

通过jenkins api 触发远程构建

原文链接（针对用到的构建部分稍作了补充）：<https://segmentfault.com/a/1190000018939447>

Jenkins 是一款流行的开源持续集成工具，可以用来做一些软件开发的自动化工作，如打包，测试，自动部署等。

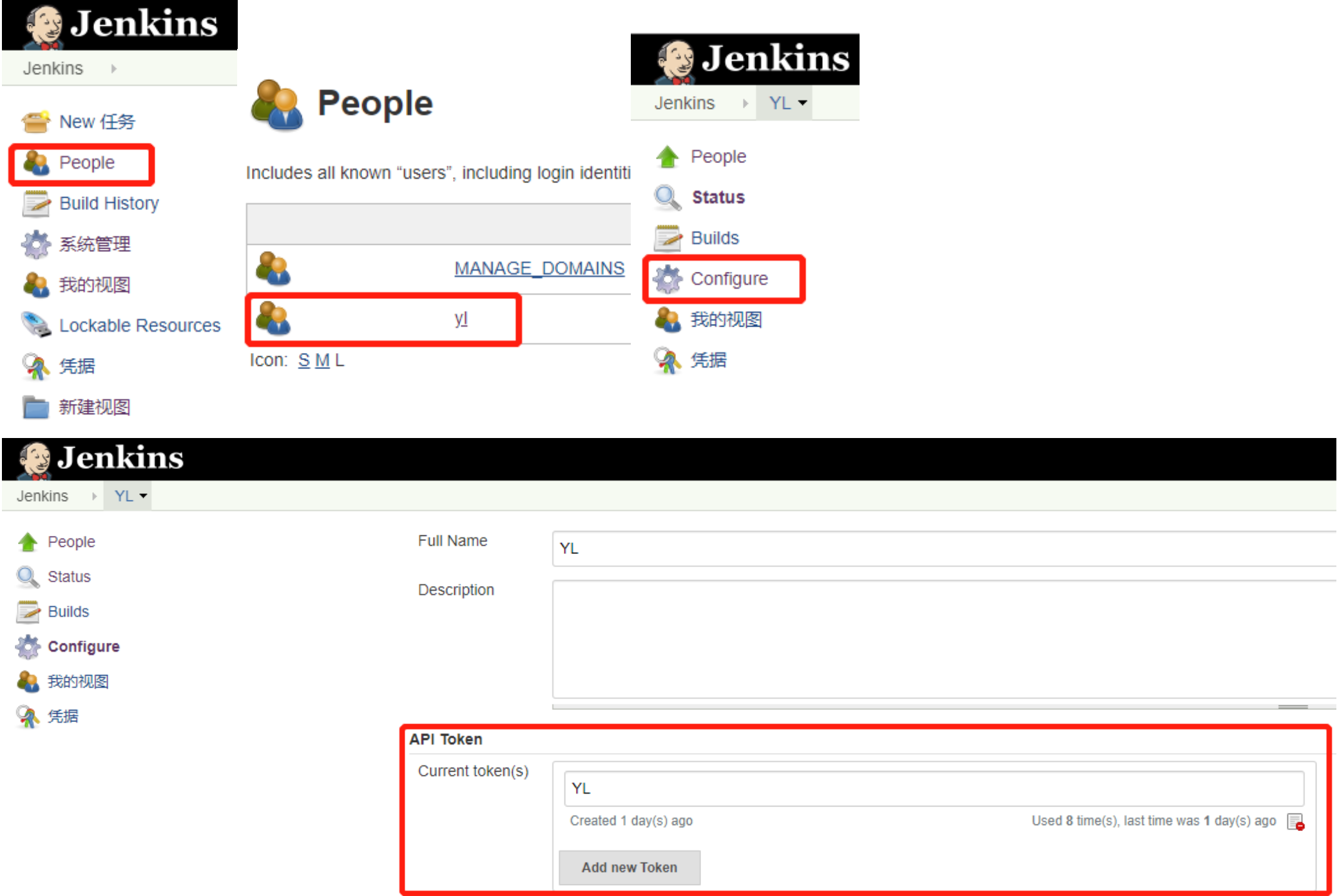
Jenkins 中有 view 和 job 的概念，view 相当于组，job 则是具体的任务。
view 下面可以创建 job，但 job 可以不在任何 view 下。

这里主要介绍 Jenkins 提供的 HTTP API，至于如何使用 Jenkins 请参看 [Jenkins User Documentation](#)。

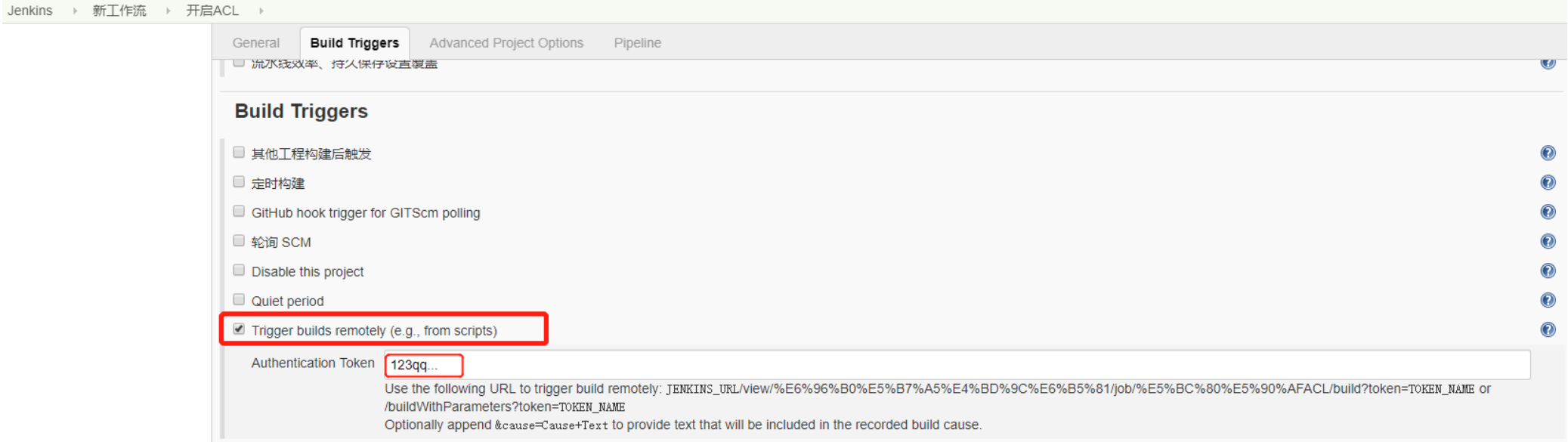
API

鉴权

Jenkins 使用 Basic Auth 的权限验证方式，需要传入 username 和 api token。
其中 api token 需要在用户的设置界面去创建。具体见下面截图引导：



但在 Job 的远程触发中，可以设置用于远程触发的 token (在 Job 的配置页面设置)，这样在触发 Job 时就不需要传入 Basic Auth 了。以下是在job中创建token的图示：



远程触发的 token 使用 urlencode 的方式放在请求的 body 中，其原始数据为：token=<Token Value>

下面给出两种方式触发 Job 的例子：

• Basic Auth

curl -X POST <jenkins url>/view/<view name>/job/<job name>/build --user <username>:<api token>


实例：

<view name>和<job name>的值是直接复制的浏览器地址栏中的url，不是手工输入的，手工输入真实的字符串运行不成功。

```
curl -X POST http://192.168.2.17:8080/jenkins/view/%E6%96%B0%E5%B7%A5%E4%BD%9C%E6%B5%81/job/%E5%BC%80%E5%90%AFACL/build --user YL:11f9f1616fa554b4bb3dfb3f515c07b490
```

• Token


curl -X POST <jenkins url>/view/<view name>/job/<job name>/build --data-urlencode token=<Token Value>




```
curl -X POST http://192.168.2.17:8080/jenkins/view/%E6%96%B0%E5%B7%A5%E4%BD%9C%E6%B5%81/job/%E5%BC%80%E5%90%AFACL/build --data-urlencode token=123qq...
```

这种方式去直接远程触发job会提示Authentication required，文章最后部分有不推荐的临时解决方法，以下是推荐的方法，再额外加上身份验证：

```
curl -X POST http://192.168.2.17:8080/jenkins/view/%E6%96%B0%E5%B7%A5%E4%BD%9C%E6%B5%81/job/%E5%BC%80%E5%90%AFACL/build --data-urlencode token=123qq... -u YL:11f9f1616fa554b4bb3dfb3f515c07b490
```



补充：



Jenkins的API支持如下两种方式进行认证：


用户名 + 用户密码（不安全）
以curl客户端访问为例，通过如下方式进行相关的用户认证

```
curl -u 用户名:用户密码 http://Jenkins服务的URL/具体操作的URI
```

用户名 + Token信息（安全）
以curl客户端访问为例，通过如下方式进行相关的用户认证

```
curl -u 用户名:用户Token http://Jenkins服务的URL/具体操作的URI
```

注：此处使用的Token为在Jenkins中可以查询到的（用户的配置页面可以查询到的API Token即为此处认证所使用的信息，而Token的修改也可以在此页面通过点击Change API Token按钮来进行）



View

• 创建

API: <jenkins url>/createView

curl: curl -X POST <jenkins url>/createView?name=<view name> --data-binary "@viewCofnig.xml" -H "Content-Type: text/xml" --auth <u>:<p>

• 删除

API: <jenkins url>/view/<view name>/doDelete

curl: curl -X POST <jenkins url>/view/<view name>/doDelete --auth <u>:<p>

• 查询状态

API: <jenkins url>/view/<view name>/api/json

curl: curl -X GET <jenkins url>/view/<view name>/api/json --auth <u>:<p>

• 查询配置

API: <jenkins url>/view/<view name>/config.xml

curl: curl -X GET <jenkins url>/view/<view name>/config.xml --auth <u>:<p>

• 更新配置

API: <jenkins url>/view/<view name>/config.xml

curl: curl -X POST <jenkins url>/view/<view name>/config.xml --data-binary "@newCofnig.xml" -H "Content-Type: text/xml" --auth <u>:<p>

Job

• 创建

API: <jenkins url>/createItem

curl: curl -X POST <jenkins url>/createItem?name=<job name> --data-binary "@jobConfig.xml" -H "Content-Type: text/xml" --auth <u>:<p>

- 删除
API: <jenkins url>/job/<job name>/doDelete
curl: curl -X POST <jenkins url>/job/<job name>/doDelete --auth <u>:<p>
- 查询状态
API: <jenkins url>/view/<view name>/job/<job name>/api/json
curl: curl -X GET <jenkins url>/view/<view name>/job/<job name>/api/json --auth <u>:<p>
- 启用
API: <jenkins url>/view/<view name>/job/<job name>/enable
- 禁用
API: <jenkins url>/view/<view name>/job/<job name>/disable
- 查询配置
API: <jenkins url>/view/<view name>/job/<job name>/config.xml
curl: curl -X GET <jenkins url>/view/<view name>/job/<job name>/config.xml --auth <u>:<p>
- 更新配置
API: <jenkins url>/view/<view name>/job/<job name>/config.xml
curl: curl -X POST <jenkins url>/view/<view name>/job/<job name>/config.xml --data-binary "@newCofnig.xml" -H "Content-Type: text/xml" --auth <u>:<p>

Job Build

为了描述方便，这里先定义 `baseUrl=<jenkins url>/view/<view name>/job/<job name>`，下面的 `API` 都需要加上 `baseUrl` 才是完整的 `=URL=`。


- 触发构建
API: `/build`
支持 `token` 触发，支持填入构建参数，构建参数是在 `Job` 配置页面创建的。
如使用 `token` 并有字符参数 `branch` 和 `target` 的例子：

```
curl -X POST <api> --data-urlencode token=<token value> --data-urlencode json='{"parameters": [{"name": "branch", "value": "master"}, {"name": "target", "value": "mirror"}]}'
```

触发成功后，会在 `header` 的 `Location` 字段指明 `queue url`，再通过查询 `queue` 就可获取到 `build id` 了。
- 停止构建
API: `/<id>/stop`
- 删除构建
API: `/<id>/doDelete`
- 构建状态
API: `/<id>/api/json`
- 最后一次构建
API: `/lastBuild/api/json`

附录

使用 Go 编写触发构建的程序

 View Code

参考资料

没有在 [Jenkins REST API 文档](#) 中找到过多的 `API` 信息，但在 [python-jenkins](#) 的 [代码](#) 中找到，所以想要了解更多请看代码。

以下方式不推荐，只是临时解决思路。

原文链接：<https://www.cnblogs.com/luckyhome168/p/7455482.html>

问题：通过Postman访问 Jenkins api时出现访问权限问题



```
<html>
  <head>
    <meta http-equiv='refresh' content='1;url=/jenkins/login?from=%2Fjenkins%2Fapi%2Fjson' />
    <script>window.location.replace('/jenkins/login?from=%2Fjenkins%2Fapi%2Fjson');</script>
  </head>
  <body style='background-color:white; color:white;'>

Authentication required

    <!--
You are authenticated as: anonymous
Groups that you are in:

Permission you need to have (but didn't): hudson.model.Hudson.Read
... which is implied by: hudson.security.Permission.GenericRead
... which is implied by: hudson.model.Hudson.Administer
-->




  </body>
</html>
```



解决方法：Jenkins 系统设置中设置未授权用户可以读访问。



管理Jenkins

-  [系统设置](#)
全局设置&路径
-  [Configure Global Security](#)
Secure Jenkins; define who is allowed to access/use the system.
-  [Configure Credentials](#)
Configure the credential providers and types



Configure Global Security

☒ 启用安全

TCP port for JNLP agents ☐ 指定端口: ☐ 随机选取 ☒ 禁用

Agent protocols...

Disable remember me ☐

访问控制

安全域

☒ Jenkins专有用户数据库

☐ 允许用户注册

☐ LDAP

☐ Servlet容器代理

授权策略

☐ 任何用户可以做任何事(没有任何限制)

☐ 安全矩阵

☒ 登录用户可以做任何事

☒ Allow anonymous read access

☐ 遗留模式

然后Apply后，再次访问可以正常了

原文地址: <https://www.cnblogs.com/tianleblog/p/11792977.html>