

MeterSphere使用规范

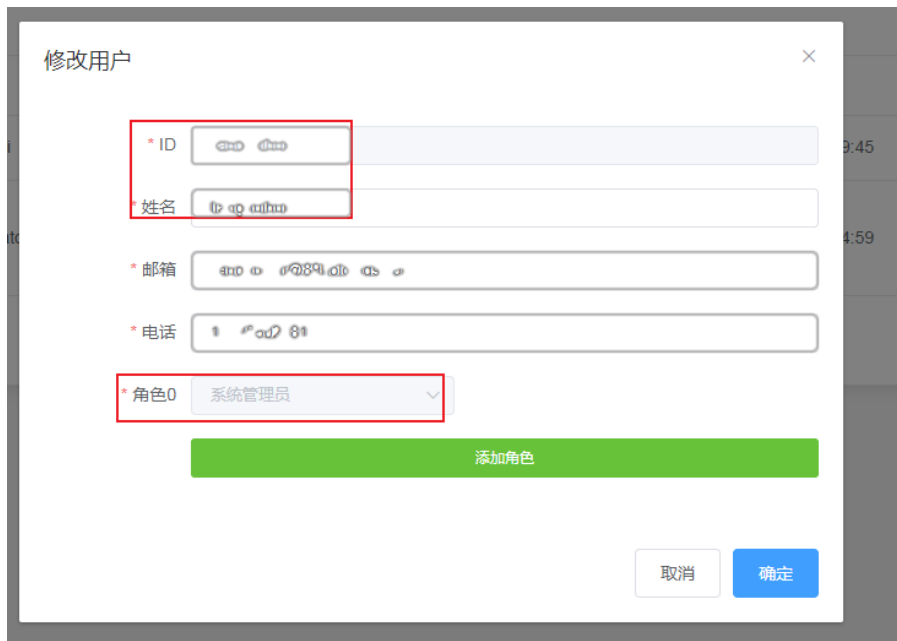
MeterSphere使用规范

一、系统设置及需求规范

1. 用户及其他命名规范

【测试人员】ID和姓名均设置为自己的名字全拼，角色分配为管理员；

【非测试人员】ID和姓名均设置为自己的名字全拼，角色分配为只读用户；



The screenshot shows a '修改用户' (Edit User) dialog box. It contains the following fields:

- * ID: A text input field with a red box around it.
- * 姓名: A text input field with a red box around it.
- * 邮箱: A text input field.
- * 电话: A text input field.
- * 角色0: A dropdown menu currently showing '系统管理员', with a red box around it.

Below the role field is a green button labeled '添加角色'. At the bottom right are '取消' (Cancel) and '确定' (Confirm) buttons.

【组织】只新建一个组织，名称为“公司名称-九里云科技（深圳）有限公司”，组织添加测试组人员；

【工作空间】工作空间命名为“质量支撑组”，工作空间添加测试组人员；

【项目】根据需求创建项目，命名方式为“需求名称项目”；另外增加一个项目名称为“自动化接口项目”

【自动化接口项目】所有测试通过的测试接口需要放置于该项目下，用于自动化接口测试；

【接口自动化】在项目下新建用例模块，然后在用例模块下新建场景，一个场景即为一个接口，接口的命名为“融担平台-接口功能-普通接口/性能接口”：，同一个接口要命名一致；

2. 消息设置

- Jenkins接口调用任务通知：jenkins集成自动化job失败，发送钉钉信息到测试组群里，另外发送邮件信息给指定人员，邮件包含测试报告；

消息设置

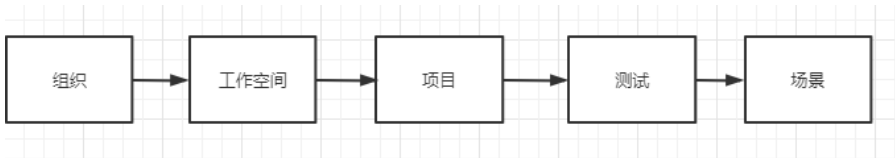
注意：1. 钉钉和企业群里新建一个自定义机器人，然后复制 webhook 地址在我们平台上；2. 机器人选择为群机器人，安全验证选择“自定义关键词”；“任务通知”；3. 选择接收人时必须是你所建的群里的成员，接收人手机号为必填项且为钉钉企业所使用的手机号。

Jenkins接口调用任务通知

创建新通知

事件	接收人	接收方式	webhook	操作
执行失败	Administrator	钉钉机器人	https://oapi.dingtalk.com/robot/send?ac...	<input type="button" value="编辑"/> <input type="button" value="删除"/>

整体的测试项目框架：



- 测试计划任务通知：测试计划有了更新（状态），对测试计划的创建人通过钉钉通知

测试计划任务通知

创建新通知

事件	接收人	接收方式	webhook	操作
更新	创建人	钉钉机器人	https://oapi.dingtalk.com/robot/send?ac...	<input type="button" value="编辑"/> <input type="button" value="删除"/>

3. 测试报告模板

自定义一套通用的测试用例模板

4. 项目管理

4.1 创建项目

创建项目 ×

* 名称

描述

接口定义URL可重复

取消

确定

4.2 jar包管理

在项目中添加jar包，可以在项目中的任意接口进行全局调用

jar包管理 ×

* 名称

描述



将文件拖到此处，或点击上传

文件大小不超过 20 M

添加





名称	Jar包	描述	更新时间	创建人	修改人	操作
重构后的jar包	auto_interface...		2021-02-04 1...	admin	admin	
随机生成身份...	java-generato...		2021-01-13 1...	weimenghua	weimenghua	

4.3项目设置

4.3.1HTTP配置

环境配置

环境列表 +




- 
- 
- 
- 

环境名称

通用配置 **HTTP配置** 数据库配置 TCP配置

环境域名

请求头

<input checked="" type="checkbox"/>	Content-Type	12/200	<input type="text" value="application/json"/>	
<input checked="" type="checkbox"/>	isTest	6/200	<input type="text" value="true"/>	
	键	0/200	<input type="text" value="值"/>	

4.3.2数据库配置

数据库连接URL:jdbc:mysql://xxxx.xxx.xxx.xxx:3306/xxx?characterEncoding=UTF-8

The screenshot shows a configuration window with the following fields and values:

- 数据源名称: [Empty]
- 数据库连接URL: jdbc:mysql://xxxx.xxx.xxx.xxx:3306/xxx?characterEncoding=UTF-8
- 数据库驱动: com.mysql.jdbc.Driver
- 用户名: [Empty]
- 密码: [Empty]
- 最大连接数: 10
- 超时时间(ms): 3000

Buttons: 校验, 更新, 清空

二、测试跟踪

1. 测试用例

- 将已经测试通过的接口测试或者性能测试，根据需求的测试维度创建一级模块
- 在对应得模块下，添加测试用例，规范如图：

The screenshot shows a form for creating a new test case with the following fields and values:

- 用例名称: 测试用例名称就跟项目下的测试保持一致
- 所属模块: 该模块就是在当前计划下新建的模块
- 维护人: admin (Administrator)
- 用例等级: P3
- 类型: 接口测试
- 测试方式: 自动
- 关联测试: [Empty]
- 前置条件: 请输入前置条件
- 备注: 请输入内容
- 附件: [Empty]

Buttons: 保存并继续创建, 取消, 确定

2. 测试计划

- 创建测试计划，关联测试计划，并将其集成到jenkins。

创建测试计划

创建测试计划

* 计划名称 命名: 系统名称+需求名称

* 负责人 负责人跟创建人保持一致

计划开始 计划开始

所属项目 选择项目

测试阶段 测试阶段选择回归测试

计划结束 计划结束

描述 请输入内容

取消 确定

给测试计划关联测试用例

关联测试用例

搜索用例

关联用例计划

借款-XNA-360

111

模块1

用例名称

用例等级

类型

已经关联了 1 共 0 条

暂无数据

取消 确定

- 将测试计划集成到jenkins见 "六、jenkins集成自动化测试"

三、接口测试

1. 创建项目

根据需求创建项目，命名方式为“需求名称项目”；

创建项目

* 名称

描述

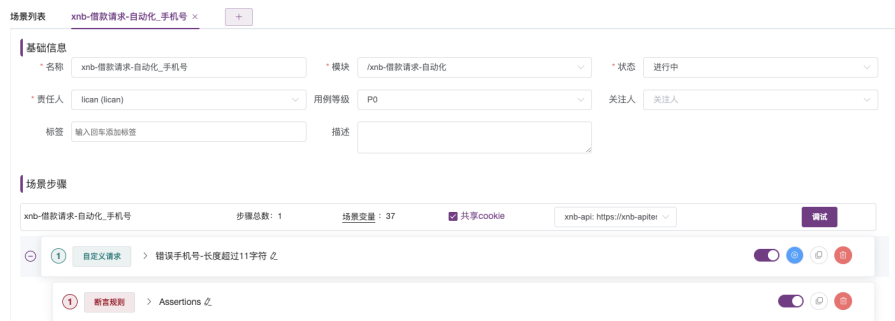
取消 确定



2. 创建测试

2.1 直接创建场景

在项目下新建测试，一个测试为一个场景，普通接口名称命名规范：融担平台-接口功能-普通接口



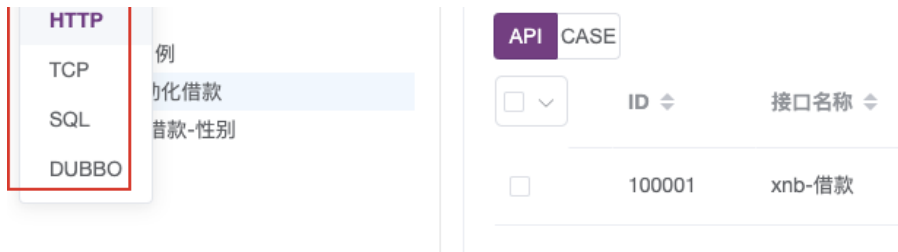
2.2 旧版转场景测试



3. 接口定义

3.1 接口类型





3.2功能模块

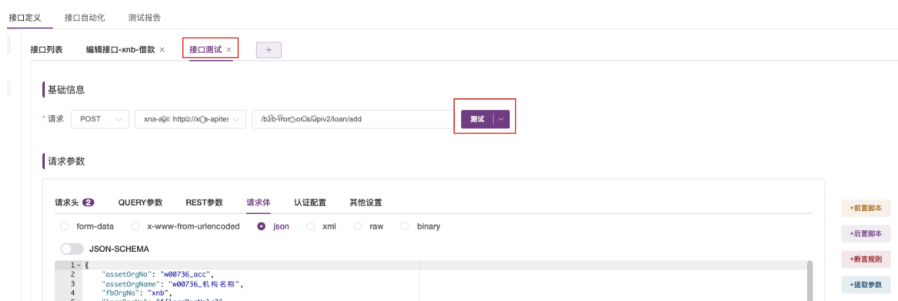
左侧的用例模块支持添加子模块，同时也支持上下拖拽，便于调整层级顺序



3.3API接口



测试它的接口时，系统会生成一条测试接口，同时可以将单个接口另存为接口用例



3.4接口用例



接口用例会增加运行环境地址、前置脚本、断言等内容



4.接口自动化

4.1回收站

删掉后的接口，会自动转入回收站中，可以将其恢复或者彻底删除



4.2用例模块

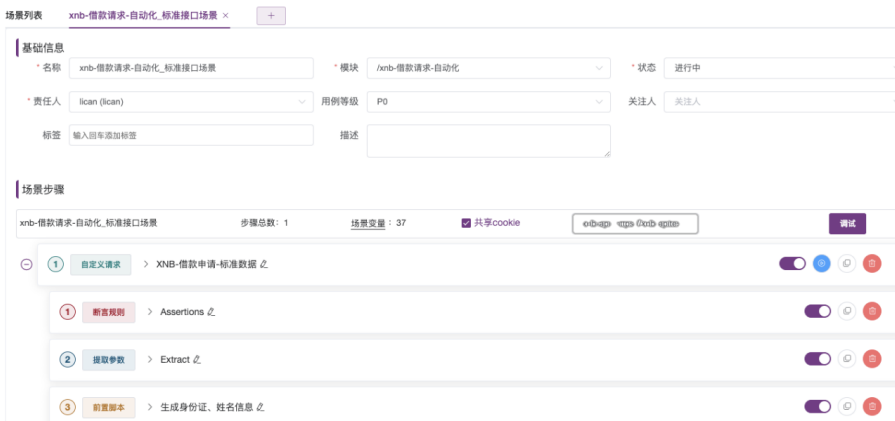
每一条用例模块，会对应多个场景



4.3接口场景列表

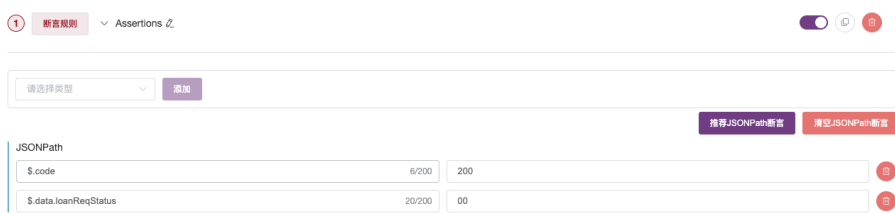


4.4接口详情



4.4.1断言规则

4.4.1.1JSONPath断言



响应时间
0

4.4.1.2脚本断言

脚本

asset_loan_order_no_1 等于 loan20210127122

``${asset_loan_order_no_1}` == 'loan20210127122'`

参考文档

```
1 value = vars.get("asset_loan_order_no_1");
2 result = "loan20210127122".equals(value);
3 if (!result){
4     msg = "assertion [`${asset_loan_order_no_1}` == 'loan20210127122']: false;";
5     AssertionResult.setFailureMessage(msg);
6     AssertionResult.setFailure(true);
7 }
```

beanshell

- 代码模板
- 获取变量
- 设置变量
- 获取全局变量

4.4.2前置脚本

前置脚本 个人信息

参考文档

```
1 import cn.binarywang.tools.generator.*;
2
3 String custName = ChineseNameGenerator.getInstance().generate();
4 vars.put("custName", custName);
5 log.info("姓名: " + custName);
6
7 String loanReqNo = "loanOrderNo${_time(YMD)}${_counter(FALSE)}";
8 vars.put("loanReqNo", loanReqNo);
9 log.info("借款订单号: " + loanReqNo);
10
11 String id=ChineseIDCardNumberGenerator.getInstance().generate();
12 vars.put("id", id);
13 log.info("身份证: " + id);
```

beanshell

- 代码模板
- 获取变量
- 设置变量
- 获取全局变量
- 设置全局变量
- 获取响应头
- 获取响应码
- 获取响应结果

4.5场景变量

自定义增加全局变量，可以引入常量、列表、csv、计数器、随机数等

ID	变量名	类型	值
1	loanReqNo	常量	<code>`\${_time(YMD)}`</code>
2	sourceCode	常量	QH
3	custName	常量	@cname
4	idType	常量	1
5	id	常量	@id
6	sex	常量	F
7	dbBankCode	常量	1

变量名:

描述:

值:

删除 添加

- 常量
- 列表
- CSV
- 计数器
- 随机数

5.场景的导入方式

5.1接口列表导入

5.1.1复制接口

接口导入

HTTP 搜索模块

- 全部用例
- 自动化借款

API	CASE	请搜索						
<input checked="" type="checkbox"/>	接口名称	接口状态	请求类型	路径	最后更新时间	用例数	用例状态	用例通过率
<input checked="" type="checkbox"/>	xnb-借款	进行中	POST	/b2b-fr...	2021-02-22 14:34:44	1	未执行	0.00%

共 1 条 10条/页 < 1 > 前往 1 页

已选中 1 条数据

复制

5.1.2复制或引用接口用例

接口导入

HTTP 搜索模块

- 全部用例
- 自动化借款

API	CASE	搜索用例			
<input checked="" type="checkbox"/>	接口名称	用例等级	路径	创建人	最后更新时间
<input checked="" type="checkbox"/>	借款-正常场景	P0	/b2b-front-oms/a...	lican	2021-02-19 16:47:52

共 1 条 10条/页 < 1 > 前往 1 页

已选中 1 条数据

复制 引用

5.2场景导入

场景导入

搜索模块

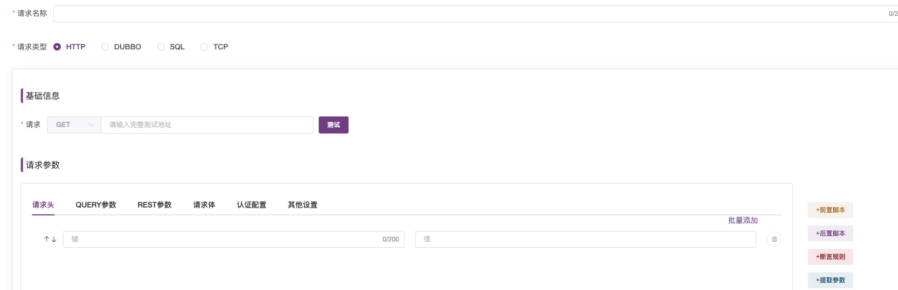
- 全部用例
- xnb-loan-用于压测
- xnb-repay-用于压测
- xnb-借款请求-自动化
- xnb-还款请求-自动化
- 测试demo
- 验证压力测试demo

根据 ID/名称/标签 搜索 高级搜索

<input checked="" type="checkbox"/>	ID	场景名称	用例等级	当前状态	标签	创建人
<input type="checkbox"/>	100006	xnb-借款请...	P0	进行中		lican
<input type="checkbox"/>	100004	xnb-loan-用...	P0	进行中		lican
<input type="checkbox"/>	100002	xnb-repay-用...	P0	进行中		lican

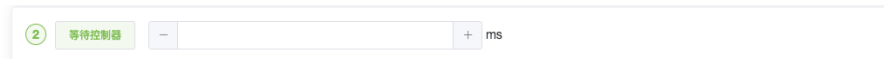
5.3自定义请求

通过添加自定义请求，用户可以自由编辑该请求的详细信息，该请求不需要事先存在于接口定义中。目前支持 HTTP、TCP、DUBBO、SQL 等不同类型的请求



5.4等待控制器

当某个步骤执行后需要等待一段时间时使用。作为某个步骤的子步骤添加时，该步骤将等待若干时间后再执行；当作为上级步骤添加时，与之同级的所有步骤均会等待若干时间



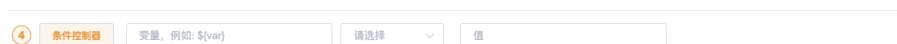
5.5循环控制器

目前 MeterSphere 提供了次数循环、ForEach 循环和 While 循环三种循环方式



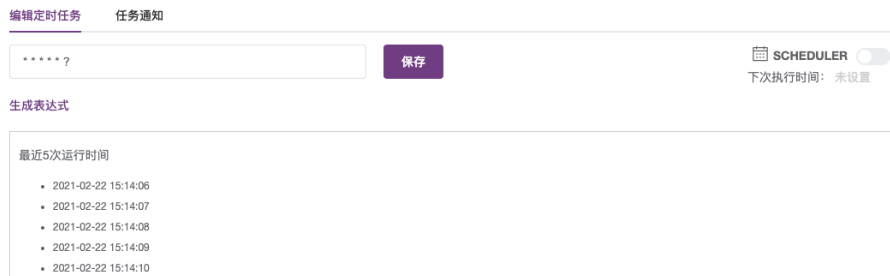
5.6条件控制器

当条件控制器中配置的条件满足时，条件控制器下的子步骤才会执行，否则子步骤会被跳过



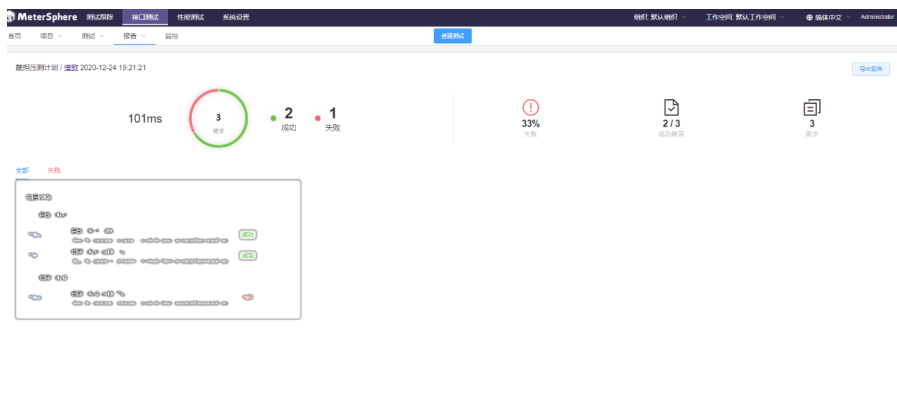
6.定时执行

请求页的右上角，可以设置定时任务，并选择将任务成功或者失败，以钉钉的形式通知到指定的成员。



7.测试报告

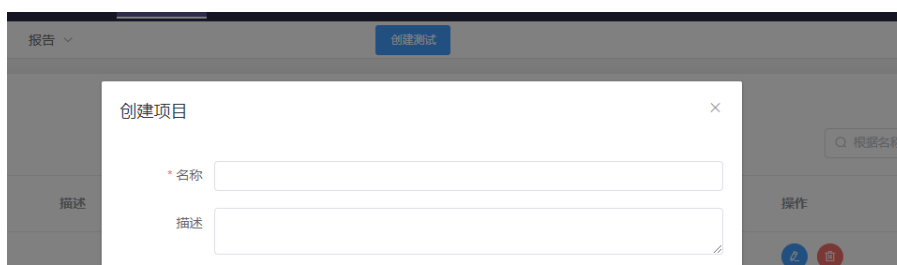
该测试报告可以查看测试执行的具体结果和请求信息。



四、性能测试

1. 创建或者选择项目

根据需求创建项目，命名方式为“需求名称项目”；

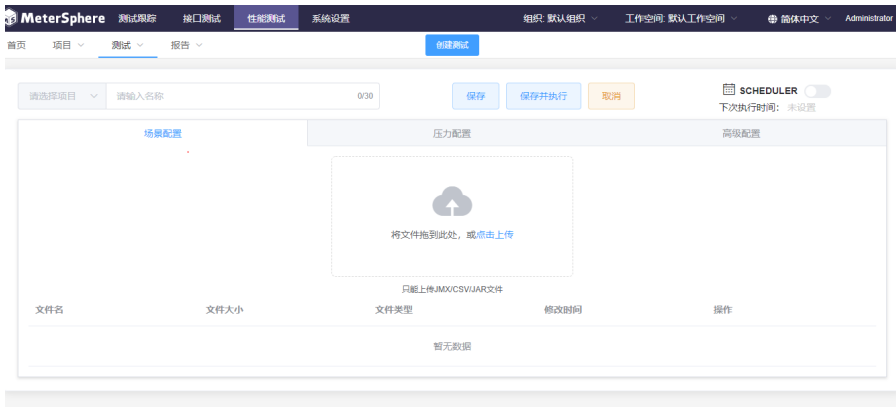




2. 创建测试

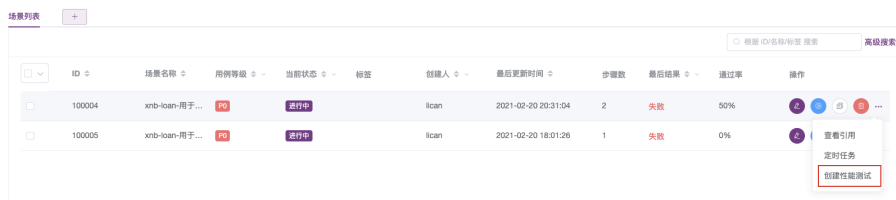
- 场景配置有且必须上传一个jmx文件，也可以选择上传参数文件csv或者jar

2.1 直接创建测试

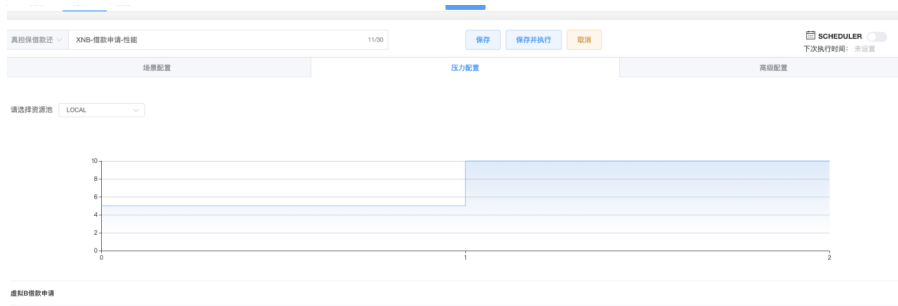


2.2 接口测试转性能测试（推荐）

从接口测试中选择对应的接口，然后点击“创建性能测试”



性能接口名称命名规范：融担平台-接口功能-压测



3. 场景配置

增加其jar包java-generator.jar用于性能测试

```
-rw-r--r-- 1 root root 4671521 1月 19 20:45 creat_date.jar
-rw-r--r-- 1 root root 123 1月 19 20:46 Dockerfile
```

注：接口测试转性能测试时，本资源池无法读取csv/jar文件，需新增资源池用于存放包

3.1借款-请求体

性能接口不支持mock数据引用

```
{
  "id": "${id}",
  "sex": "F",
  "dbBankCode": "0001",
  "dbAcct": "6217002020039444",
  "dbAcctName": "银行卡账户名XNA_360",
  "loanDate": "2020-01-01 01:01:01",
  "loanAmt": "100.00",
  "lnTerm": "3",
  "rpyType": "02",
}
```

3.2借款-预执行脚本

性能接口不支持mock数据引用

```
import cn.binarywang.tools.generator.*;

String custName = ChineseNameGenerator.getInstance().generate();

vars.put("custName",custName);

log.info("姓名： " + custName);

String loanReqNo = "orderNo${__time(YMD,)}"+"${__counter(FALSE,)}";

vars.put("loanReqNo",loanReqNo);

log.info("订单号： " + loanReqNo);

String id=ChineseIDCardNumberGenerator.getInstance().generate();

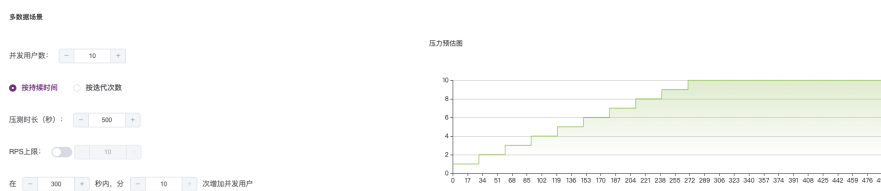
vars.put("id",id);

log.info("身份证： " + id);
```

4.压力配置

【资源池】：为性能测试提供独立节点类型的测试资源池,接收来自系统的性能测试任务,动态的启动 JMeter 容器完成性能测试,可以理解为压测的服务器,在 系统设置->系统->测试资源池 中进行管理

压力设置详情如图所示：RPS-每秒钟的请求数



4. 高级设置

- 自定义属性：自定义变量，定义的变量可以在JMX脚本中引用，暂只支持常量赋值
- 其他设置：根据实际请求设置

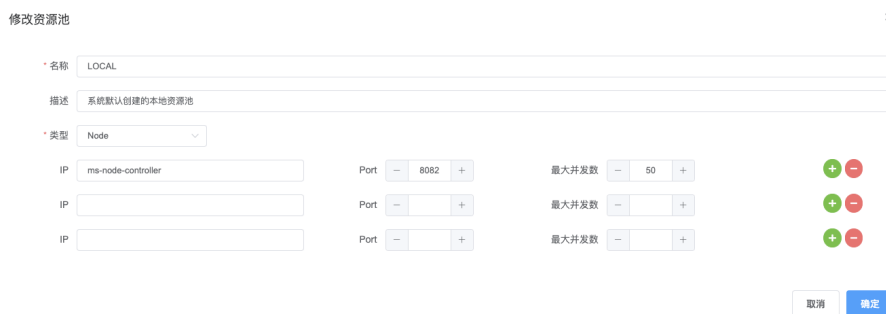




5. 支持分布式的性能测试

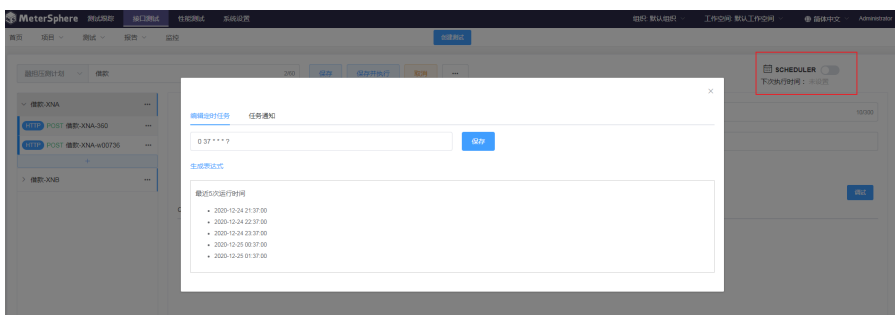
通过在测试资源池中添加多个测试执行节点的方式来支持分布式的性能测试

当我们在执行性能测试的过程中选择了某个测试资源池时，MeterSphere 会将本次性能测试定义的并发用户数，按照所选测试资源池的节点支持的最大并发数进行按比例拆分



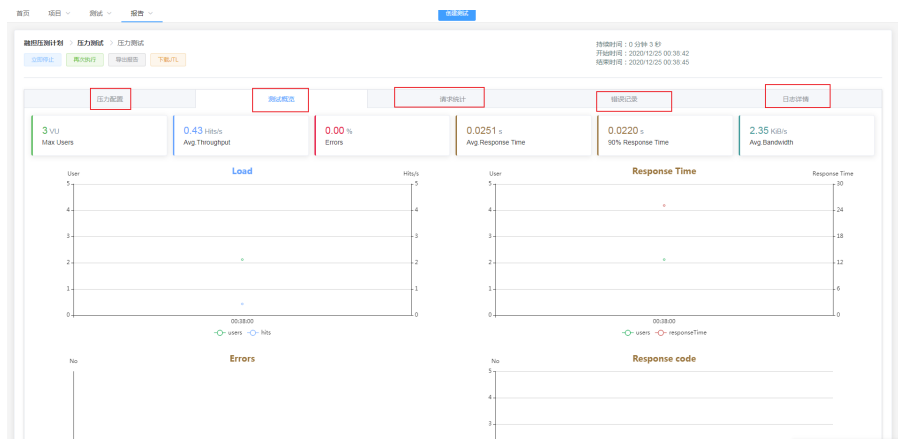
6. 定时执行（后续通过Jenkins集成）

请求页的右上角，可以设置定时任务，并选择将任务成功或者失败，以钉钉的形式通知到指定的成员。



7. 测试报告

在测试报告可以查看测试报告：



五、自动化项目

【说明】每一个完整业务流程都会有一个对应的自动化项目，命名为“业务名称+自动化项目”

名称	描述	创建时间	更新时间	操作
自动化接口	这里是发布所有测试通过并需要被自动化的接口	2020-12-29 10:06:37	2020-12-29 10:06:37	[+][●][●]
具保保借款还款项目		2020-12-28 19:12:38	2020-12-28 19:12:38	[+][●][●]

【业务增加新的接口】增加了新的接口，在测试通过之后，把该接口复制到该业务的自动化项目中；

自动化接口命名规范：名称+“-自动化”

名称	所属项目	创建人	创建时间	更新时间	状态	操作
XNB-还款申请-自动化	自动化接口	lican	2020-12-29 16:02:51	2020-12-29 17:14:10	Completed	[+][●][●]
XNA-还款申请-自动化	自动化接口	hangshuhui	2020-12-29 15:56:12	2020-12-29 17:13:52	Completed	[+][●][●]
XNB-借款申请-自动化	自动化接口	lican	2020-12-29 16:02:18	2020-12-29 16:47:15	Completed	[+][●][●]
XNA-借款申请-自动化	自动化接口	hangshuhui	2020-12-29 15:53:36	2020-12-29 16:21:10	Completed	[+][●][●]

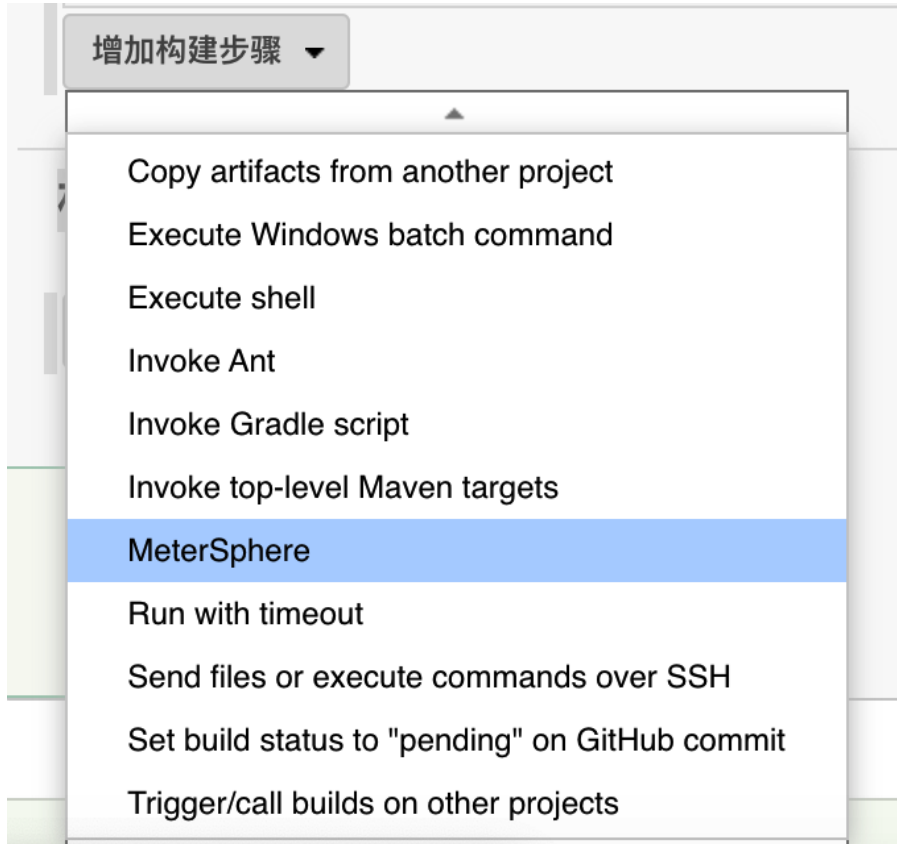
【业务修改旧接口】修改了旧接口，在测试通过之后，需要更新该业务的自动化项目中对应的接口

六、jenkins集成自动化测试

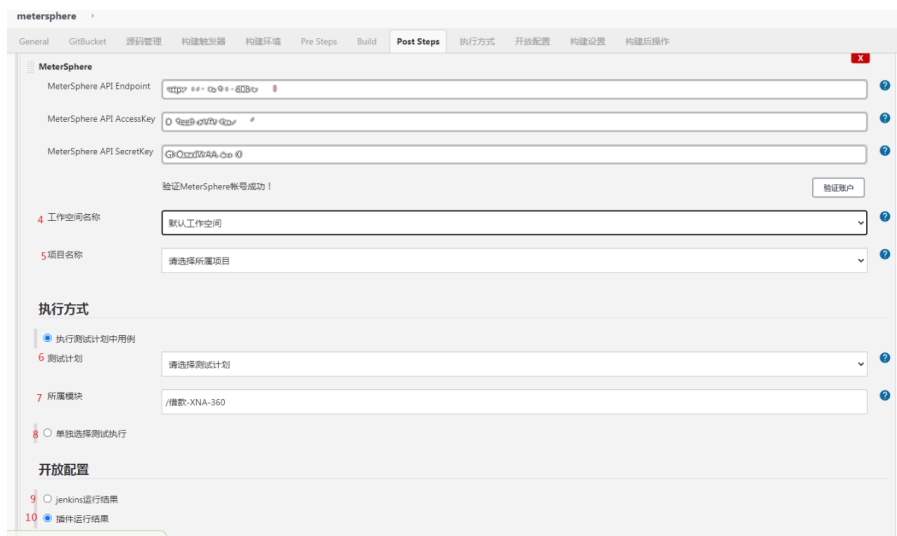
1. jenkins插件

2. jenkins配置

在job中选择“增加构建步骤”的MeterSphere；

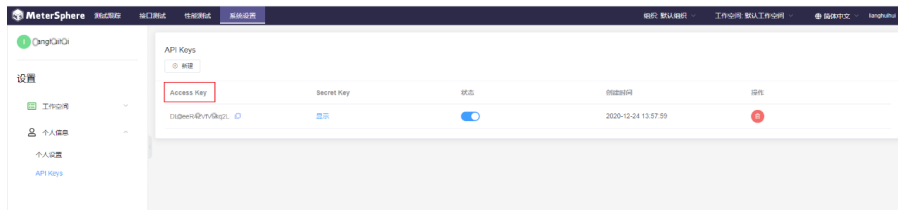


配置如图：

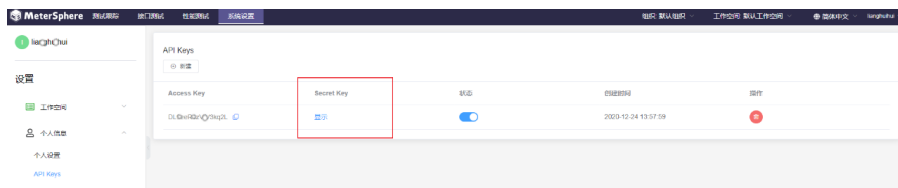


【1-MeterSphere API Endpoint】：MeterSphere的访问地址；

【2-MeterSphere API AccessKey】：个人设置-》个人信息-》API Keys-》Access Keys



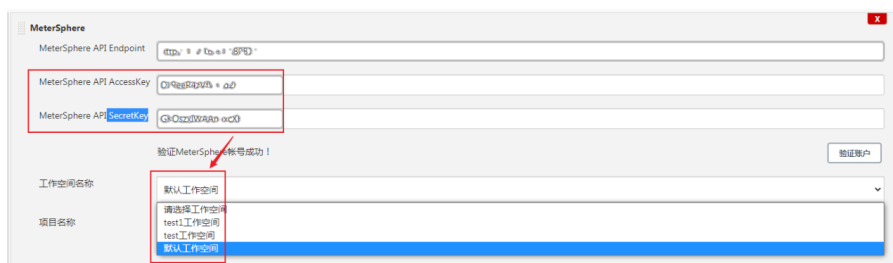
【3-MeterSphere API SecretKey】：个人设置-》个人信息-》API Keys-》Secret Key



每个账户都有MeterSphere API AccessKey 和MeterSphere API SecretKey，添加之后，点击验证账户验证



【4-工作空间名称】展示的是3、4输入的Key的账户下的所属工作空间



【5-项目名称】展示的是所选的工作空间的项目

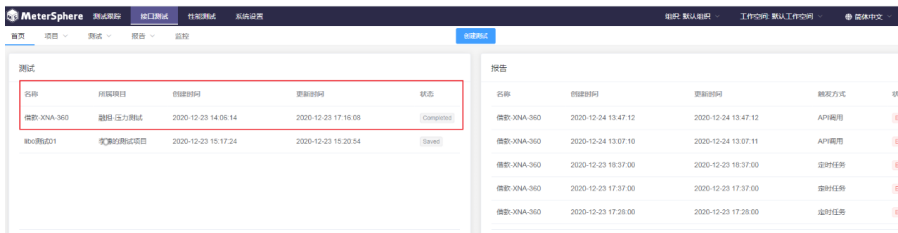
【6-测试计划】展示的是对应工作空间下和所属项目的测试计划



【7-所属模块】写入的是该项目的测试计划属于的模块



【8-测试名称】选择具体的测试执行



【9-jenkins运行结果】不依赖测试用例的运行结果

【10-插件运行结果】测试用例成功与否决定job的运行结果

3. 接口和性能自动化测试规范

- 将已经测试通过的接口测试或者性能测试，根据需求的测试维度创建一级模块
- 在对应得模块下，添加测试用例：



备注

请输入内容 **根据实际情况填写**

附件

+ 只能上传jpg、jpeg、png、docx、doc、pdf、xlsx文件

文件名	文件大小	文件类型	上传时间	操作
暂无数据				

保存并继续创建

· 创建测试计划

创建测试计划

* 计划名称 **命名：系统名称+需求名称**

* 所属项目 **选择项目**

* 负责人 **请负责人跟创建人保持一致**

* 测试阶段 **测试阶段选择回归测试**

计划开始

计划结束

描述 请输入内容

· 给测试计划关联测试用例

关联测试用例

搜索测试用例

用例名称	用例等级	类型
暂无数据		

已经到底了！
共 0 条

- 于jenkins配置自动化执行job

- jenkins执行策略：

【执行时间】：每天凌晨3: 50左右执行；代码master分支有更新，接口测试和性能测试执行部署；

【执行环境】：测试环境

【执行结果通知】：job构建失败，以邮件的方式通知并附带测试报告，测试报告将直接展示在邮件内容中；以钉钉的形式通知测试组并附带测试报告链接

【执行结果通知对象】：测试组及对应开发负责人

七、Chrome plugin: 浏览器插件

【功能】录制 Web 访问请求生成 JMeter 脚本并导入到 MeterSphere 中用于接口测试及性能测试

【使用方法】见链接：<https://github.com/metersphere/chrome-extensions>

八、测试使用脚本和jar管理

1. 脚本管理

接口和性能测试：

【预执行脚本】目前放置于具体请求的预执行选项下

【后执行脚本】暂时没有使用

性能测试：

【JMX】第一种：上传JMX文件，现由jmeter生成，文件上传不要轻易删除，以留后用

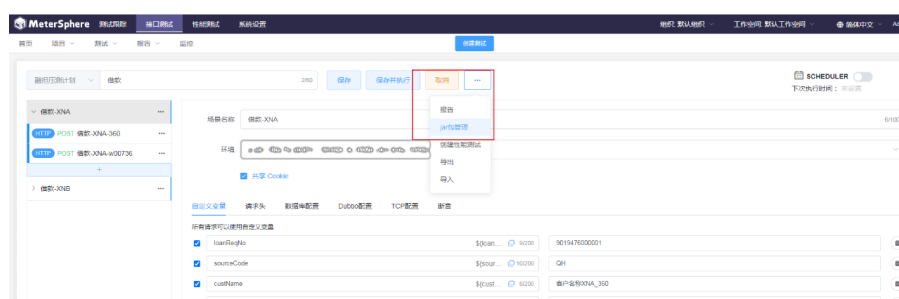
第二种：在接口测试模块中，点击“创建性能测试”，系统自动生成JMX文件

【CSV参数文件】上传CSV参数文件，文件上传不要轻易删除，以留后用

2. jar管理

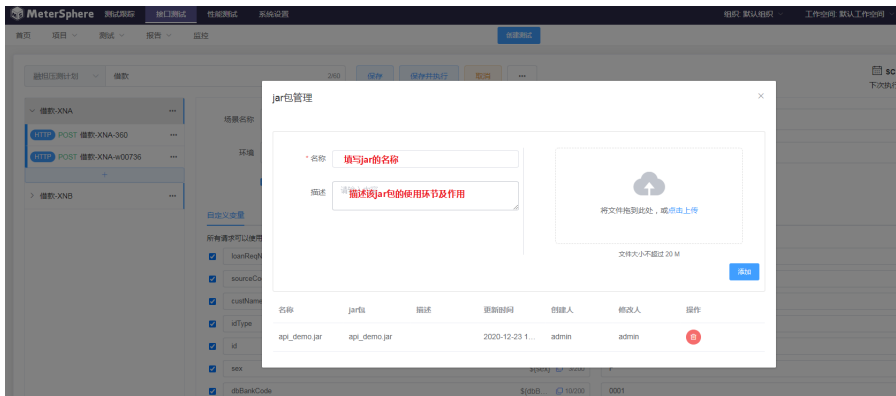
接口测试：

【进入jar包管理】



【jar包上传规范】

上传使用的jar包不要轻易删除，留以后使用



性能测试：

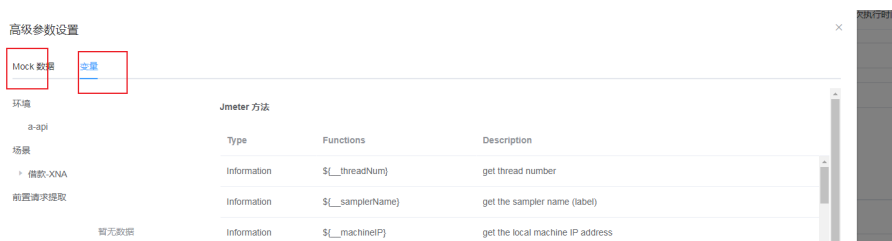
在性能测试的场景配置下，上传jar包

上传使用的jar包不要轻易删除，留以后使用



九、测试数据管理

9.1接口和性能测试在场景中自定义变量，对于有规律变化或者特殊规定的变量，均使用mock或者jmeter实现



Information	\${__machineName}	get the local machine name
Information	\${__time}	return current time in various formats
Information	\${__log}	log (or display) a message (and return the value)

值

保存

目前会用到：

【累加】

【身份证变化】

9.2性能测试中的CSV文件生成

暂通过idea自动生成csv文件，关于id和loanReqNo数据需要特殊管理；后期会使用Python脚本来维护测试数据

十、接口用例的规范

1. 接口名称命名

用例名称：融担平台+接口功能+普通接口，例：XNB-借款申请-自动化

<input type="checkbox"/>	名称	所属项目	创建人	创建时间	更新时间	状态
<input type="checkbox"/>	xnb-借款请求-自动化	自动化接口项目	Administrator	2021-01-19 17:26:43	2021-01-21 10:03:56	Saved
<input type="checkbox"/>	xnb-还款请求-自动化	自动化接口项目	Administrator	2021-01-19 17:28:28	2021-01-21 10:03:33	Saved

2. 测试场景的使用

2.1、显示一个标准接口作为单独场景

测试名称 dev-XNB-借款申请-普通接口 17/60 保存 保存并执行 取消 ...

标准接口场景

HTTP POST dev-XNB-借款申请-标准...

请求名称 dev-XNB-借款申请-标准数据

请求路径 POST /b2b-011-0-nw-acl-0-03-0dd

请求地址 dev-xnb-011-0-nw-acl-0-03-0dd

引用环境 跟随重定向 对 POST 使用 multipart/form-data



2.2、以参数字段维度来划分，每一个参数字段有不同的测试点，包含正常的、异常的、边界值等等场景



2.3、增加业务逻辑场景

若接口需要判断多条件的业务逻辑时，将对应的测试接口放入此场景里



2.4、增加幂等测试场景





3、接口测试-请求头

请求头免密设置: isTest=true

<input checked="" type="checkbox"/>	Content-Type	12/200	application/json
<input checked="" type="checkbox"/>	isTest	6/200	true

disconf设置免密

4、接口测试-请求体

设置免密后的请求体脚本

```
{  
  "assetOrgNo": "w00736_acc",  
  "assetOrgName": "w00736_机构名称",  
  "fbOrgNo": "xnb",  
  "loanReqNo": "${loanReqNo}",  
  "sourceCode": "QH",  
  "custApplyScore": "123456",  
  "custActScore": "123456",  
  "agreementNo": "7419476000001"  
}
```

5、接口测试-断言

添加其脚本断言, 根据返回结果校验接口测试是否执行正确

JSONPath		
\$.code	6/200	200
\$.data.rpyStatus	16/200	00

十一、测试资源池

性能测试接口为了成功调用csv文件、jar包内容，现需要新增一个资源池，使用新的镜像文件
什么是 Dockerfile?

Dockerfile是一个用来构建镜像的文本文件，文本内容包含了一条条构建镜像所需的指令说明
ps注：

显示所有docker镜像文件：docker images

删除docker镜像文件：docker rmi jmeter2:v6

1. 使用Dockerfile定制一个新镜像

jar包存放地址和Dockerfile 文件同目录

```
-rw-r--r-- 1 root root 4671521 1月 19 20:45 creat_date.jar  
-rw-r--r-- 1 root root 123 1月 19 20:46 Dockerfile
```

在一个空目录下，新建一个名为 Dockerfile 文件，并在文件内添加以下内容：

新建文件：vi Dockerfile

FROM registry.cn-qingdao.aliyuncs.com/metersphere/jmeter-master:5.3-ms14

ADD creat_date.jar /opt/jmeter/lib/creat_date.jar

释：拷贝一份原始的镜像文件，同时将creat_date.jar放入新的镜像文件中

2. 构建镜像文件

在 Dockerfile 文件的存放目录下，通过 Dockerfile 构建一个 jmeter:v1 （镜像名称:镜像标签）

docker build -t jmeter:v1 .

3. 新建资源池

* 名称

描述

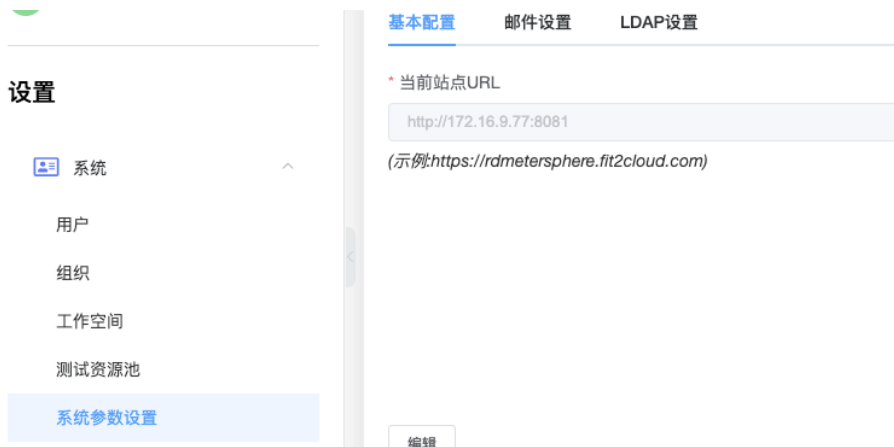
镜像

* 类型

* IP * Port * 最大并发数

十二、接口测试报告

1、设置-系统参数设置-基本配置



2、设置-组织-消息设置



设置完成后，系统会根据接口测试的配置，每触发一条就会在钉钉上发出一条消息，目前Jenkins不支持meterSphere的测试报告发送

钉钉上设置的关键词：任务通知

* 安全设置 ?

自定义关键词

[说明文档](#)

任务通知

十三、测试结束

导出测试用例和测试报告与需求测试报告一起发送

测试报告定时删除

数据库定时备份