


Jira - ScriptRunner 활용하여 Worflow Action 시 HTML 메일 보내기

이 문서는 Jira Script Runner 활용하여 HTML 메일 보내기 가이드를 공유하기 위해 작성되었다.

도구명	Jira, Script Runner
비고	

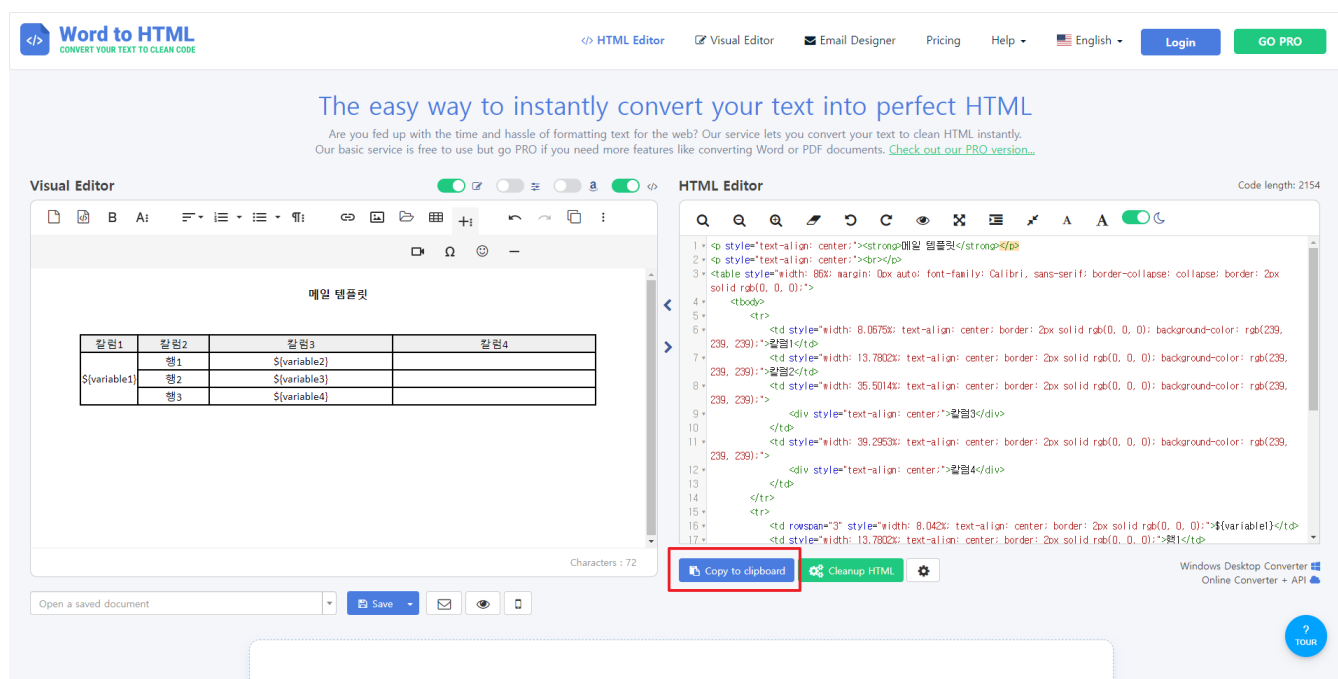
- Workflow Action 시 HTML 메일 보내기
 - 메일 템플릿 만들기
 - Script Runner Workflow Action 만들기
- Script 내용
 - Html 전체 글자수가 65535 이하인 경우
 - Html 전체 글자수가 65535 이상인 경우
- 참조 링크

 메일 서버 설정이 되어있어야 메일 보내기가 가능

Workflow Action 시 HTML 메일 보내기

메일 템플릿 만들기

- <https://wordtohtml.net/> 접속 후 원하는 형식의 메일 템플릿을 작성 후 HTML Editor 부분을 따로 저장한다.
- 변수를 사용하려면 해당 변수자리에 \${} 을 넣는다.



The screenshot shows the Word to HTML converter interface. The 'Visual Editor' on the left displays a table with 4 columns and 3 rows. The 'HTML Editor' on the right shows the corresponding HTML code. A red box highlights the 'Copy to clipboard' button at the bottom of the HTML Editor.

Script Runner Workflow Action 만들기

- 시스템 → 앱 관리 → Script Runner → Workflow → Create Workflow functions 선택

관리 Q Jira 관리도구 검색

응용프로그램 프로젝트 1 **앱 관리** 사용자 관리 최신 업그레이드 보고서 시스템 Configuration Manager

2 **ScriptRunner** Settings ...

Browse Console Built-in Scripts Jobs Listeners Fields Behaviours **Workflows** Fragments JQL Functions REST Endpoints Resources Mail Handler Script Editor

Workflows
Enhance and automate your workflows beyond Jira's native possibilities using ScriptRunner's workflow functions: script conditions, script validators, and script post functions.

Documentation & Tips Show

Search Applied to

Name	Applied to	Workflow	Transition	Type	Performance	History	Actions
Custom script post-function	IN	(INTERPARK) Change Request (Draft)	요청서 작성 → 요청서 승인 → 요청서 승인	Post function	100%	Has not run yet	...
Custom script post-function (Disabled)	IN	(INTERPARK) Change Request (Draft)	요청서 작성 → 요청서 승인 → 요청서 승인	Post function	100%	Has not run yet	...
Custom script post-function	IN	(INTERPARK) Change Request (Draft)	기술상실/작업 계획 수립 → 작업계획 승인 → 작업계획 승인	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G) Sub_Task Workflow (Draft)	필리츠 승인 완료 → 티켓 완료 → 완료	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G)Project Workflow (Draft)	필리츠 승인 → 필리츠 승인 완료 → 필리츠 승인 완료	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G)Project Workflow (Draft)	변경승인 완료 → 필리츠 승인 → 필리츠 승인	Post function	100%	Has not run yet	...

3 **Create Workflow Function**

- 해당하는 Workflow와 Transition 선택 후 Validator 선택 후 Create

관리 Q Jira 관리도구 검색

응용프로그램 프로젝트 이슈 **앱 관리** 사용자 관리 최신 업그레이드 보고서 시스템 Configuration Manager

3 **ScriptRunner** Settings ...

Browse Console Built-in Scripts Jobs Listeners Fields Behaviours **Workflows** Fragments JQL Functions REST Endpoints Resources Mail Handler Script Editor

Workflows
Enhance and automate your workflows beyond Jira's native possibilities using ScriptRunner's workflow functions: script conditions, script validators, and script post functions.

Documentation & Tips Show

Search Applied to

Create Workflow function

Workflows
(KT&G)Project Workflow

Transitions
변경승인완료 (변경완료/변경승인 요청 → 변경승인완료)

Workflow function type
Select

Condition

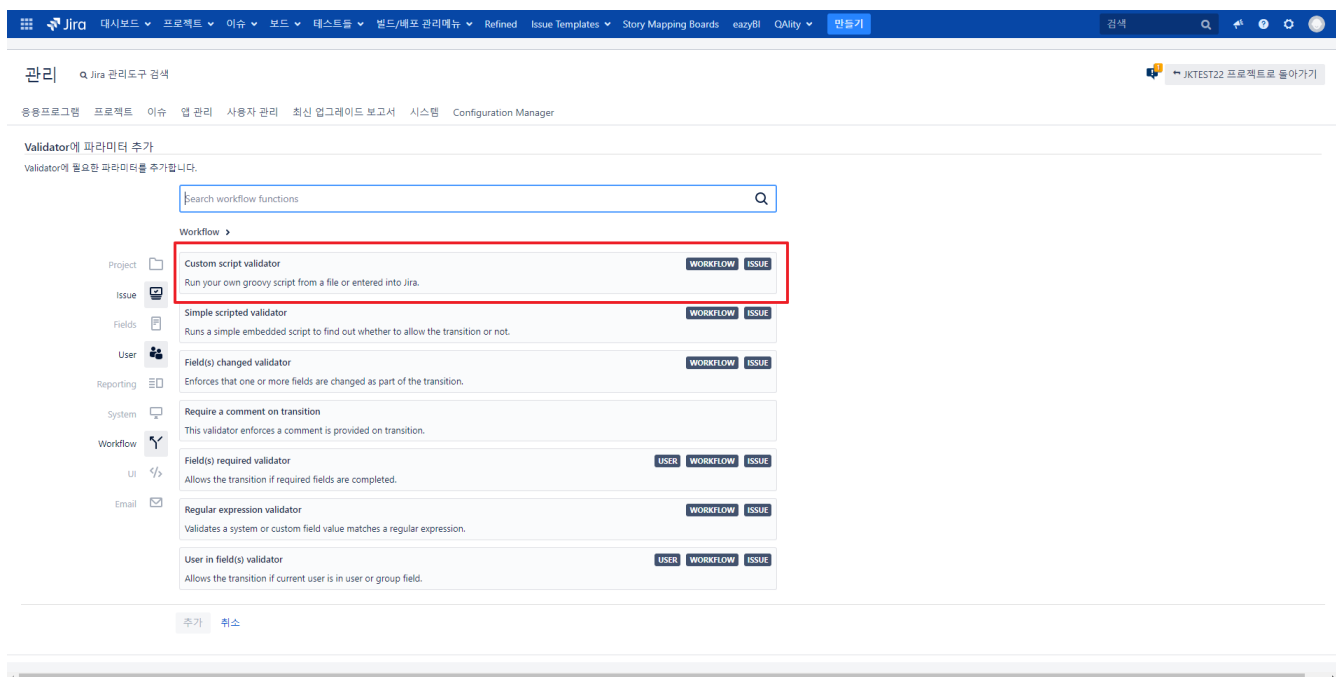
Validator

Post function

2 **Create** Cancel

Name	Applied to	Workflow	Transition	Type	Performance	History	Actions
Custom script post-function	IN	(INTERPARK) Change Request (Draft)	요청서 작성 → 요청서 승인 → 요청서 승인	Post function	100%	Has not run yet	...
Custom script post-function (Disabled)	IN	(INTERPARK) Change Request (Draft)	요청서 작성 → 요청서 승인 → 요청서 승인	Post function	100%	Has not run yet	...
Custom script post-function	IN	(INTERPARK) Change Request (Draft)	기술상실/작업 계획 수립 → 작업계획 승인 → 작업계획 승인	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G) Sub_Task Workflow (Draft)	필리츠 승인 완료 → 티켓 완료 → 완료	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G)Project Workflow (Draft)	필리츠 승인 → 필리츠 승인 완료 → 필리츠 승인 완료	Post function	100%	Has not run yet	...
Custom script post-function	KT	(KT&G)Project Workflow (Draft)	변경승인 완료 → 필리츠 승인 → 필리츠 승인	Post function	100%	Has not run yet	...

- Custom script validator 선택



Script 내용

Html 전체 글자수가 65535 이하인 경우

```
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.mail.Email
import com.atlassian.jira.mail.settings.MailSettings
import com.atlassian.mail.MailException
import com.atlassian.mail.server.SMTPMailServer
import com.atlassian.plugin.util.ContextClassLoaderSwitchingUtil
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.opensymphony.workflow.InvalidInputException

def issueManager = ComponentAccessor.getIssueManager()
// Summary
def summaryfield = issue.getSummary()
//
def variable1 = "1 "
def variable2 = "1 "
def variable3 = "2 "
def variable4 = "3 "

// HTML
def body = ""
<p style="text-align: center;"><strong> </strong></p>
<p style="text-align: center;"><br></p>
<table style="width: 86%; margin: 0px auto; font-family: Calibri, sans-serif; border-collapse: collapse;
border: 2px solid rgb(0, 0, 0);">
  <tbody>
    <tr>
      <td style="width: 8.0675%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color: rgb(
(239, 239, 239);">1</td>
      <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">2</td>
      <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">
        <div style="text-align: center;">3</div>
      </td>
      <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">
        <div style="text-align: center;">4</div>
    </tr>
  </tbody>
</table>
```

```

        </td>
    </tr>
    <tr>
        <td rowspan="3" style="width: 8.042%; text-align: center; border: 2px solid rgb(0, 0, 0);"
>${variable1}</td>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">1</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable2}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
    <tr>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">2</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable3}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
    <tr>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">3</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable4}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
</tbody>
</table>
<p><br></p> ""

// ( )
def userManager = ComponentAccessor.getUserManager()
def user = issue.getAssignee()

// ,
if(user == null) {
    user = issue.getReporter()
}
def receiver = user.getEmailAddress()

// , ,
sendEmail(receiver.toString(), summary ,body)

//
String sendEmail(String emailAddr, String subject, String body) {
    def logger = Logger.getLogger(getClass())
    logger.setLevel(Level.DEBUG)

    // Stop emails being sent if the outgoing mail server gets disabled (useful if you start a script sending
    emails and need to stop it)
    def mailSettings = ComponentAccessor.getComponent(MailSettings)
    if (mailSettings?.send()?.disabled) {
        logger.debug(" .")
        throw new InvalidInputException(" .")
    }

    def mailServer = ComponentAccessor.mailServerManager.defaultSMTPMailServer
    if (!mailServer) {
        logger.debug(" .")
        throw new InvalidInputException(" .")
    }

    def email = new Email(emailAddr)
    email.setMimeType('text/html')
    email.setSubject(subject)
    email.setBody(body)
    try {
        // This is needed to avoid the exception about IMAPProvider
        ContextClassLoaderSwitchingUtil.runInContext(SMTPMailServer.classLoader) {
            mailServer.send(email)
        }
        logger.debug('Mail sent')
    } catch (MailException e) {
        logger.debug("Send mail failed with error: ${e.message}")
        throw new InvalidInputException(" .")
    }
}

```

}

Html 전체 글자수가 65535 이상인 경우

- HTML 내용이 큰 경우 위의 방법으로 진행 시 에러가 발생

```
Enter your script here
The template expects your script to return: a String, or an Object that .toString() will be called on

The script could not be compiled:
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
Script120.groovy: 8: String too long. The given string is 69388 Unicode code units long, but only a maximum of 65535 is allowed.
@ line 8, column 12.
    def body = '''
        ^

1 error
```

- 방법 1: 에러를 방지하기 위해 해당 HTML을 파일로 저장 후 사용 - 파일에서 읽을 때 변수가 치환되지 않아 replace를 사용하여 변수를 치환

```
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.mail.Email
import com.atlassian.jira.mail.settings.MailSettings
import com.atlassian.mail.MailException
import com.atlassian.mail.server.SMTPMailServer
import com.atlassian.plugin.util.ContextClassLoaderSwitchingUtil
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.opensymphony.workflow.InvalidInputException

def issueManager = ComponentAccessor.getIssueManager()
// Summary
def summaryfield = issue.getSummary()
//
def variable1 = "1 "
def variable2 = "1 "
def variable3 = "2 "
def variable4 = "3 "

// HTML
def body = new File('/atlassian/application-data/jira-home/scripts/jktest.groovy').text
// replace
body = body.replace('${variable1}', "1 ")
body = body.replace('${variable2}', "1 ")
body = body.replace('${variable3}', "2 ")
body = body.replace('${variable4}', "3 ")

// ( )
def userManager = ComponentAccessor.getUserManager()
def user = issue.getAssignee()

// ,
if(user == null) {
    user = issue.getReporter()
}
def receiver = user.getEmailAddress()

// , ,
sendEmail(receiver.toString(), summary ,body)

//
String sendEmail(String emailAddr, String subject, String body) {
    def logger = Logger.getLogger(getClass())
    logger.setLevel(Level.DEBUG)

    // Stop emails being sent if the outgoing mail server gets disabled (useful if you start a script sending
    emails and need to stop it)
```

```

def mailSettings = ComponentAccessor.getComponent(MailSettings)
if (mailSettings?.send()?.disabled) {
    logger.debug("    .")
    throw new InvalidInputException("    .")
}

def mailServer = ComponentAccessor.mailServerManager.defaultSMTPMailServer
if (!mailServer) {
    logger.debug("    .")
    throw new InvalidInputException("    .")
}

def email = new Email(emailAddr)
email.setMimeType('text/html')
email.setSubject(subject)
email.setBody(body)
try {
    // This is needed to avoid the exception about IMAPProvider
    ContextClassLoaderSwitchingUtil.runInContext(SMTPMailServer.classLoader) {
        mailServer.send(email)
    }
    logger.debug('Mail sent')
} catch (MailException e) {
    logger.debug("Send mail failed with error: ${e.message}")
    throw new InvalidInputException("    .")
}
}

```

▪ 방법 2: body를 분할하여 저장

```

import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.mail.Email
import com.atlassian.jira.mail.settings.MailSettings
import com.atlassian.mail.MailException
import com.atlassian.mail.server.SMTPMailServer
import com.atlassian.plugin.util.ContextClassLoaderSwitchingUtil
import org.apache.log4j.Level
import org.apache.log4j.Logger
import com.opensymphony.workflow.InvalidInputException

def issueManager = ComponentAccessor.getIssueManager()
// Summary
def summaryfield = issue.getSummary()
//
def variable1 = "1 "
def variable2 = "1 "
def variable3 = "2 "
def variable4 = "3 "

// HTML
def body1 = ""
<p style="text-align: center;"><strong> </strong></p>
<p style="text-align: center;"><br></p>
<table style="width: 86%; margin: 0px auto; font-family: Calibri, sans-serif; border-collapse: collapse;
border: 2px solid rgb(0, 0, 0);">
    <tbody>
        <tr>
            <td style="width: 8.0675%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color: rgb
(239, 239, 239);">1</td>
            <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">2</td>
            <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">
                <div style="text-align: center;">3</div>
            </td>
            <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0); background-color:
rgb(239, 239, 239);">
                <div style="text-align: center;">4</div>

```

```

        </td>
    </tr>
    """
def body2 = """
    <tr>
        <td rowspan="3" style="width: 8.042%; text-align: center; border: 2px solid rgb(0, 0, 0);"
>${variable1}</td>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">1</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable2}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
    <tr>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">2</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable3}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
    <tr>
        <td style="width: 13.7802%; text-align: center; border: 2px solid rgb(0, 0, 0);">3</td>
        <td style="width: 35.5014%; text-align: center; border: 2px solid rgb(0, 0, 0);">${variable4}</td>
        <td style="width: 39.2953%; text-align: center; border: 2px solid rgb(0, 0, 0);"><br></td>
    </tr>
</tbody>
</table>
<p><br></p>
    """
def body = body1 + body2

// ( )
def userManager = ComponentAccessor.getUserManager()
def user = issue.getAssignee()

// ,
if(user == null) {
    user = issue.getReporter()
}
def receiver = user.getEmailAddress()

// , ,
sendEmail(receiver.toString(), summary ,body)

//
String sendEmail(String emailAddr, String subject, String body) {
    def logger = Logger.getLogger(getClass())
    logger.setLevel(Level.DEBUG)

    // Stop emails being sent if the outgoing mail server gets disabled (useful if you start a script sending
    emails and need to stop it)
    def mailSettings = ComponentAccessor.getComponent(MailSettings)
    if (mailSettings?.send()?.disabled) {
        logger.debug(" .")
        throw new InvalidInputException(" .")
    }

    def mailServer = ComponentAccessor.mailServerManager.defaultSMTPMailServer
    if (!mailServer) {
        logger.debug(" .")
        throw new InvalidInputException(" .")
    }

    def email = new Email(emailAddr)
    email.setMimeType('text/html')
    email.setSubject(subject)
    email.setBody(body)
    try {
        // This is needed to avoid the exception about IMAPProvider
        ContextClassLoaderSwitchingUtil.runInContext(SMTPMailServer.classLoader) {
            mailServer.send(email)
        }
    }
    logger.debug('Mail sent')
}

```

```
    } catch (MailException e) {  
        logger.debug("Send mail failed with error: ${e.message}")  
        throw new InvalidInputException("  .")  
    }  
}
```

참조 링크

- [Send Custom Email](#)
- [HTML 만들기](#)