

接口自动化测试执行规范

质量线

文档修订记录

（状态：C―创建，A—增加，M—修改，D—删除，R—发布）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **版本** | **状态** | **简要说明（内容和范围）** | **更改人** | **更改日期** | **审批人** | **审批日期** |
| V1.0 | C | 建立接口规范 | 张文俊 | 4月13日 | 樊兆妮 | 4月14日 |
| V1.1 |  | 修改规范 | 张文俊 |  |  |  |

目录

[1 概述 4](#_Toc28010)

[1.1 文档目的 4](#_Toc2948)

[1.2 文档适用人群 4](#_Toc2934)

[2 执行规范 4](#_Toc15892)

[2.1 准备及配置管理 4](#_Toc2892)

[2.1.1 脚本归档管理 4](#_Toc16156)

[2.1.2 Jenkins项目配置管理 5](#_Toc12597)

[2.2 脚本设计 6](#_Toc21115)

[2.2.1 脚本设计原理 6](#_Toc26470)

[2.2.2 脚本设计评审准则 6](#_Toc24607)

[2.2.3 脚本执行与维护 6](#_Toc10872)

[2.3 附录 7](#_Toc23830)

# 概述

## 文档目的

主要是统一规范接口自动化执行、编写与维护，减少迭代项目测试过程中测试人员重复劳动工作，以自动化的方式验证流程的可行性而编写该文档。

## 文档适用人群

本规范适用于测试人员。

# 执行规范

## 准备及配置管理

登录说明：

地址：[http://172.16.0.8:8080/](http://172.16.0.7:8080/)

用户名：个人工号

密码：123456（首次登录必须修改密码）

### 脚本归档管理

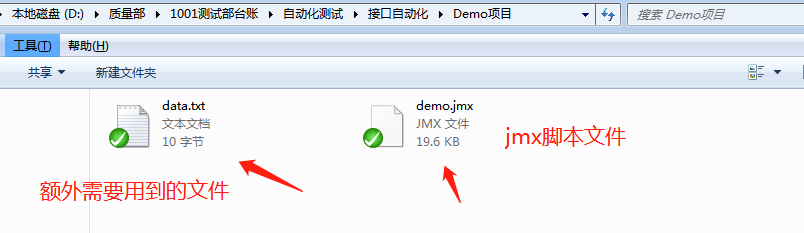
1. 脚本存放目录：

../接口自动化/系统名称/功能【模块】/脚本相关文件

1. 脚本相关文件：

存放脚本相关文件，如用户参数化文件、外部jar包、其他说明等

示例：



PS：如果需要使用当前脚本执行绝对路径，请在脚本中加上${\_\_P(Jworkspace,)}

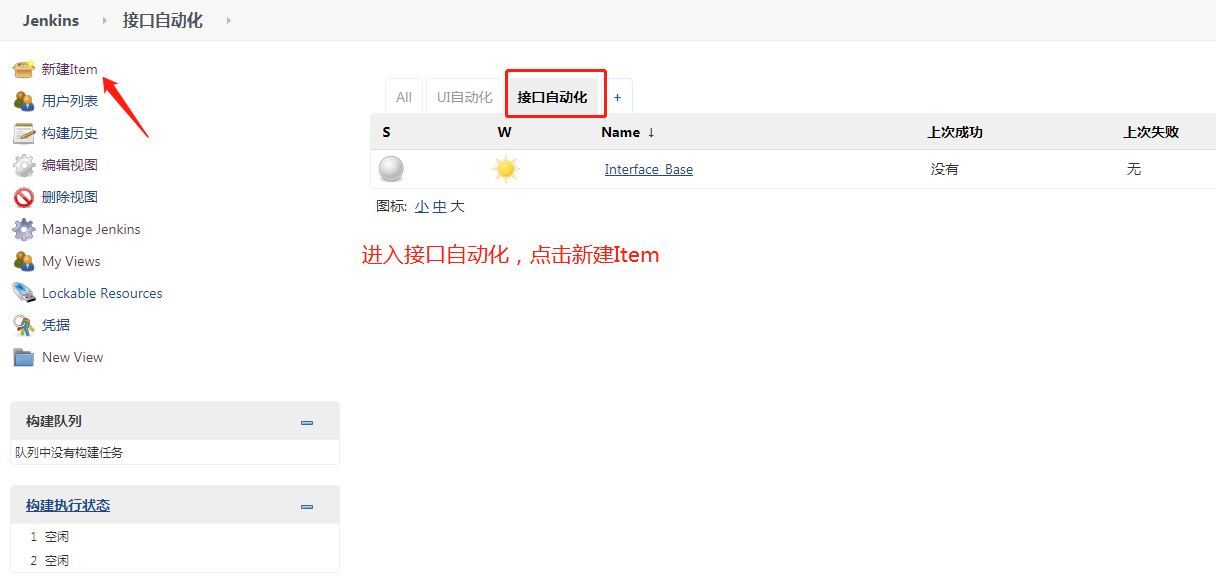
如data.txt绝对路径：${\_\_P(Jworkspace,)}\data.txt

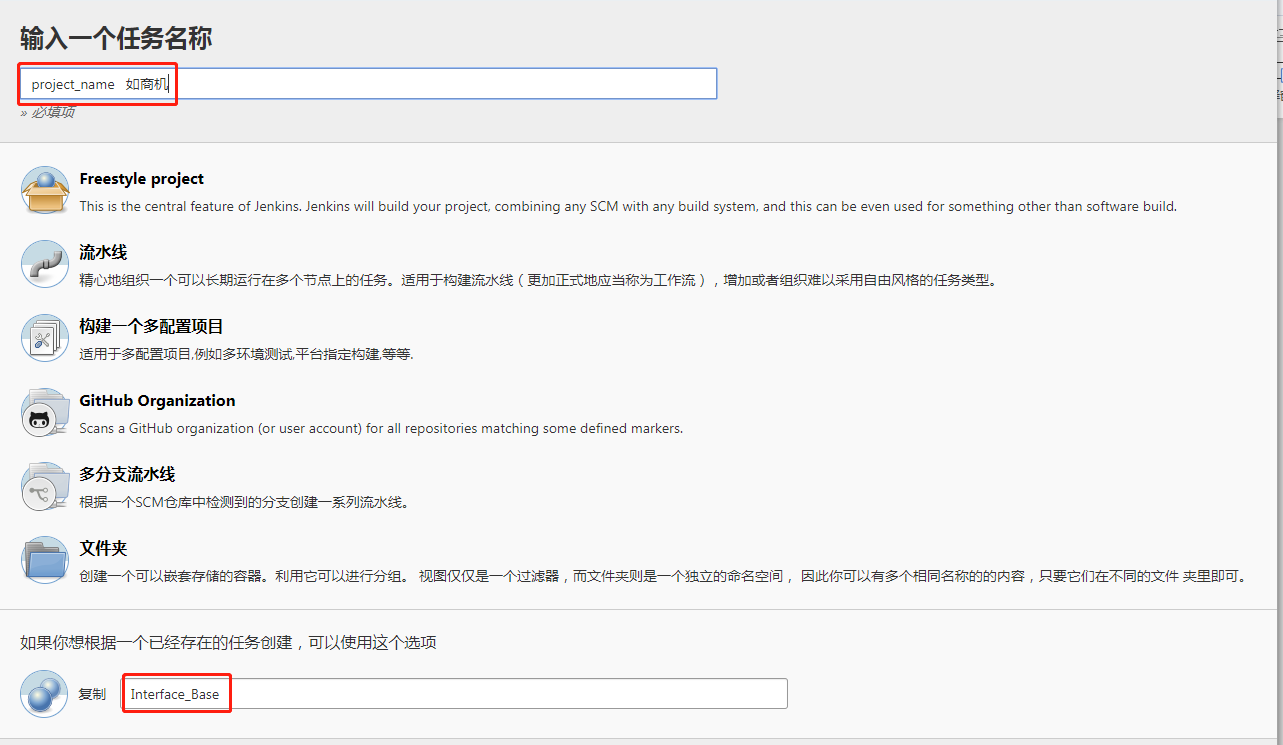
### Jenkins项目配置管理

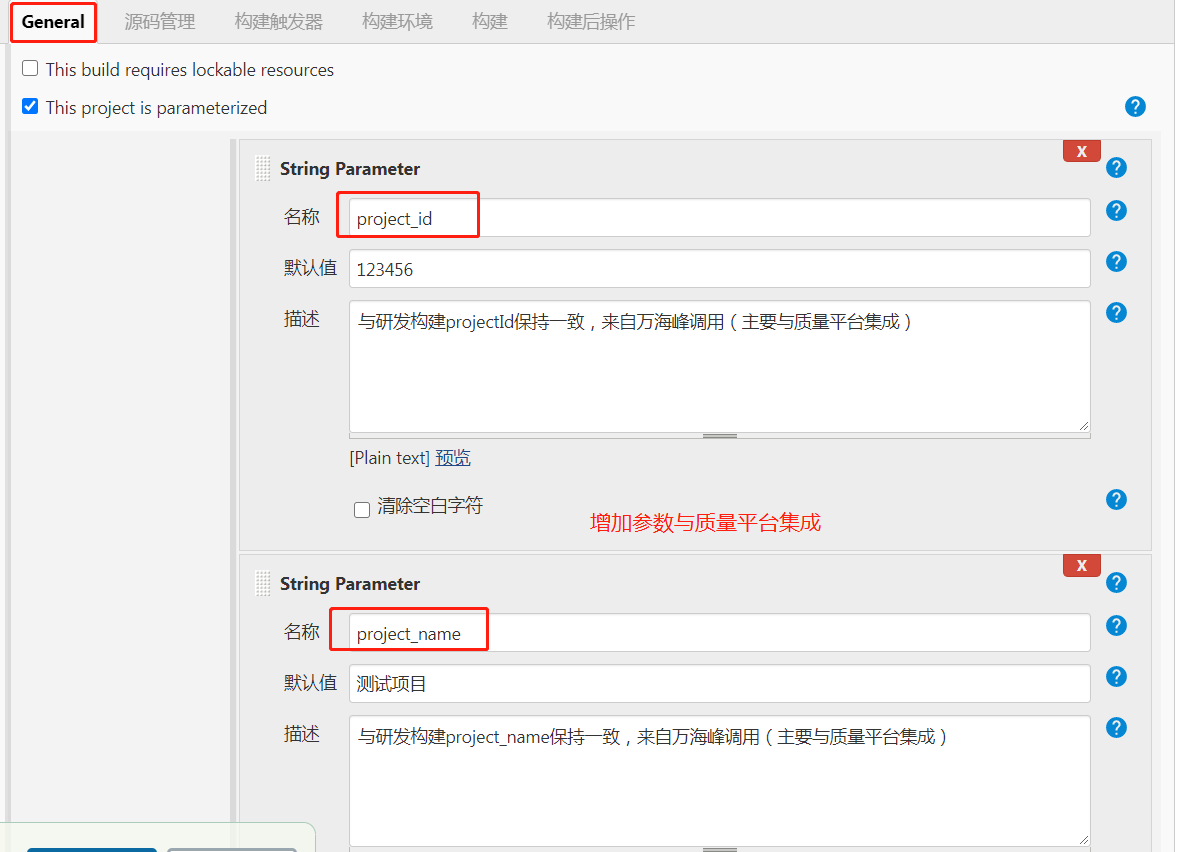
1. **项目配置**

新建项目：新建Item【接口自动化列表下】 >输入名称【项目名称】 >复制【Interface\_Base】

**注意：项目名称命名：Interface\_项目名称 如: Interface\_商机、Interface\_小顶云**



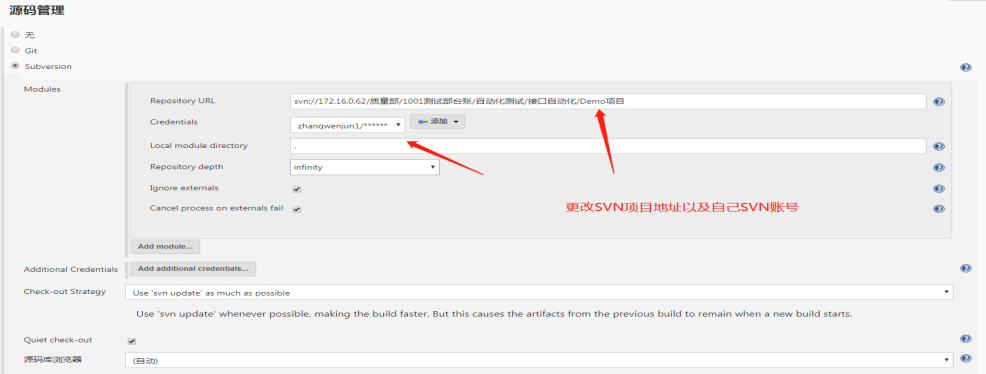






1. **源码管理**

**必须修改为自己的SVN账号**



1. **构建**

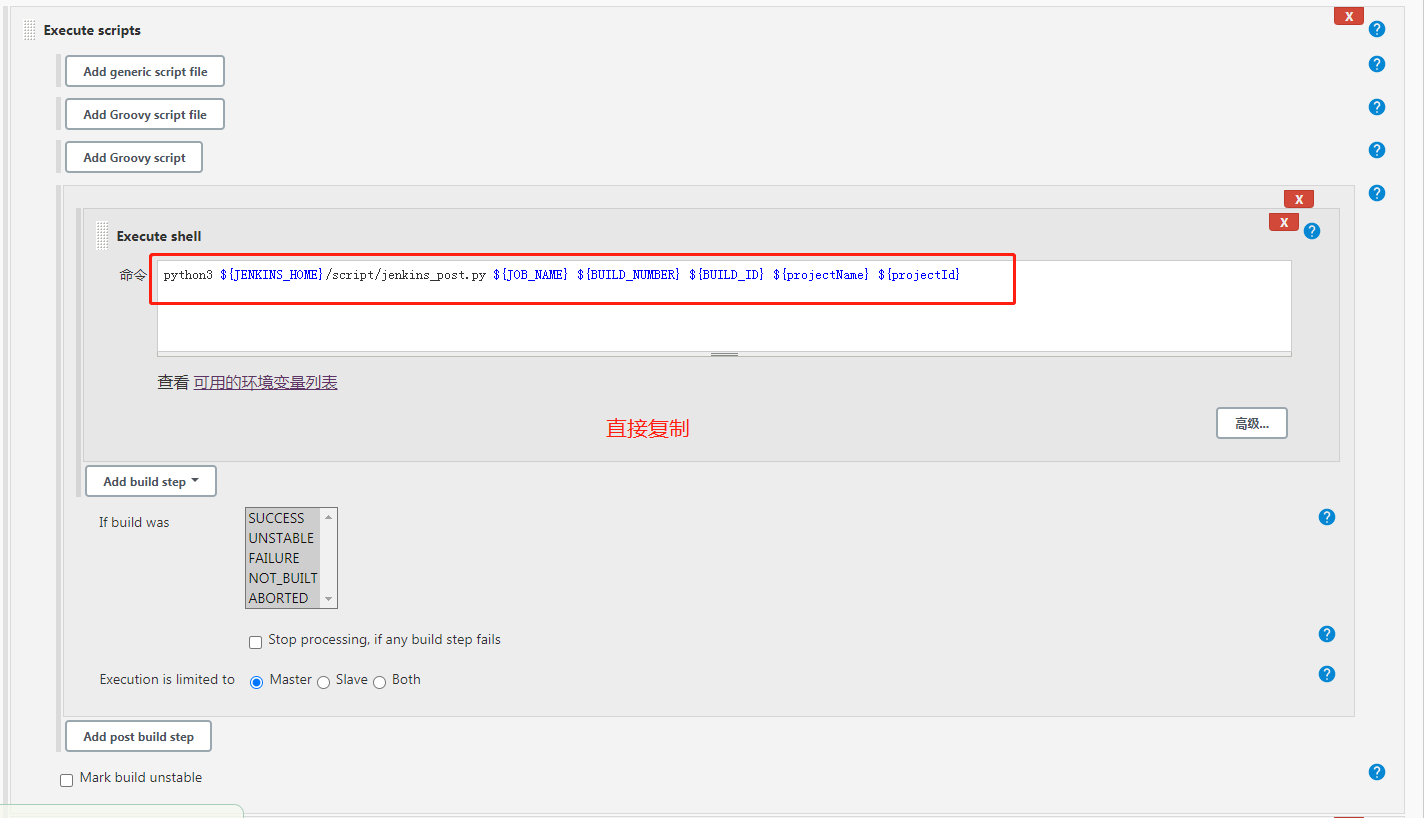
接收人用空格隔开。

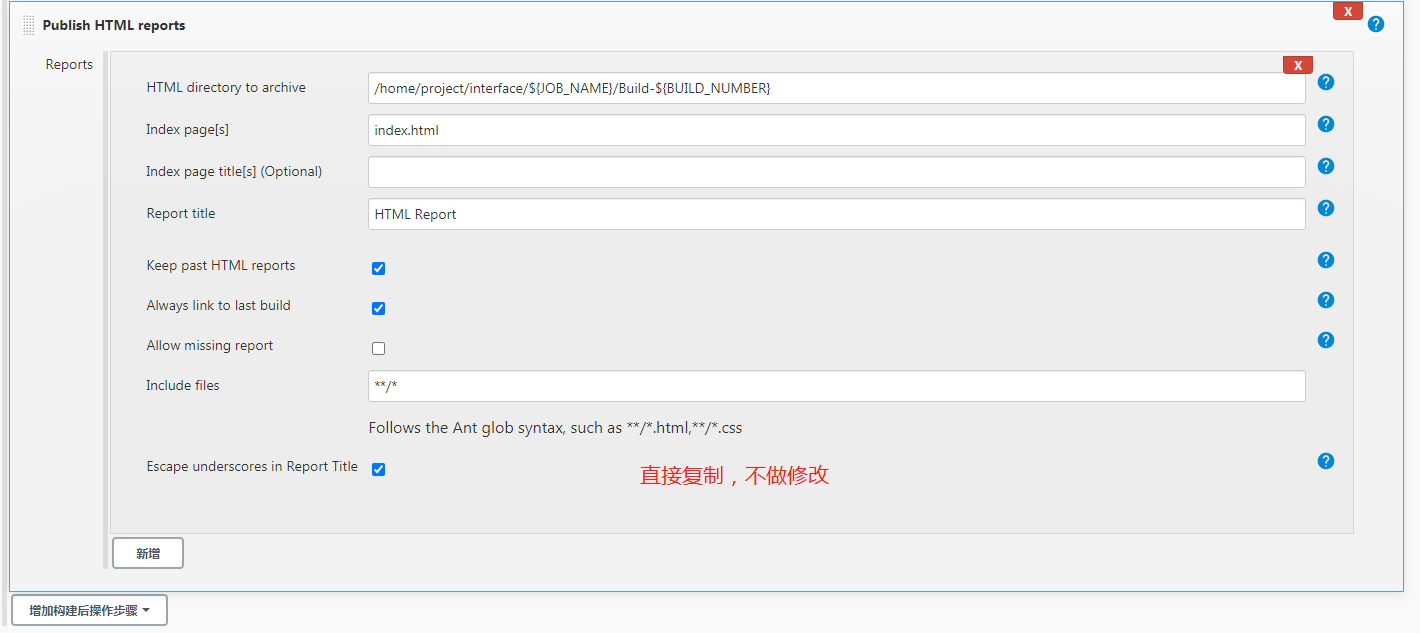
python3 ${JENKINS\_HOME}/script/jenkins\_jmeter\_linux.py ${JENKINS\_URL} ${JOB\_NAME} ${BUILD\_NUMBER} ${WORKSPACE} ${BUILD\_ID} ${project\_name} ${project\_id} demo zhangwenjun1@dgg.net 984783751



1. **构建后操作（5个构建状态都得选中）**

python3 ${JENKINS\_HOME}/script/jenkins\_post.py ${JOB\_NAME} ${BUILD\_NUMBER} ${BUILD\_ID} ${project\_name} ${project\_id}





## 脚本设计

### 脚本设计原理

1. **测试点设计**

每个项目脚本设计之前需提前完成测试点的设计，建议使用xmind，需测试点设计评审通过之后方可开展脚本设计工作。

1. **脚本设计**

每个项目包含两种脚本：主流程脚本、全覆盖脚本

**主流程脚本：**项目主流程，根据项目迭代影响程度进行维护。

**全覆盖脚本：**项目功能点覆盖80%以上，需要实时维护。

1. **检查点设计**

数据去向全覆盖检查、平时测试过程中容易出细节问题的地方、容易忽略的地方等，尽可能全面的进行断言。

### 脚本设计评审准则

评审对象：测试点、主流程脚本

参与人：质量总监、测试经理、专业督导、相关测试人员。

通过准则：参与人全票通过。

### 脚本执行与维护

1. **新项目**

新项目在测试报告输出后两周内完成主流程脚本，一个月内完成全覆盖脚本的维护。

1. **迭代项目**

迭代过程中主要维护全覆盖脚本，执行流程如下：



## 附录

1、为什么需要jenkins，直接本地执行不是更方便？

答：jenkins可以保留历史执行记录，这样可以用于溯源，且配置也不用花费多少时间

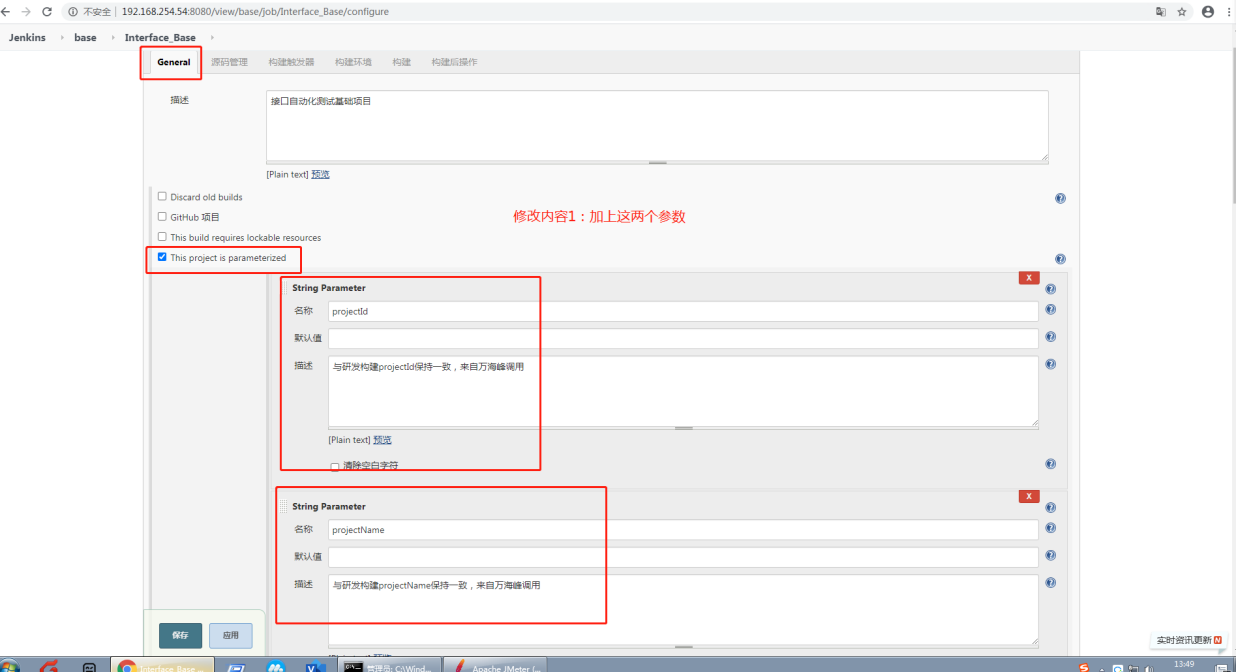
2、接口自动化真能减轻我们的工作么？

答：迭代项目是能减轻一定工作的，如果项目复杂度高，迭代频繁，且迭代内容又很细小，这样更能减轻大量工作

# 迭代更新（V1.2）

修改一：

常规里面增加2个外部字符串参数：projectId、projectName



修改二：

构建传参，增加参数：%BUILD\_ID% %projectName% %projectId%



# 迭代更新（V1.3）

说明：由于windows下jenkins服务器经常出现异常，故本次更新后，jenkins部署在linux服务器上，需要对项目进行迁移。

特殊说明：如果使用了Jworkspace内置属性，则调用服务器文件使用/作为路径分隔符，windows下使用\作为文件分隔符（本服务器是linux）

所有测试项目需要与万海峰对接，当研发构建项目后，自动触发测试执行（如crm项目，研发这边可能存在多个job，而测试这边就只有一个job:interface\_crm。）

1. 登录地址及用户名迁移：

登录说明：

地址：[http://172.16.0.8:8080/](http://172.16.0.7:8080/)

用户名：个人工号

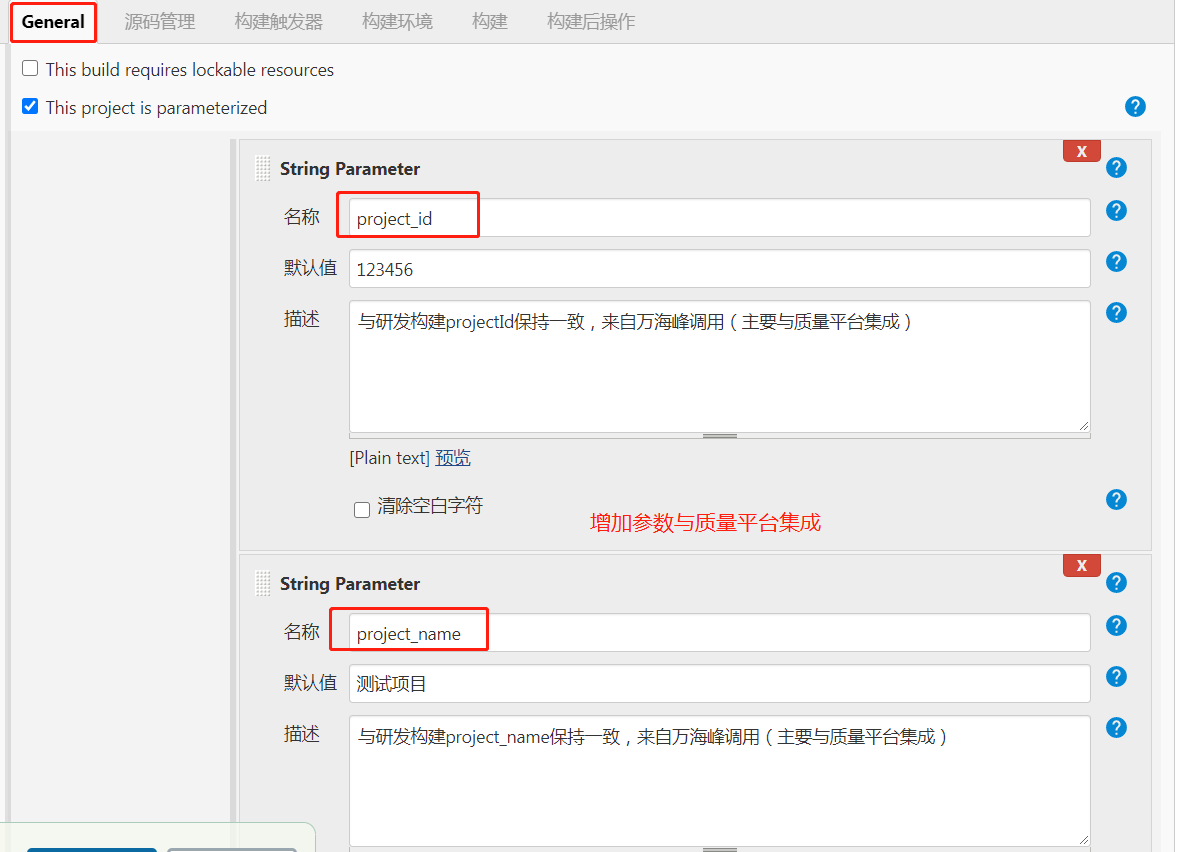
密码：123456（首次登录必须修改密码）

1. 个人项目可以参考老地址，copy过来即可，第一次创建项目请从头到尾阅读

旧地址：<http://192.168.254.54:8080/>

1. 额外修改内容如下（可以参考base\_interface）

* General修改





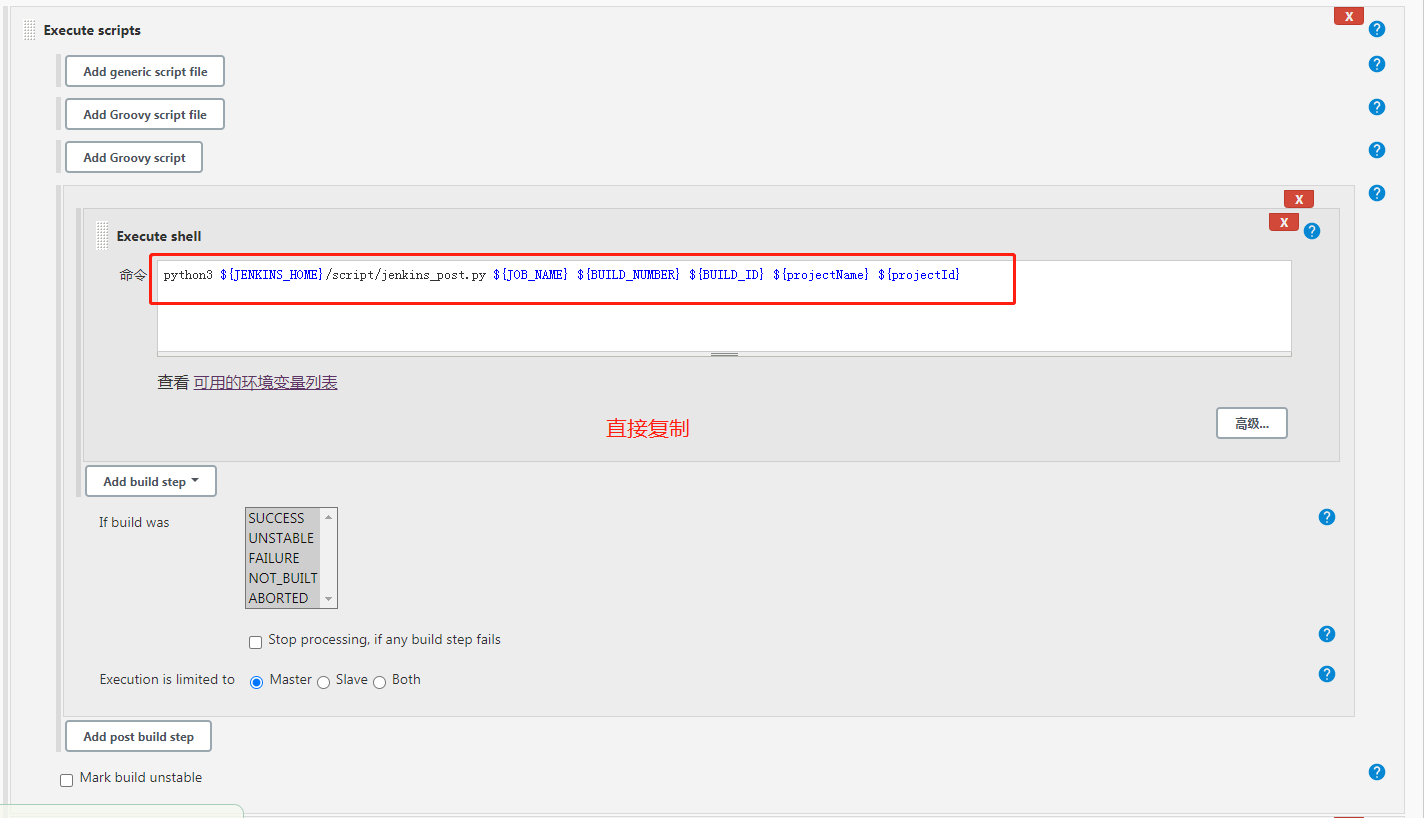
* 构建修改

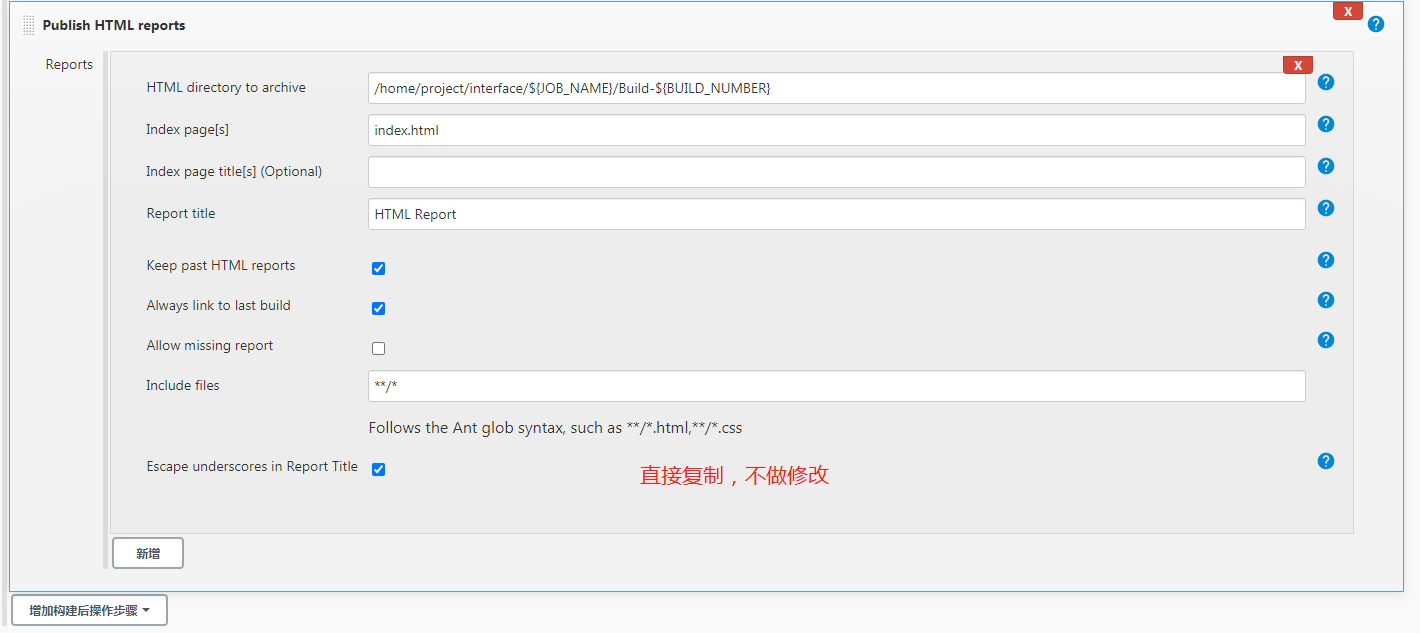
python3 ${JENKINS\_HOME}/script/jenkins\_jmeter\_linux.py ${JENKINS\_URL} ${JOB\_NAME} ${BUILD\_NUMBER} ${WORKSPACE} ${BUILD\_ID} ${project\_name} ${project\_id} demo zhangwenjun1@dgg.net 984783751



* 构建后操作修改

python3 ${JENKINS\_HOME}/script/jenkins\_post.py ${JOB\_NAME} ${BUILD\_NUMBER} ${BUILD\_ID} ${project\_name} ${project\_id}





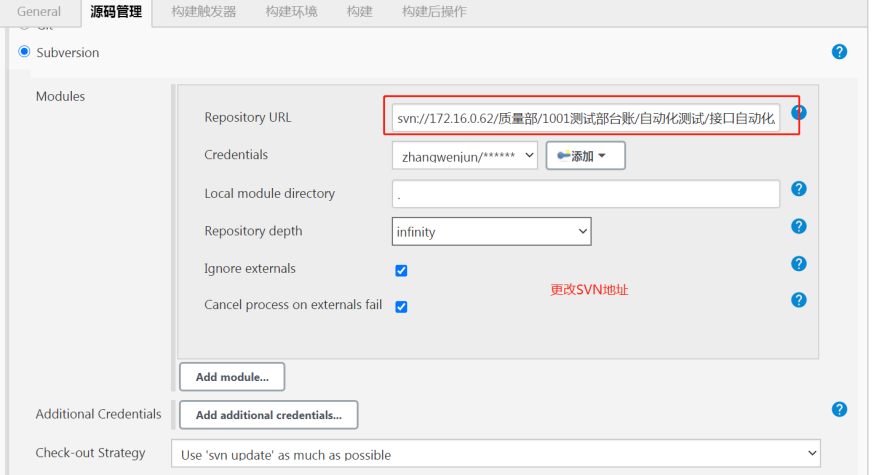
# 快速搭建CI持续集成（V1.4）

登录说明：

地址：[http://172.16.0.8:8080/](http://172.16.0.7:8080/)

用户名：个人工号

密码：123456（首次登录必须修改密码）

1. 复制base\_interface项目
2. 
3. 
4. 进入质量平台【http://tqp.dgg188.cn/ci\_count】，添加项目关联

说明：部署jenkins与测试jenkins需要关联（具体细节可以进入质量平台查看帮助文档）

如：CRM项目（项目名称）

开发job：[crm-oqm35006【CRM订单】](http://192.168.254.162:8080/view/CRM/job/crm-oqm35006%E3%80%90CRM%E8%AE%A2%E5%8D%95%E3%80%91/)、[crm-bus11175【CRM商机】](http://192.168.254.162:8080/view/CRM/job/crm-bus11175%E3%80%90CRM%E5%95%86%E6%9C%BA%E3%80%91/)……

测试job：Interface\_CRM