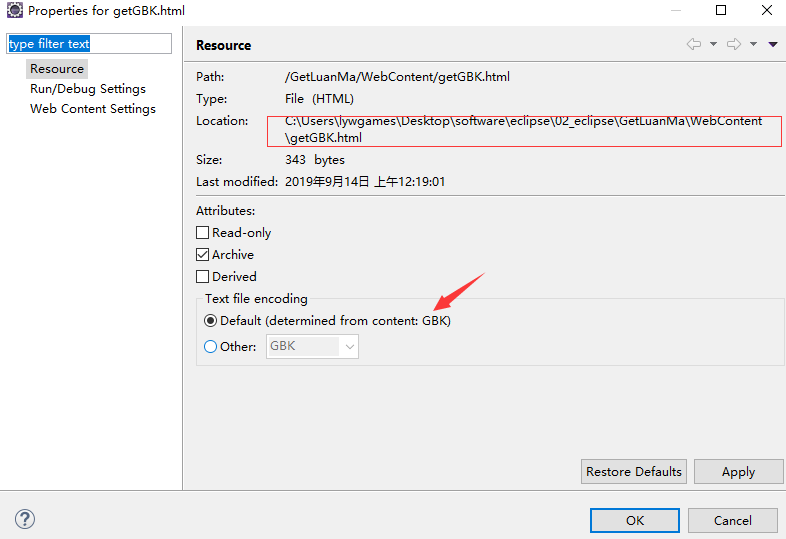
1.新建一个GetLuanMa的Web工程



2.新建一个getGBK.html文件



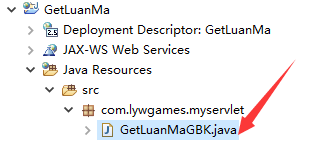
3.更改getGBK.html编码为GBK



4.编写getGBK.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="GBK">  <title>get请求中午乱码处理</title>  </head>  <body>  <form action="GLMGBK.action" method="get">  用户名:<input type="text" name="username" /><br />  密码:<input type="password" name="password" /><br />  <input type="submit" value="登录">  </form>  </body>  </html> |

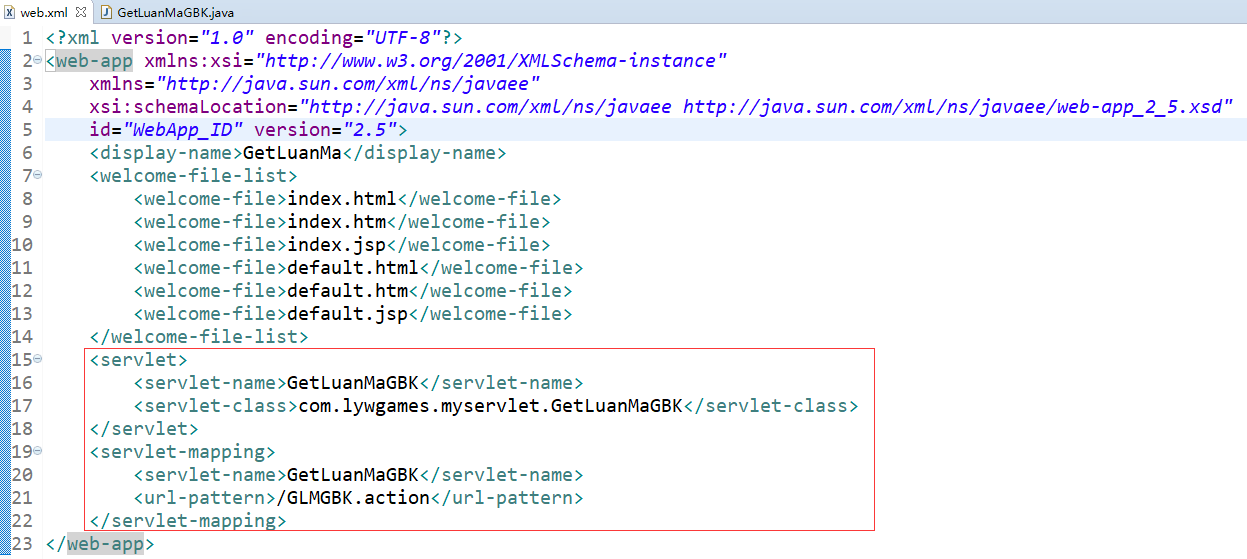
5.新建一个GetLuanMaGBK.java类



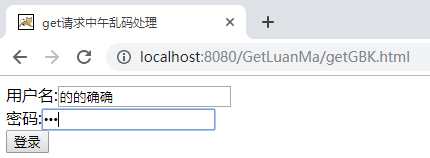
6.编写GetLuanMaGBK.java

|  |
| --- |
| package com.lywgames.myservlet;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class GetLuanMaGBK extends HttpServlet {  private static final long serialVersionUID = 1L;  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  String username = req.getParameter("username");  String password = req.getParameter("password");    System.out.println("username = " + username + " password = " + password);  }    @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  doGet(req, resp);  }  } |

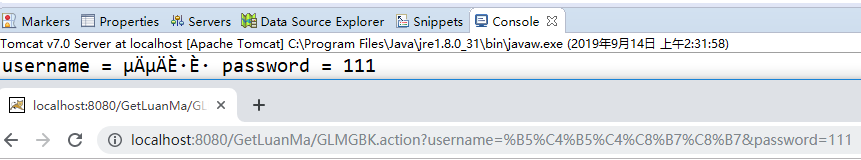
7.编写web.xml



8.部署运行, 访问getGBK.html

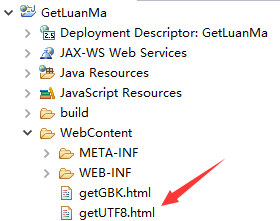


9.点击登录按钮, 控制台打印



二.创建一个UTF-8编码的html文件, 发送get请求

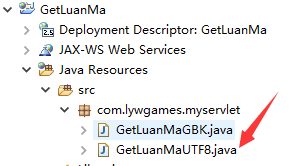
1.新建一个getUTF8.html



2.编辑getUTF8.html

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>get请求中午乱码处理</title>  </head>  <body>  <form action="GLMUTF8.action" method="get">  用户名:<input type="text" name="username" /><br />  密码:<input type="password" name="password" /><br />  <input type="submit" value="登录">  </form>  </body>  </html> |

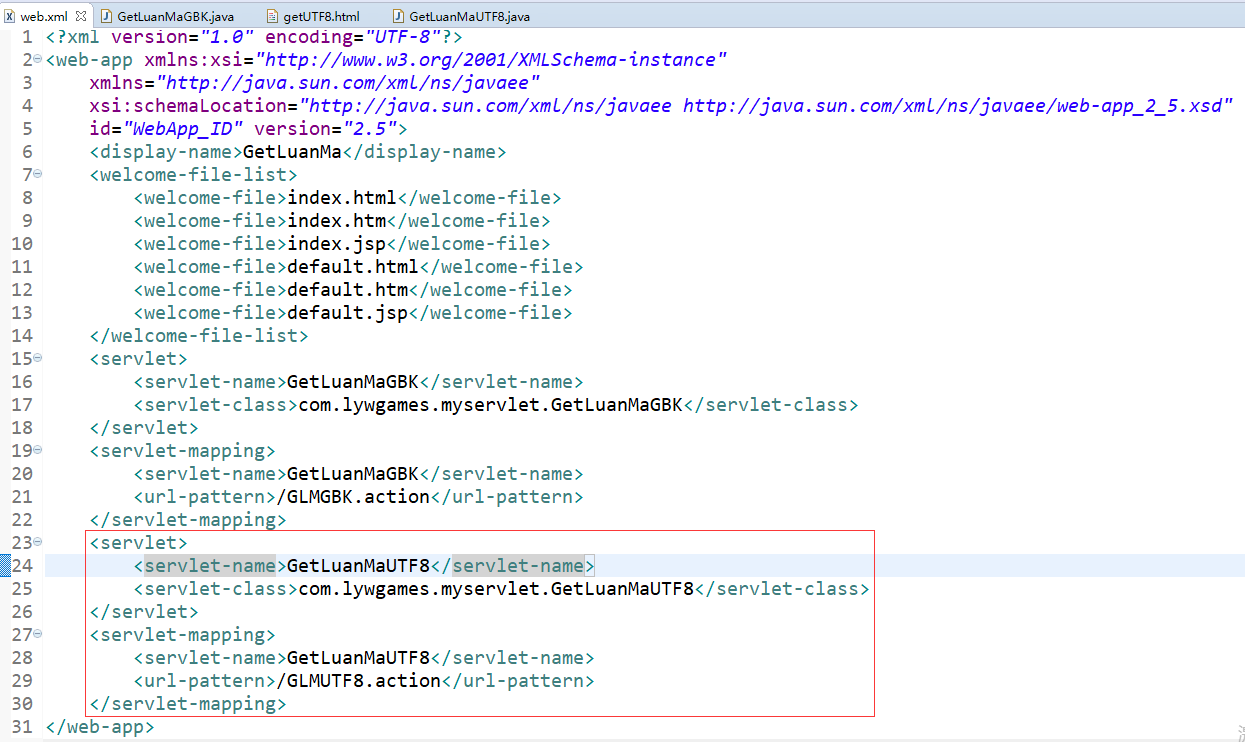
3.新建GetLuanMaUTF8.java类



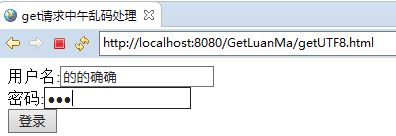
4.编辑GetLuanMaUTF8.java

|  |
| --- |
| package com.lywgames.myservlet;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  public class GetLuanMaUTF8 extends HttpServlet {  private static final long serialVersionUID = 1L;  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  String username = req.getParameter("username");  String password = req.getParameter("password");    System.out.println("username = " + username + " password = " + password);  }    @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  doGet(req, resp);  }  } |

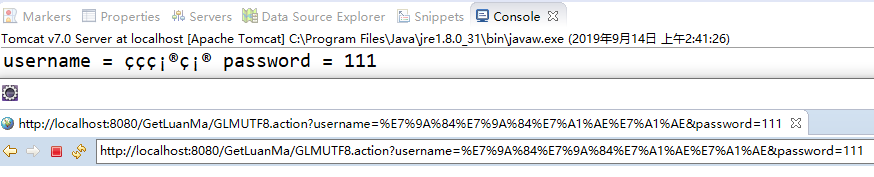
5.编辑web.xml



6.部署运行, 访问getUTF-8.html



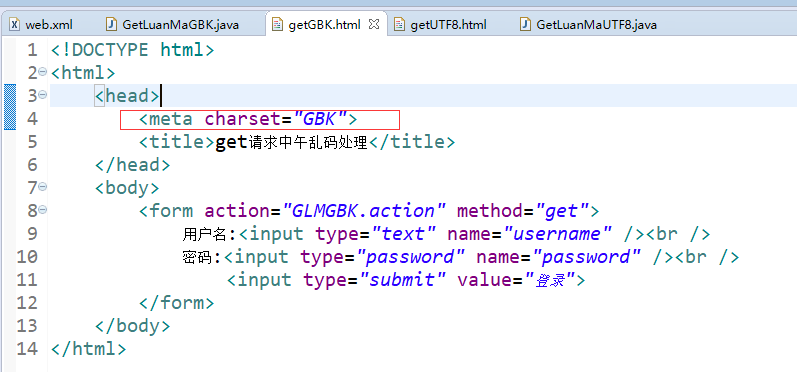
7.点击登录按钮, 控制台打印



三.分析以上编码为GBK和UTF-8的网页发送的请求

1.浏览器会对请求的URL中非ASCII码字符进行编码。

(1).编码为GBK的网页

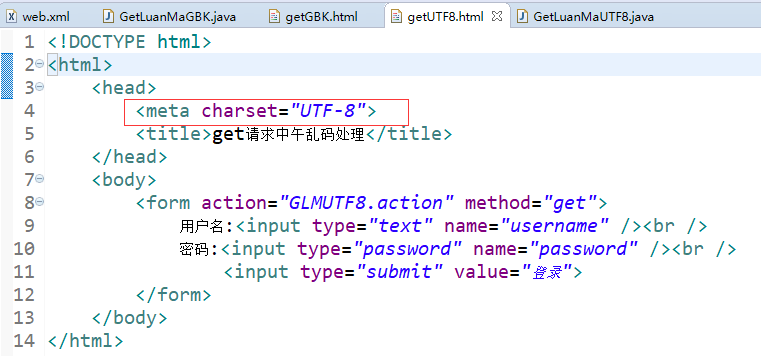


(2).编码为GBK的网页, 发送请求, 浏览器地址栏的信息如下:

http://localhost:8080/GetPostLuanMa/GLMGBK.action?username=%B5%C4%B5%C4%C8%B7%C8%B7&password=111

(3).在GBK的网页中, 用户名”的的确确”, 在经过GBK编码后变成了%B5%C4%B5%C4%C8%B7%C8%B7。

(4).编码为UTF-8的网页



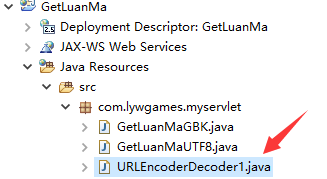
(5).编码为UTF-8的网页, 发送请求, 浏览器地址栏的信息如下:

http://localhost:8080/GetPostLuanMa/GLMUTF8.action?username=%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE&password=111

(6).在UTF-8的网页中, 用户名”的的确确”, 在经过UTF-8编码后变成了%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE。

2.新建一个URLEncoderDecoder1.java类, 对”的的确确”进行GBK和UTF-8的URLEncoder.encode和URLDecoder.decode, 观察是不是浏览器对”的的确确”进行GBK和UTF-8编码。

(1).新建一个URLEncoderDecoder1.java类



(2).编写URLEncoderDecoder1.java类

|  |
| --- |
| package com.lywgames.myservlet;  import java.io.UnsupportedEncodingException;  import java.net.URLDecoder;  import java.net.URLEncoder;  public class URLEncoderDecoder1 {  public static void main(String[] args) {  try {  // URLEncoder.encode("的的确确", "utf-8") ==> %E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE  System.out.println(URLEncoder.encode("的的确确", "utf-8"));  System.out.println(URLDecoder.decode("%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE", "utf-8"));    // URLEncoder.encode("的的确确", "gbk") ==> %B5%C4%B5%C4%C8%B7%C8%B7  System.out.println(URLEncoder.encode("的的确确", "gbk"));  System.out.println(URLDecoder.decode("%B5%C4%B5%C4%C8%B7%C8%B7", "gbk"));  } catch (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  } |

(3).运行程序



四.解决get请求的中午乱码

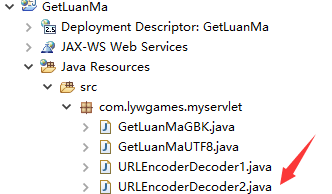
1.tomcat收到客户端的数据后, 默认使用ISO-8859-1去解码。

2.服务器在接收到浏览器经过UTF-8编码后的数据, 进行一次ISO-8859-1解码, 打印出来的数据为: ççç¡®ç¡®。

3.服务器在接收到浏览器经过UTF-8编码后的数据, 进行一次ISO-8859-1解码, 打印出来的数据为: µÄµÄÈ·È·。

4.编写URLEncoderDecoder2.java类, 对%B5%C4%B5%C4%C8%B7%C8%B7和%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE, 进行ISO-8859-1的URLDecoder.decode解码。查看控制台的输出信息和Servlet请求时的信息是否一致。

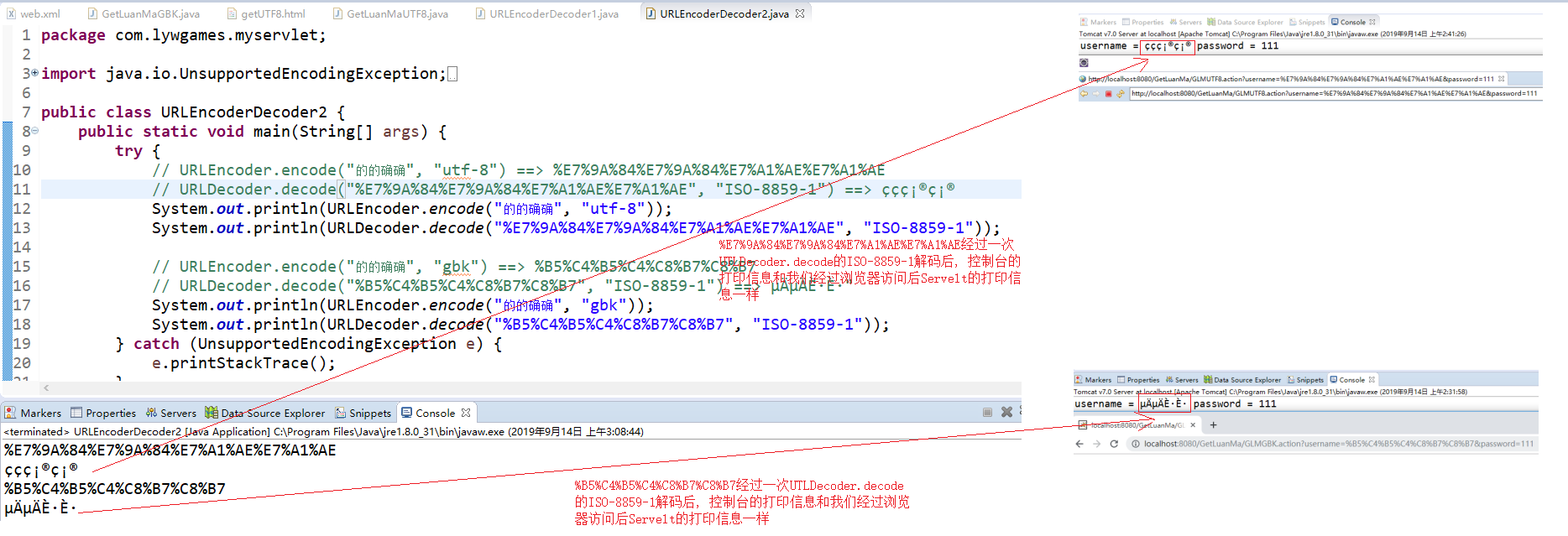
(1).新建URLEncoderDecoder2.java类



(2).编写URLEncoderDecoder2.java类

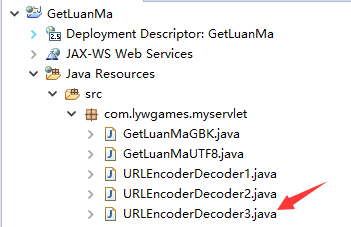
|  |
| --- |
| package com.lywgames.myservlet;  import java.io.UnsupportedEncodingException;  import java.net.URLDecoder;  import java.net.URLEncoder;  public class URLEncoderDecoder2 {  public static void main(String[] args) {  try {  // URLEncoder.encode("的的确确", "utf-8") ==> %E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE  // URLDecoder.decode("%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE", "ISO-8859-1") ==> ççç¡®ç¡®  System.out.println(URLEncoder.encode("的的确确", "utf-8"));  System.out.println(URLDecoder.decode("%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE", "ISO-8859-1"));    // URLEncoder.encode("的的确确", "gbk") ==> %B5%C4%B5%C4%C8%B7%C8%B7  // URLDecoder.decode("%B5%C4%B5%C4%C8%B7%C8%B7", "ISO-8859-1") ==> µÄµÄÈ·È·"  System.out.println(URLEncoder.encode("的的确确", "gbk"));  System.out.println(URLDecoder.decode("%B5%C4%B5%C4%C8%B7%C8%B7", "ISO-8859-1"));  } catch (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  } |

(3).运行程序



5.使用new String("数据信息".getBytes("ISO-8859-1"), "utf-8"), 把"数据信息"打散到ISO-8859-1对应的字节数组 , 然后再按utf-8组拼字符串, 解决乱码问题。

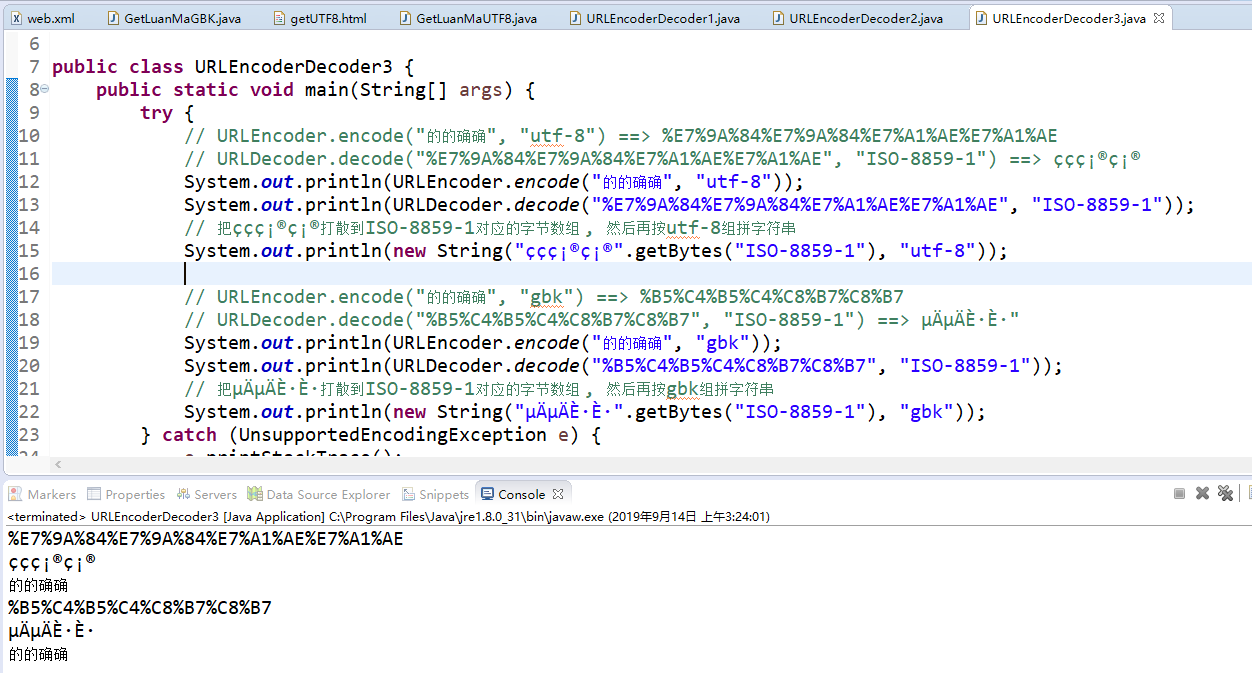
(1).新建一个URLEncoderDecoder3.java类



1. .编写URLEncoderDecoder3.java

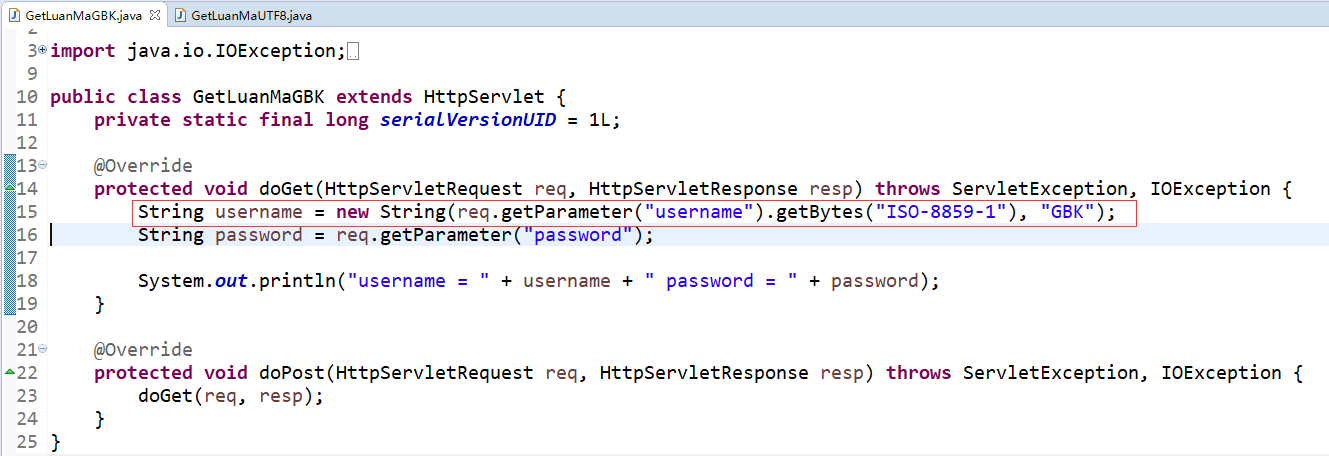
|  |
| --- |
| package com.lywgames.myservlet;  import java.io.UnsupportedEncodingException;  import java.net.URLDecoder;  import java.net.URLEncoder;  public class URLEncoderDecoder3 {  public static void main(String[] args) {  try {  // URLEncoder.encode("的的确确", "utf-8") ==> %E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE  // URLDecoder.decode("%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE", "ISO-8859-1") ==> ççç¡®ç¡®  System.out.println(URLEncoder.encode("的的确确", "utf-8"));  System.out.println(URLDecoder.decode("%E7%9A%84%E7%9A%84%E7%A1%AE%E7%A1%AE", "ISO-8859-1"));  // 把ççç¡®ç¡®打散到ISO-8859-1对应的字节数组 , 然后再按utf-8组拼字符串  System.out.println(new String("ççç¡®ç¡®".getBytes("ISO-8859-1"), "utf-8"));    // URLEncoder.encode("的的确确", "gbk") ==> %B5%C4%B5%C4%C8%B7%C8%B7  // URLDecoder.decode("%B5%C4%B5%C4%C8%B7%C8%B7", "ISO-8859-1") ==> µÄµÄÈ·È·"  System.out.println(URLEncoder.encode("的的确确", "gbk"));  System.out.println(URLDecoder.decode("%B5%C4%B5%C4%C8%B7%C8%B7", "ISO-8859-1"));  // 把µÄµÄÈ·È·打散到ISO-8859-1对应的字节数组 , 然后再按gbk组拼字符串  System.out.println(new String("µÄµÄÈ·È·".getBytes("ISO-8859-1"), "gbk"));  } catch (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  } |

(3). 运行程序



6.修改Servlet解决乱码问题

(1).修改GetLuanMaGBK.java



(2).修改GetLuanMaUTF8.java

