

Bitbucket ScriptRunner를 사용하여 브랜치 생성 시 날짜 Commit 하기

이 문서는 Bitbucket ScriptRunner를 사용하여 브랜치 생성 시 날짜 Commit 가이드를 공유하기 위해 작성되었다.

도구명	Bitbucket, ScriptRunner for Bitbucket
비고	

- ScriptRunner를 사용하여 브랜치 생성 시 날짜 Commit
 - Custom Listener 만들기
- [참조링크](#)

ScriptRunner를 사용하여 브랜치 생성 시 날짜 Commit

Custom Listener 만들기

- Script Runner → Listeners → Create Listener 선택

The screenshot shows the Bitbucket ScriptRunner interface. On the left is the 'Administration' sidebar with various settings. The main area is titled 'ScriptRunner' and has a tabbed interface. The 'Listeners' tab is selected, indicated by a blue circle with the number '1' and a red box. Below the tabs, there's a 'Listeners' section with a description and a 'Documentation & Tips' link. At the bottom right, a 'Create Listener' button is highlighted with a blue circle with the number '2' and a red box. Below this, there's a table listing existing listeners.

Name	Created in	Applied to	Events	Performance	History	Actions
Check delete branch checkbox No note	Global	All	1 Event	100%	✓ No failures in the last 15 execution(s)	...
Custom listener (Disabled) No note	Global	All	1 Event	100%	⚠ 1 of 15 executions have failed	...
Custom listener No note	Global	All	1 Event	100%	⚠ 1 of 15 executions have failed	...

- Custom Listener 선택

Administration

ScriptRunner

Listeners > Create

Create Listeners

ScriptRunner allows you to respond to Bitbucket events through built-in and custom event listeners. To write your own choose *Custom listener*.

Search Listeners

Listeners >

- System**
 - Check delete branch checkbox
 - Check the delete branch checkbox on a pull request to be enabled
- Project**
 - Custom listener
 - Run your own groovy scripts in response to events
- Repository**
 - Auto-add reviewers to pull requests
 - Automatically add reviewers, depending on your conditions
- Branch**
 - Auto-merge pull requests on approval
 - Automatically merge a pull request upon approval
- Tag**
 - Remote custom listener dispatcher
 - Listener for dispatching remote events - these are created when you add a Remote Custom Listener
- Pull Request**
 - Block out-of-date pull requests
 - Prevent creation of a pull request when it has fallen behind the target branch

■ BranchCreatedEvent 이벤트 선택

ScriptRunner

Custom listener

Run your own groovy scripts in response to events

Documentation & Tips

Note

An optional note, used only for your reference.

Events

BranchCreatedEvent

Select the events this code will handle

Script

1

Enter your script here. If you are using a file then use the path to the script accessible on the server - or the fully-qualified class name for the class in the form com.acme.MyListener

Add Cancel

■ 해당 스크립트 사용

- 스크립트 내용
- 32줄: 내용을 기록할 파일명 입력
- 40줄: 원하는 날짜포맷 입력
- 44줄: 파일에 기록될 내용 입력
- 47줄: Commit Message 입력
- 54줄: 적용되는 브랜치 패턴 입력

```
import com.atlassian.sal.api.net.ResponseException
import com.atlassian.plugin.PluginAccessor
import com.onresolve.scriptrunner.canned.bitbucket.util.BitbucketBaseScript
import com.onresolve.scriptrunner.runner.ScriptRunnerImpl
```

```

import groovy.transform.BaseScript

import java.util.regex.Pattern
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.atlassian.sal.api.component.ComponentLocator
import com.atlassian.sal.api.component.ComponentLocator
import com.atlassian.bitbucket.event.branch.BranchCreatedEvent
import com.atlassian.bitbucket.event.ApplicationEvent

import com.atlassian.bitbucket.content.EditFileRequest
import java.nio.file.FileSystems
import java.nio.file.PathMatcher
import com.atlassian.bitbucket.content.ContentService
import com.atlassian.bitbucket.io.TypeAwareOutputSupplier
import com.atlassian.bitbucket.util.MoreSuppliers
import com.atlassian.bitbucket.io.InputSupplier
import java.io.ByteArrayOutputStream
import java.io.ByteArrayInputStream
import com.atlassian.bitbucket.avatar.SimpleAvatarSupplier
import java.text.SimpleDateFormat
@BaseScript BitbucketBaseScript baseScript
//
def event = event as BranchCreatedEvent
def log = Logger.getLogger("com.curvc")
def repository = event.getRepository()
//
def filePath = 'readme.txt'

log.setLevel(Level.DEBUG)
//
log.warn("Date: " + event.getDate())
def branch = event.getBranch()
String branchId = branch.getDisplayId()
//
SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy MM dd HH MM");
String strNowDate = simpleDateFormat.format(event.getDate())

//
String addStr = "\nCreate Branch Name: " + branchId + " Date: " + strNowDate

// Commit Message
String commitMessage = addStr

//
log.warn("BranchName: " + branchId)

//

def branchNamepattern = [~/release\/.*/ , ~/product\/.*/ , ~/develop\/.*/]
boolean branchNameCheck = false
branchNamepattern.each{ def pattern ->
    if( branchId =~ pattern){
        log.warn(branchId)
        branchNameCheck = true
    }
}
//      return
if(!branchNameCheck){
    log.warn("return")
    return
}

//      commit
def contentService = ComponentLocator.getComponent(ContentService)
def lastCommit = branch.getLatestCommit().toString()

ByteArrayOutputStream outputStream = new ByteArrayOutputStream()
TypeAwareOutputSupplier os = MoreSuppliers.newTypeAwareOutputSupplierOf(outputStream)
contentService.streamFile(repository, lastCommit, filePath, os);

```

```
def newline = addStr.bytes
outputStream.write(newline)
outputStream.flush()
log.warn(outputStream.toString())

byte[] bytes = outputStream.toByteArray()
InputStream inputStream = new ByteArrayInputStream(bytes)
InputSupplier lastContent = new SimpleAvatarSupplier(inputStream)

def editFileCommand = contentService.editFile(
    new EditFileRequest.Builder(branchId, filePath, repository)
        .content(lastContent)
        .message(commitMessage)
        .sourceCommitId(lastCommit)
        .build()
)
outputStream.close()
inputStream.close()
```



script에 설정한 FilePath에 해당 파일이 없을 시 Commit 할 수 없음

참조링크

- [Bitbucket API](#)
- [Script Runner](#)
- [날짜 포맷 변경 방법](#)
- [How to convert OutputStream to InputStream?](#)
- [GitHub - yaml-validator-hook](#)