

Jira - Scriptrunner 활용하여 Bitbucket Branch 생성 시 해당 이슈에 댓글 남기기

이 문서는 Jira Scriptrunner 활용하여 Bitbucket Branch 생성 시 해당 이슈에 댓글 남기기 가이드를 공유하기 위해 작성되었다.

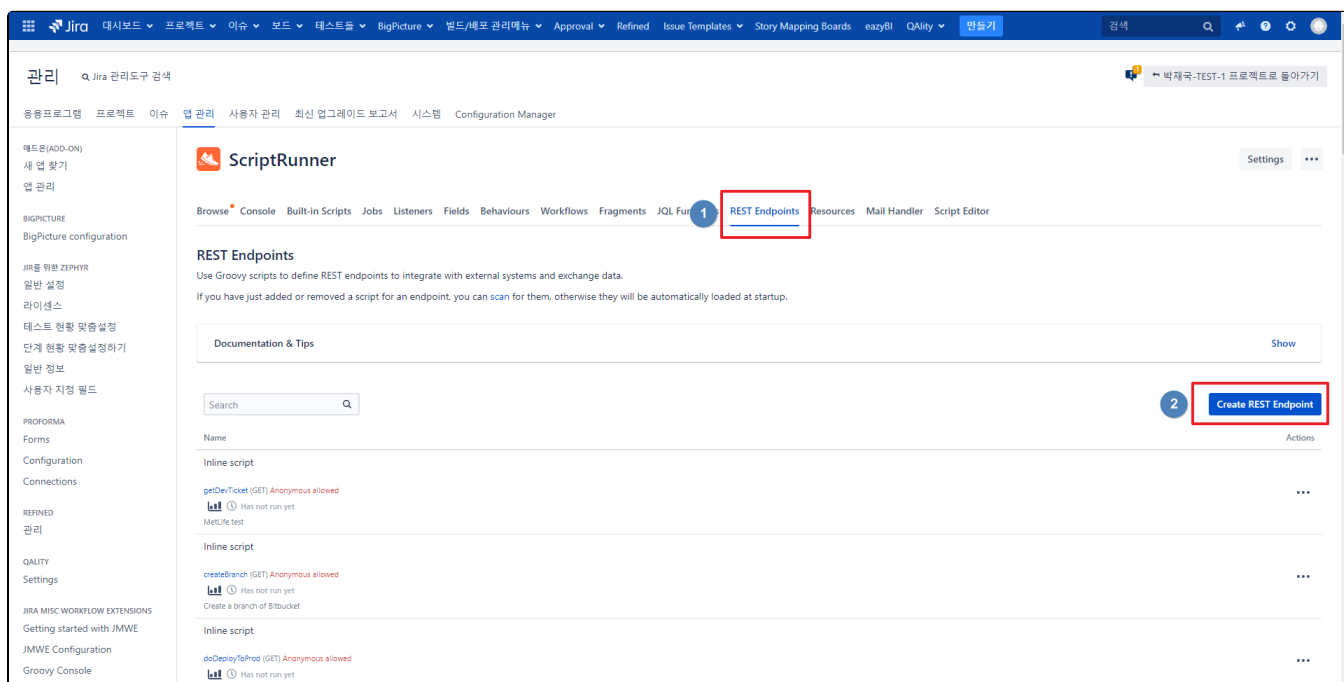
도구명	Jira, Scriptrunner, Bitbucket
비고	

- [Jira Scriptrunner 활용하여 Bitbucket Branch 생성 시 해당 이슈에 댓글 남기기](#)
 - [Scriptrunner REST endpoint 만들기](#)
 - [Bitbucket 저장소에서 Webhook 만들기](#)
 - [확인](#)
- [참조링크](#)

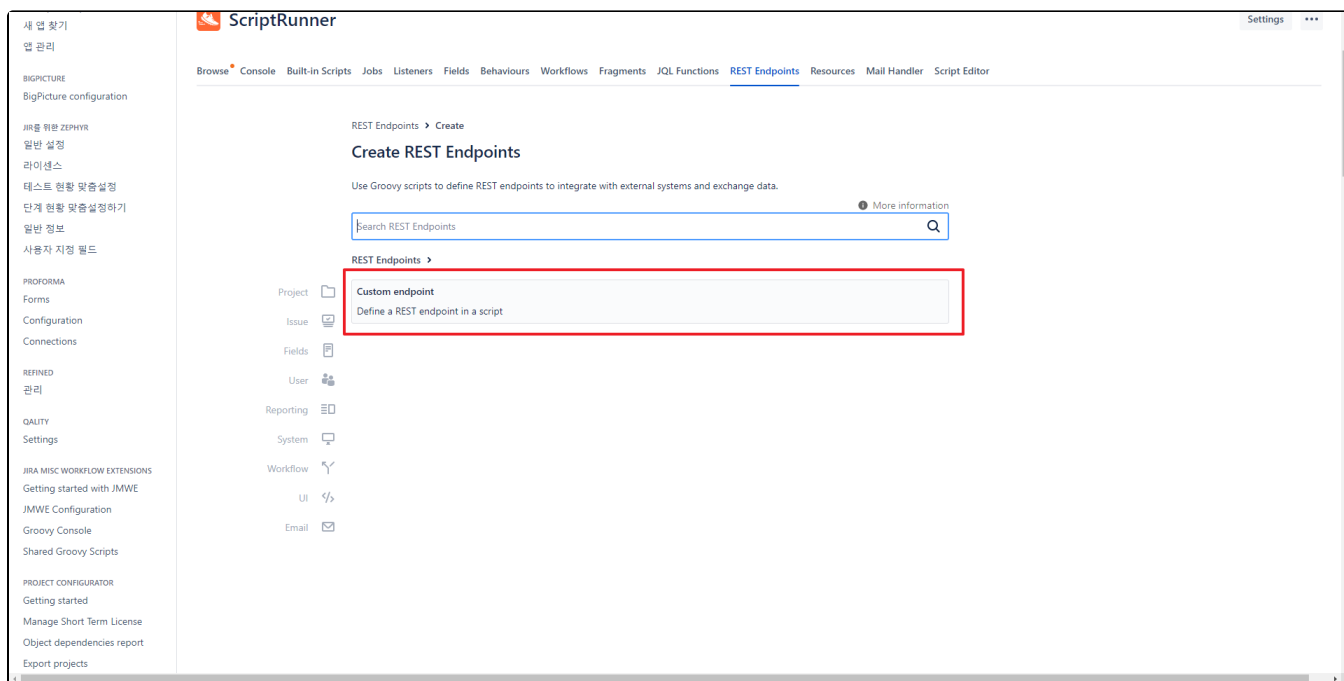
Jira Scriptrunner 활용하여 Bitbucket Branch 생성 시 해당 이슈에 댓글 남기기

Scriptrunner REST endpoint 만들기

- Scriptrunner → REST Endpoints → Create REST Endpoint 선택



- Custom endpoint 선택



■ 스크립트 내용

33번 줄 'repositoryPush': Scriptrunner rest Endpoint의 주소, {jira_url}/rest/scriptrunner/latest/custom/repositoryPush

64번 줄: Jira Issue에 댓글 남길 Id 입력

```
import com.onresolve.scriptrunner.runner.rest.common.CustomEndpointDelegate
import groovy.json.JsonBuilder
import groovy.json.JsonSlurper
import groovy.transform.BaseScript

import javax.ws.rs.core.MultivaluedMap
import javax.ws.rs.core.Response

import com.atlassian.jira.issue.Issue
import com.atlassian.jira.issue.IssueManager
import com.atlassian.jira.issue.MutableIssue
import com.atlassian.jira.component.ComponentAccessor
import org.apache.log4j.Level
import org.apache.log4j.Logger

import com.atlassian.applinks.api.ApplicationLinkResponseHandler
import com.atlassian.applinks.api.ApplicationLinkService
import com.atlassian.applinks.api.application.bitbucket.BitbucketApplicationType
import com.atlassian.sal.api.component.ComponentLocator
import com.atlassian.sal.api.net.Request
import com.atlassian.sal.api.net.Response
import com.atlassian.sal.api.net.ResponseException
import com.atlassian.sal.api.net.ResponseHandler

import java.util.Date
import java.text.SimpleDateFormat

def log = Logger.getLogger("com.curvc")
log.setLevel(Level.DEBUG)

@BaseScript CustomEndpointDelegate delegate

repositoryPush(httpMethod: "POST") { MultivaluedMap queryParams, String body ->
    JsonSlurper slurper = new JsonSlurper()
    Map map = slurper.parseText(body)

    //      return
    if(map.changes.type.toString().equals("[ADD]")&&map.changes.ref.type.toString().equals("[BRANCH]")){
```

```

        log.info("Branch Created")
    } else {
        return
    }

    def issueManager = ComponentAccessor.getIssueManager()
    def userManager = ComponentAccessor.getUserManager()
    def commentManager = ComponentAccessor.getCommentManager()

    //
    String projectKey = map.repository.project.key.toString()
    String repostiroySlug = map.repository.slug.toString()
    String repostiroyName = map.repository.name.toString()
    String tempBranchName = map.changes.ref.displayId.toString()
    String branchName = tempBranchName.substring(1, tempBranchName.size() - 1)
    String branchCreator = map.actor.displayName

    //
    String tempDate = map.date.toString().split('\\+')[0]
    Date date = Date.parse("yyyy-MM-dd'T'HH:MM:ss", tempDate)
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy MM dd HH MM")
    String strDate = simpleDateFormat.format(date)
    log.info(": " + projectKey + "/" + repostiroySlug + " : " + repostiroyName + " Create Branch: " +
branchName + " : " + branchCreator)

    // Jira issue
    String userName = "jk.park"
    def user = userManager.getUserByName(userName)
    //
    def message = ""
    : ${repostiroyName}
    : ${branchName}
    : ${branchCreator}
    : ${strDate}
    ""

    // bitbucket Jira Issue
    def applicationLinkService = ComponentLocator.getComponent(ApplicationLinkService)
    final def bitbucketLink = applicationLinkService.getPrimaryApplicationLink(BitbucketApplicationType)
    ComponentAccessor.getJiraAuthenticationContext().setLoggedInUser(user)
    def authenticatedRequestFactory = bitbucketLink.createImpersonatingAuthenticatedRequestFactory():
bitbucketLink.createAuthenticatedRequestFactory()
    authenticatedRequestFactory
        .createRequest(Request.MethodType.GET, "/rest/api/1.0/projects/MAV/repos/testtest/branches?filterText="
+ branchName + "&details=true")
        .addHeader("Content-Type", "application/json")
        .execute(new ResponseHandler<Response>(){
            @Override
            void handle(Response response) throws ResponseException{
                if(response.statusCode != HttpURLConnection.HTTP_OK){
                    throw new Exception(response.responseBodyAsString)
                } else{
                    responseBody = new JsonSlurper().parseText(response.responseBodyAsString)
                }
            }
        })

    log.info(responseBody)
    if(responseBody.size == 0){
        log.info(" .")
        return
    }
    if(responseBody.values.metadata["com.atlassian.bitbucket.server.bitbucket-jira:branch-list-jira-issues"][0].
key.size == 0){
        log.info(" Jira Issue .")
        return
    } else if(responseBody.values.metadata["com.atlassian.bitbucket.server.bitbucket-jira:branch-list-jira-
issues"][0].key.size > 1){
        log.info(" Jira Issue 2 .")
        return
    }
}

```

```
String issueKey = responseBody.values.metadata["com.atlassian.bitbucket.server.bitbucket-jira:branch-list-jira-issues"][0].key[0]
    log.info("issue: " + issueKey)

def issue = issueManager.getIssueByCurrentKey(issueKey)

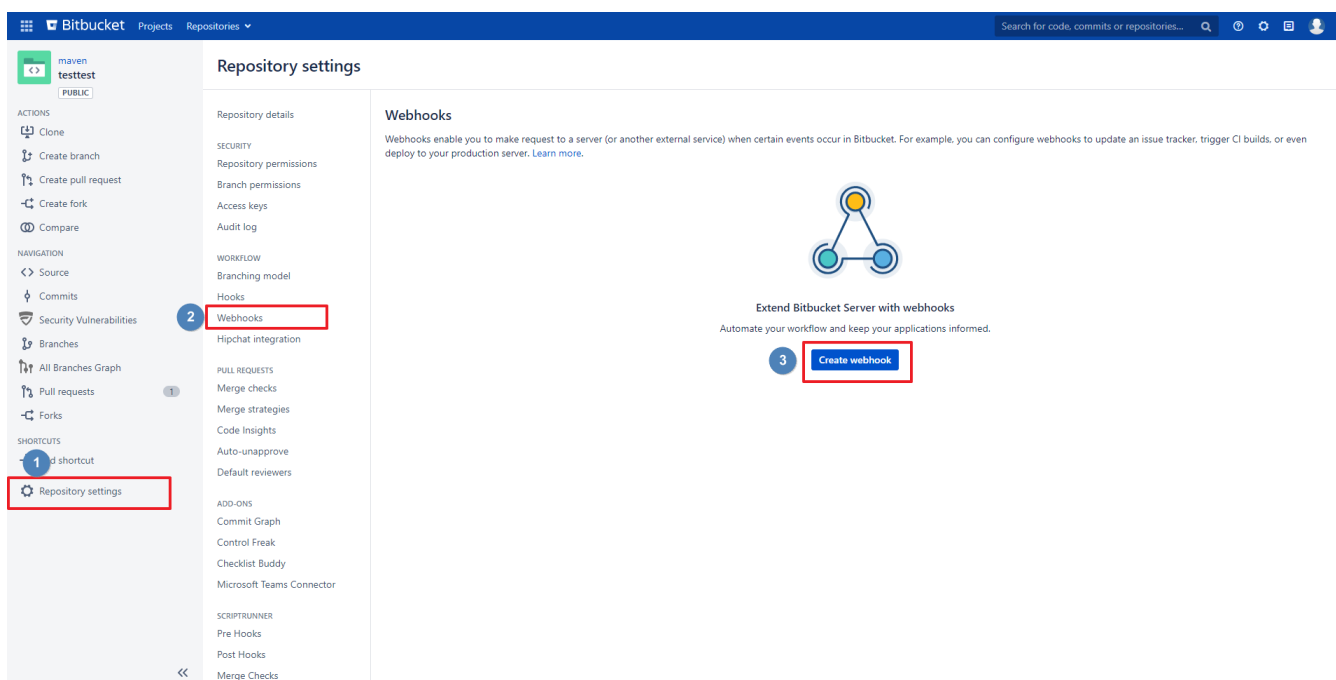
commentManager.create(issue, user, message, false)
log.info("comment Add")

return javax.ws.rs.core.Response.ok(new JsonBuilder([success:"comment Add"]).toString()).build();
}
```

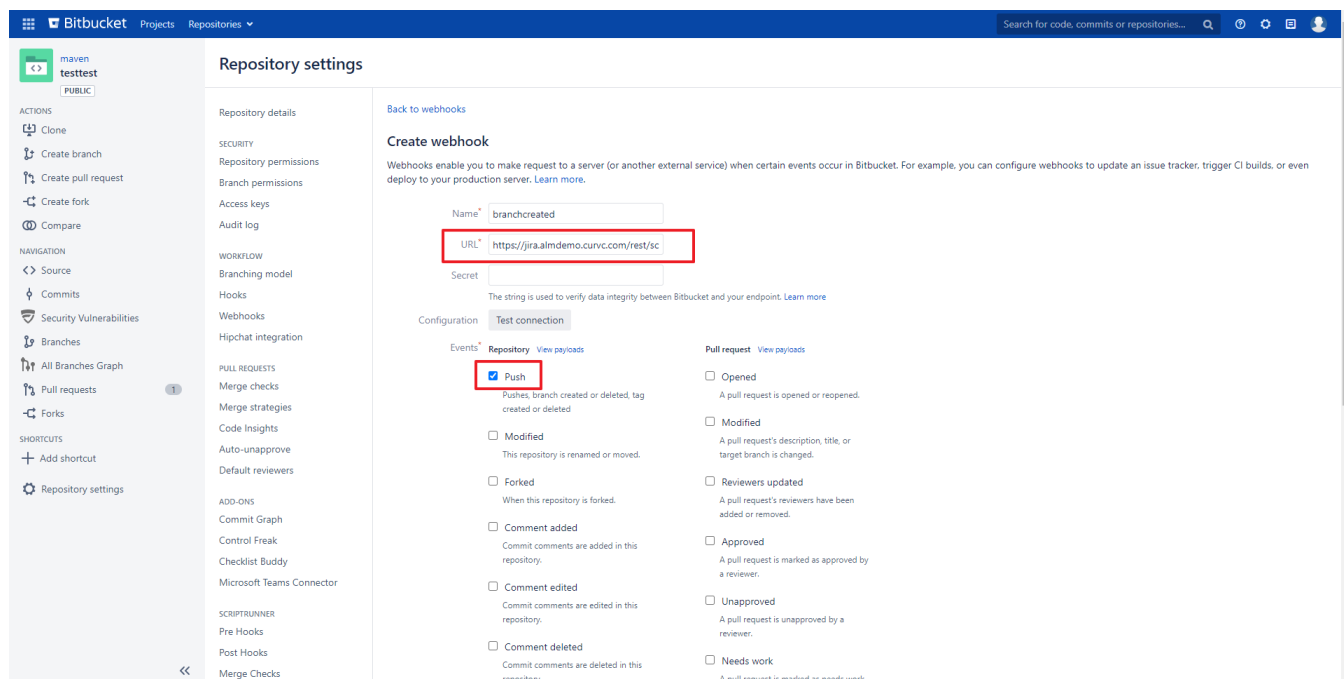
Bitbucket 저장소에서 Webhook 만들기

i Webhook은 저장소 단위로만 줄 수 있다. 프로젝트에서는 줄 수 없다.

- Bitbucket 해당 저장소 → Repository Settings → Webhooks → Create webhooks

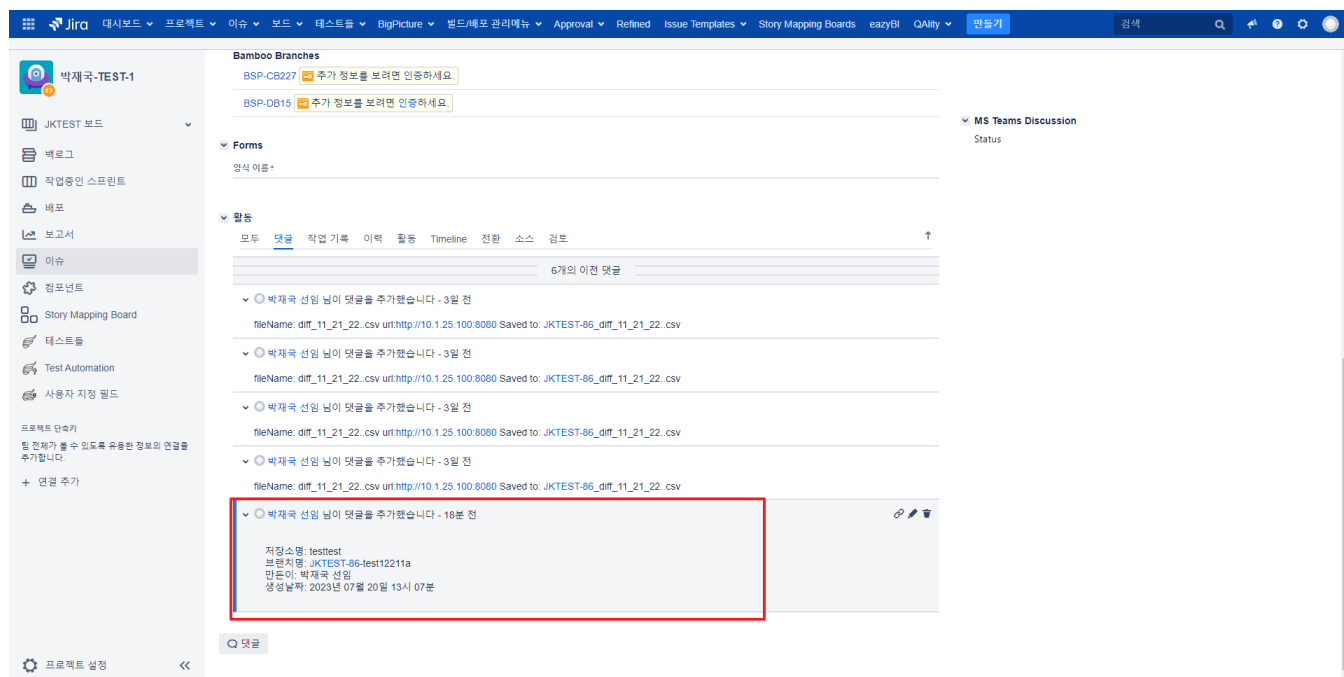


- Url: Jira Scriptrunner Endpoint Url
Events: Push 선택



확인

- Webhook을 설정한 저장소에서 신규로 Branch 생성 시 연동된 Issue에 댓글을 남긴다



참조링크

- bitbucket rest api
- AuthenticatedRequestFactory with BasicAuth

