

Jmeter测试工 具培训

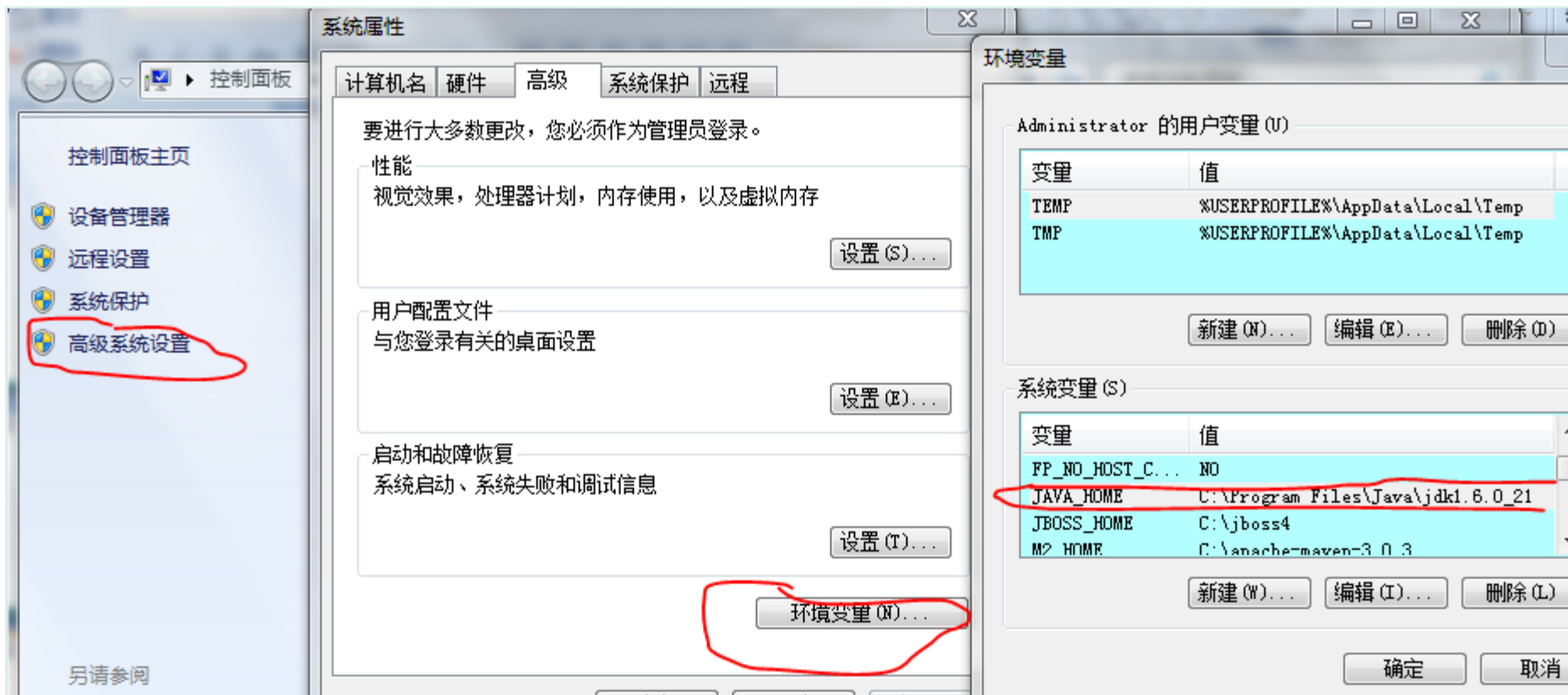
柯婷
2012-7

- 1、Jmeter介绍
- 2、Jmeter主要组件
- 3、Jmeter用例创建和测试

- Jmeter是基于Java的，开放源代码的测试工具
- Jmeter 可以用于测试静态或者动态资源的性能（文件、SOAP、Web、Perl脚本、java对象、数据库和查询、ftp服务器或者其他的资源）
- Jmeter 用于模拟在服务器、网络或者其他对象上附加高负载以测试他们提供服务的受压能力，或者分析他们提供的服务在不同负载条件下的总性能情况
- Jmeter的工作原理：向服务器提交请求；从服务器获取请求返回的结果
- 版本获取：官网地址
http://jmeter.apache.org/download_jmeter.cgi。下载后，直接解压，运行bin/jmeter.bat即可
- <http://code.google.com/p/jmeter-plugins/downloads/list>安装jmeter plugins（第三方插件）

Jmeter介绍二：环境配置

- Jmeter要求支持JAVA运行环境变量，且版本满足JVM1.3或更高



- 完成环境配置检查后，在windows则直接运行apache-jmeter\bin\jmeter.bat 就启动了该软件；linux系统下，运行jmeter.sh

- 1、测试计划 (Test Plan)
- 2、线程组 (Thread Group)
- 3、配置元件 (Configuration)
- 4、逻辑控制器 (Logic Controller)
- 5、取样器 (Sampler)
- 6、断言 (Assertion)
- 7、监听器 (monitor)
- 8、正则表达式提取器

Jmeter主要组件—测试计划

测试计划

名称: 测试计划

注释:

用户定义的变量

名称:	值
IP	192.168.1.1
port	8081

Detail

添加

Add from Clipboard

删除

Up

Down

☒ 独立运行每个线程组（例如在一个组运行结束后启动下一个）

☐ 函数测试模式

只有当你需要记录每个请求从服务器取得的数据到文件时才需要选择函数测试模式。

选择这个选项很影响性能。

Add directory or jar to classpath

浏览...

删除

清除

Library

- 测试计划：是使用Jmeter进行测试的起点，是其他测试元件的容器，一个完整的测试计划包括一个或者多个线程组、逻辑控制器、取样器、监听器、配置元件
- 用户自定义的变量：用户可以自己定义的变量，使用此变量时，直接`${变量名}`引用。 eg：
`${IP}`, `${port}`

Jmeter主要组件—线程组

线程组

名称: 线程组

注释:

在取样器错误后要执行的动作

☒ 继续 ☐ Start Next Thread Loop ☐ 停止线程 ☐ 停止测试 ☐ Stop Test Now

线程属性

线程数: 1

Ramp-Up Period (in seconds): 4

循环次数 ☐ 永远 2

☒ 调度器

调度器配置

启动时间 2012/06/26 17:13:21

结束时间 2012/06/26 17:13:21

持续时间 (秒)

启动延迟 (秒)

- 线程组：测试里每个任务都要线程去处理，所有任务必须在线程组下面创建。
- 线程数：设置发送请求的用户数目，即并发数
- Ramp-Up Period(in seconds)：创建完所有线程的时间，防止对服务器有太大的负载。
- 循环次数：请求的重复次数
- 启动时间：测试计划的启动时间
- 结束时间：测试计划结束时间
- 持续时间：测试计划持续时间（覆盖结束时间）
- 启动延迟：测试计划延迟启动时间（覆盖启动时间）

Jmeter主要组件—配置元件

用户定义的变量

名称: 地址配置

注释:

用户定义的变量

名称:	值	Description
url216	http://172.17.252.216:10080/app	
urly	https://58.246.226.97:8443/app	
url	http://172.17.252.214:10080/app	
url214	http://172.17.252.214:10080/app	
merchant	http://172.17.255.27:10080/app/merc...	
MerchantSimulator	http://172.17.249.95:8081/MerchantSi...	
encryptUtil	http://172.17.249.95:8081/MobileSim...	
MobileSimulator	http://172.17.249.95:8081/MobileSim...	

添加

Add from Clipboard

删除

Up

Down

- 配置元件维护Sampler需要的配置信息，配置元件并不发送请求（除了HTTP代理服务器例外），并根据实际的需要会添加或修改请求的内容
- 配置元件“用户定义的变量”会在测试的初始阶段执行（无论它处于测试树的哪个位置）。
- 配置元件仅对其所在的测试树分支有效

Jmeter主要组件—逻辑控制器

循环控制器

名称: 一期消费_首次

注释:

循环次数 ☐ 永远 1

简单控制器

名称: 信息获取

注释:

事务控制器

名称: 事务控制器

注释:

☒ Generate parent sample

☒ Include duration of timer and pre-post processors in generated sample

- 逻辑控制器可以控制JMeter发送请求的测试逻辑，它与Sampler结合使用可以模拟复杂的请求序列
- 循环逻辑器：设置请求的循环次数或永远循环，可用于压力测试
- 简单控制器：测试用例顺序执行
- 事务控制器：将多个请求放在同一个事务中。选中Generate parent sample，则聚合报告中只显示事务控制器的数据，不会显示其中的各个请求的数据，反之则全部显示。
- 逻辑控制器可以组合使用

Jmeter主要组件—取样器

SOAP/XML-RPC Request

名称: 支付初始化【小额】

注释:

URL: \${url}/entry

☒ Send SOAPAction:

☒ Use KeepAlive

Soap/XML-RPC Data

```
<?xml version="1.0" encoding="UTF-8"?>
<cupMobiles>
<cupMobile application="UPNoCard" version="1.01">
  <transaction type="PayInit.APReq">
    ${submitTime}
    ${order}
    ${transAmount}
    <terminal id="01042900"/>
    ${merchant}
    <mobileTerminal type="XT800" os="Android 2.1" imei="A00000015CD36DD"/>

    <sysProvide>12345678</sysProvide>
    ${serviceInfo}
    ${encryptedSessionKey}
  </transaction>
  ${senderSignature}
</cupMobile>
</cupMobiles>
```

文件名:

- 取样器 (Sampler) : 可以认为所有的测试任务都由取样器承担, 如: HTTP 请求、SOAP/XML-RPC Request。
- 取样器告诉Jmeter发送一个请求到指定服务器, 并等待服务器的请求。取样器会按照其在测试树中的顺序去执行, 还可以用逻辑控制器来改变取样器器运行的重复次数。

Jmeter主要组件—断言

响应断言

名称: 订单断言

注释:

Apply to:

☒ Main sample only ☐ Sub-samples only ☐ Main sample and sub-samples ☐ JMeter Variable

要测试的响应字段

☒ 响应文本 ☐ URL样本 ☐ 响应代码 ☐ 响应信息 ☐ Response Headers ☐ Ignore Status

模式匹配规则

☒ 包括 ☐ 匹配 ☐ Equals ☐ Substring ☐ 否

要测试的模式

要测试的模式

<submitTime>

<sessionKey>

添加

删除

- 断言：对取样器返回的请求结果给出判断，是否正确。
- 用户可以使用断言来检查从服务器获得的响应内容，用来判断请求响应的结果是否如用户所期望的

- 断言会影响作用域内的所有采样器。如果要让断言只影响某个采样器，需要将断言作为该采样器的子项
- 断言中设置要测试的模式,则会对响应的内容(如http请求后,服务器返回的内容)进行与字符串进行匹配. 如果返回的内容包括了字符串,则认为该次测试通过, 否则就是失败。一般通过添加监听器来监听响应断言

聚合报告

名称: 聚合报告

注释:

所有数据写入一个文件

文件名

Log/Display Only: ☐ 仅日志错误 ☐ Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
总体	0	0	0	0	9223372...	-9223372...	0.00%	.0/hour	.0

☐ Include group name in label? ☒ Save Table Header

- Label：取样器名称

➤ Samples：运行过程中一共发出了多少个请求，一般来说是其值=线程数*循环次数（前提：测试按计划跑完，其中没有异常结束的线程）。总体Samples是各请求Samples之和。

➤ Average：平均响应时间

➤ Median：响应时间中间值

➤ 90%Line：响应时间90%线

➤ Min/ Max：最小/最大响应时间
- Error%：出错率（出错的Request数/所有的request数）

➤ Throughput：吞吐量，每秒/每分钟（具体看“/”后面的单位）处理的Request数

➤ KB/sec：每秒从服务器端接收到的数据

➤ 【注意】关于总体值并不是各列的累加。是以所有Samples为样本的统计值，如：总体Min=min{各个Samples的Min}，总体Max=max{各个Samples的Max}



- 样本数目：发出去的请求数
- 最新样本：最近一个Request的响应时间
- 平均：所有Request响应时间的平均值
- 偏离：所有Request响应时间的标准差
- 吞吐量：单位时间内（一般是1分钟）处理Request的个数
- 中值：所有Request响应时间的中间值

监听器——用表格查看结果

用表格察看结果

名称: 用表格察看结果

注释:

所有数据写入一个文件

文件名

浏览...

Log/Display Only: ☐ 仅日志错误 ☐ Successes

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes

➤ 显示每一个请求的信息，包括发起请求的县城，开始时间，请求响应时间，请求字节数

Jmeter主要组件—正则表达式提取器



中国银联
China Unionpay

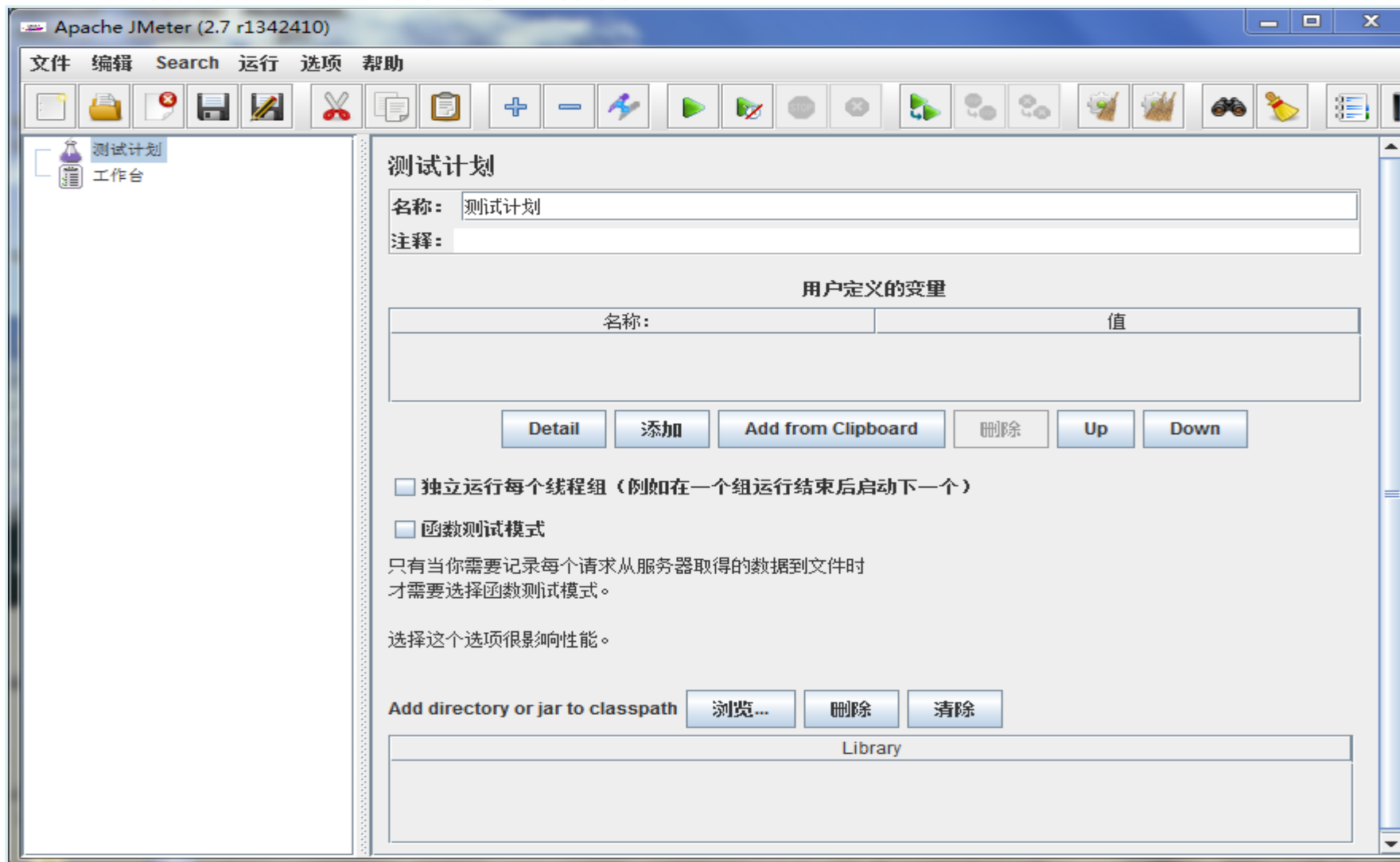
正则表达式提取器	
名称:	submitTime
注释:	
Apply to:	
<input type="radio"/> Main sample only <input type="radio"/> Sub-samples only <input checked="" type="radio"/> Main sample and sub-samples <input type="radio"/> JMeter Variable	
要检查的响应字段	
<input checked="" type="radio"/> 主体 <input type="radio"/> Body (unescaped) <input type="radio"/> 信息头 <input type="radio"/> URL <input type="radio"/> 响应代码 <input type="radio"/> 响应信息	
引用名称:	submitTime
正则表达式:	<submitTime>(.*?)</submitTime>
模板:	\$0\$
匹配数字 (0代表随机):	1
缺省值:	0

- 提取器作用：从请求的响应结果中取到需要的内容，从而实现关联
- 要检查的响应字段：正则表达式提取内容的范围

- 引用名称：其他地方引用提取值的变量名称，具体的引用方式是\${变量名}，eg:\${submitTime}
- 正则表达式：提取内容的正则表达式，()表示提取，对于你要提前的内容需要用小括号括起来
- 模板：用\$\$引用起来，如果在正则表达式中有多个提取表达式（多个括号括起来的东西），则可以是\$1\$，\$2\$等等，表示解析到的第几个值给变量，正则表达式的提取模式，值从1开始，值0对应的是整个匹配的表达式 如对于表达式s(.) 值0对应s(.*)，值1对应(.*)
- 匹配数字(0代表随机)：0代表随机，-1代表所有，其余正整数代表将在已提取的内容中，第几个匹配的内容。
- 缺省值：正则匹配失败时，取的值

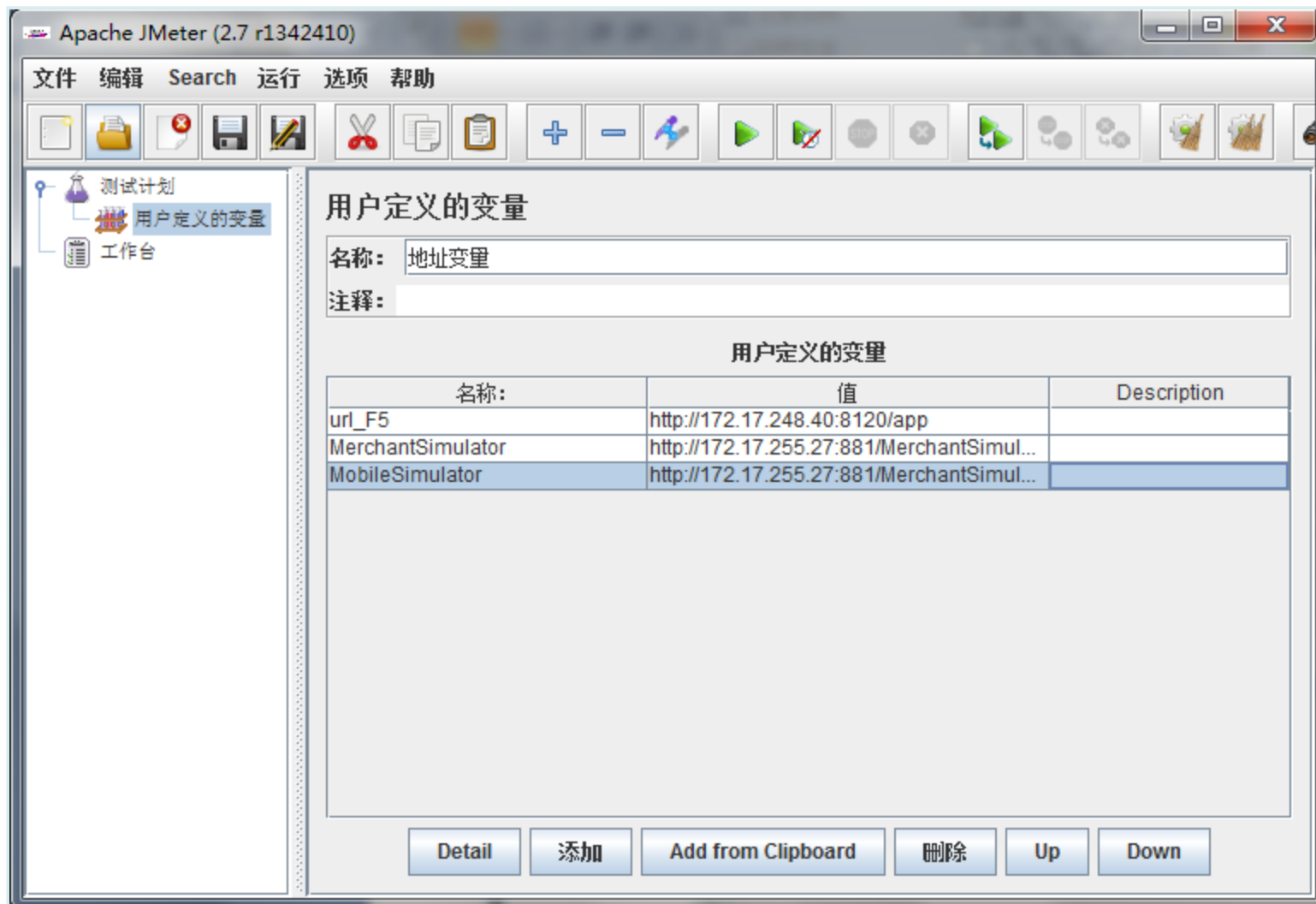
Jmeter用例创建之测试计划

- 新建测试计划。修改测试计划名称和运行模式
- 初始界面显示测试计划和工作台



Jmeter用例创建之用户变量

- 右键点击测试计划，添加->配置元件->用户定义的变量。配置用例需要的全局用户变量or地址变量



Jmeter用例创建之线程组&循环控制器



中国银联
China Unionpay

- 右键测试计划->添加->Threads (Users) ->线程组，建立线程组以及逻辑控制器
- 右键线程组->添加->逻辑控制器->循环控制器/简单控制器。建立控制器

The screenshot shows the JMeter GUI with the 'Thread Group' configuration panel. On the left, the tree view shows 'Test Plan' (测试计划) containing 'Address Variable' (地址变量), 'Thread Group' (线程组), and 'Loop Controller' (循环控制器). The 'Thread Group' is selected. The main panel has the following fields:

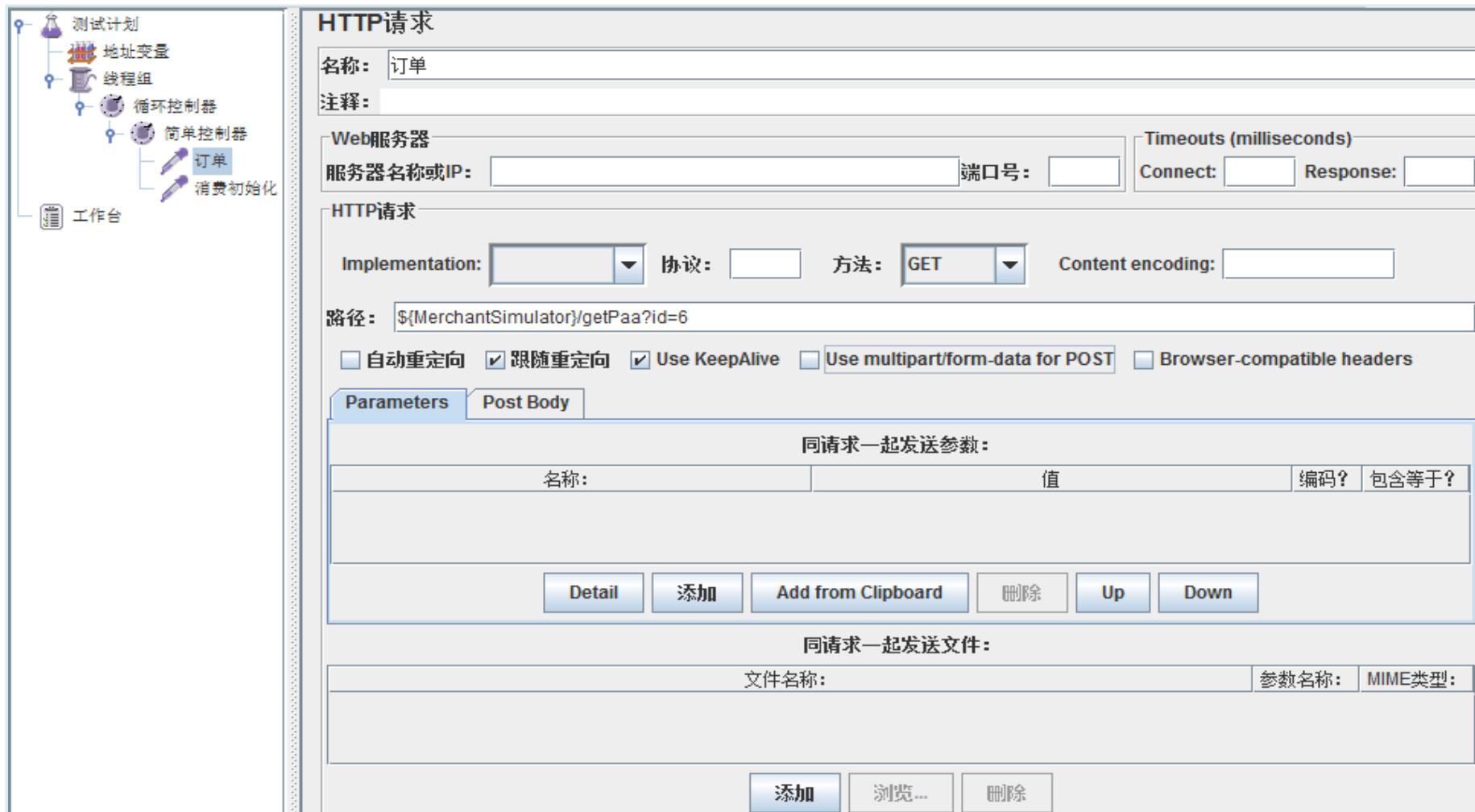
- 名称:** 线程组
- 注释:**
- 在取样器错误后要执行的动作:**
 - ☒ 继续
 - ☐ Start Next Thread Loop
 - ☐ 停止线程
 - ☐ 停止测试
 - ☐ Stop Test Now
- 线程属性:**
 - 线程数:** 1
 - Ramp-Up Period (in seconds):** 1
 - 循环次数:** ☐ 永远 1
 - ☐ 调度器

The screenshot shows the JMeter GUI with the 'Loop Controller' configuration panel. On the left, the tree view shows 'Test Plan' (测试计划) containing 'Address Variable' (地址变量), 'Thread Group' (线程组), and 'Loop Controller' (循环控制器). The 'Loop Controller' is selected. The main panel has the following fields:

- 名称:** 循环控制器
- 注释:**
- 循环次数:** ☐ 永远 1

Jmeter用例创建之取样器

- 右键控制器->添加->Sampler->HTTP请求or SOAP/XML-RPC Request，添加取样器
- 完成取样器配置和请求报文命令编写



The screenshot shows the JMeter configuration window for an HTTP Request sampler. The left sidebar displays the test plan structure: 测试计划 (Test Plan) -> 地址变量 (Address Variables) -> 线程组 (Thread Group) -> 循环控制器 (Loop Controller) -> 简单控制器 (Simple Controller) -> 订单 (Order) -> 消费初始化 (Initialize Consumption). The main configuration area is titled "HTTP请求" (HTTP Request).

HTTP请求配置:

- 名称:** 订单
- 注释:**
- Web服务器:**
 - 服务器名称或IP:**
 - 端口号:**
 - Timeouts (milliseconds):**
 - Connect:**
 - Response:**
- HTTP请求:**
 - Implementation:** (dropdown)
 - 协议:** (dropdown)
 - 方法:** GET (dropdown)
 - Content encoding:** (text field)
 - 路径:** \${MerchantSimulator}/getPaa?id=6
 - ☐ 自动重定向 ☒ 跟随重定向 ☒ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers
- Parameters / Post Body:**
 - 同请求一起发送参数:**

名称:	值	编码?	包含等于?

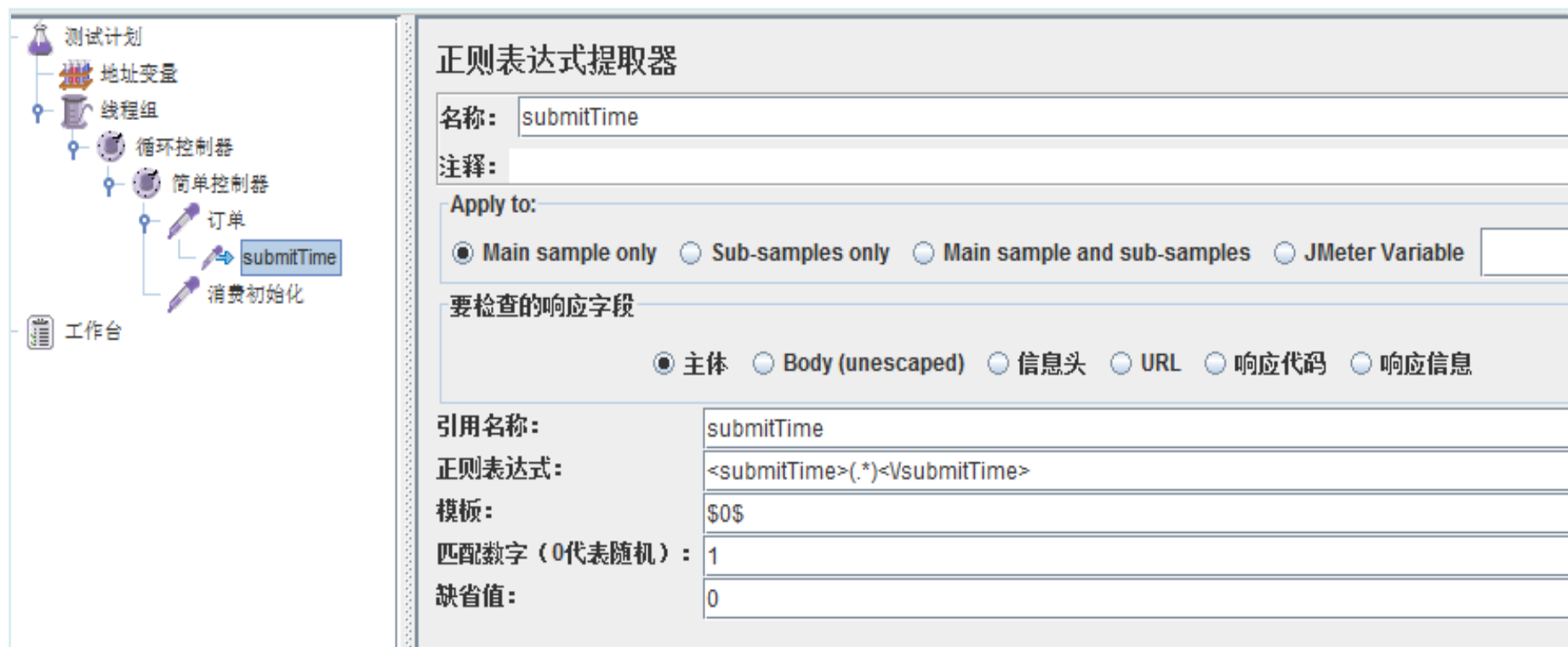
Buttons: Detail, 添加, Add from Clipboard, 删除, Up, Down
 - 同请求一起发送文件:**

文件名称:	参数名称:	MIME类型:

Buttons: 添加, 浏览..., 删除

Jmeter用例创建之正则表达式提取器

- 右键取样器->添加->后置处理器->正则表达式提取器，添加正则表达式提取器
- 进行正则表达式提取器模式和命令配置



The screenshot shows the JMeter GUI with a test plan on the left and the configuration for a Regular Expression Extractor on the right.

测试计划 (Test Plan):

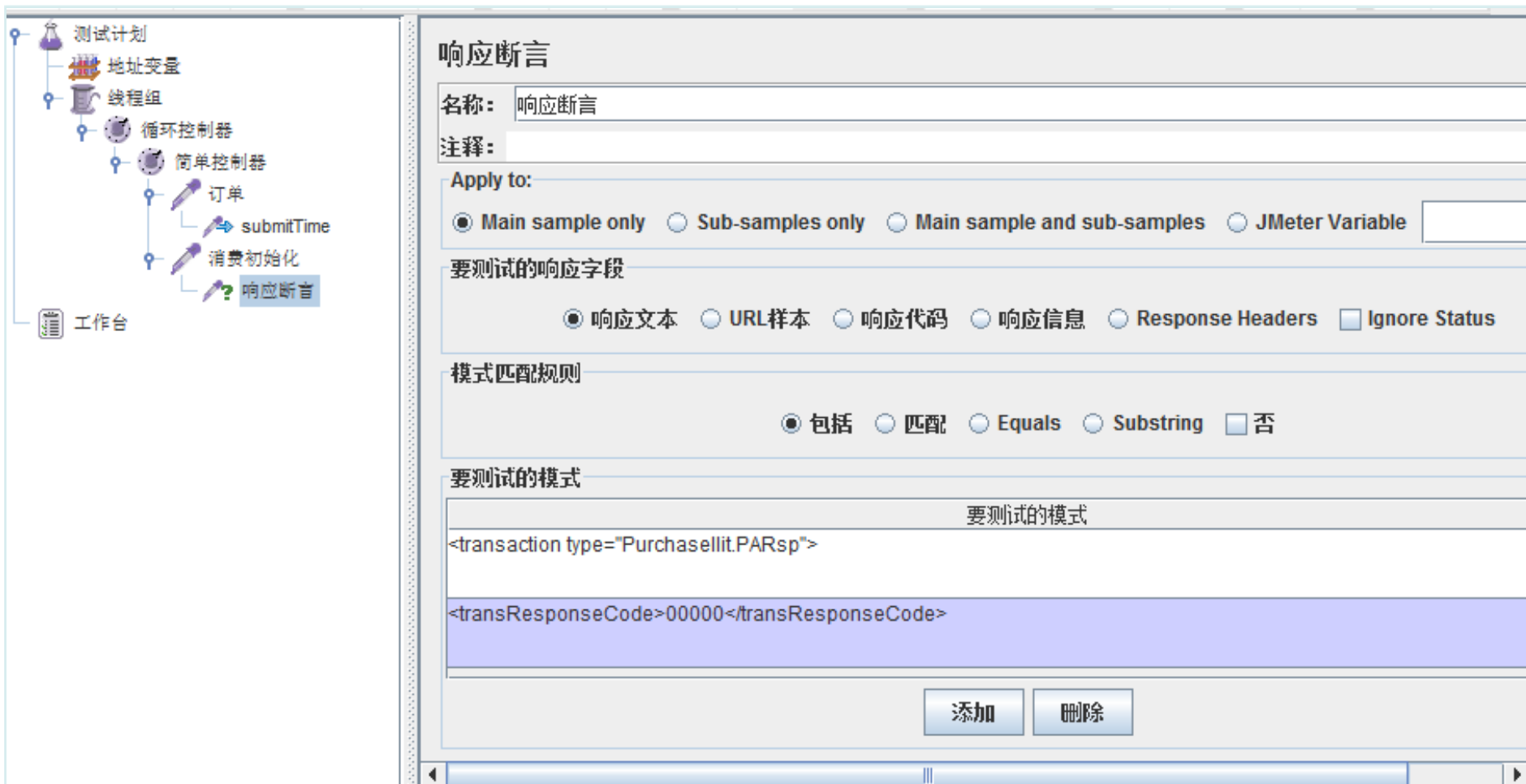
- 地址变量 (Address Variable)
- 线程组 (Thread Group)
- 循环控制器 (Loop Controller)
- 简单控制器 (Simple Controller)
- 订单 (Order)
- consumeInit (消费初始化)
- submitTime (Regular Expression Extractor)

正则表达式提取器 (Regular Expression Extractor) Configuration:

名称 (Name):	submitTime
注释 (Comment):	
Apply to:	<input checked="" type="radio"/> Main sample only <input type="radio"/> Sub-samples only <input type="radio"/> Main sample and sub-samples <input type="radio"/> JMeter Variable
要检查的响应字段 (Response field to check):	<input checked="" type="radio"/> 主体 (Body) <input type="radio"/> Body (unescaped) <input type="radio"/> 信息头 (Header) <input type="radio"/> URL <input type="radio"/> 响应代码 (Response code) <input type="radio"/> 响应信息 (Response message)
引用名称 (Reference name):	submitTime
正则表达式 (Regular expression):	<submitTime>(.*?)</submitTime>
模板 (Template):	\$0\$
匹配数字 (0代表随机) (Match number (0 represents random)):	1
缺省值 (Default value):	0

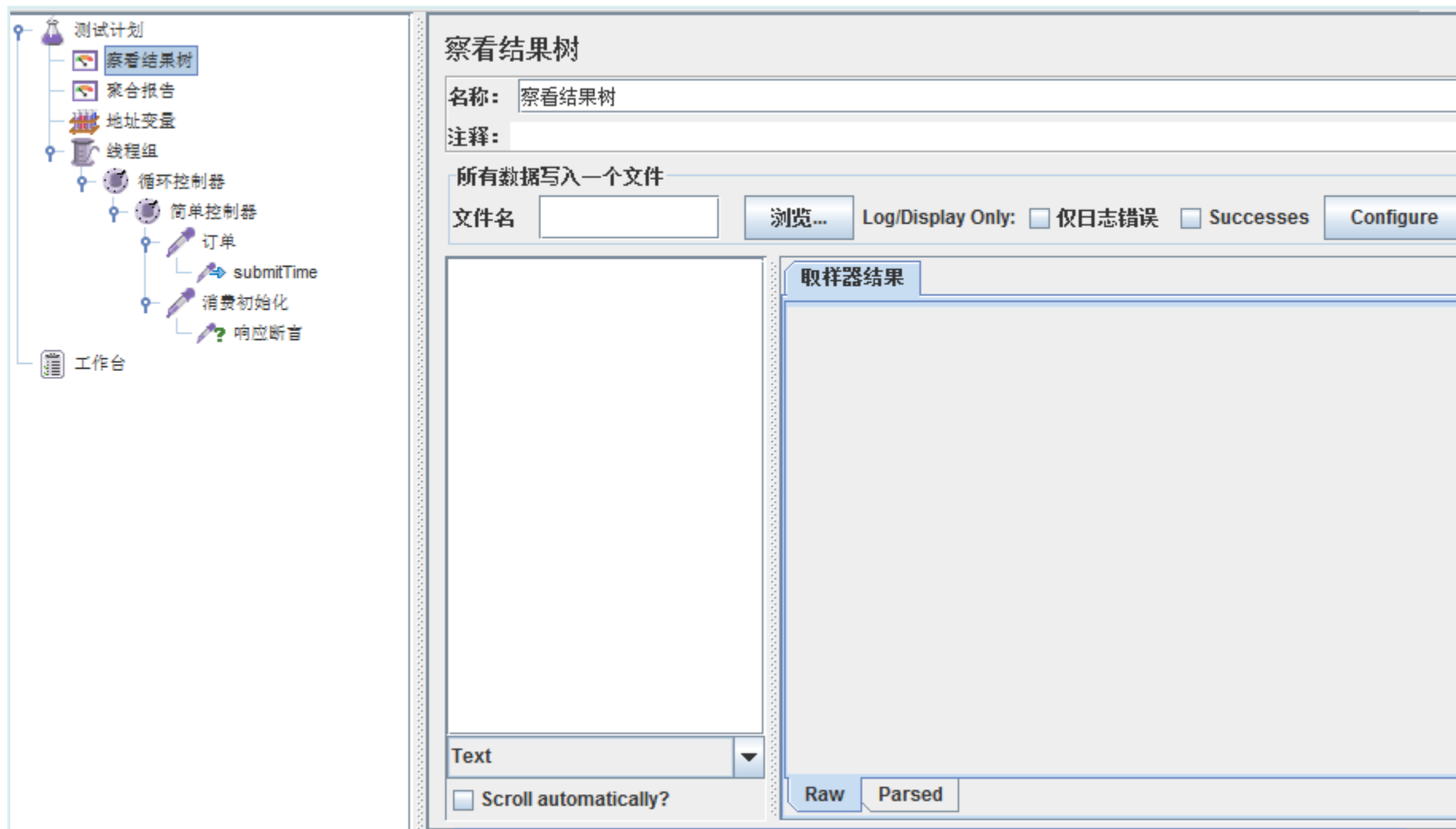
Jmeter用例创建之响应断言

- 右键取样器->添加->断言->响应断言，添加对应取样器的断言
- 进行断言内容定义和模式配置



Jmeter用例创建之监听器

- 右键测试计划->添加->监听器->察看结果树/聚合报告等 添加对应测试的监听器
- 根据要求配置监听器的写入文件等





Q&A