

# Web Rest Api实践

---

API首页: <http://127.0.0.1:8080/api/>

## 项目API

---

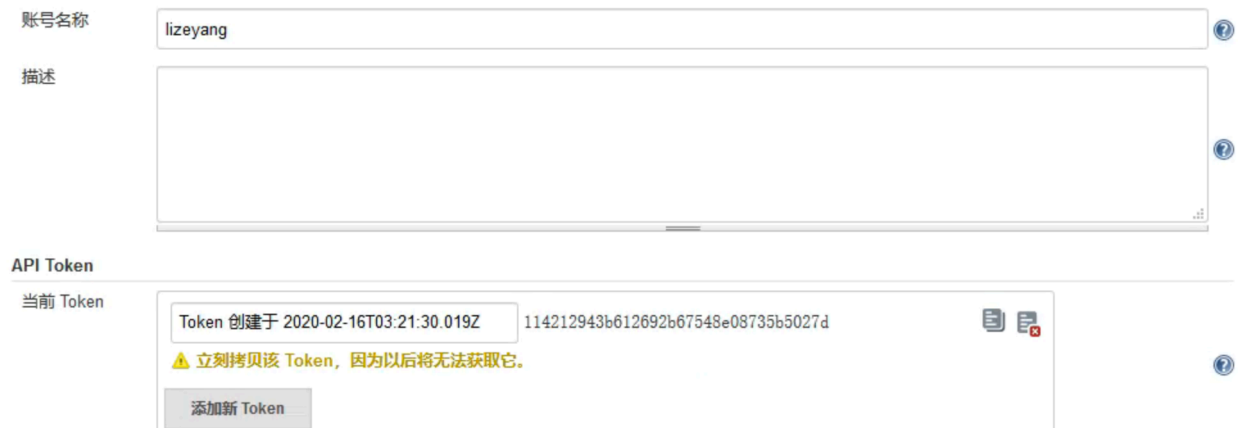
- 获取项目信息
  - 接口: <http://127.0.0.1:8080/job/{projectName}/api/json>
  - 方式: GET
  - 实例: <http://127.0.0.1:8080/job/demo-test-03/api/json>
- 获取项目构建信息
  - 接口: <http://127.0.0.1:8080/job/{projectName}/{buildNumber}/api/json>
  - 实例: <http://127.0.0.1:8080/job/test-devops-service/1/api/json>
  - 方式: GET
- 获取项目配置
  - 接口: <http://127.0.0.1:8080/job/{projectName}/config.xml>
  - 实例: <http://127.0.0.1:8080/job/test-devops-service/config.xml>
  - 方式: GET
- 创建项目
  - 接口: <http://127.0.0.1:8080/createItem?name={projectName}>
  - 参数: --data-binary @config.xml
  - 头部: -H "Content-Type:text/xml"
  - 方式: POST
- 禁用项目
  - 接口: <http://127.0.0.1:8080/job/{projectName}/disable>
  - 方式: POST
- 启用项目
  - 接口: <http://127.0.0.1:8080/job/{projectName}/enable>
  - 方式: POST
- 删除项目
  - 接口: <http://127.0.0.1:8080/job/{projectName}/doDelete>
  - 方式: POST
- 构建项目
  - 接口: <http://127.0.0.1:8080/job/{projectName}/build>
  - 方式: POST
- 参数化构建
  - 接口: <http://127.0.0.1:8080/job/{projectName}/buildWithParameters>

- 方式：POST

## 使用httprequest封装请求

如果没有做认证集成： 用户名+密码 。如果做了集成： 用户名+token。

由于本地做了github认证集成，所以这里要用一个token。如果没有做集成不需要创建token。



114212943b612692b67548e08735b5027d

创建一个凭据保存jenkins账号密码。（注意如果是github集成一定要填写用户ID而不是用户名称）



使用httprequest封装请求。

```
pipeline {  
  agent {node {label "master"}}}
```



```

        ignoreSslErrors: true,
        requestBody: reqBody,
        url: "${jenkinsServer}/${reqUrl}"
        //quiet: true
    }

}

//新建项目

def CreateProject(projectName){

    withCredentials([usernamePassword(credentialsId: 'jenkins-zeyang-admin',
passwordVariable: 'password', usernameVariable: 'username')]) {

        sh """
            curl -u ${username}:${password} -X GET
            'http://192.168.1.200:30080/job/demo-project-manage/config.xml' -o config.xml
            ls -l

            curl -u ${username}:${password} -X POST
            'http://192.168.1.200:30080/createItem?name=${projectName}' -H 'Content-
            Type:text/xml' --data-binary @config.xml

        """
    }
}

//项目操作

def Project(projectName,option){
    println(projectName)
    println(option)

    options = [ "DisableProject": "disable",
                "EnableProject": "enable",
                "DeleteProject": "doDelete",
                "BuildProject": "build" ]

    result = HttpReq('POST',"job/${projectName}/${options[option]}",'')

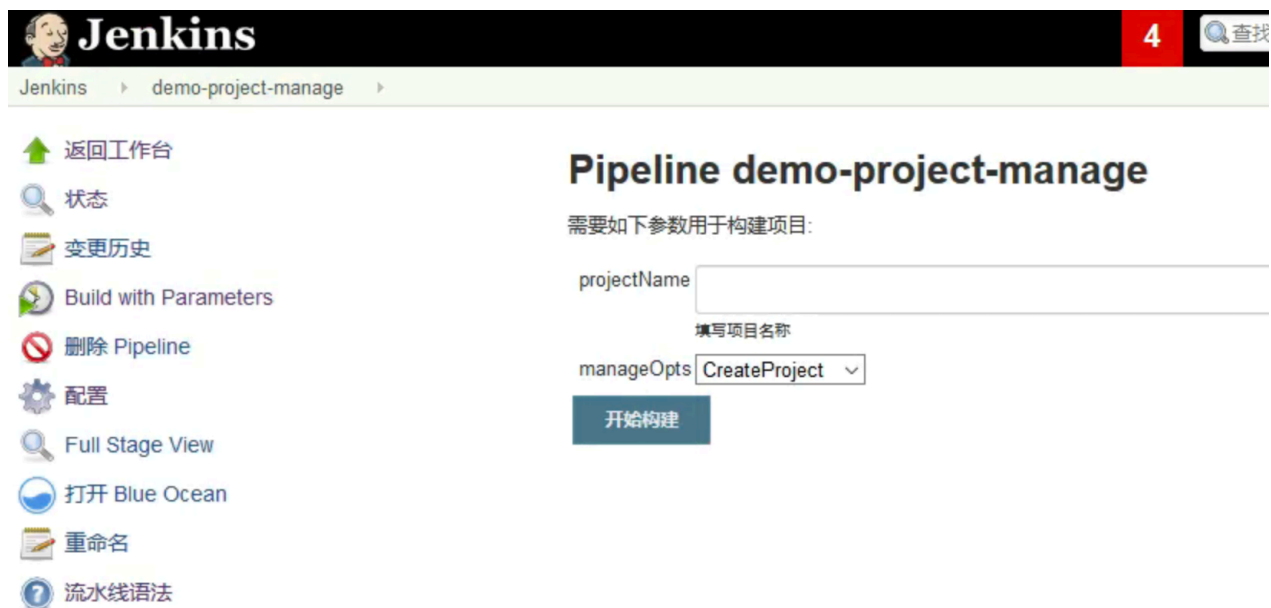
}

```

创建一个流水线项目具有两个参数,一个字符参数 `projectName` 用于填写项目名称, 一个选项参数 `manageOpts` 用于项目的操作。

manageOpts:

- CreateProject 创建项目
- DisableProject 禁用项目
- EnableProject 启动项目
- DeleteProject 删除项目
- BuildProject 构建项目



The image shows the Jenkins web interface for configuring a pipeline named 'demo-project-manage'. The top navigation bar includes the Jenkins logo, a breadcrumb trail 'Jenkins > demo-project-manage', a red tab with the number '4', and a search icon labeled '查找'. On the left sidebar, there are several icons and links: '返回工作台' (Return to Dashboard), '状态' (Status), '变更历史' (Change History), 'Build with Parameters' (selected), '删除 Pipeline' (Delete Pipeline), '配置' (Configure), 'Full Stage View', '打开 Blue Ocean' (Open Blue Ocean), '重命名' (Rename), and '流水线语法' (Pipeline Syntax). The main content area is titled 'Pipeline demo-project-manage' and contains the text '需要如下参数用于构建项目:' (Need the following parameters for building the project:). Below this, there are two input fields: 'projectName' with a text input box and a label '填写项目名称' (Fill in project name), and 'manageOpts' with a dropdown menu currently showing 'CreateProject'. At the bottom of the configuration area is a blue button labeled '开始构建' (Start Building).

Jenkinsfile

```
@Library("jenkinslibrary@master") _

def jenkinsapi = new org.devops.jenkinsapi()

String projectName = "${env.projectName}"
String manageOpts = "${env.manageOpts}"

pipeline {

    agent { node {label "master"}}

    stages{
        stage("test"){
            steps{
                script{
```

```
        if (manageOpts == "CreateProject"){
            jenkinsapi.CreateProject(projectName)
        } else {
            jenkinsapi.Project(projectName,manageOpts)
        }
    }
}
}
```