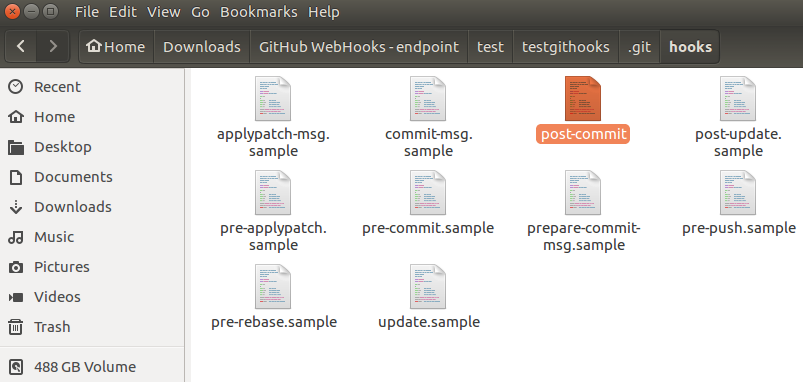
Git Hooks

Use case

We have our personal website repository and every time we commit changes we want to upload files to FTP server - also the place we store our website is configured on nginx to be exposed on the web.

1 We need to use post-commit git hook:



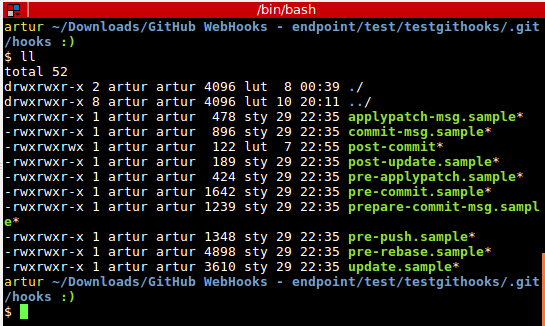
Create file named ‘post-commit’ - if it is not already present in .git/hooks directory:

touch post-commit

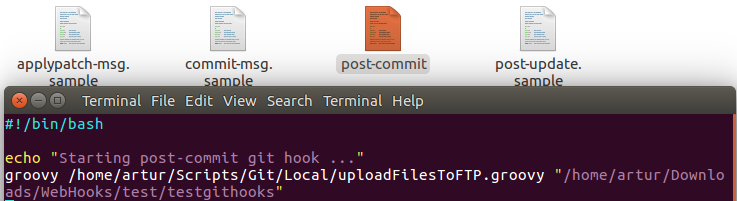
or if there is ‘post-commit.sample’ file - remove ‘.sample’ from file name.

Make sure that file is executable:

chmod +x post-commit



Enter command(s) that you wish to execute every time you commit something or execute script outside of ‘post-commit’ hook

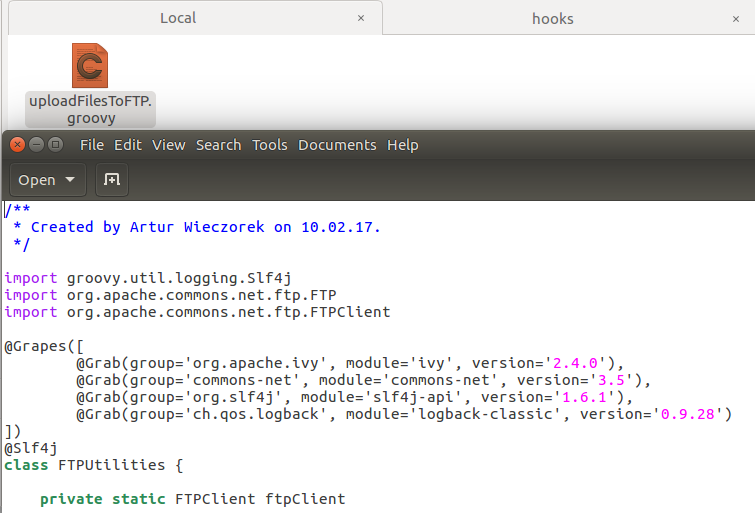


#!/bin/bash

echo "Starting post-commit git hook ..."

groovy /home/artur/Scripts/Git/Local/uploadFilesToFTP.groovy "/home/artur/Downloads/WebHooks/test/testgithooks"

Here we are running groovy script to upload files to FTP server and we are passing path to directory we want to upload as an script argument:



/\*\*

\* Created by Artur Wieczorek on 10.02.17.

\*/

import groovy.util.logging.Slf4j

import org.apache.commons.net.ftp.FTP

import org.apache.commons.net.ftp.FTPClient

@Grapes([

@Grab(group='org.apache.ivy', module='ivy', version='2.4.0'),

@Grab(group='commons-net', module='commons-net', version='3.5'),

@Grab(group='org.slf4j', module='slf4j-api', version='1.6.1'),

@Grab(group='ch.qos.logback', module='logback-classic', version='0.9.28')

])

@Slf4j

class FTPUtilities {

private static FTPClient ftpClient

static void login(String server, int port = 21, String username = 'ftp\_user', String password = 'your\_password') throws IOException {

ftpClient = new FTPClient()

try {

ftpClient.connect(server, port)

ftpClient.login(username, password)

ftpClient.enterLocalPassiveMode()

log.info 'Connected and successfully logged to FTP server'

}

catch (IOException ex) {

ex.printStackTrace()

}

}

static void uploadDirectory(String remoteDirPath, String localDirPath) throws IOException {

log.info 'Listing directory: ' + localDirPath

File localDir = new File(localDirPath)

File[] subFiles = localDir.listFiles()

if (subFiles) {

for (File item : subFiles) {

String remoteFilePath = remoteDirPath + '/' + item.getName()

if (item.isFile()) {

// upload the file

String localFilePath = item.getAbsolutePath()

log.info 'Starting upload of the file: ' + localFilePath

boolean uploaded = uploadSingleFile(localFilePath, remoteFilePath)

if (uploaded)

log.info 'File uploaded to: ' + remoteFilePath

else

log.warn 'Could not upload file: ' + localFilePath

} else {

// ignore .git directory

if(remoteFilePath.contains('.git'))

continue

// create directory on the server

boolean created = ftpClient.makeDirectory(remoteFilePath)

if (created)

log.info 'Created directory: ' + remoteFilePath

else

log.warn 'Could not create the directory: ' + remoteFilePath

// upload the sub directory

localDirPath = item.getAbsolutePath()

uploadDirectory(remoteFilePath, localDirPath)

}

}

}

}

static boolean uploadSingleFile(String localFilePath, String remoteFilePath) throws IOException {

File localFile = new File(localFilePath)

InputStream inputStream = new FileInputStream(localFile)

try {

ftpClient.setFileType(FTP.BINARY\_FILE\_TYPE)

return ftpClient.storeFile(remoteFilePath, inputStream)

} finally {

inputStream.close()

}

}

static void disconnect(){

ftpClient.logout()

ftpClient.disconnect()

}

}

String server = '111.22.33.44'

FTPUtilities.login(server)

for(String arg : args ){

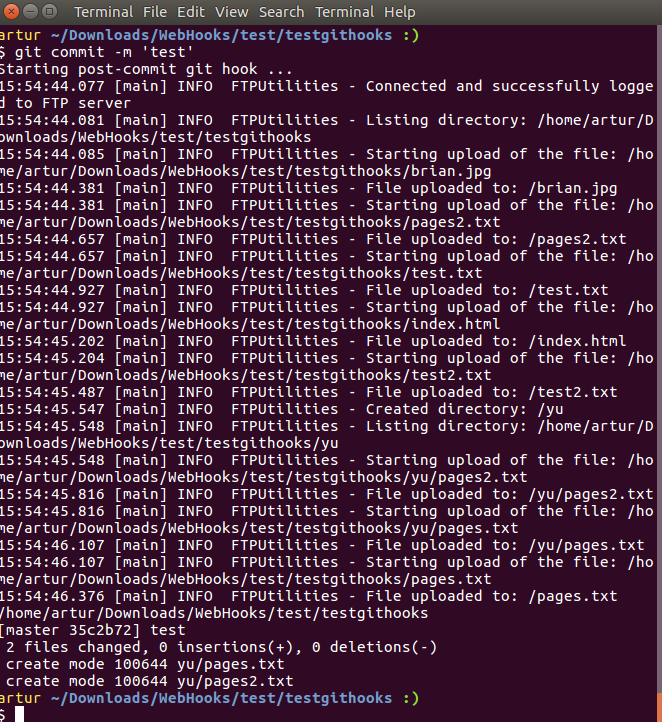
FTPUtilities.uploadDirectory('', arg)

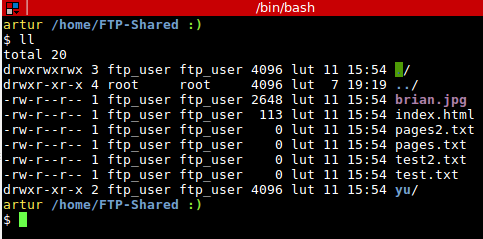
println arg

}

FTPUtilities.disconnect()

Now, when you commit changes the script will be executed and files uploaded to server:





JAVA version of this file:

import org.apache.commons.net.ftp.FTP;

import org.apache.commons.net.ftp.FTPClient;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

*/\*\**

*\* Created by Artur on 06.02.17.*

*\*/*

public class FTPUtils {

public static void uploadDirectory(FTPClient ftpClient,

String remoteDirPath, String localParentDir, String remoteParentDir)

throws IOException {

System.*out*.println("LISTING directory: " + localParentDir);

File localDir = new File(localParentDir);

File[] subFiles = localDir.listFiles();

if (subFiles != null && subFiles.length > 0) {

for (File item : subFiles) {

String remoteFilePath = remoteDirPath + "/" + remoteParentDir

+ "/" + item.getName();

if (remoteParentDir.equals("")) {

remoteFilePath = remoteDirPath + "/" + item.getName();

}

if (item.isFile()) {

// upload the file

String localFilePath = item.getAbsolutePath();

System.*out*.println("About to upload the file: " + localFilePath);

boolean uploaded = *uploadSingleFile*(ftpClient,

localFilePath, remoteFilePath);

if (uploaded) {

System.*out*.println("UPLOADED a file to: "

+ remoteFilePath);

} else {

System.*out*.println("COULD NOT upload the file: "

+ localFilePath);

}

} else {

// create directory on the server

boolean created = ftpClient.makeDirectory(remoteFilePath);

if (created) {

System.*out*.println("CREATED the directory: "

+ remoteFilePath);

} else {

System.*out*.println("COULD NOT create the directory: "

+ remoteFilePath);

}

// upload the sub directory

String parent = remoteParentDir + "/" + item.getName();

if (remoteParentDir.equals("")) {

parent = item.getName();

}

localParentDir = item.getAbsolutePath();

*uploadDirectory*(ftpClient, remoteDirPath, localParentDir,

parent);

}

}

}

}

public static boolean uploadSingleFile(FTPClient ftpClient,

String localFilePath, String remoteFilePath) throws IOException {

File localFile = new File(localFilePath);

InputStream inputStream = new FileInputStream(localFile);

try {

ftpClient.setFileType(FTP.*BINARY\_FILE\_TYPE*);

return ftpClient.storeFile(remoteFilePath, inputStream);

} finally {

inputStream.close();

}

}

public static void main(String[] args) {

String server = "138.68.122.58";

int port = 21;

String user = "ftp\_user";

String pass = "ftp\_password";

FTPClient ftpClient = new FTPClient();

try {

// login and login to the server

ftpClient.connect(server, port);

ftpClient.login(user, pass);

// use local passive mode to pass firewall

ftpClient.enterLocalPassiveMode();

System.*out*.println("Connected");

String remoteDirPath = "/";

String localDirPath = "/home/artur/Downloads/Test/";

FTPUtils.*uploadDirectory*(ftpClient, remoteDirPath, localDirPath, "");

// log out and disconnect from the server

ftpClient.logout();

ftpClient.disconnect();

System.*out*.println("Disconnected");

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

And bash script to execute it:

#!/bin/bash

java -cp ".:/home/artur/.gradle/caches/modules-2/files-2.1/commons-net/commons-net/3.5/342fc284019f590e1308056990fdb24a08f06318/commons-net-3.5.jar:/home/artur/IdeaProjects/dependency-injection/build/classes/main/" FTPUtils

We need to pass location of Apache library (jar file) we use for FTP connections to cp - classpath argument and add : and then location of the compiled java class