

```

package com.sxt.server;

import java.io.IOException;
import java.io.InputStream;
import java.io.UnsupportedEncodingException;
import java.net.Socket;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * 封装请求协议:封装请求参数
 * @author 江
 *
 */

public class Request02 {
    //协议信息
    private String requestInfo;
    //请求方式
    private String method;
    //请求url
    private String url;
    //请求参数
    private String queryStr="";
    //存储参数
    private Map<String,List<String>> parameterMap;
    //换行
    private final String CRLF="\r\n";

    public Request02(InputStream is) {
        parameterMap=new HashMap<>();
        byte[] datas=new byte[1024*1024];
        int len;
    }

```

```

try {
    len=is.read(datas);
    this.requestInfo=new String(datas,0,len);
    System.err.println(requestInfo);
} catch (IOException e) {
    e.printStackTrace();
    return ;
}

//分解字符串
parseRequestInfo();
}

```

//分解字符串

```

private void parseRequestInfo() {
    System.err.println("-----分解-----");
    System.err.println("--1. 获取请求方式:开头到第一个/--");
}

```

```

this.method=this.requestInfo.substring(0, this.requestInfo.indexOf("/")).toLowerCase();
this.method=this.method.trim();
System.err.println("--2. 获取请求url:第一个/到HTTP/--");
System.err.println("--可能包含请求参数?前面的为url--");
//1). 获取/的位置
int idx1=this.requestInfo.indexOf("/") +1;
//2). 获取HTTP的位置
int endIdx=this.requestInfo.indexOf("HTTP/");
//3). 分割字符串
this.url=this.requestInfo.substring(idx1, endIdx);
System.err.println(this.url);
//4). 获取?的位置
int queryIdx=this.url.indexOf("?");
if(queryIdx>=0) { //表示存在请求参数
    String[] urlArray=url.split("\\?"); //转义
    this.url=urlArray[0];
    queryStr=urlArray[1].trim();
}

System.err.println(this.url);

```

```

        System.err.println("—3. 获取请求参数:如果Get已经获取, 如果是post可能在请求
中-----");
        if(method.equals("post")) {
            String
qStr=this.requestInfo.substring(this.requestInfo.lastIndexOf(CRLF)).trim();
            System.err.println(qStr+"-->");
            if(queryStr==null) {
                queryStr=qStr;
            }else {
                queryStr+="&"+qStr;
            }
        }
        System.err.println(method+"-->"+url+"-->"+queryStr);

        //转成Map    fav=1&fav=2&uname=shsxt&age=18&others=
        convertMap();

    }

    //处理请求参数为Map
    private void convertMap() {
        //1. 分割字符串  &
        String[] keyValues=this.queryStr.split("&");
        for(String queryStr:keyValues) {
            //2. 再次分割字符串 =
            String[] kv=queryStr.split("=");
            kv=Arrays.copyOf(kv, 2);    //为空时可以进行填充
            //获取key和value
            String key=kv[0];
            String value=kv[1]==null?null:decode(kv[1], "utf-8");
            //存储到Map中
            if(!parameterMap.containsKey(key)) { //第一次
                parameterMap.put(key, new ArrayList<String>());
            }
            parameterMap.get(key).add(value);
        }
    }
}

```

```

/**
 * 处理中文
 * @return
 */
private String decode(String value,String enc) {
    try {
        return java.net.URLDecoder.decode(value, enc);
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
        return null;
    }
}

```

```

/**
 * 通过name获取对应的多个值
 * @param key
 * @return
 */
public String[] getParameterValues(String key) {
    List<String> values=this.parameterMap.get(key);
    if(null==values||values.size()<1) {
        return null;
    }
    return values.toArray(new String[0]);
}

```

```

/**
 * 通过name获取对应的一个值
 * @param key
 * @return
 */
public String getParameter(String key) {
    String[] values=getParameterValues(key);
    return values==null?null:values[0];
}

```

```
public String getMethod() {  
    return method;  
}
```

```
public String getUrl() {  
    return url;  
}
```

```
public String getQueryStr() {  
    return queryStr;  
}
```

```
public Request02(Socket client) throws IOException {  
    this(client.getInputStream());  
}  
}
```