# Java web http server 详细设计说明书

目录

**1引言2**

1.1编写目的2

1.2背景2

1.3目标读者2

**2系统总体设计3**

2.1启动器设计3

2.2服务器设计4

2.3系统 业务流程5

**3系统数据结构及接口5**

启动器类5

Request解析类5

Respone响应6

ConfigBean配置类7

httpSever类7

Handle业务逻辑类8

**4测试计划8**

**5尚未解决的问题8**

## 1引言

#### 1.1编写目的

在前一阶段概要设计中已经解决了整个实现该系统需要所需要的模块设计、系统流程问题，在一下的详细设计报告中将对在本阶段中对系统所做的所有详细设计进行说明，确定系统该如何具体实现所需要的系统，从而可以在编码阶段把这个描述直接翻译成用成具体语言书写的程序，主要包括对业务处理流程，总体结构模块设计，系统数据接口与接口设计及各程序块的具体描述，像类的功能，性能，方法的输入输出，程序的算法，程序的逻辑，相互之间的接口等等。。。

#### 1.2背景

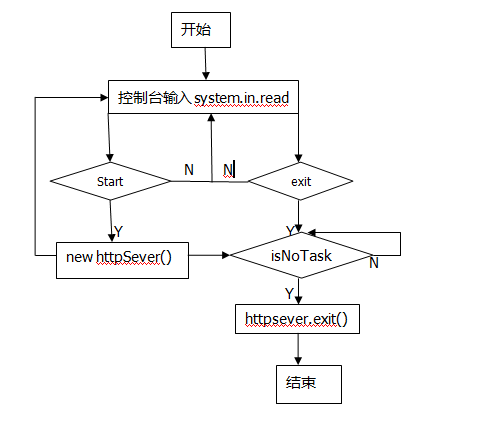
Java web http server设计很多方法的必备基础知识，包括像http协议，java多线程，网络套接字Socket，接口设计等等，利用这些只是，完成一个web服务器，处理 用户的请求，同时按照软件开发各个阶段，掌握软件开发的流程。

#### 1.3目标读者

项目开发人员，测试人员等

## 2系统总体设计

#### 2.1启停器设计

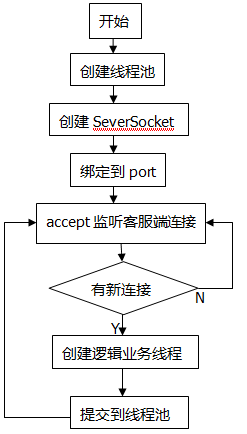


20160315修改方案：将new Thread（httpSever）.start()放在httpServer类的构造函数中，它是单例模式，在这里也可以通过这个类的静态工厂方法获得该对象，如果存在，则不予重新创建,exit命令来时先检查httpSever对象是否还有任务正在执行，否则直到等待任务执行完毕，关闭资源，退出程序

#### 2.2服务器程序

通过端口监听http请求，当有新的请求时将任务线程提交给线程池处理

该模块通过实例httpSever类完成，该类实现runable接口，读取ConfigBean进行初始化



#### 2.3系统业务流程设计

业务流程处理模块主要由Handle.handleRequest()方法完成，需要使用的类有Resquest解析类，Respones响应类

根据http请求，通过Reques解析类的解析，并对解析结果经行逻辑处理，最后做出响应

（Accept进来的新request）

getMethod()

getConnection()

其他

getFile()

非GET方法 文件不存 文件存在 true

目录

getFileType()

outNotExist()

outNotGet()

Keep-alive处理

getDisk()

静态文件 非静态文件

outPathList()

outIllegalType()

非D盘

getParameter()

outNoPower()

无 预览参数true

outFile()

outFilePreview()

## 3系统数据结构及接口设计

#### 启动器类

方法：唯一方法，程序入口函数static void main()

作用：是监控控制台的输入，控制httpSever的开始与结束；

#### Request解析类

通过构造函数注入Socket的输入流对象

成员变量：

都是私有变量，请求方法method，请求的资源file，请求的文件类型fileType，请求的资源所在磁盘分区disk，请求的附加预览参数preview，请求的keep-alive设置情况

keepAliveTimeOut过期时长，keepAliveMax最大连接次数，还有主机，协议版本等，和请求头的一些相关信息 /\*\*其中请求头的RANGE指定请求的资源开始下载的位置，如：bytes—2000 -，配合响应头RANGE:bytes—2000 – 4000，完成断点续传？\*\*/

getter方法：

获得请求的资源getFile();

获得文件的类型getFileType()；

获得请求的磁盘分区getDisk()；

获得请求参数getParamers()；

获得请求的连接方式keep-alive设置getConnection()；

获得请求的主机getHost等等。

作用：对客服端发来的http请求进行解析，打包封装，以便得到详细的信息

#### Respones响应类

通过构造方法注入Socket输出流对象

方法：

包含很多输出方法，其中它们的参数都需要客服端Socket；请求是非get输出用outNotGet()；请求不是D盘的资源时输出用outNoPower()；请求的只是一个路径时输出用outPathList()，参数为路径path;请求的是正确的路劲下的文件时，如果文件不存在输出用outNotExist(),如果存在输出使用下载输出方法outFile()参数为文件名file；如果请求中包含预览参数，则用输出用outFilePreview()等等

方法介绍：

outNotGet()｛响应体：<html><body>请求方法不对，请检查您的提交方式！<body></html>｝

outIllegalType(){响应体：html><body>不支持该文件类型！ <body></html>}

outNoPower(){ 响应体：<html><body>您没有权限访问D盘以外的资源！ <body></html> }

outNotExist(){响应状态行：状态码404，响应体：<html><body>页面提示“文件找不到！”</body></html>}

outPathFile(){响应体：<html><body>/\*\*获得该目录下的文件列表，并以超连接的方式反映给用户\*\*/<body></html>}

outFile(){响应头：指定文件的格式，响应体：读取文件的类容，写入socket的输出流}

通过指定文件的类型，然后将文件类型写入Socket输出流，浏览器就可以下载了

outPreviewFile(){响应头：指定文件的格式为text/html,响应体：<html><body>/\*\*文件的内容\*\*/</body><html>},通过指定文件类型给html，将内容下载html页面当中，在浏览器中就可以预览了

#### ConfigBean配置文件类

配置服务端口号port=80，与服务目录D盘

#### httpServer类

接口：

其实现runable接口，可由启动器主线程创建或者销毁，

成员变量：

由Executors的工厂方法 所创建的一个合适大小的线程池对象pool；和一个全局变量serverSocket

方法：

构造方法 httpServer()，将serverSocket绑定到PORT；

实现的run()方法，监控客服端的连接，一旦有新的连接，则创建执行handle任务的线程，并将其提交给线程池pool

#### Handle核心逻辑类

/\*\*

通过构造函数注入 private SelectionKey key;

        public Handle(SelectionKey key){

            this.key = key;

        }

\*\*/

接口：实现Ruanable接口，完成业务逻辑的处理的任务的线程，由httpSever启动

方法：实现的run()方法，和一个私有handleRequest()，在run()中调用

作用：业务逻辑处理，调用不同的方法完成用户的功能

例如： if ( !getMethod ) throw Expection outNotGet()

If ( !D盘/!服务目录 ) outNoPower()

If ( file not exist ) outNotExist()

If ( file is direct) outPathList()

If( file != null and !preview) outFile()

If( file != null and preview ) outFilePreview()

## 4调试、测试计划

代码开发中使用日志记录

使用juint4测试，代码编写中就完成有关方法的测试，当各个功能，类完成之后再组合测试，完成整体测试，最后通过浏览器访问相关资源，完成用户测试

## 5尚未解决的问题

Keep-alive处理，

断点续传功能，

exit命令来时的等待已有任务的结束然后退出的具体实现

NIO使用的体现