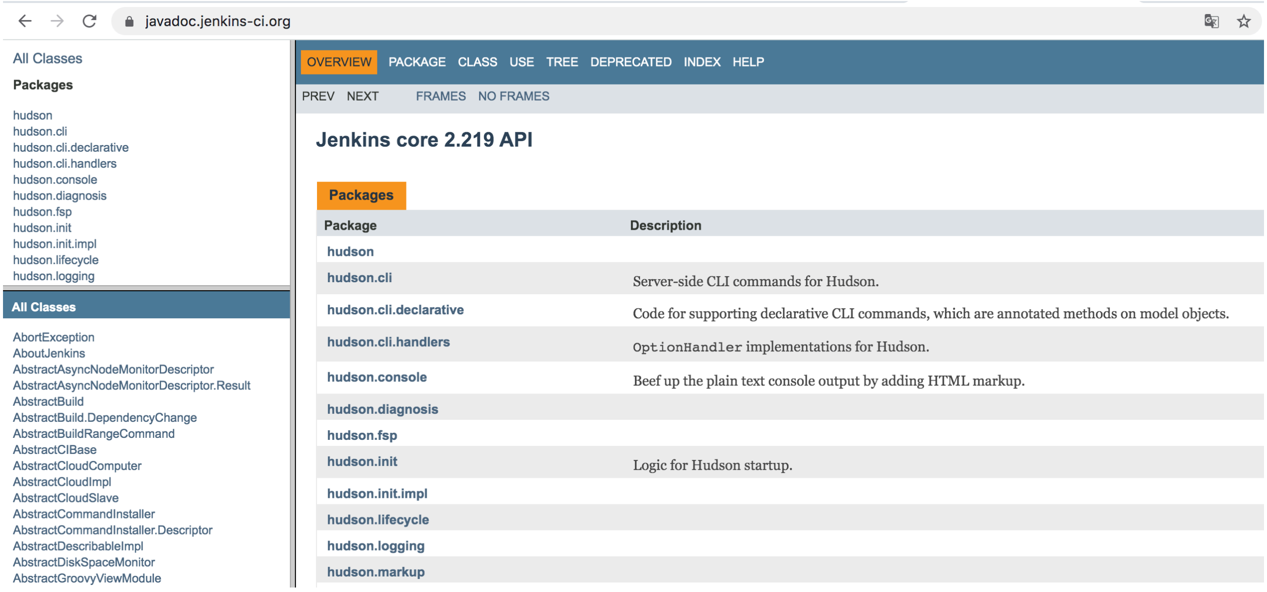
Jenkins Core API

简介

API文档： https://javadoc.jenkins-ci.org/



All Classes

Packages

hudson
hudson.cli
hudson.cli.declarative
hudson.cli.handlers
hudson.console
hudson.diagnosis
hudson.fsp
hudson.init
hudson.init.impl
hudson.lifecycle
hudson.logging

All Classes

AbortException
AboutJenkins
AbstractAsyncNodeMonitorDescriptor
AbstractAsyncNodeMonitorDescriptor.Result
AbstractBuild
AbstractBuild.DependencyChange
AbstractBuildRangeCommand
AbstractCIBase
AbstractCloudComputer
AbstractCloudImpl
AbstractCloudSlave
AbstractCommandInstaller
AbstractCommandInstaller.Descriptor
AbstractDescribableImpl
AbstractDiskSpaceMonitor
AbstractGroovyViewModule

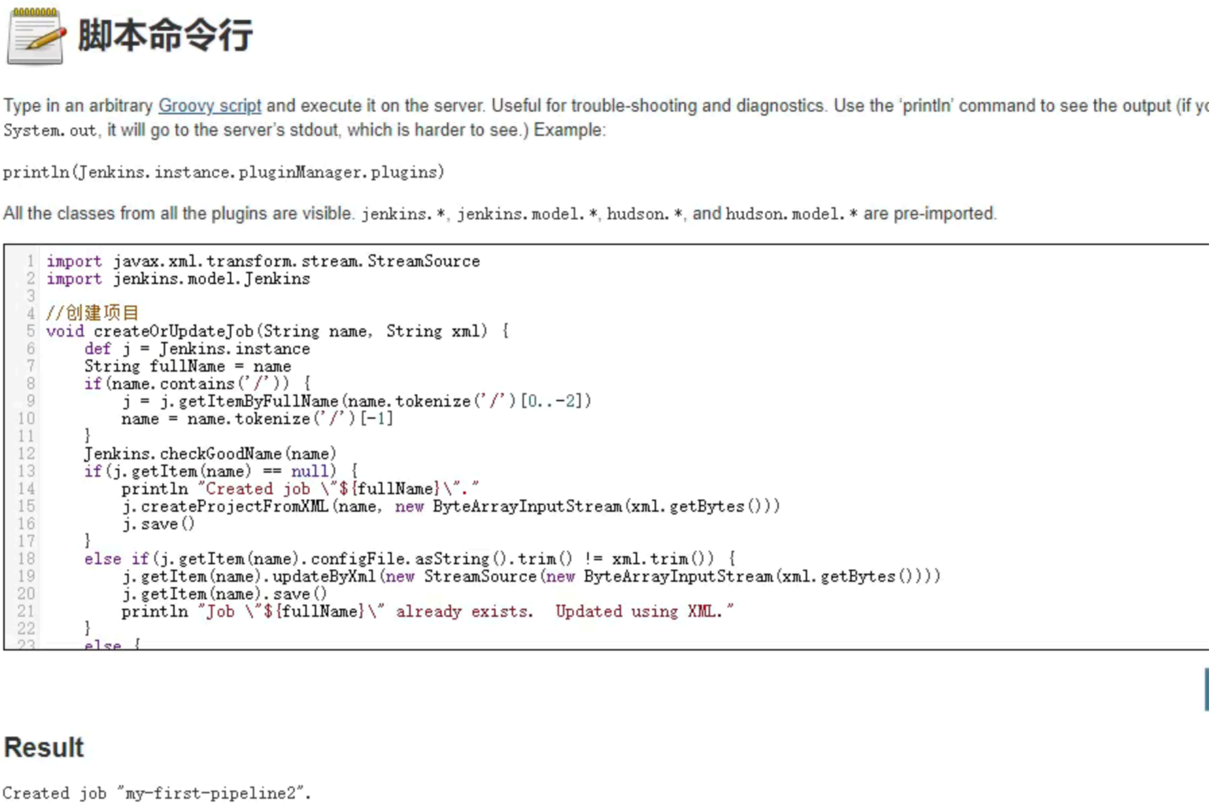
Jenkins core 2.219 API

Packages

Package	Description
hudson	
hudson.cli	Server-side CLI commands for Hudson.
hudson.cli.declarative	Code for supporting declarative CLI commands, which are annotated methods on model objects.
hudson.cli.handlers	`OptionHandler` implementations for Hudson.
hudson.console	Beef up the plain text console output by adding HTML markup.
hudson.diagnosis	
hudson.fsp	
hudson.init	Logic for Hudson startup.
hudson.init.impl	
hudson.lifecycle	
hudson.logging	
hudson.markup	

```
import javax.xml.transform.stream.StreamSource
import jenkins.model.Jenkins

//创建项目
void createOrUpdateJob(String name, String xml) {
    def j = Jenkins.instance
    String fullName = name
    if(name.contains('/')) {
        j = j.getItemByFullName(name.tokenize('/')[0..-2])
        name = name.tokenize('/')[-1]
    }
    Jenkins.checkGoodName(name)
    if(j.getItem(name) == null) {
        println "Created job \"${fullName}\"."
        j.createProjectFromXML(name, new ByteArrayInputStream(xml.getBytes()))
        j.save()
    }
    else if(j.getItem(name).configFile.asString().trim() != xml.trim()) {
        j.getItem(name).updateByXml(new StreamSource(new
ByteArrayInputStream(xml.getBytes())))
        j.getItem(name).save()
        println "Job \"${fullName}\" already exists.  Updated using XML."
    }
    else {
        println "Nothing changed.  Job \"${fullName}\" already exists."
    }
}

try {
    //just by trying to access properties should throw an exception
} catch(MissingPropertyException e) {
    println 'ERROR Can\'t create job.'
    println 'ERROR Missing properties: itemName, xmlData'
    return
}

String xmlData = """<!-- 1. test-schdule-service -->
<flow-definition>
    <actions></actions>
    <description>this is my first job</description>
    <keepDependencies>false</keepDependencies>
    <properties>
        <hudson.model.ParametersDefinitionProperty>
            <parameterDefinitions>
                <hudson.model.ChoiceParameterDefinition>
                    <choices class='java.util.Arrays$ArrayList'>
                        <a class='string-array'>
                            <string>1</string>
                            <string>2</string>
                            <string>3</string>
                        </a>
                    </choices>
                    <name>test</name>
                    <description></description>
                </hudson.model.ChoiceParameterDefinition>
            </parameterDefinitions>
        </hudson.model.ParametersDefinitionProperty>
        <com.coravy.hudson.plugins.github.GithubProjectProperty>

<projectUrl>https://github.com/https://gitlab.com/xxx/xxx.git/</projectUrl>
        </com.coravy.hudson.plugins.github.GithubProjectProperty>
    </properties>
    <triggers></triggers>
    <definition
class='org.jenkinsci.plugins.workflow.cps.CpsScmFlowDefinition'>
        <scriptPath>Jenkinsfile</scriptPath>
        <lightweight>false</lightweight>
        <scm class='hudson.plugins.git.GitSCM'>
            <userRemoteConfigs>
                <hudson.plugins.git.UserRemoteConfig>
```

```
<url>https://github.com/https://gitlab.com/xxx/xxx.git.git</url>
                    <credentialsId>24982560-17fc-4589-819b-
bc5bea89da77</credentialsId>
                </hudson.plugins.git.UserRemoteConfig>
            </userRemoteConfigs>
            <branches>
                <hudson.plugins.git.BranchSpec>
                    <name>*/master</name>
                </hudson.plugins.git.BranchSpec>
            </branches>
            <configVersion>2</configVersion>

<doGenerateSubmoduleConfigurations>false</doGenerateSubmoduleConfigurations>
            <gitTool>Default</gitTool>
            <browser class='hudson.plugins.git.browser.GithubWeb'>
                <url>https://github.com/https://gitlab.com/xxx/xxx.git/</url>
            </browser>
        </scm>
    </definition>
    <disabled>false</disabled>
</flow-definition>
"""
String itemName = "my-first-pipeline"

createOrUpdateJob(itemName, xmlData)
```



脚本命令行

Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if yo
System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

```
import javax.xml.transform.stream.StreamSource
import jenkins.model.Jenkins

//创建项目
void createOrUpdateJob(String name, String xml) {
    def j = Jenkins.instance
    String fullName = name
    if(name.contains('/')) {
        j = j.getItemByFullName(name.tokenize('/')[0..-2])
        name = name.tokenize('/')[-1]
    }
    Jenkins.checkGoodName(name)
    if(j.getItem(name) == null) {
        println "Created job \"${fullName}\"."
        j.createProjectFromXML(name, new ByteArrayInputStream(xml.getBytes()))
        j.save()
    }
    else if(j.getItem(name).configFile.asString().trim() != xml.trim()) {
        j.getItem(name).updateByXml(new StreamSource(new ByteArrayInputStream(xml.getBytes())))
        j.getItem(name).save()
        println "Job \"${fullName}\" already exists.  Updated using XML."
    }
    else {
```

Result

```
Created job "my-first-pipeline2".
```