开发者指南

[1. 搭建环境](#_Toc370997525)

[2. 几个重要类](#_Toc370997526)

[3. 开发工具](#_Toc370997527)

[4. 开发示例](#_Toc370997528)

1. 搭建环境

工具云是基于Apache License2.0的开源Java项目，代码位于GitHub（<https://github.com/siddur/toolcloud.git>）上。 您可以通过Git客户端（Eclipse自带）下载到本地。导入成Eclipse项目。启动siddur.tool.jetty.JettyStart的main方法就能运行。

1. 几个重要类
   1. ITool 是一个接口，所有工具的开发都要继承该接口。换句话说，所有继承该接口的工具都能被平台识别。
   2. ConsoleTool 是一个能调用控制台的ITool的实现。如果您希望在工具运行时将日志实时打印到浏览器的模拟控制台，您可以直接继承此类。否则，不要继承此类，而是直接实现ITool接口。
   3. ScriptTool是一个包装脚本（Python、Perl等）的类，它继承ConsoleTool。因此能调用控制台打印实时信息。所有脚本工具注册后都会自动由ScriptTool管理调用。此类无须开发者调用。
2. 开发工具

开发工具最重要的是要考虑输入和输出。浏览器将用户输入的数据进行验证，然后发送给平台，平台根据工具的配置信息，解析、验证数据，然后传给代表工具的类或脚本。代表工具的类或脚本运行后将产生的结果平台，平台再一次根据配置信息解析、验证后将输入发送给浏览器。

* 1. 开发脚本

选择平台支持的脚本语言及语言版本，如Python27。脚本文件只能是单个文件，并且脚本的输入输出参数的设计需要与注册工具时填写的输入输出信息一致。

由于脚本的实时信息能够打印在浏览器端的控制台，为了达到这一点，需要注意的是，标准输出需要随时flush，不能让它们缓存。

* 1. 开发Jar

继承ConsoleTool或实现ITool，编写excute方法的代码。因为Java代码是在安全沙箱里运行，它并不能随意读写磁盘文件，平台提供了一个TempFileUtil类用来处理文件的读写。

1. 开发示例
   1. HelloWorld

实现ITool的execute方法。execute的参数String[] inputs是从用户界面收集来的输入参数，具体的个数由发布该工具时定义。此方法返回的也是一个数组，作为结果打印在页面，以什么形式（html代码、文件连接）同样也是有发布该工具时定义的。

**package** siddur.tool.sample;

**import** siddur.tool.core.ITool;

**public** **class** HelloWorld **implements** ITool{

@Override

**public** String[] execute(String[] inputs, IToolWrapper toolWrapper, Map<String, Object> context) {

String[] output = **new** String[2];

output[0] = "Hello World";

output[1] = inputs[0];

**return** output;

}

@Override

**public** **void** init() {

}

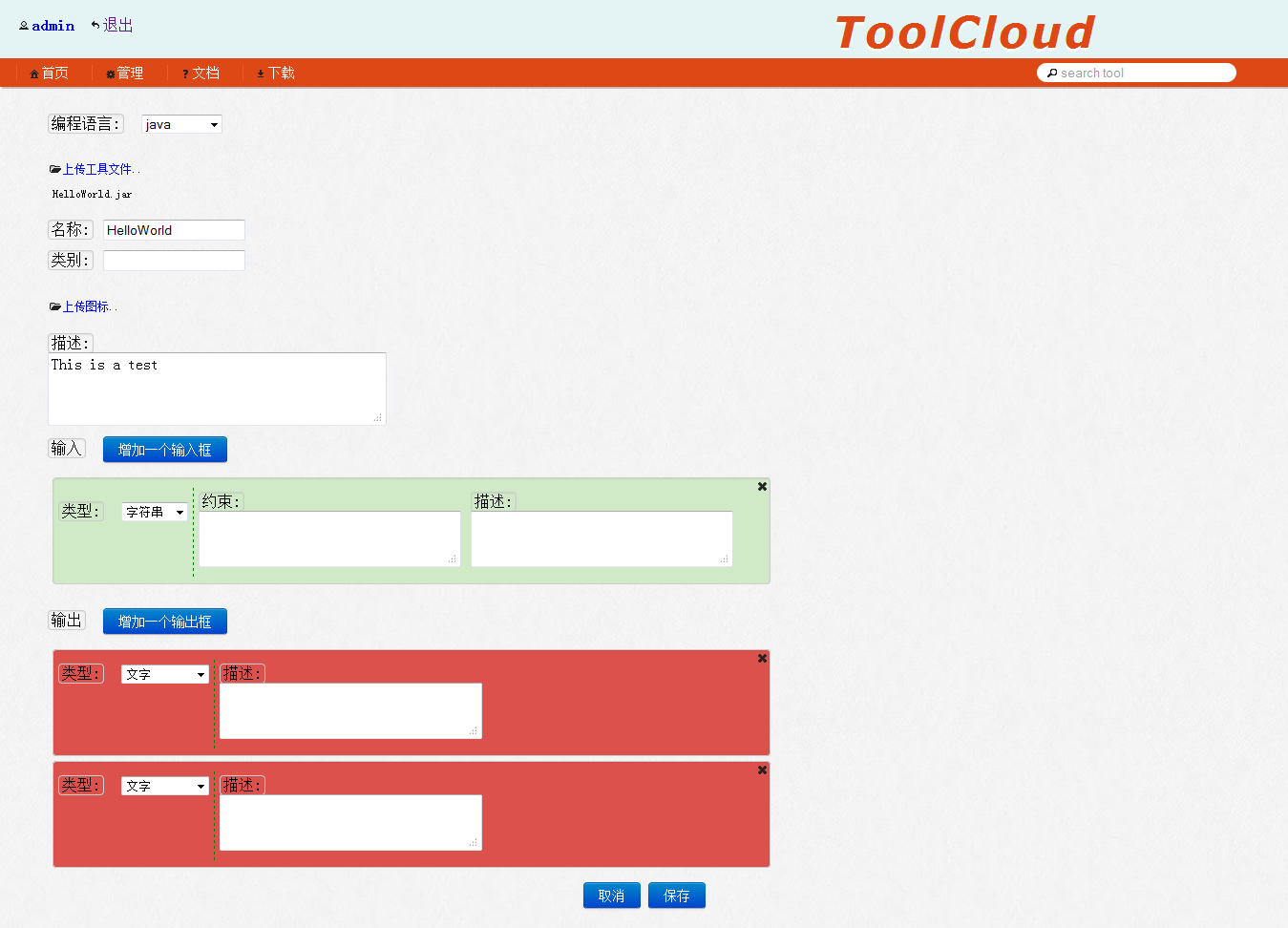
@Override

**public** **void** destroy() {

}

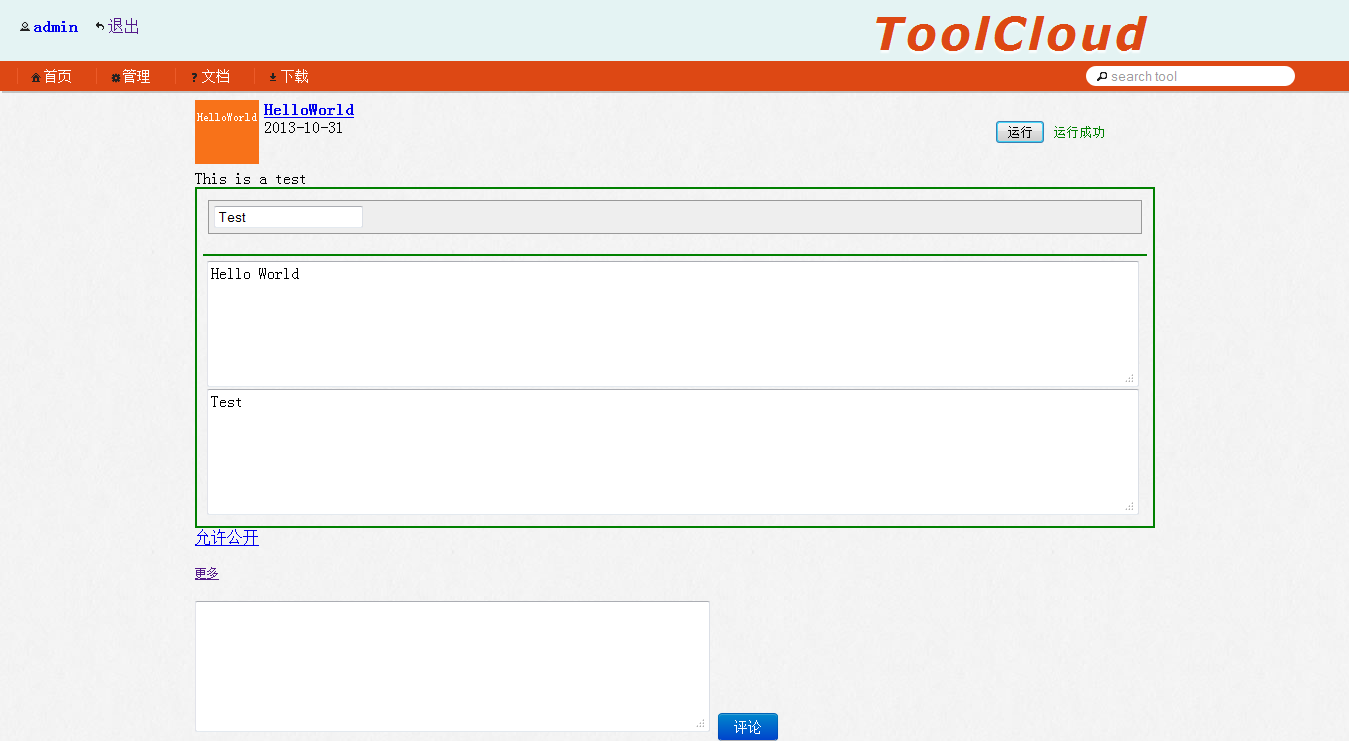
}

我们将Helloworld.class打成Jar包。进入发布页面，按图所示输入相关信息：



需要注意的是此工具有一个字符串输入和两个文字输出。

在运行界面输入一个字符串，运行查看结果：



* 1. 文件输入输出

首先继承ConsoleTool类

**class** FileReadAndWrite **extends** ConsoleTool

然后实现execute方法

**public** String[] execute(String[] inputs, IToolWrapper toolWrapper, Map<String, Object> context) **throws** Exception {

log("开始---->");

//通过TempFileUtil获取File对象

File dir = TempFileUtil.*findFile*(inputs[0]);

File file = TempFileUtil.*findFile*(inputs[1]);

String s = inputs[2];

//通过TempFileUtil产生临时文件和文件夹

File newDir = TempFileUtil.*createEmptyDir*();

File newFile = TempFileUtil.*createEmptyFile*("txt");

BufferedWriter bw = **new** BufferedWriter(**new** OutputStreamWriter(**new** FileOutputStream(newFile)));

bw.write(s);

**if**(dir.isDirectory()){

**for**(File f : dir.listFiles()){

readAndWrite(bw, f);

}

}

**if**(file.isFile()){

readAndWrite(bw, file);

}

bw.flush();

bw.close();

String[] results = **new** String[3];

//通过TempFileUtil将File对象转换成String对象返回给系统

results[0] = TempFileUtil.*file2String*(newFile);

FileUtils.*copyFileToDirectory*(newFile, newDir);

results[1] = TempFileUtil.*file2String*(newDir);

results[2] = s;

//通过log方法打印到页面模拟控制台

log(results[0]);

log(results[1]);

log("----->结束");

**return** results;

}

**private** **void** readAndWrite(BufferedWriter bw, File f)**throws** Exception{

BufferedReader r = **new** BufferedReader(**new** InputStreamReader(**new** FileInputStream(f)));

String line = **null**;

log("读取文件" + f.getCanonicalPath());

**while**((line = r.readLine()) != **null**){

//print to console

log(line);

//print to a file

bw.append(line);

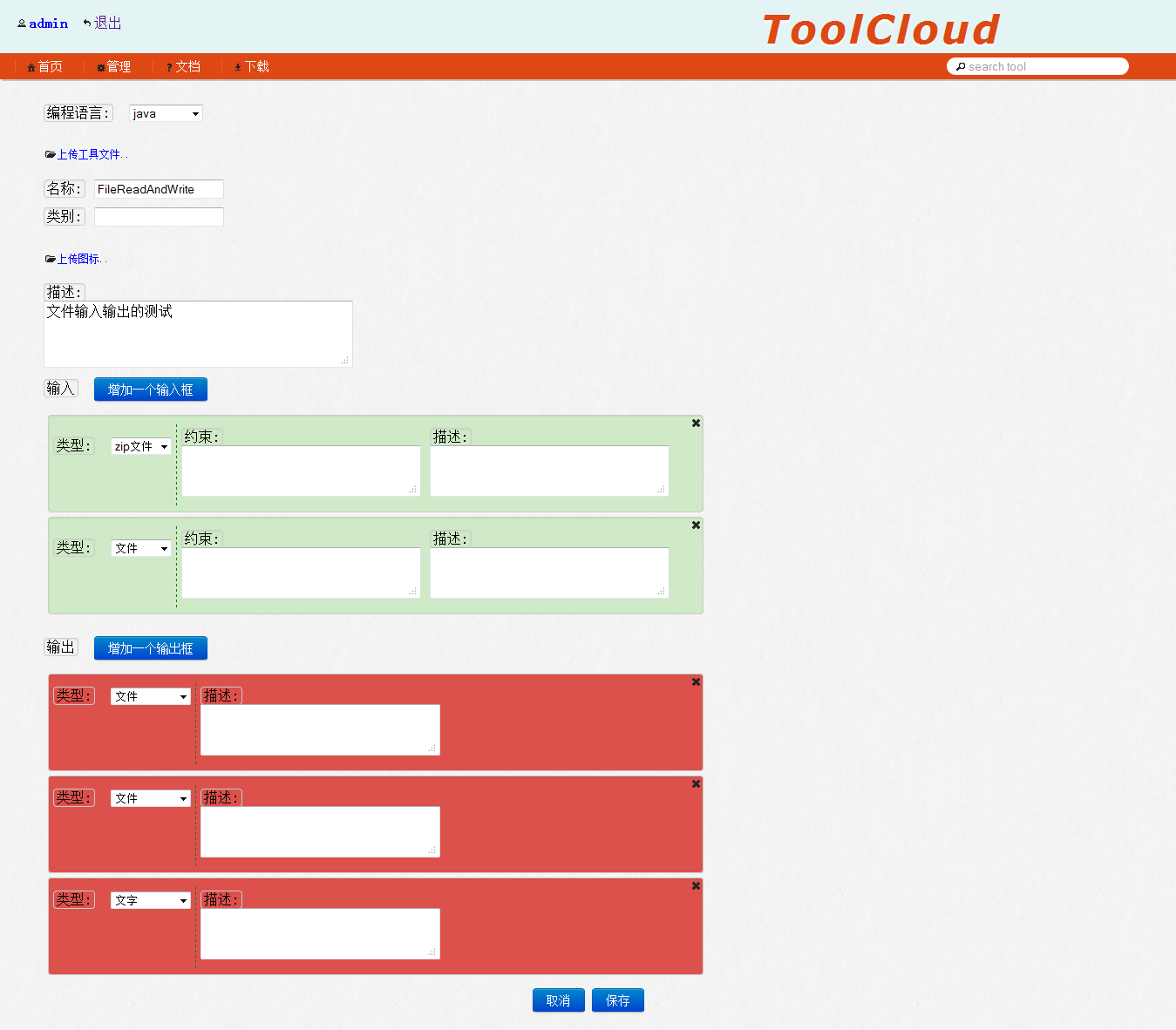
}

bw.flush();

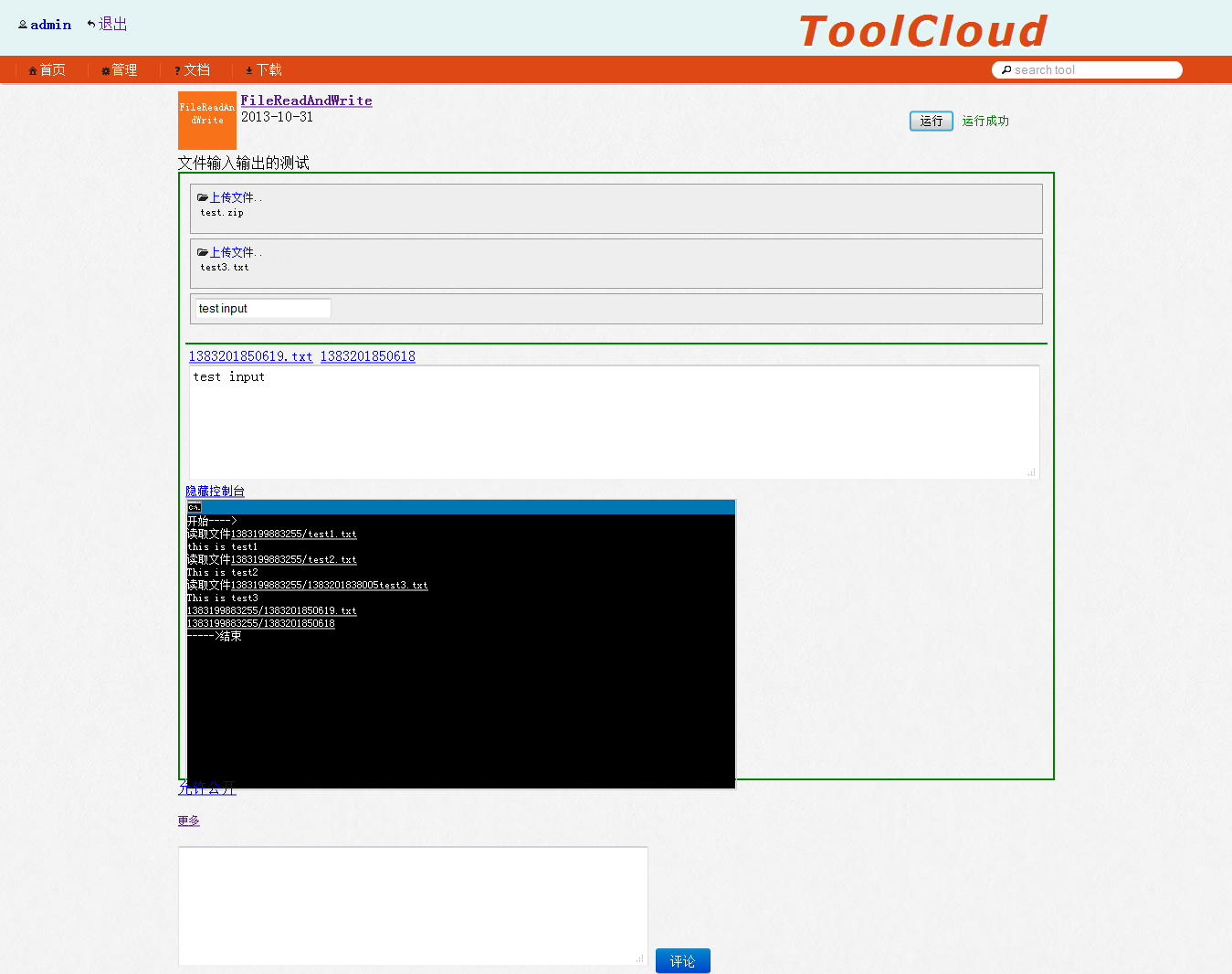
r.close();

}

然后打成Jar包，进入发布界面，设置三个输入（分别是zip文件、文件、字符串）和三个输出（分别是文件、文件、文字）。



进入运行界面，上传一个zip包，一个文件，并输入一段文字，打开控制台，看结果如图：



结果中有超链接，第一个是一个txt文件，第二个是zip文件。