Jira ScriptRunner 활용하여 첨부파일 백업 및 삭제하기

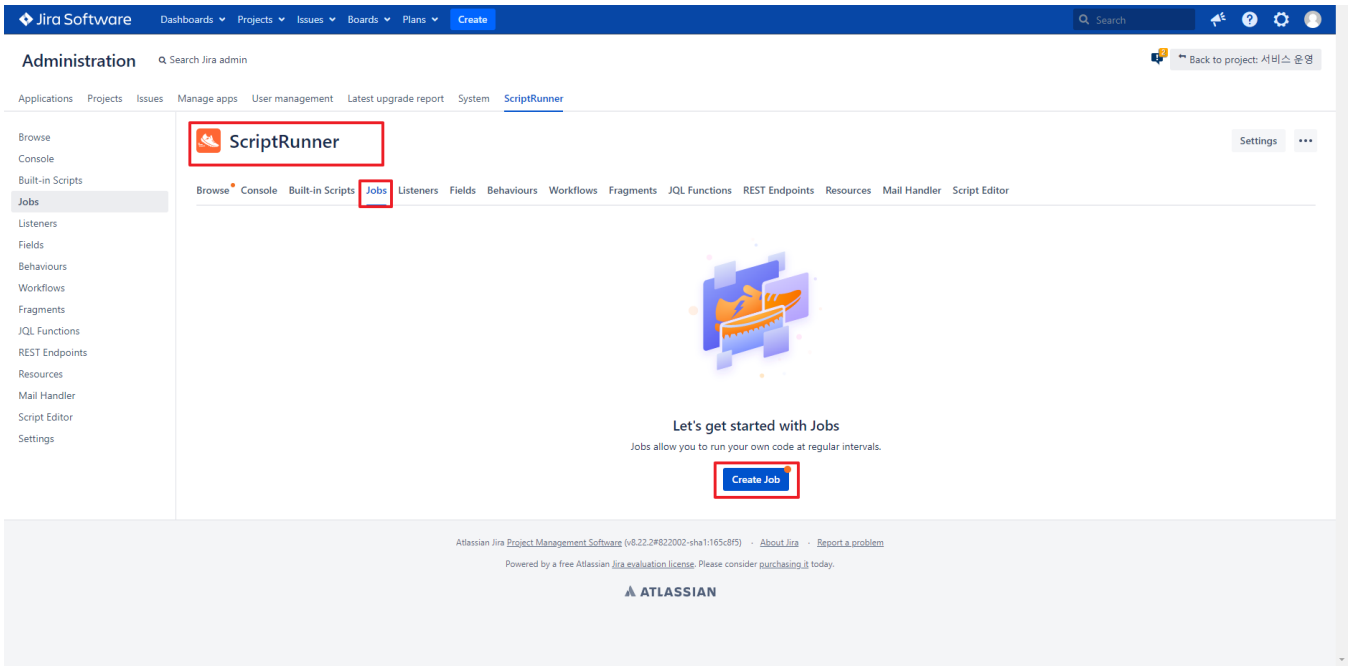
이 문서는 Jira ScriptRunner 활용하여 첨부파일 백업 및 삭제하기 가이드를 공유하기 위해 작성되었다.

도구명	Jira, Scriptrunner
버전	

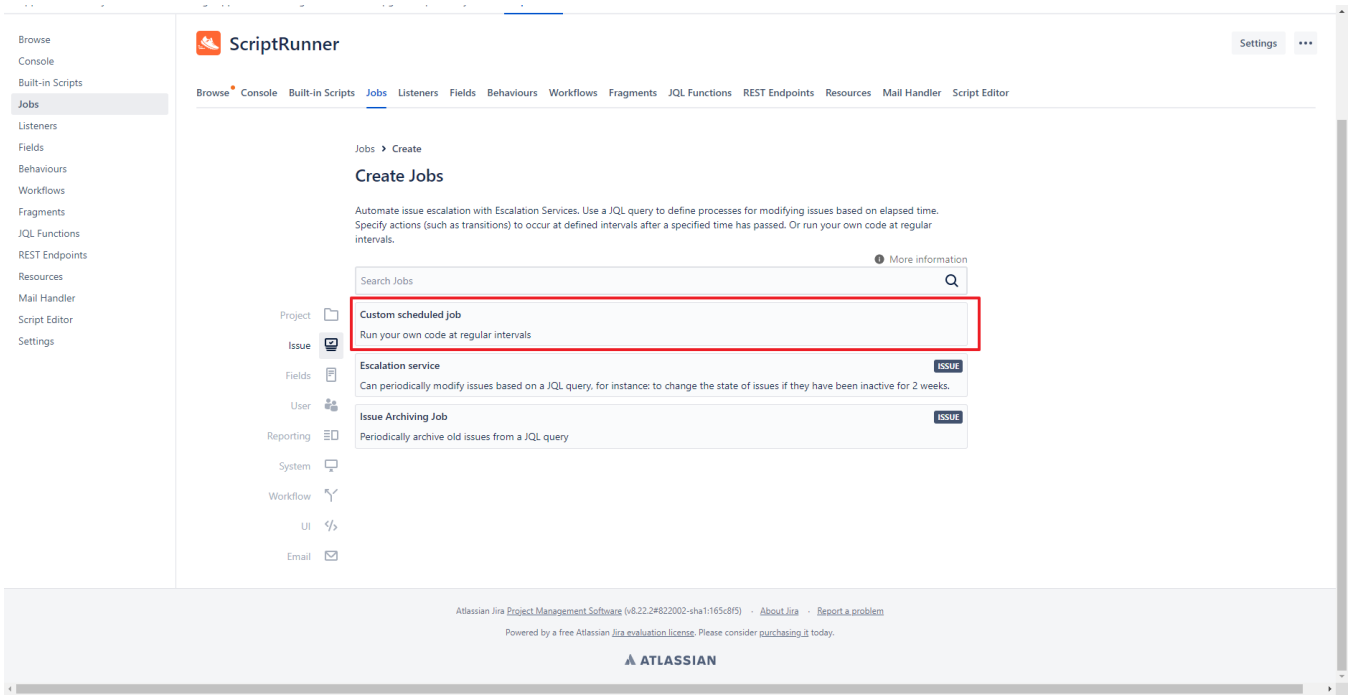
ScriptRunner 활용하여 첨부파일 백업 및 삭제하기

Scheduled Job 만들기

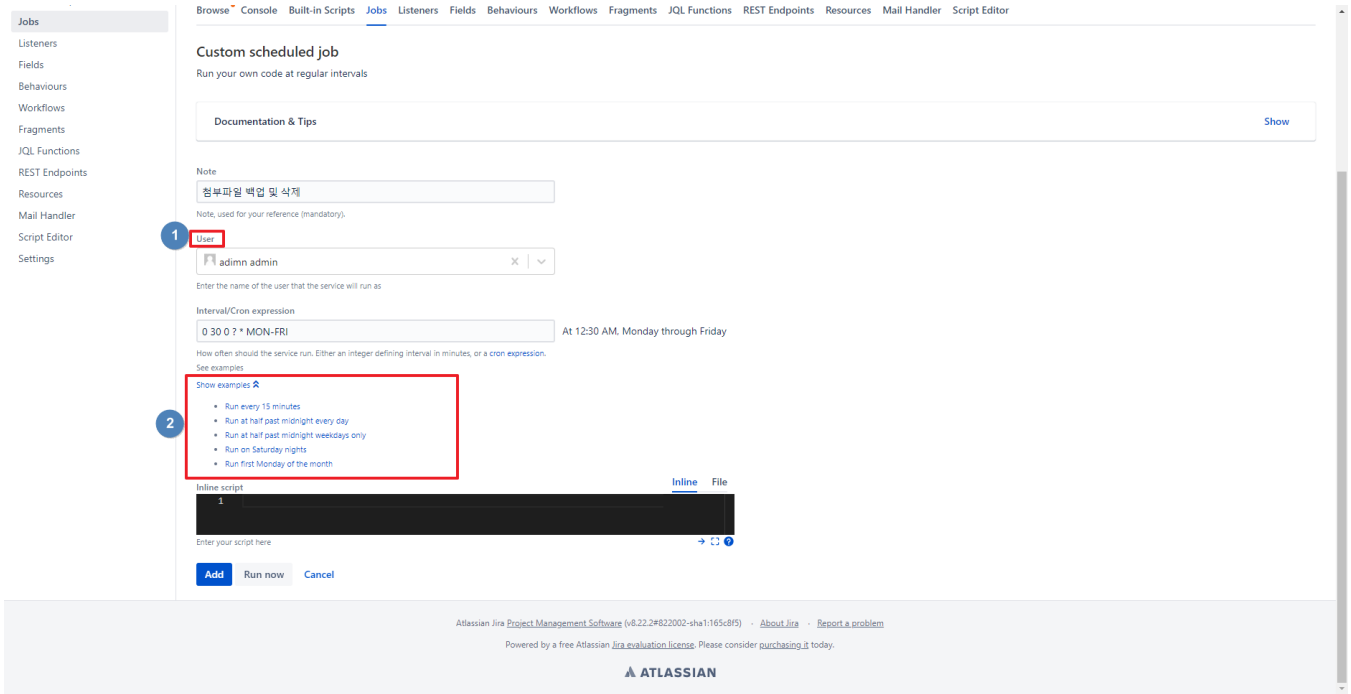
- ScriptRunner → Jobs → Create Job



- Custom Scheduled Job 선택



- 1: 작업을 실행할 유저
  2: Cron 형식 예제
  알맞은 값 선택 및 수정



Script 내용

- 해당 스크립트 실행 시 정의한 JQL문에서 나온 이슈들의 첨부파일을 특정 폴더로 이동 후 삭제한다.

- Script 주요내용
  20번 줄: JQL을 실행할 유저 정의
  25번 줄: JQL문
  29번 줄: 첨부파일을 이동시킬 위치(Jira를 실행하고 있는 linux user의 해당 폴더 Read, Write 권한 필요)

```
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.bc.issue.search.SearchService
import com.atlassian.jira.jql.parser.JqlQueryParser
import com.atlassian.jira.web.bean.PagerFilter
import com.onresove.scriptrunner.runner.util.UserMessageUtil
import com.atlassian.jira.issue.attachment.FileSystemAttachmentDirectoryAccessor
import org.apache.log4j.Level
import org.apache.log4j.Logger
import java.io.File

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

def user = userManager.getUserByName("jql  ")
def jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser)
def searchService = ComponentAccessor.getComponent(SearchService)

//  JQL
def qstr = ''
def query = jqlQueryParser.parseQuery(qstr)
def search = searchService.search(user, query, PagerFilter.getUnlimitedFilter())
//
def newfilepath = "/tmp/"
//
int deletedAttachmentsCount = 0;
for(int i = 0;i < search.total;i++){
    def issue = issueManager.getIssueObject(search.results[i].key)
    log.info(issue)
    def attachments = ComponentAccessor.attachmentManager.getAttachments(issue)
    for (attachment in attachments) {
        def attachmentname = attachment.getFilename().toString()
        log.info(attachmentname)
        def attachmentsize = attachment.getFilesize()
        def attachmentid = attachment.getId().toString()
        //
        def filelocate = ComponentAccessor.getComponent(FileSystemAttachmentDirectoryAccessor.class).
getAttachmentDirectory(issue)
        def filepath = filelocate.toString() + '/' + attachmentid
        File file = new File(filepath)
        log.info(filepath)
        //
        File tempPath = new File(newfilepath + issue.getKey().toString().split('-')[0])
        if (!tempPath.exists()){
            tempPath.mkdirs()
        }
        //
        String path = tempPath.toString() + '/' + issue.getKey().toString() + '_' + attachmentname
        //
        File copyfile = new File(path)
        copyfile << file.bytes
        deletedAttachmentsCount = deletedAttachmentsCount + 1
        log.info(issue.getKey() + " Name:" + attachmentname + " Size: " + attachmentsize + " bytes ")

        //Jira
        ComponentAccessor.attachmentManager.deleteAttachment(attachment)
    }
}
log.info(" : " +  deletedAttachmentsCount)
```

참조링크

- Atlassian Community
- Atlassian Community 2
- Atlassian Community 3
- Groovy web console