Servlet增强功能

# 文件上传

## 客户端编程

|  |
| --- |
| <form action=*"singleUpload"* enctype=*"multipart/form-data"*  method=*"post"*>  Author: <input type=*"text"* name=*"author"*/><br/>  Select file to upload <input type=*"file"* name=*"filename"*/><br/>  <input type=*"submit"* value=*"Upload"*/>  </form> |

## 服务器端编程

|  |
| --- |
| #HttpServletRequest接口  Part getPart(Strng name); //根据input 中的name值获取part  Collection<Part> getPart(); //获取所有的Part  #Part接口方法  getName()：表单中name值  getContentType()：查看Part内容类型  getHeader(String headerName)： |
| @WebServlet(urlPatterns = { "/singleUpload" })  @MultipartConfig  **public** **class** SingleUploadServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 8593038L;  **private** String getFilename(Part part) {  String contentDispositionHeader =  part.getHeader("content-disposition");  String[] elements = contentDispositionHeader.split(";");  **for** (String element : elements) {  **if** (element.trim().startsWith("filename")) {  **return** element.substring(element.indexOf('=') + 1)  .trim().replace("\"", "");  }  }  **return** **null**;  }  **public** **void** doPost(HttpServletRequest request,  HttpServletResponse response) **throws** ServletException,  IOException {    // save uploaded file to WEB-INF  Part part = request.getPart("filename");  String fileName = getFilename(part);  **if** (fileName != **null** && !fileName.isEmpty()) {  part.write(getServletContext().getRealPath(  "/WEB-INF") + "/" + fileName);  }    // write to browser  response.setContentType("text/html");  PrintWriter writer = response.getWriter();  writer.print("<br/>Uploaded file name: " + fileName);  writer.print("<br/>Size: " + part.getSize());    String author = request.getParameter("author");  writer.print("<br/>Author: " + author);  }  } |

# 异步请求处理

任务执行时间过长，可以使用异步方式。（**不让长任务一直占用线程**）

|  |
| --- |
| @WebServlet(name = "AsyncDispatchServlet",  urlPatterns = { "/asyncDispatch" },  asyncSupported = **true**)  doGet or doPost方法中主要内容  final AsyncContext asyncContext = request.startAsync();  asyncContext.setTimeout(60000);  asyncContext.start(new Runnable() {  @Override  public void run() {  //long run task  asyncContext.complete() or asyncContext.dispatch();  }  }); |
| @WebServlet(name = "AsyncListenerServlet",  urlPatterns = { "/asyncListener" },  asyncSupported = **true**)  **public** **class** AsyncListenerServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 62738L;  @Override  **public** **void** doGet(**final** HttpServletRequest request,  HttpServletResponse response)  **throws** ServletException, IOException {  **final** AsyncContext asyncContext = request.startAsync();  asyncContext.setTimeout(5000);    asyncContext.addListener(**new** MyAsyncListener());  asyncContext.start(**new** Runnable() {  @Override  **public** **void** run() {  **try** {  Thread.*sleep*(3000);  } **catch** (InterruptedException e) {  }  String greeting = "hi from listener";  System.***out***.println("wait....");  request.setAttribute("greeting", greeting);  asyncContext.dispatch("/test.jsp");  }  });  }  } |
| **public** **class** MyAsyncListener **implements** AsyncListener {  @Override  **public** **void** onComplete(AsyncEvent asyncEvent)  **throws** IOException {  System.***out***.println("onComplete");  }  @Override  **public** **void** onError(AsyncEvent asyncEvent)  **throws** IOException {  System.***out***.println("onError");  }  @Override  **public** **void** onStartAsync(AsyncEvent asyncEvent)  **throws** IOException {  System.***out***.println("onStartAsync");  }  @Override  **public** **void** onTimeout(AsyncEvent asyncEvent)  **throws** IOException {  System.***out***.println("onTimeout");  }  } |

# webSocket

双向通讯