

Jira - Scriptrunner 활용하여 기존 첨부파일, 댓글 소유주 변경하여 새로 생성하기

이 문서는 Scriptrunner 활용하여 기존 첨부파일(attachment), 댓글(comment) 소유주(creator) 변경하여 새로 생성하기 가이드를 공유하기 위해 작성되었다.

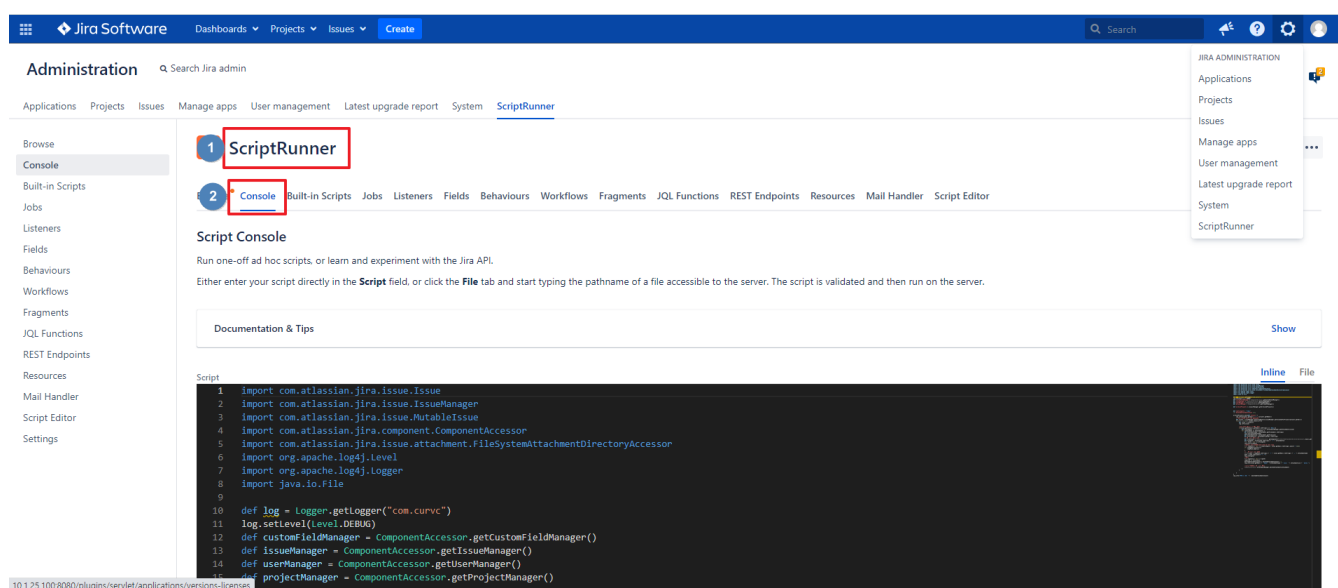
도구명	Jira, scriptrunner
비고	

- [Scriptrunner 활용하여 기존 첨부파일, 댓글 소유주 변경하여 새로 생성하기](#)
 - [ScriptRunner Console 들어가기](#)
 - [프로젝트 지정하여 변경하기](#)
 - [이슈 지정하여 변경하기](#)
- [참조링크](#)

Scriptrunner 활용하여 기존 첨부파일, 댓글 소유주 변경하여 새로 생성하기

ScriptRunner Console 들어가기

- Scriptrunner → Console



프로젝트 지정하여 변경하기

- Console 창에 다음 코드 입력

```

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 com.atlassian.jira.issue.attachment.FileSystemAttachmentDirectoryAccessor
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.atlassian.jira.issue.attachment.CreateAttachmentParamsBean
import java.io.File
import com.atlassian.jira.issue.comments.MutableComment
  
```

```

import com.atlassian.jira.issue.comments.Comment
import com.atlassian.jira.user.ApplicationUser
import com.atlassian.jira.event.type.EventDispatchOption
import com.onresolve.scriptrunner.parameters.annotation.*
import com.atlassian.jira.project.Project

def log = Logger.getLogger("com.curvc")
log.setLevel(Level.DEBUG)
def customFieldManager = ComponentAccessor.getCustomFieldManager()
def issueManager = ComponentAccessor.getIssueManager()
def userManager = ComponentAccessor.getUserManager()
def projectManager = ComponentAccessor.getProjectManager()
def commentManager = ComponentAccessor.getCommentManager()

//
@ProjectPicker(
    label = 'Project', description = 'Pick a project', placeholder = 'Pick a project', includeArchived = false
)
Project project
String projectName = "SERVICE"
//
@UserPicker(label = "Original User", description = "Select a user")
ApplicationUser user
log.info(" : " + user.displayName)
//
@UserPicker(label = "Change User", description = "Select a user")
ApplicationUser changeUser
log.info(" : " + changeUser.displayName)
log.info(project.getId())

// admin user
@UserPicker(label = "Admin User", description = "Select a admin user")
ApplicationUser adminuser

def issues = issueManager.getIssueObjects(issueManager.getIssueIdsForProject(project.getId()))

for(int i = 0; i < issues.size(); i++){
    def issue = issues[i] as MutableIssue
    log.info(issue.getKey())
    if(issue.reporter == user){
        //
        def attachments = ComponentAccessor.attachmentManager.getAttachments(issue)
        log.info(" : " + attachments.size())
        for (attachment in attachments) {
            String type = attachment.getMimetype()
            def attachmentname = attachment.getFilename().toString()
            log.info(attachmentname)
            def attachmentUser = userManager.getUserByKey(attachment.getAuthorKey())
            log.info(attachmentUser)
            //
            String fileName = attachmentname.toString()
            def attachmentid = attachment.getId().toString()
            //
            def filelocate = ComponentAccessor.getComponent(FileSystemAttachmentDirectoryAccessor.class).
getAttachmentDirectory(issue)
            def filepath = filelocate.toString() + '/' + attachmentid
            File oriFile = new File(filepath)
            log.info(filepath)
            //
            def bean = new CreateAttachmentParamsBean.Builder()
                .file(oriFile)
                .filename(fileName)
                .contentType(type)
                .author(changeUser)
                .issue(issue)
                .createTime(attachment.getCreated())
                .build()
            ComponentAccessor.attachmentManager.createAttachment(bean)
            log.info("create attachment " + fileName)
            //Jira
            ComponentAccessor.attachmentManager.deleteAttachment(attachment)
        }
    }
}

```

```

        log.info("attahment: " + attachmentname + " recreated")
    }
    //
    def comments = commentManager.getCommentsForUser(issue, user)
    log.info(issue.getKey() + " : " + comments.size())
    for (int j = 0; j < comments.size(); j++){
        def mutableComment = commentManager.getMutableComment(comments[j].getId())
        log.info(mutableComment.getAuthorFullName())
        if(mutableComment.getAuthorApplicationUser().getKey() == user.getKey()){
            commentManager.create(issue, changeUser, changeUser, mutableComment.getBody(), mutableComment.
getGroupLevel(), mutableComment.getRoleLevelId(), mutableComment.getCreated(), mutableComment.getUpdated(),
false)

            commentManager.delete(mutableComment)
            log.info("comment recreated")

        }
    }
    //
    issueManager.updateIssue(adminuser, issue, EventDispatchOption.ISSUE_UPDATED, false)
}
}

```

- Project: 변경할 프로젝트 선택
- Original User: 기존 유저 선택
- Change User: 변경할 유저 선택
- Admin User: 관리자 유저 선택

The screenshot shows the Jira Script Console interface. The 'Documentation & Tips' section is highlighted with a red box, showing configuration fields for Project, Original User, Change User, and Admin User. The Project field is set to '서비스 운영', Original User to 'admin admin', Change User to 'jk park', and Admin User to 'jk park'. Below this, a script is visible in the 'Script' field.

이슈 지정하여 변경하기

- Console창에 다음 코드 입력

```

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 com.atlassian.jira.issue.attachment.FileSystemAttachmentDirectoryAccessor
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.atlassian.jira.issue.attachment.CreateAttachmentParamsBean
import java.io.File
import com.atlassian.jira.issue.comments.MutableComment
import com.atlassian.jira.issue.comments.Comment
import com.atlassian.jira.user.ApplicationUser

```

```

import com.atlassian.jira.event.type.EventDispatchOption
import com.onresolve.scriptrunner.parameters.annotation.*
import com.atlassian.jira.project.Project

def log = Logger.getLogger("com.curvc")
log.setLevel(Level.DEBUG)
def customFieldManager = ComponentAccessor.getCustomFieldManager()
def issueManager = ComponentAccessor.getIssueManager()
def userManager = ComponentAccessor.getUserManager()
def projectManager = ComponentAccessor.getProjectManager()
def commentManager = ComponentAccessor.getCommentManager()

//
@ShortTextInput(label = "Issue", description = "Enter a issueKey")
String issueKey

//
@UserPicker(label = "Original User", description = "Select a user")
ApplicationUser user
log.info(" : " + user.displayName)
//
@UserPicker(label = "Change User", description = "Select a Chnage user")
ApplicationUser changeUser
log.info(" : " + changeUser.displayName)

// admin user
@UserPicker(label = "Admin User", description = "Select a admin user")
ApplicationUser adminuser

def issue = issueManager.getIssueByCurrentKey(issueKey)
if(issue instanceof MutableIssue){
    issue as MutableIssue
} else {
    log.info(" issueKey Issue .")
    return
}

log.info(issue.getKey())
if(issue.reporter == user){
    //
    issue.setReporter(changeUser)
    //
    def attachments = ComponentAccessor.attachmentManager.getAttachments(issue)
    log.info(" : " + attachments.size())
    for (attachment in attachments) {
        String type = attachment.getMimetype()
        def attachmentname = attachment.getFilename().toString()
        log.info(attachmentname)
        def attachmentUser = userManager.getUserByKey(attachment.getAuthorKey())
        log.info(attachmentUser)
        //
        String fileName = attachmentname.toString()
        def attachmentid = attachment.getId().toString()
        //
        def filelocate = ComponentAccessor.getComponent(FileSystemAttachmentDirectoryAccessor.class).
getAttachmentDirectory(issue)
        def filepath = filelocate.toString() + '/' + attachmentid
        File oriFile = new File(filepath)
        log.info(filepath)
        //
        def bean = new CreateAttachmentParamsBean.Builder()
            .file(oriFile)
            .filename(fileName)
            .contentType(type)
            .author(changeUser)
            .issue(issue)
            .createdTime(attachment.getCreated())
            .build()
        ComponentAccessor.attachmentManager.createAttachment(bean)
        log.info("create attachment " + fileName)
    }
}
//Jira

```

```

        ComponentAccessor.attachmentManager.deleteAttachment(attachment)
        log.info("attachment: " + attachmentname + " recreated")
    }
    //
    def comments = commentManager.getCommentsForUser(issue, user)
    log.info(issue.getKey() + " : " + comments.size())
    for (int i = 0; i < comments.size(); i++){
        def mutableComment = commentManager.getMutableComment(comments[i].getId())
        log.info(mutableComment.getAuthorFullName())
        if(mutableComment.getAuthorApplicationUser().getKey() == user.getKey()){
            commentManager.create(issue, changeUser, changeUser, mutableComment.getBody(), mutableComment.
getGroupLevel(), mutableComment.getRoleLevelId(), mutableComment.getCreated(), mutableComment.getUpdated(),
false)

            commentManager.delete(mutableComment)
            log.info("comment recreated")
        }
    }
    //
    issueManager.updateIssue(adminuser, issue, EventDispatchOption.ISSUE_UPDATED, false)
}

```

- Issue: 변경 할 IssueKey 입력
Original User: 기존 유저 선택
Change User: 변경할 유저 선택
Admin User: 관리자 유저 선택

Fields
Behaviours
Workflows
Fragments
JQL Functions
REST Endpoints
Resources
Mail Handler
Script Editor
Settings

Script Editor

Run one-off ad hoc scripts, or learn and experiment with the Jira API.

Either enter your script directly in the **Script** field, or click the **File** tab and start typing the pathname of a file accessible to the server. The script is validated and then run on the server.

Documentation & Tips Show

Issue

SERVICE-112

Enter a IssueKey

Original User

admin admin

Select a user

Change User

jk park

Select a Change user

Admin User

jk park

Select a admin user

Script

```

60 // 첨부파일 위치 지정
61 def filelocate = ComponentAccessor.getComponent(FileSystemAttachmentDirectoryAccessor.class).getAttachmentDirectory(issue)
62 def filepath = filelocate.toString() + '/' + attachmentid
63 File oriFile = new File(filepath)
64 log.info(filepath)
65 // 기존첨부파일은 삭제 등록
66 def bean = new CreateAttachmentParamsBean.Builder()
67     .file(oriFile)
68     .filename(fileName)
69     .contentType(type)
70     .author(changeUser)
71     .issue(issue)
72     .createTime(attachment.getCreated())
73     .build()
74 ComponentAccessor.attachmentManager.createAttachment(bean)
75 log.info("create attachment " + fileName)

```

Inline File

참조링크

- [Dynamic Forms](#)