

# Crowd Powershell 이용하여 Excel로 사용자 생성 및 그룹 추가

이 문서는 Crowd Powershell 활용하여 Excel로 사용자 생성 및 그룹 추가 가이드를 공유하기 위해 작성되었다.

도구명	Crowd, Powershell
비고	

- [Powershell 이용하여 Excel로 사용자 생성 및 그룹 추가](#)
  - [설정](#)
  - [Excel에 정보 넣기](#)
  - [Powershell Script](#)
- [참조 링크](#)

## Powershell 이용하여 Excel로 사용자 생성 및 그룹 추가

### 설정

- Application → Crowd 선택

The screenshot shows the Atlassian Crowd web interface. The top navigation bar includes 'Crowd', 'Applications', 'Users', 'Groups', 'Directories', and 'Audit log'. The 'Applications' tab is active. On the left, there's a sidebar with 'Search applications' and 'Add application'. The main area displays a table of applications. The 'crowd' application is highlighted with a red box and a red circle with the number 2 next to it. The table columns are Name, Description, and Status. The status for all listed applications is 'ACTIVE'.

Name	Description	Status
bamboo	bamboo	ACTIVE
bitbucket		ACTIVE
confluence		ACTIVE
crowd	Crowd console	ACTIVE
google-apps	Google Applications Connector	ACTIVE
jira		ACTIVE

At the bottom of the page, it says 'Powered by Atlassian Crowd Version: 4.4.1 (Build#1688 - 2022-03-02) 28255e25-4664-4a36-8a7b-bf1693ab6379'. There are links for 'Report a bug', 'Request a feature', 'About', and 'Contact Atlassian'. The Atlassian logo is also present.

- Application Name, Password 사용

Crowd Applications Users Groups Directories Audit log

Search applications

Add application

**crowd**

Details Directories & groups Administrators Users Permissions Remote addresses Authentication test Options

Change the details of your application here.

Name crowd

The unique name that the application will use to authenticate against the Crowd framework as a client.

Description Crowd console

A short description of the application. Often a URL is helpful.

Application type Crowd

Active

Conception 11 5월 2022, 20:38:55

Last modified 24 10월 2022, 09:07:25

Password \*\*\*\*

Change password

To set a new password, enter the password and confirm.

Confirm password \*\*\*\*

Save Cancel

Powered by Atlassian Crowd Version: 4.4.1 (Build#1688 - 2022-03-02) 28255e25-4664-4a36-8a7b-bf1693ab63f9

Report a bug Request a feature About Contact Atlassian

ATLASSIAN

- Remote address 선택 후 Script를 실행 할 서버 IP 입력

Crowd Applications Users Groups Directories Audit log

Search applications

Add application

**crowd**

Details Directories & groups Administrators Users Permissions Remote addresses Authentication test Options

Specify the IP address(es) of each application that should be allowed to authenticate to Crowd. For an application to be able to use Crowd, the application's address must be valid.

Address	Action
10.1.25.100	Remove
10.0.100.18	Remove
10.0.100.113	Remove
<input type="text"/>	Add

Powered by Atlassian Crowd Version: 4.4.1 (Build#1688 - 2022-03-02) 28255e25-4664-4a36-8a7b-bf1693ab63f9

Report a bug Request a feature About Contact Atlassian

ATLASSIAN

## Excel에 정보 넣기

- 해당 형식으로 생성(첨부파일). 각 행마다 그룹 칸에 값이 없을 때까지 그룹 추가됨

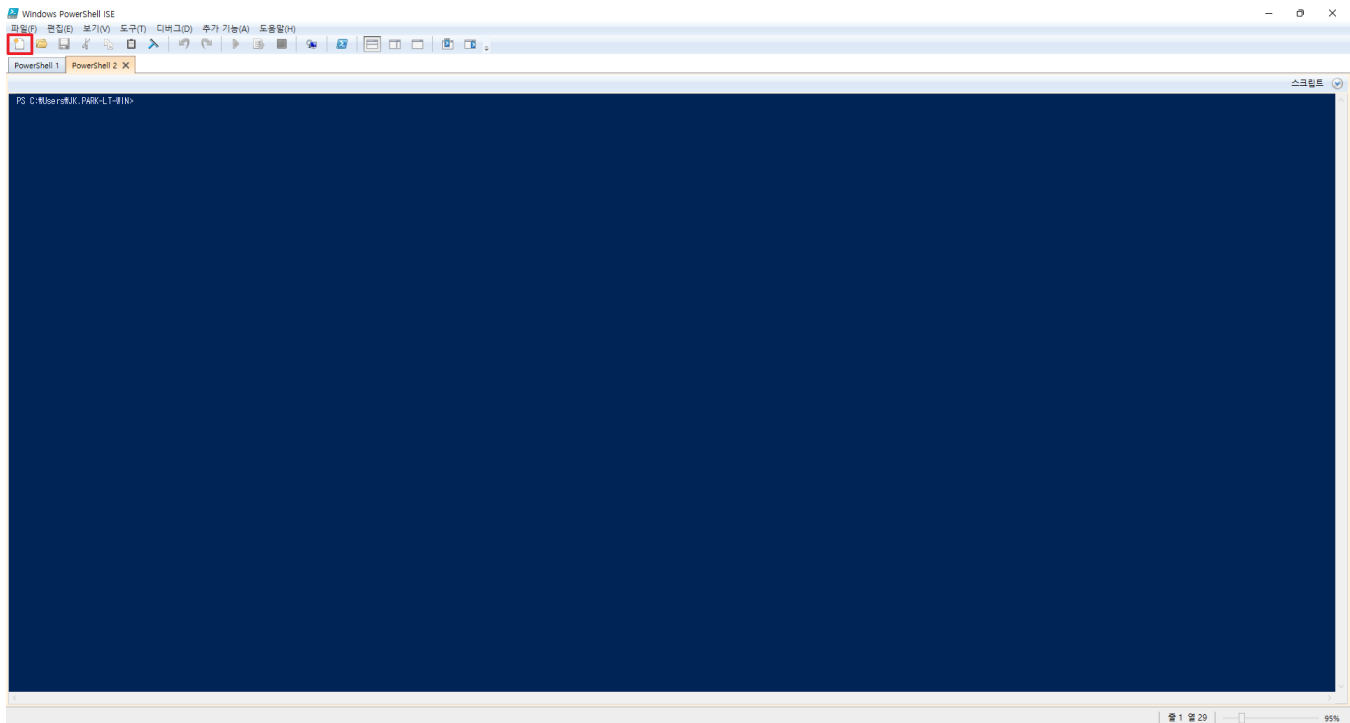
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	name	password	first-name	last-name	display-name	email	group1	group2	...	group...										
2	test1	test123	test1	test1	test1	test1@test.com	stash-users	jira-software-users	Confluence-users											
3	test2	test123	test2	test2	test2	test2@test.com	stash-users	jira-software-users	Confluence-users											
4	test3	test123	test3	test3	test3	test3@test.com	stash-users	jira-software-users	Confluence-users	Confluence-administrators										
5	test4	test123	test4	test4	test4	test4@test.com	stash-users	jira-software-users	Confluence-users											
6	test5	test123	test5	test5	test5	test5@test.com	stash-users	jira-software-users	Confluence-users											
7	한글테스트	test123	한	글	한글테스트	test6@test.com	stash-users	jira-software-users	Confluence-users											
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				
31																				
32																				



Crowd 유저... 추가.xlsx

## Powershell Script

- 윈도우 키 → Powershell ISE 실행 → 새로 만들기



- 2행에 Crowd 주소  
6,7 행에 crowd application ID, 및 Password  
Script 실행 후 해당 Excel 파일 선택 가능

```
#Url api
$base_url = "http://10.1.25.100:8095"
$user_api = "/crowd/rest/usermanagement/1/user"

# crowd application
$user = "crowd"
$password = "crowd"

$LogArray = @()
# SSL , SSL X
Add-Type @"
using System.Net;
using System.Security.Cryptography.X509Certificates;
public class TrustAllCertsPolicy : ICertificatePolicy {
    public bool CheckValidationResult(
        ServicePoint srvPoint, X509Certificate certificate,
        WebRequest request, int certificateProblem){
        return true;
    }
}
"@

[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls, [Net.SecurityProtocolType]::Tls11, [Net.SecurityProtocolType]::Tls12, [Net.SecurityProtocolType]::Tls13, [Net.SecurityProtocolType]::Ssl3

#
function Get-BasicAuthCreds {
    param([string]$Username, [string]$Password)
    $AuthString = "{0}:{1}" -f $Username,$Password
    $AuthBytes = [System.Text.Encoding]::ASCII.GetBytes($AuthString)
    return [Convert]::ToBase64String($AuthBytes)
}
$BasicCreds = Get-BasicAuthCreds -Username $user -Password $password;

#
$FileBrowser = New-Object System.Windows.Forms.OpenFileDialog
$FileBrowser.Title = ""
$FileBrowser.Filter = "xlsx (*.xlsx)| *.xlsx"
[void]$FileBrowser.ShowDialog()
$FilePath = $FileBrowser.FileName.ToString()
$Excel = New-Object -ComObject Excel.Application
$Excel.Visible = $false
$WorkBook = $Excel.Workbooks.add($FilePath)
$sheet = $WorkBook.worksheets.Item(1)

#
$row = 2
$usernumber = 0
# , 1

while($Sheet.Cells.Item($row,1).Text -ne ""){
    # ,
    $groupcol = 7
    #
    $Log = ""
    #
    $adduser = $Sheet.Cells.Item($row,1).Text
    $pass = $Sheet.Cells.Item($row,2).Text
    $first_name = $Sheet.Cells.Item($row,3).Text
    $last_name = $Sheet.Cells.Item($row,4).Text
    $display_name = $Sheet.Cells.Item($row,5).Text
    $email = $Sheet.Cells.Item($row,6).Text

    #
    try{
        $user_check = Invoke-WebRequest -Uri "${base_url}${user_api}?username=${adduser}" -Method Get -
        ContentType 'application/json' -Headers @{"Authorization"="Basic $BasicCreds"}
    }
}
```

```

    } catch {
        $user_check = $null
    }
    # user
    if($user_check -eq $null){
        $Body1 = '{"name":"' + ${adduser} + '", "password":{"value":"' + ${pass} + '"}, "active": true, "first-name": "' + ${first_name} + '", "last-name": "' + ${last_name} + '", "display-name": "' + ${display_name} + '", "email": "' + ${email} + '"}'
        #
        $Body = [System.Text.Encoding]::UTF8.GetBytes($Body1)
        $user_add = Invoke-WebRequest -Uri "${base_url}${user_api}.json" -Method Post -ContentType 'application/json' -Headers @{"Authorization"="Basic $BasicCreds"} -Body $Body
        $result = [System.Text.Encoding]::UTF8.GetString($user_add.RawContentStream.ToArray()) | ConvertFrom-Json
        $username = $result.name
        $Log += "Username: ${username} . "

        #Group
        while($Sheet.Cells.Item($row,$groupcol).Text -ne ""){
            $group = $Sheet.Cells.Item($row,$groupcol).Text
            $group_list = "/crowd/rest/usermanagement/1/group.json?groupname=${group}"
            $group_check = $null
            try{
                $group_check = Invoke-WebRequest -Uri "${base_url}${group_list}" -Method Get -ContentType 'application/json' -Headers @{"Authorization"="Basic $BasicCreds"}
            } catch{
                $group_check = $null
                $Log += "${group} . "
            }
            $group_api = "/crowd/rest/usermanagement/1/group/user/direct.json?groupname=${group}"
            #Group
            if($group_check -ne $null){
                try{
                    $form1 = '{"name":"' + ${adduser} + '"}'
                    $form = [System.Text.Encoding]::UTF8.GetBytes($form1)
                    $group_add = Invoke-WebRequest -Uri "${base_url}${group_api}" -Method Post -ContentType 'application/json' -Headers @{"Authorization"="Basic $BasicCreds"} -Body $form
                    $Log += "${group} . "
                } catch{
                    $Log += "${group} . "
                }
            }
            $groupcol++
        }
        $LogArray += $Log
        $usernumber++
    } else {
        $LogArray += "${adduser} is already exist"
    }
    $row++
}
$row -= 2
$LogArray += ": ${row} : ${usernumber}"
$date = Get-Date -UFormat %Y%m%d%H%M
$LogArray > .\${date}_CrowdAdd.log

```

## 참조 링크

- [Crowd REST API Reference](#)