**背景:**

   Jenkins具有丰富的插件生态，足以满足我们日常工作的需求，但如果我们想通过具体的Jenkins任务直接对外提供服务，而不想将内部的具体实现对外暴露(否则，需添加对应的用户权限，通过页面执行job)；可以对外直接提供接口，第三方直接调用接口（比如提供给开发，提测前回归冒烟用例集），执行相应的Jenkins任务并获取任务结果

   Jenkins本身支持丰富的API接口，我们通过调用接口，基本可以实现所有需要的功能，包括获取JOB，触发JOB执行，甚至删除JOB等

   Jenkins的Remote API以REST-like的形式进行提供，通过对特定的API执行POST请求即可

**站点说明:**

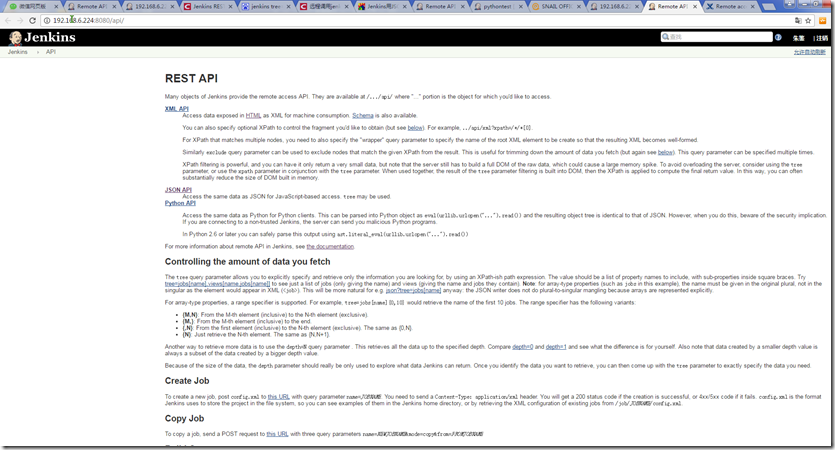
如果搭建的Jenkins站点为 [http://192.168.6.224:8080](http://192.168.6.224:8080/)

其中创建了一个job，job name为"pythontest"，还配置了一个字符参数test，job只是执行一个简单的python script

之后所有模拟的请求，都通过postman工具进行

**REST API:**

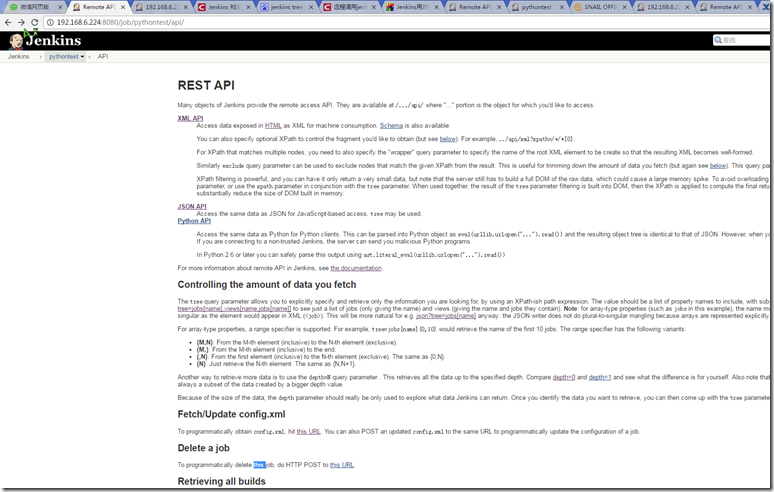
站点所有支持的API都可以通过地址http://192.168.6.224:8080/api 获取，如图

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105504169-1077032470.png)

       某一个job的所有相关API，可以通过地址 [http://192.168.6.224:8080/job/{jobname}/api/来获取](http://192.168.6.224:8080/job/%7Bjobname%7D/api/%E6%9D%A5%E8%8E%B7%E5%8F%96)

如我们的job name是pythontest，则地址是 <http://192.168.6.224:8080/job/pythontest/api/>

所有支持的API如图

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105508263-255215953.png)

**JOB API---获取JOB相关信息:**

    我们通过url直接获取job的相关信息，生成xml格式或者JSON，甚至可以用python、ruby脚本获取

    如获取job(pythontest)的信息，要获取json的地址

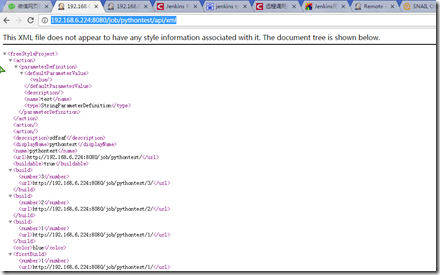
<http://192.168.6.224:8080/job/pythontest/api/json?pretty=true>

可以获取name，build等的所有信息，后面说明都使用json格式

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105510466-1763850036.png)

  XML的地址

http://192.168.6.224:8080/job/pythontest/api/xml

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105512357-1952719846.png)

    python <http://192.168.6.224:8080/job/pythontest/api/python?pretty=true>

**JOB API---获取Build相关信息:**

  刚才我们获取的是job所有的相关信息，可能我们还需要获取某一次build的信息

   则可以通过地址 [http://192.168.6.224:8080/job/pythontest/{build\_number}/api/json?pretty=true](http://192.168.6.224:8080/job/pythontest/%7Bbuild_number%7D/api/json?pretty=true)

   如果我们要获得这个pythontest第一次build的信息

<http://192.168.6.224:8080/job/pythontest/1/api/json?pretty=true>

**JOB API--信息过滤:**

之前获取的所有相关信息，可以通过添加参数tree=XXX[]，来过滤出你所感兴趣的节点，类似于正则表达式

详细的可以参考，之前站点API 说明中的"Controlling the amount of data you fetch"这一节

如何用，假设通过 [http://192.168.6.224:8080/job/pythontest/api/json?pretty=true](http://192.168.6.224:8080/job/pythontest/api/json?pretty=true%E8%8E%B7%E5%8F%96%E7%9A%84%E6%98%AF%E6%89%80%E6%9C%89%E7%9A%84%E4%BF%A1%E6%81%AF) 会获取所有的信息

那么添加   tree=builds[\*]  可以获取所有builds下的节点

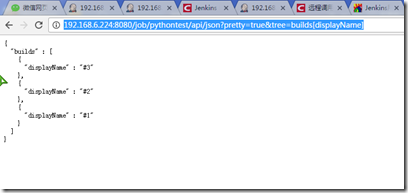
地址拼接为 ，直接http get请求

  http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds[\*]

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105514748-989450297.png)

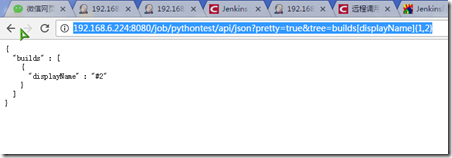
通过地址 http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds[displayName]

   可以获取builds下，所有displayName的节点，其中一共有三个displayName

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105516576-2099354735.png)

如果要获取三个displayName节点中第二个节点，可以通过{X,Y}

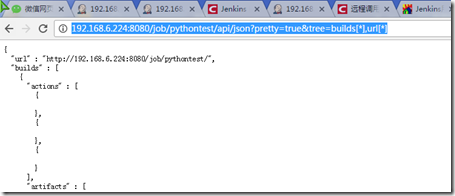
[http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds[displayName]{1,2}](http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds%5bdisplayName%5d%7B1,2%7D)

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105517810-1209376452.png)

如果要获取两个相关的节点，通过,连接两个过滤节点

[http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds[\*],url[\*]](http://192.168.6.224:8080/job/pythontest/api/json?pretty=true&tree=builds%5b*%5d,url%5b*%5d)

  这样就可以获取build和url的信息

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105519685-2120203803.png)

**JOB API--执行Build:**

  1 直接执行，不包括参数

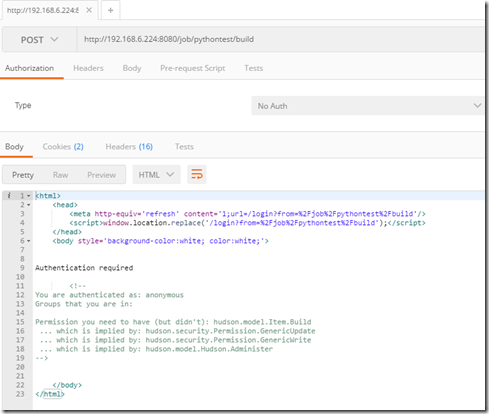
    此时暂时删除pytest的参数,使它可以直接执行

    那么执行build的API地址就是

<http://192.168.6.224:8080/job/pythontest/build>

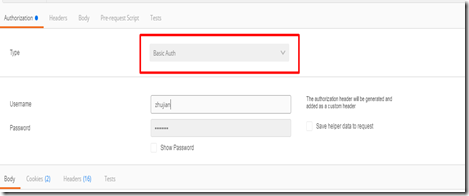
请求方法为POST，通过post man发送这个请求

会发现请求失败，返回"Authentication required"

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105521623-2034256647.png)

此提示信息说明执行job的时候需要身份认证，可以通过postman的"Authorization"，选择“Basic Auth”， 再输入登录jeknins，可以有权限

执行job的用户名和密码

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105523169-1939497361.png)

此时再次提交,Job执行成功

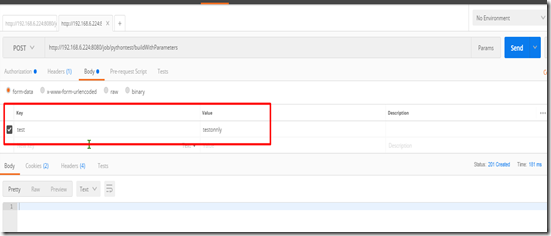
2 包括参数执行

    仍然是用pytest这个job，只是添加了参数test(一个字符串的参数)

    带参数执行的API地址，请求仍然是POST

<http://192.168.6.224:8080/job/pythontest/buildWithParameters>

仍然需要认证，参数可以设置为test="testonly"

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105524935-2029459178.png)

此时提交job执行成功

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105526498-961004504.png)

**JOB API--获取和更新描述信息:**

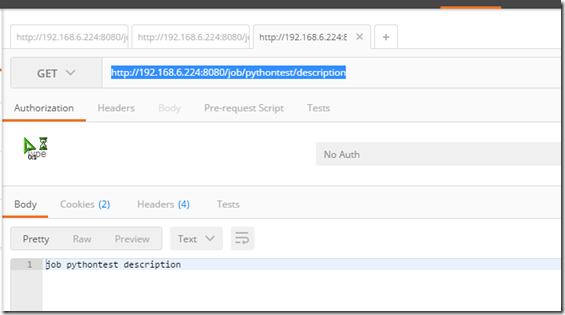
  通过API可以获取和更新JOB的description的信息

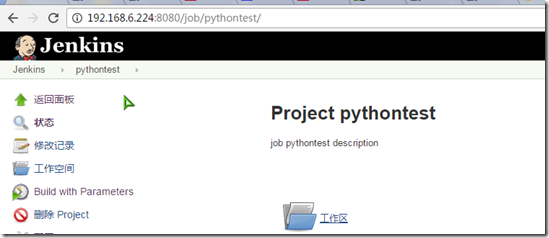
  1.获取Description信息

    API地址:[http://192.168.6.224:8080/job/pythontest/description](http://192.168.6.224:8080/job/pythontest/description(pythontest%E6%98%AFjob)

   （pythontest是job name)

    请求方法"GET"，不需要认证

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105528357-1061493210.png)

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105530169-1768747560.png)

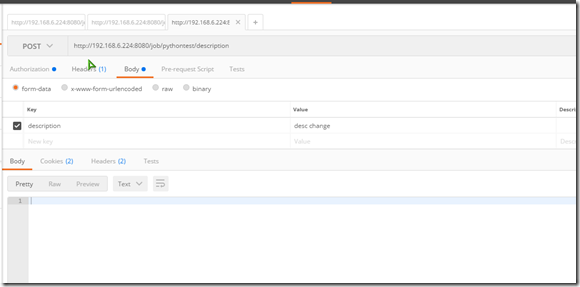
2 更新Description信息

    更新description的时候，API地址还是

<http://192.168.6.224:8080/job/pythontest/description>(pythontest是jobname)

    只是请求方法为POST,添加一个参数 description，填写要更新的description的内容

    同样需要身份验证

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105532076-170714659.png)

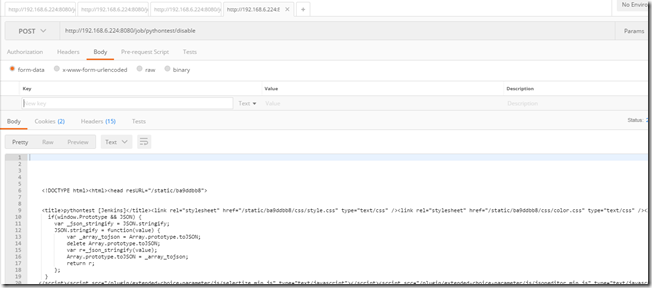
**JOB API--禁用和启动Job:**

  1 禁用job

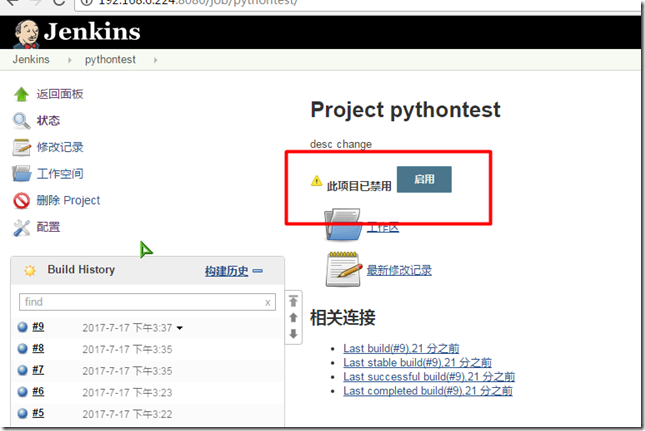
    要禁用某一个job API地址 [http://192.168.6.224:8080/job/pythontest/disable](http://192.168.6.224:8080/job/pythontest/disable(pythontest%E6%98%AF%7Bjobname%7D))

[(pythontest是job name)](http://192.168.6.224:8080/job/pythontest/disable(pythontest%E6%98%AF%7Bjobname%7D))

请求方法是POST

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105533669-1454653843.png)

执行成功，在jenkins中查看job ,此时已禁用

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105535076-1904401842.png)

2 启用Job

    之前的job,pythntest已经禁用，要重新启用这个job

    API地址 [http://192.168.6.224:8080/job/pythontest/enable（pythontest是job name)](http://192.168.6.224:8080/job/pythontest/enable%EF%BC%88pythontest%E6%98%AF%7Bjobname%7D))

请求方法是POST，执行后，job变成正常状态

**JOB API--删除JOB:**

要删除一个job，仍然可以通过API

  地址  <http://192.168.6.224:8080/job/pythontest/doDelete> (pythntest是job name)

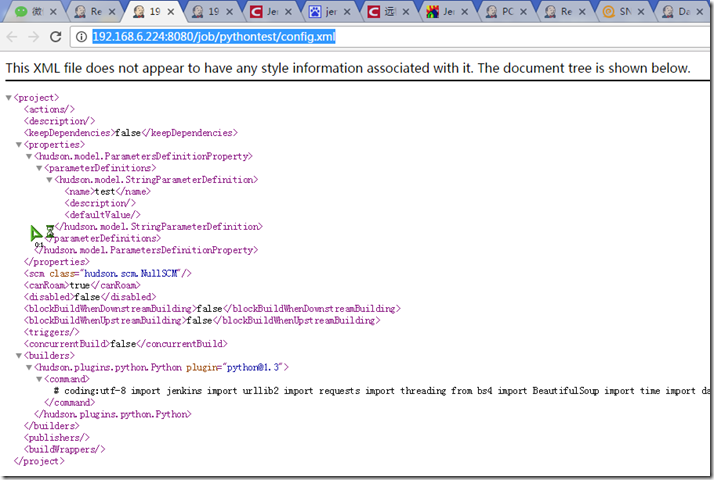
  请求方法是post，需要认证

**JOB API--JOB conofig.xml:**

  JOB的配置信息，通过地址 <http://192.168.6.224:8080/job/pythontest/config.xml(> pythntest是job name)

  获取到的是一个xml的格式，这个里面包括了详细的配置信息，同样的地址通过post可以更新配置

主要是后面用这个config.xml可以用来创建job

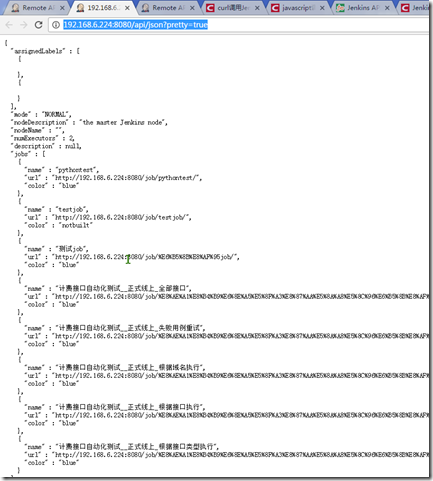
[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105536951-1781418699.png)

**站点 API:**

  站点所有相关的API，都是可以从地址[http://192.168.6.224:8080/api](http://192.168.6.224:8080/api/%E8%8E%B7%E5%8F%96%EF%BC%8C%E5%90%8C%E6%A0%B7%E5%8F%AF%E4%BB%A5%E6%9F%A5%E8%AF%A2%E5%88%B0%E7%9B%B8%E5%85%B3%E7%9A%84%E4%BF%A1%E6%81%AF%EF%BC%8C) 中查询到，同样可以查询job的相关信息等

但不同的是，还可以创建Job,拷贝Job，停止或重启Jenkis的服务

通过 [http://192.168.6.224:8080/api/json?pretty=true （JSON) 可以查询到站点中所有的job信息](http://192.168.6.224:8080/api/json?pretty=true%EF%BC%88JSON)%EF%BC%8C%E5%8F%AF%E4%BB%A5%E6%9F%A5%E8%AF%A2%E5%88%B0%E7%AB%99%E7%82%B9%E4%B8%AD%E6%89%80%E6%9C%89%E7%9A%84job%E4%BF%A1%E6%81%AF)

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105539248-1175083169.png)

同样的XML地址是 <http://192.168.6.224:8080/api/xml>

查询同样支持，通过tree进行过滤

如通过 [http://192.168.6.224:8080/api/json?pretty=true&tree=jobs[name[\*]]](http://192.168.6.224:8080/api/json?pretty=true&tree=jobs%5bname%5b*%5d%5d)

可以过滤出所有jobs的name

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105541326-1335719395.png)

**站点API\_创建Job**

要创建一个Job，首先要明白每个Job的配置信息都保存在config.xml中，如之前我们已经创建的job pythontest，通过访问地址

[http://192.168.6.224:8080/job/pythontest/config.xml](http://192.168.6.224:8080/job/pythontest/config.xml%EF%BC%8C%E5%B0%B1%E5%8F%AF%E4%BB%A5%E8%8E%B7%E5%8F%96%E8%AF%A6%E7%BB%86%E7%9A%84config.xml)

就可以获得job相关的详细配置信息，当然需要先登录

config.xml的配置信息如下,这个里面包括了job的配置，包括名称，命令等

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105542404-1023859552.png)

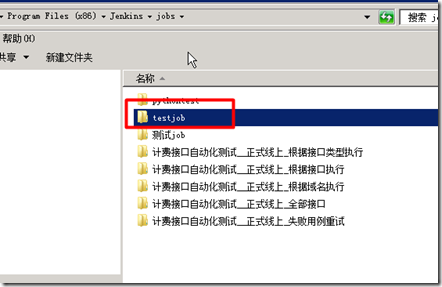
要创建一个新的Job，可以参考或复制这个config.xml文件

创建Job的API地址是<http://192.168.6.224:8080/createItem>

假设我们要新创建一个job,叫 “testjob”

那么第一步，要在Jenkins的安装目录下，找到子目录\jobs，然后在其中创建一个目录 ，就叫testjob

如果不创建这个目录，直接调用API，是会失败的

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105543310-2090298147.png)

然后准备config.xml文件，作为job的配置信息，可以从刚才的pythontest的配置直接拷贝得到，然后简单修改下name

举例通过Postman进行，访问的API地址，请求方法为POST，需要身份验证

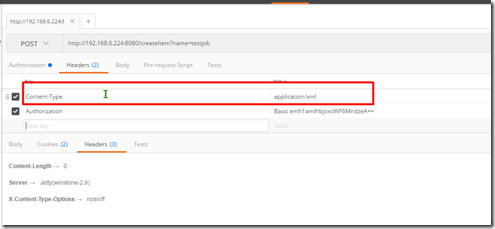
<http://192.168.6.224:8080/createItem?name=testjob>

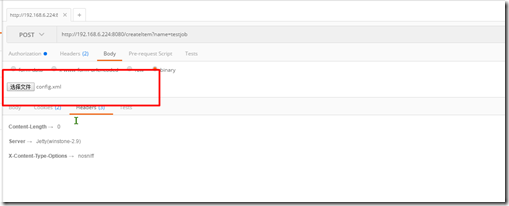
地址后面的name就是要创建的job的name

在Header中创建 Content-Type，值为application/xml

请求的Boy中类型选择为binary

然后选择刚才准备的config.xml文件

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105544294-589879294.png)

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105545607-896603245.png)

此时调用，这个接口，Job可以创建成功

**站点API\_拷贝Job**

要通过API拷贝一个已存在的Job,需要访问的API地址仍然是，请求还是POST

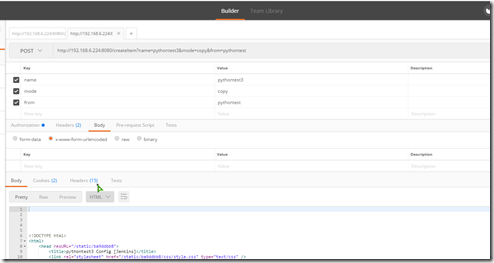
http://192.168.6.224:8080/createItem

  再需要添加三个参数

分别是 name (复制后新创建的的job name,如pythontest2

mode(模式，固定给予值copy)

from( 要复制的Job的Job name,如pythontest)

[](http://images2017.cnblogs.com/blog/1203853/201707/1203853-20170726105546732-2115637313.png)

**站点API\_重启**

通过API地址 <http://192.168.6.224:8080/restart>

可以重启Jenkins

通过API地址 http://192.168.6.224:8080/safeRestart

  可以安全重启Jenkins

**参考文档:**

1 <https://wiki.jenkins.io/display/JENKINS/Remote+access+API>

2 获取Jenkins project build结果 http://blog.csdn.net/ljj\_9/article/details/70270977