

ローカルリポジトリで **Jenkins** を使用 (**linux** 環境)

発表者: 野口耕二郎

今回行うこと

- ローカルリポジトリ(**Git**)でコミットすると自動的にテストを行う環境を作成
- Jenkins** 側の設定や **Git** 側の設定の説明を行う

今回使うコード

- ソースの構造

```
src
├── main
│   └── java
│       └── com
│           ├── example
│           └── Sample.java
└── test
    ├── java
    │   ├── com
    │   │   ├── example
    │   │   └── SampleTest.java
```

ソースコードの中身

```
package com.example;
```

```
public class Sample {  
    public int add(int a, int b) {  
        return a + b;  
    }  
}
```

```
public int returnEven(int num) {  
    if (num % 2 == 0) {  
        return num;  
    } else {  
        return -1;  
    }  
}
```

```
public String returnHello() {  
    return "Hello!!!";  
}  
}
```

テストコードの中身

```
package com.example;

import org.junit.Test;
import static org.junit.Assert.*;

public class SampleTest {
    Sample smpl = new Sample();

    @Test
    public void testAdd() {
        assertEquals(3, smpl.add(1, 2));
    }

    @Test
    public void testReturnEven1() {
        assertTrue(smpl.returnEven(4) % 2 == 0);
    }

    @Test
    public void testReturnEven2() {
        assertTrue(smpl.returnEven(3) == -1);
    }

    @Test
    public void testReturnHello() {
        assertEquals("Hello!!!", smpl.returnHello());
    }
}
```

build.gradle の中身

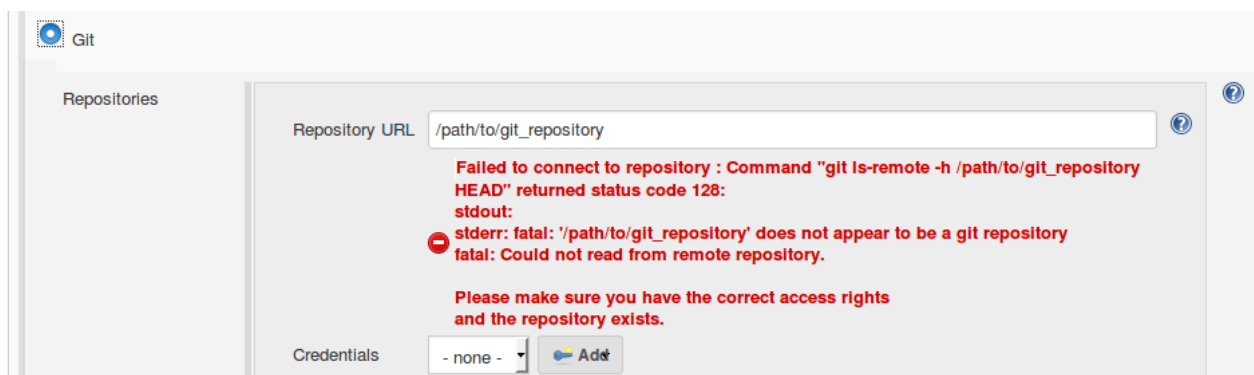
```
apply plugin: 'java'
```

```
repositories {  
    mavenCentral()  
}
```

```
dependencies {  
    testCompile 'junit:junit:4.+'  
}
```

Jenkins で Git リポジトリを登録

- ホーム画面で「 **New Item** 」というところをクリック
- プロジェクト名を書いて「 **FreeStyle project** 」をクリック
- 「 **Source Code Management** 」の部分で以下のように git リポジトリのファイルパスを入力



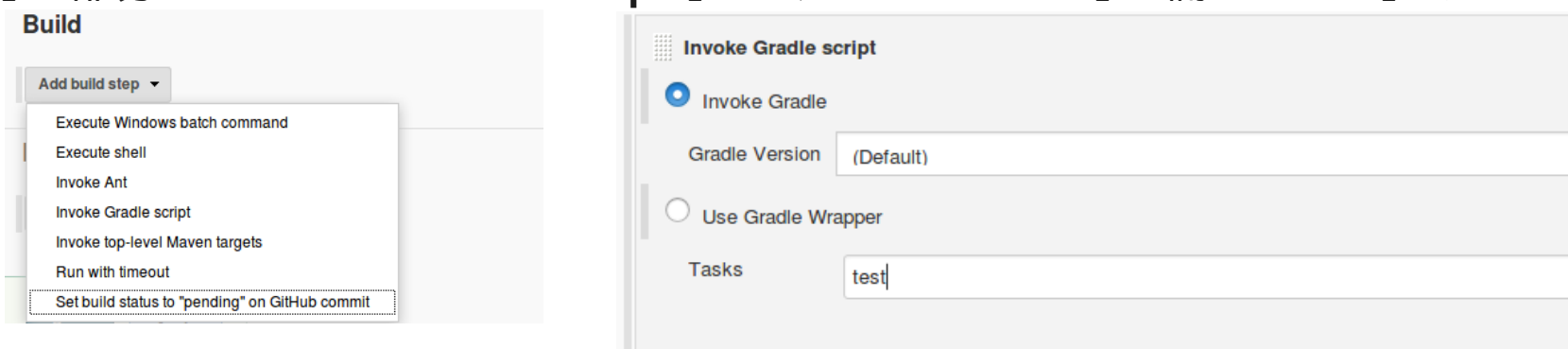
Jenkins で Git リポジトリを登録

- 「Build Triggers」の部分で「Trigger builds remotely」をチェックし好きなトークン名を書く



The screenshot shows the 'Build Triggers' configuration section in Jenkins. The 'Trigger builds remotely (e.g., from scripts)' checkbox is checked. Below it is an 'Authentication Token' text field. A note explains the URL format for triggering builds remotely: `JENKINS_URL/job/aaa/build?token=TOKEN_NAME` or `/buildWithParameters?token=TOKEN_NAME`, and mentions that appending `&cause=Cause+Text` includes the cause in the build record. Other unselected options include 'GitHub hook trigger for GITScm polling', 'Poll SCM', 'Build after other projects are built', and 'Build periodically'.

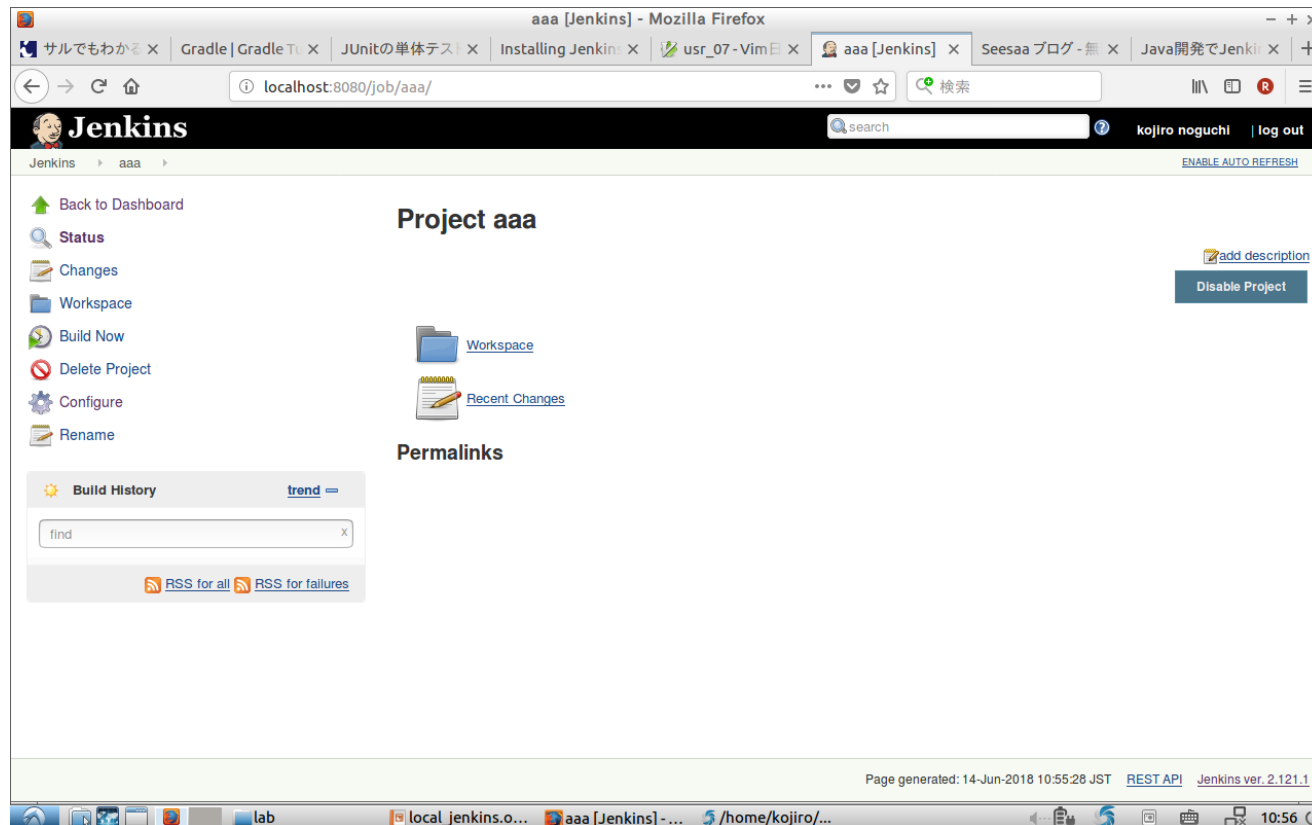
- 「Build」の部分で「invoke Gradle Script」をクリックし「Tasks」の欄に「test」と入れる



The left screenshot shows the 'Build' configuration page with the 'Add build step' dropdown menu open. The menu lists several options, with 'Invoke Gradle script' highlighted. The right screenshot shows the 'Invoke Gradle script' configuration page. The 'Invoke Gradle' radio button is selected. The 'Gradle Version' is set to '(Default)'. The 'Tasks' text field contains the word 'test'.

Jenkins で Git リポジトリを登録

- 「SAVE」をクリックして終了
- 以下のような画面に移る



Git 側の設定

- git リポジトリの「`.git/hooks/`」の下に「`post-commit`」というスクリプトを作成する
- 内容はこんな感じ

```
#!/bin/sh
echo "Hook post-commit start"
wget --auth-no-challenge --http-user=「ユーザ名」 --http-password=「パスワード」 --secure-protocol=TLSv1 'http://localhost:8080/job/「プロジェクト名」/build?token=「トークン名」' -O -
echo "Hook post-commit end"
```

- 最後に「`chmod u+x post-commit`」を実行して完了