AppPerfect

JIRACloud Bot-AppPerfect Corporation -Readme

Version 1.0

September 04,2019

Table of Contents

1. Introduction

This document contains all essential information for the user to make full use of the Bot or Digital worker. This manual includes a description of the functions and capabilities and step-by-step procedures for setup & configuration of the Bot.

Overview

This Bot reads parameters from a CSV Template based on Filter Conditions & other configurable parameters (as explained below) and creates project and creates issue. It retrieves details of all the projects and if Project Id is given then details of specific project retrieves. It also retrieves details of all the issues and if Issue Id is provided then only details of specific issue.

Detailed steps are as follows:

Create Project/Create Issue

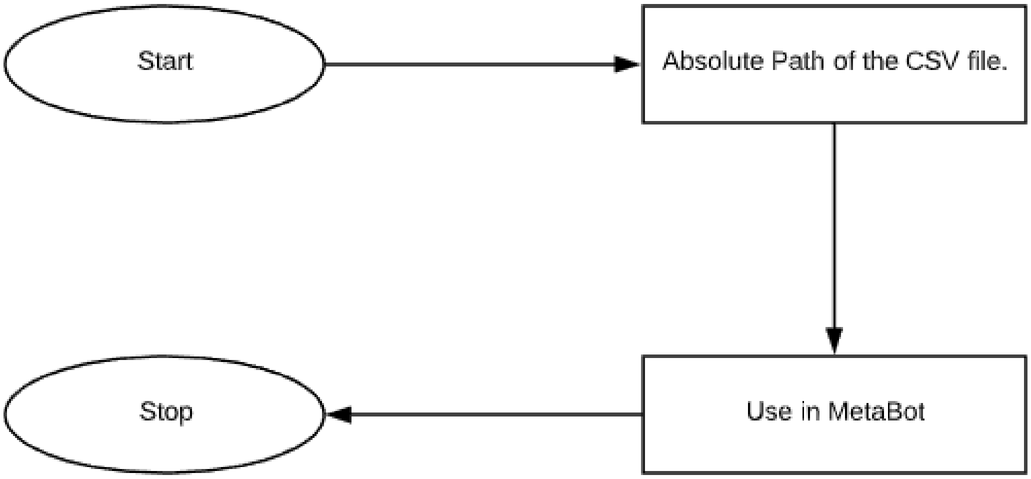
1) User provides the path of CSV file where all input parameters have been present.
2) In csv file, user can export data from the CSV Template based on Filter conditions set by a user through a Configuration File.

   Filter conditions are as below (Click individual to read details)

   - **Filter based on Column Value**

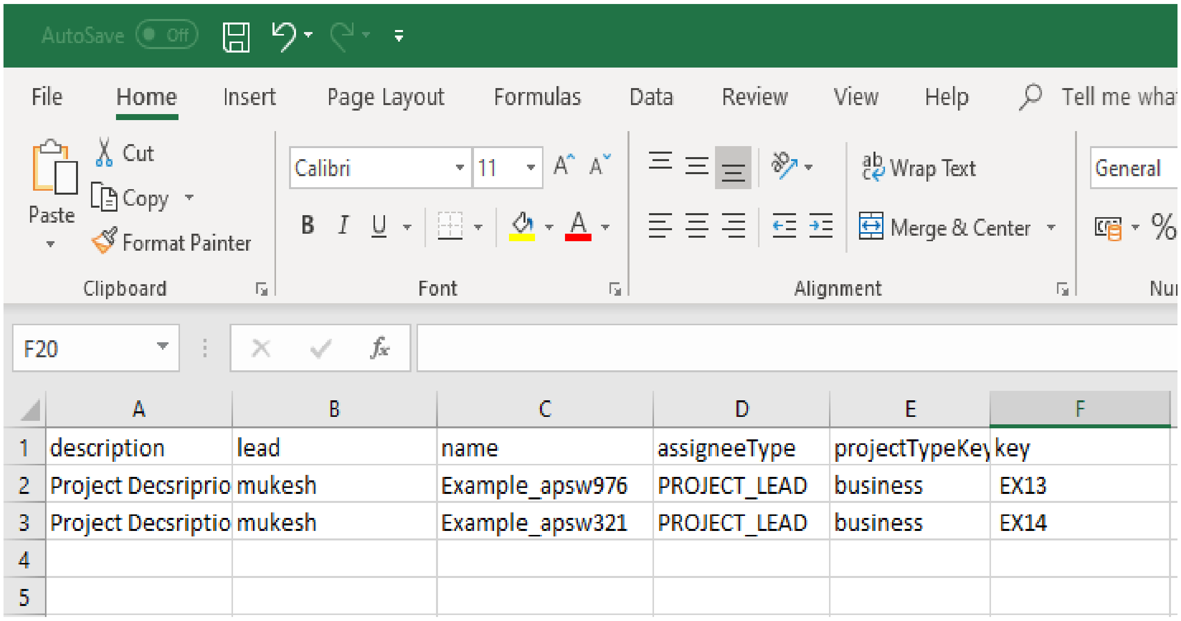
3) This returns generated Project URL and Project Id and key.

- **Visual Flow –**



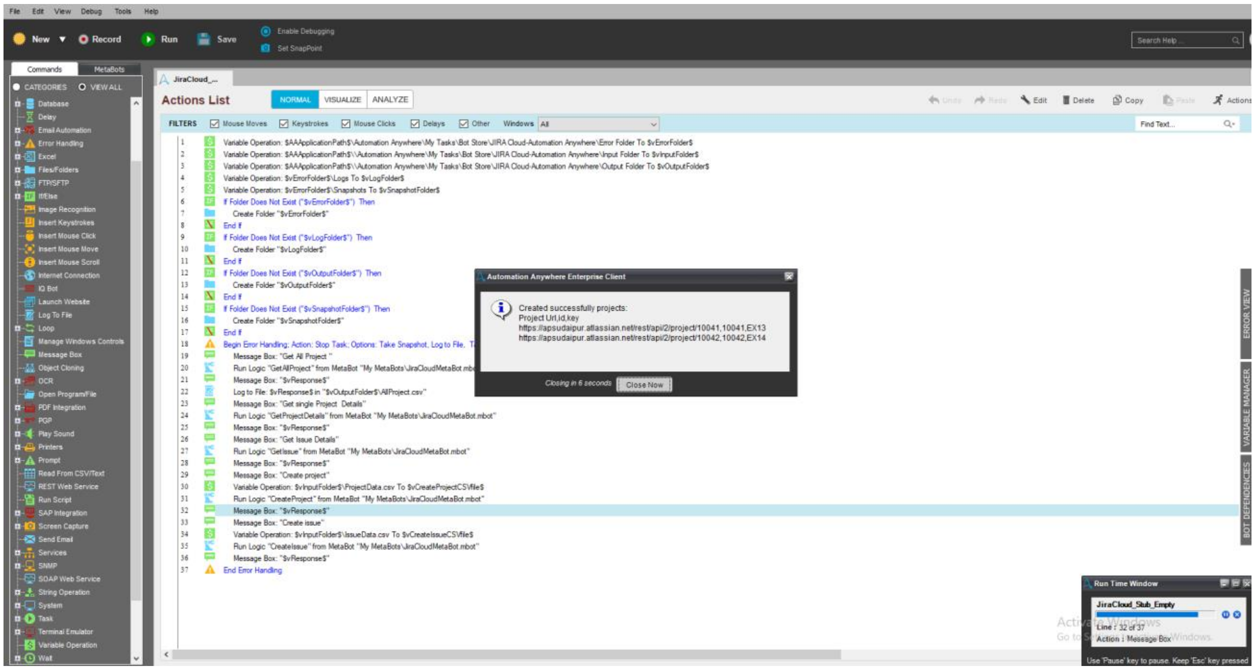
- **Input and Output Example –**

Input: ProjectData.csv



	A	B	C	D	E	F
1	description	lead	name	assigneeType	projectTypeKey	key
2	Project Decsriprio	mukesh	Example_apsw976	PROJECT_LEAD	business	EX13
3	Project Decsriptio	mukesh	Example_apsw321	PROJECT_LEAD	business	EX14
4						
5						

Output: Message box:



```
Variable Operation: $AAApplicationPath$\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Error Folder To $vErrorFolder$
Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Input Folder To $vInputFolder$
Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Output Folder To $vOutputFolder$
Variable Operation: $vErrorFolder$\Logs To $vLogFolder$
Variable Operation: $vErrorFolder$\Snapshots To $vSnapshotFolder$
If Folder Does Not Exist ("$vErrorFolder$") Then
    Create Folder "$vErrorFolder$"
End If
If Folder Does Not Exist ("$vLogFolder$") Then
    Create Folder "$vLogFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$") Then
    Create Folder "$vOutputFolder$"
End If
If Folder Does Not Exist ("$vSnapshotFolder$") Then
    Create Folder "$vSnapshotFolder$"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File,  T
    Message Box: "Get All Project "
    Run Logic "GetAllProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbo
    Message Box: "$vResponse$"
    Log to File: $vResponse$ in "$vOutputFolder$\AllProject.csv"
    Message Box: "Get single Project  Details"
    Run Logic "GetProjectDetails" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Get Issue Details"
    Run Logic "GetIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Create project"
    Variable Operation: $vInputFolder$\ProjectData.csv To $vCreateProjectCSVfile$
    Run Logic "CreateProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Create issue"
    Variable Operation: $vInputFolder$\IssueData.csv To $vCreateIssueCSVfile$
    Run Logic "CreateIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
End Error Handling
```

Automation Anywhere Enterprise Client

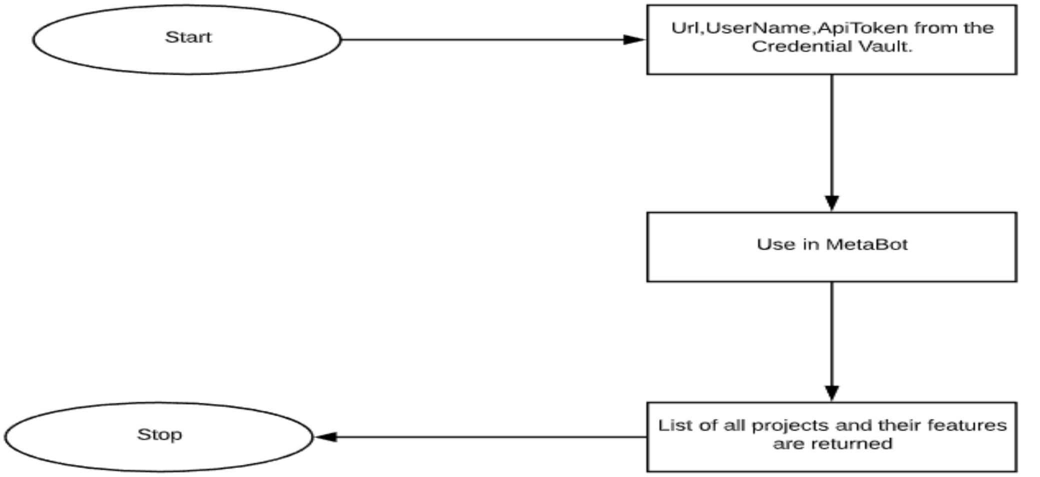
```
Created successfully projects:
Project Url,id,key
https://apsudaipur.atlassian.net/rest/api/2/project/10041,10041,EX13
https://apsudaipur.atlassian.net/rest/api/2/project/10042,10042,EX14
```

Closing in 6 seconds Close Now

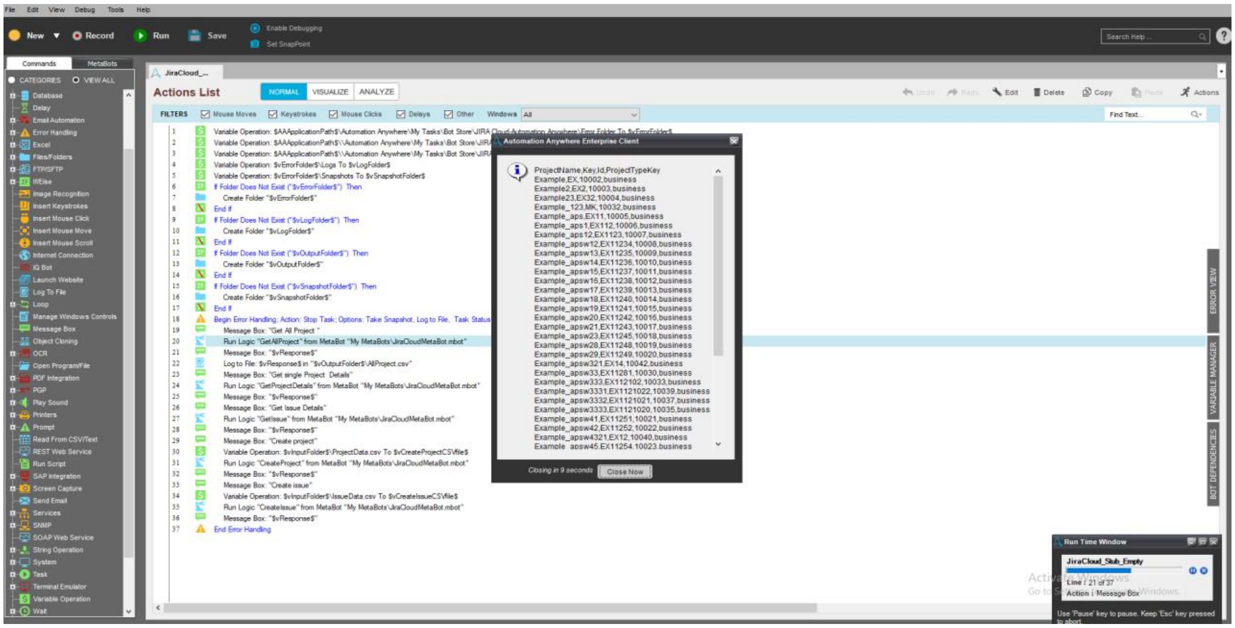
Get All Projects

1) User provides the Project Url, username, Api Token which is present in the Credential vault.
2) This will return all project information as string in the task bot.

- **Visual Flow –**


Start
Url,UserName,ApiToken from the Credential Vault.
Use in MetaBot
List of all projects and their features are returned
Stop

- **Output: Message Box**



FILTERS ☑ Mouse Moves ☑ Keystrokes ☑ Mouse Clicks ☑ Delays ☑ Other Windows All

```
Variable Operation: $AAApplicationPath$\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Error Folder To $vErrorFolder$
Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRA
Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRA
Variable Operation: $vErrorFolder$\Logs To $vLogFolder$
Variable Operation: $vErrorFolder$\Snapshots To $vSnapshotFolder$
If Folder Does Not Exist ("$vErrorFolder$") Then
    Create Folder "$vErrorFolder$"
End If
If Folder Does Not Exist ("$vLogFolder$") Then
    Create Folder "$vLogFolder$"
End If
If Folder Does Not Exist ("$vOutputFolder$") Then
    Create Folder "$vOutputFolder$"
End If
If Folder Does Not Exist ("$vSnapshotFolder$") Then
    Create Folder "$vSnapshotFolder$"
End If
Begin Error Handling; Action: Stop Task; Options: Take Snapshot, Log to File, Task Status
    Message Box: "Get All Project "
    Run Logic "GetAllProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Log to File: $vResponse$ in "$vOutputFolder$\AllProject.csv"
    Message Box: "Get single Project  Details"
    Run Logic "GetProjectDetails" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Get Issue Details"
    Run Logic "GetIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Create project"
    Variable Operation: $vInputFolder$\ProjectData.csv To $vCreateProjectCSVfile$
    Run Logic "CreateProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
    Message Box: "Create issue"
    Variable Operation: $vInputFolder$\IssueData.csv To $vCreateIssueCSVfile$
    Run Logic "CreateIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
End Error Handling
```

Automation Anywhere Enterprise Client

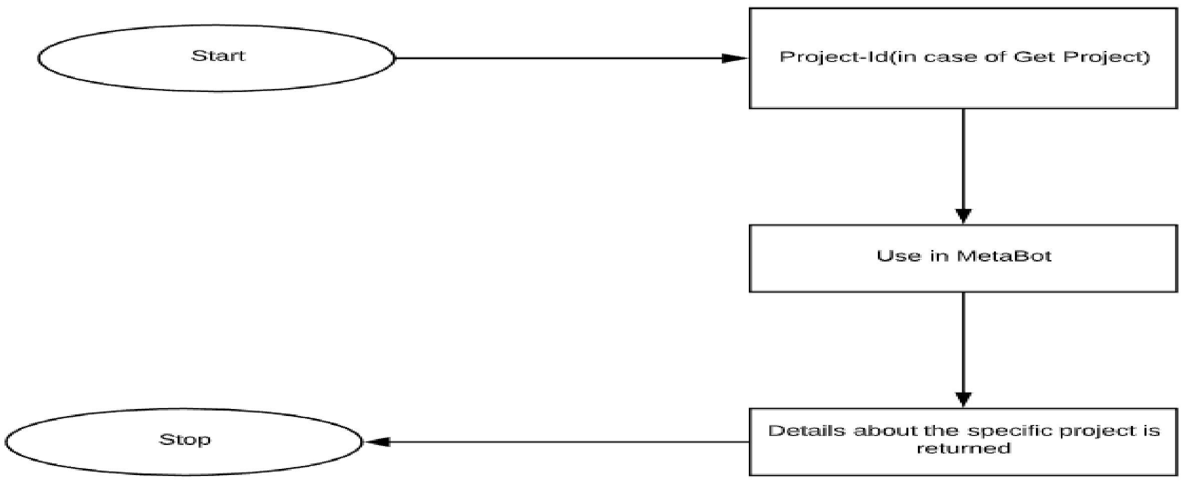
```
ProjectName,Key,Id,ProjectTypeKey
Example,EX,10002,business
Example2,EX2,10003,business
Example23,EX32,10004,business
Example_123,MK,10032,business
Example_aps,EX11,10005,business
Example_aps1,EX112,10006,business
Example_aps12,EX1123,10007,business
Example_apsw12,EX11234,10008,business
Example_apsw13,EX11235,10009,business
Example_apsw14,EX11236,10010,business
Example_apsw15,EX11237,10011,business
Example_apsw16,EX11238,10012,business
Example_apsw17,EX11239,10013,business
Example_apsw18,EX11240,10014,business
Example_apsw19,EX11241,10015,business
Example_apsw20,EX11242,10016,business
Example_apsw21,EX11243,10017,business
Example_apsw23,EX11245,10018,business
Example_apsw28,EX11248,10019,business
Example_apsw29,EX11249,10020,business
Example_apsw321,EX14,10042,business
Example_apsw33,EX11281,10030,business
Example_apsw333,EX112102,10033,business
Example_apsw3331,EX1121022,10039,business
Example_apsw3332,EX1121021,10037,business
Example_apsw3333,EX1121020,10035,business
Example_apsw41,EX11251,10021,business
Example_apsw42,EX11252,10022,business
Example_apsw4321,EX12,10040,business
Example_apsw45,EX11254,10023,business
```

Closing in 9 seconds Close Now

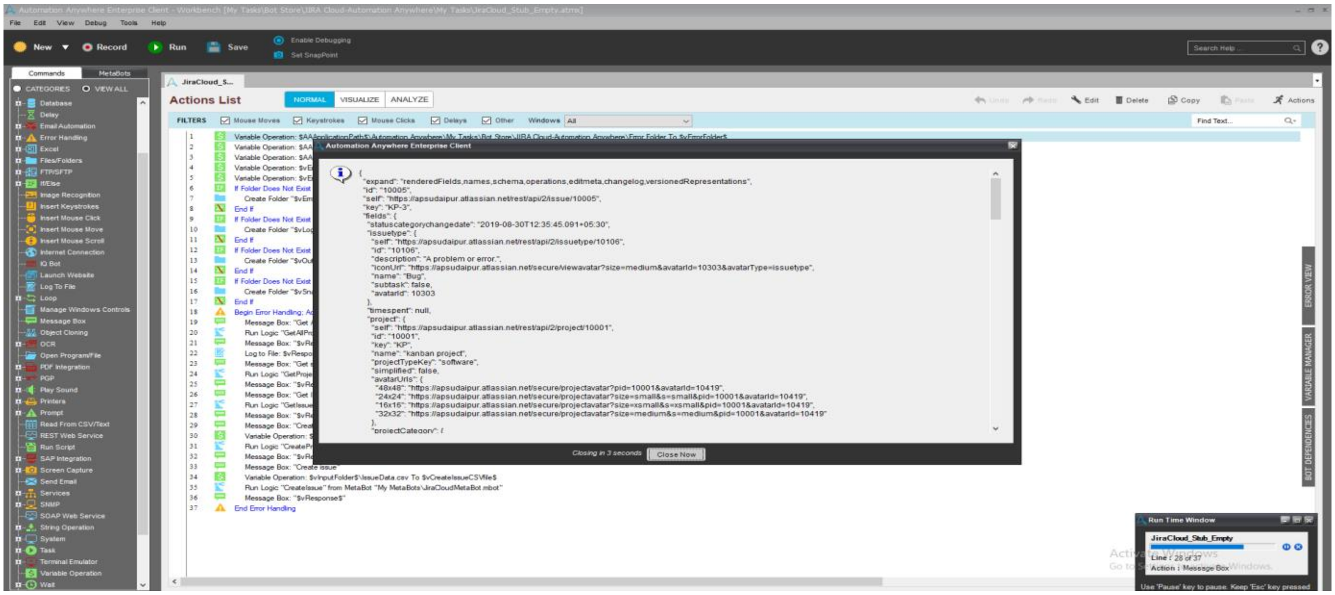
Get Project Detail/ Issue Detail

1) User provides the ProjectId (in case of Get Project Detail) or IssueId (in case of Get IssueDetail).
2) This will return details about specific project in the task bot.

- **Visual Flow –**


Start
Project-Id(in case of Get Project)
Use in MetaBot
Details about the specific project is returned
Stop

- **Output : Message Box**

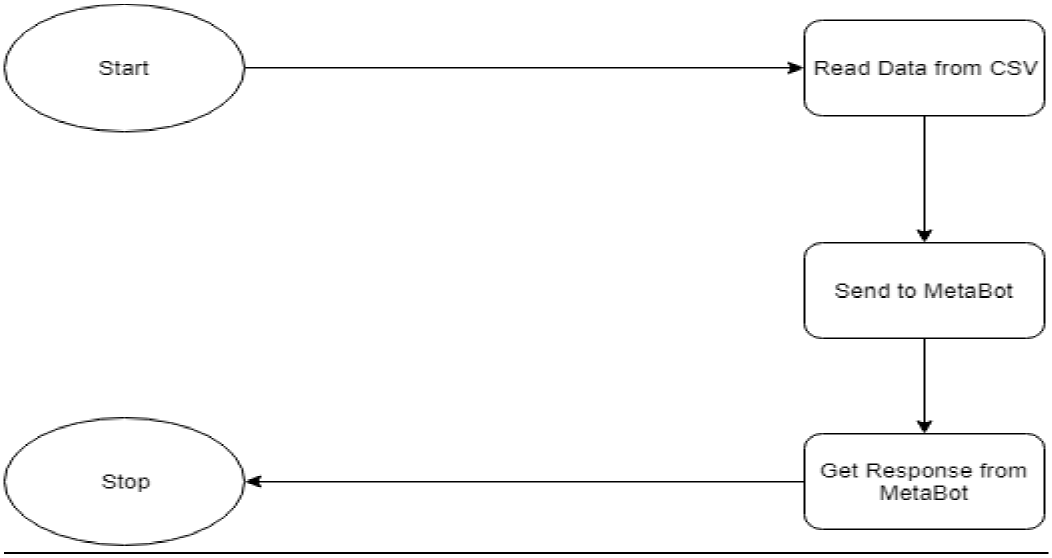


```
Variable Operation: $AAApplicationPath$\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Error Folder To $vErrorFolder$
Variable Operation: $AA
Variable Operation: $AA
Variable Operation: $vE
Variable Operation: $vE
If Folder Does Not Exist
    Create Folder "$vEm
End If
If Folder Does Not Exist
    Create Folder "$vLog
End If
If Folder Does Not Exist
    Create Folder "$vOu
End If
If Folder Does Not Exist
    Create Folder "$vSn
End If
Begin Error Handling; Ac
    Message Box: "Get A
    Run Logic "GetAllPro
    Message Box: "$vRe
    Log to File: $vRespo
    Message Box: "Get
    Run Logic "GetProje
    Message Box: "$vRe
    Message Box: "Get I
    Run Logic "GetIssue
    Message Box: "$vRe
    Message Box: "Crea
    Variable Operation: $
    Run Logic "CreatePr
    Message Box: "$vRe
    Message Box: "Create issue"
    Variable Operation: $vInputFolder$\IssueData.csv To $vCreateIssueCSVfile$
    Run Logic "CreateIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
    Message Box: "$vResponse$"
End Error Handling
```

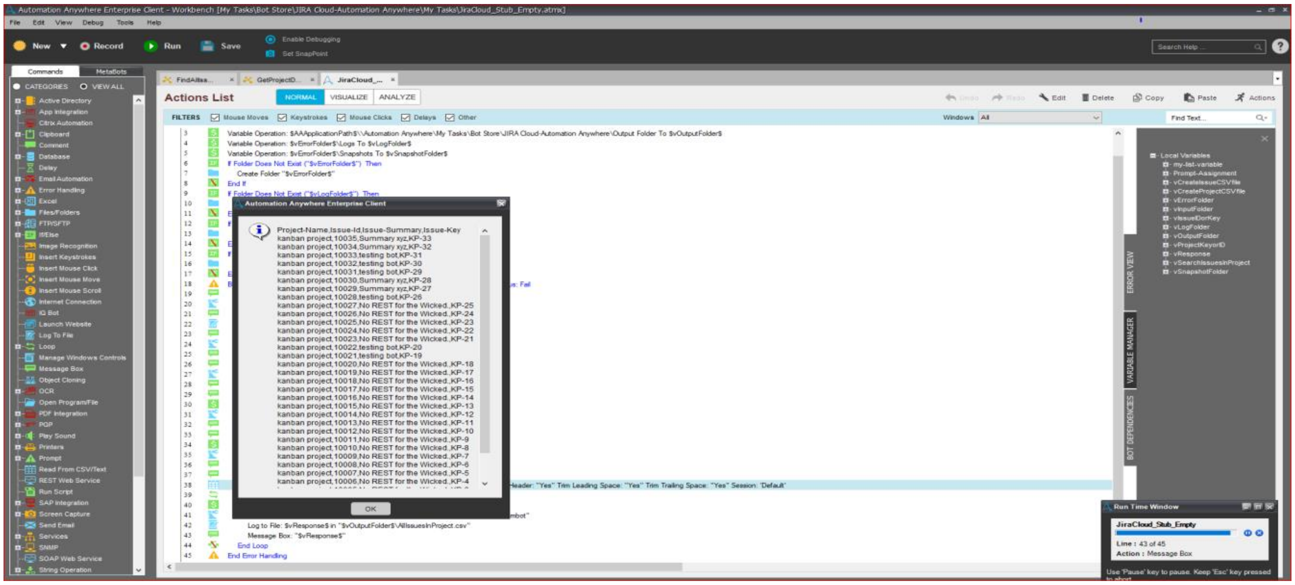
```
{
 "expand": "renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations",
 "id": "10005",
 "self": "https://apsudaipur.atlassian.net/rest/api/2/issue/10005",
 "key": "KP-3",
 "fields": {
  "statuscategorychangedate": "2019-08-30T12:35:45.091+05:30",
  "issuetype": {
   "self": "https://apsudaipur.atlassian.net/rest/api/2/issuetype/10106",
   "id": "10106",
   "description": "A problem or error.",
   "iconUrl": "https://apsudaipur.atlassian.net/secure/viewavatar?size=medium&avatarId=10303&avatarType=issuetype",
   "name": "Bug",
   "subtask": false,
   "avatarId": 10303
  },
  "timespent": null,
  "project": {
   "self": "https://apsudaipur.atlassian.net/rest/api/2/project/10001",
   "id": "10001",
   "key": "KP",
   "name": "kanban project",
   "projectTypeKey": "software",
   "simplified": false,
   "avatarUrls": {
    "48x48": "https://apsudaipur.atlassian.net/secure/projectavatar?pid=10001&avatarId=10419",
    "24x24": "https://apsudaipur.atlassian.net/secure/projectavatar?size=small&s=small&pid=10001&avatarId=10419",
    "16x16": "https://apsudaipur.atlassian.net/secure/projectavatar?size=xsmall&s=xsmall&pid=10001&avatarId=10419",
    "32x32": "https://apsudaipur.atlassian.net/secure/projectavatar?size=medium&s=medium&pid=10001&avatarId=10419"
   },
   "projectCategory": {
```

Closing in 3 seconds

- **FindAllIssuesInProject**
    1. User provides the ProjectId (in case of Get Issue Detail).
    2. This will return issue details in the task bot.
- **Visual Flow –**



- **Output : Message Box**



```
Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRA Cloud-Automation Anywhere\Output Folder To $vOutputFolder$
Variable Operation: $vErrorFolder$\Logs To $vLogFolder$
Variable Operation: $vErrorFolder$\Snapshots To $vSnapshotFolder$
If Folder Does Not Exist ("$vErrorFolder$") Then
    Create Folder "$vErrorFolder$"
End If
If Folder Does Not Exist ("$vLogFolder$") Then
```

Automation Anywhere Enterprise Client

```
Project-Name,Issue-Id,Issue-Summary,Issue-Key
kanban project,10035,Summary xyz,KP-33
kanban project,10034,Summary xyz,KP-32
kanban project,10033,testing bot,KP-31
kanban project,10032,testing bot,KP-30
kanban project,10031,testing bot,KP-29
kanban project,10030,Summary xyz,KP-28
kanban project,10029,Summary xyz,KP-27
kanban project,10028,testing bot,KP-26
kanban project,10027,No REST for the Wicked.,KP-25
kanban project,10026,No REST for the Wicked.,KP-24
kanban project,10025,No REST for the Wicked.,KP-23
kanban project,10024,No REST for the Wicked.,KP-22
kanban project,10023,No REST for the Wicked.,KP-21
kanban project,10022,testing bot,KP-20
kanban project,10021,testing bot,KP-19
kanban project,10020,No REST for the Wicked.,KP-18
kanban project,10019,No REST for the Wicked.,KP-17
kanban project,10018,No REST for the Wicked.,KP-16
kanban project,10017,No REST for the Wicked.,KP-15
kanban project,10016,No REST for the Wicked.,KP-14
kanban project,10015,No REST for the Wicked.,KP-13
kanban project,10014,No REST for the Wicked.,KP-12
kanban project,10013,No REST for the Wicked.,KP-11
kanban project,10012,No REST for the Wicked.,KP-10
kanban project,10011,No REST for the Wicked.,KP-9
kanban project,10010,No REST for the Wicked.,KP-8
kanban project,10009,No REST for the Wicked.,KP-7
kanban project,10008,No REST for the Wicked.,KP-6
kanban project,10007,No REST for the Wicked.,KP-5
kanban project,10006,No REST for the Wicked.,KP-4
```

```
    Log to File: $vResponse$ in "$vOutputFolder$\AllIssuesInProject.csv"
    Message Box: "$vResponse$"
  End Loop
End Error Handling
```

Run Time Window

JiraCloud_Stub_Empty

Line : 43 of 45

Action : Message Box

Use 'Pause' key to pause. Keep 'Esc' key pressed to abort

Common Use cases

The fields in the CSV can be changed to whatever data that needs to be added in list

- Create Project in the User’s Jira Cloud Account.
- Create Issue in the User’s Jira Cloud Account.
- Get details about all the project and their specifications.
- Get details about a specific project and its specifications.
- Get details about a specific issue and its specifications.
- Get details about all the issues and their specifications.

2. Requirements & Prerequisites

System Requirements

For the PC or server where the bot needs to run:

- ✓ RAM: 8GB or higher
- ✓ PROCESSOR: Intel Core i5 or higher and equivalent for any other OS
- ✓ Hard Disk: Up to 2GB of overall free space in the AA Client installation drive.

Reference below for Enterprise Client & Control Room system requirements.

Prerequisites

- ✓ ***Software's needed -***
- ✓ AA Enterprise Client 11.3.3
- ✓ AA Enterprise Control Room 11.3.3
- **Accounts/License needed –**
- ✓ Jira Cloud Account. Jira Cloud account can be opened at here
- ✓ AA Enterprise License

Security Measures

There are some security best practice recommendations that you may follow with your bot.

- It is not recommended to provide admin access to the Windows User Account executing the Bots, to avoid unintended data changes with the CSV Template.

It is recommended to update Jira Cloud API Tokens twice or more in a quarter to ensure data safety.

Note: For regenerating API token:

Login to your JIRA account and then Click here

I. Revoke previous token, if already exists

II. Click on "**Create API Token**" button and with appropriate label create new API token, store in credential vault as well.

3. Getting Started

Skill Matrix

The functionality of the Bot has been divided into set of skills.

Below is an overview of how the task bots and meta bots map to these skills:

Skill	Task Files	MetaBot Files
Create Project, Create Issue, GetProjectDetail, GetAllProjects, GetIssueDetails,GetAllIssues.	JiraCloud_Stub_Empty.atmx	JiraCloudMetaBot.mbot

Installation Hierarchy

Once the bot is downloaded and installed, the installer creates the files in the folder structure as shown below.

Folder Structure on your Machine

Folder Structure	Description
<AA Application Path> **1. My Tasks** **1.1. Bot Store** **1.1.1. JIRACloudBot-AppPerfectCorporation** ➢ **Error Folder** • **Logs** ○ Error logs Month-Day-Year.txt • **Snapshots** ○ Error Snap Month-Day-Year Hour Min Sec.png ➢ **Input Folder** • ProjectData.csv • IssueData.csv • FindAllIssuesInProject.csv ➢ **My Tasks** • JiraCloud_Stub_Empty.atmx	**<AA Application Path>** is the location where AA files are stored on your machine **1. My Tasks** My Task Folder is the default directory where Bot files are saved. **1.1 Bot Store** Bot Store Folder contains the Bot Name Folder which the installer creates during installation of the Bot. **1.1.1 JIRACloudBot-AppPerfectCorporation** This folder gets created by the installer and contains files and folders needed for the bot execution. ➢ **Error folder** is where logs and snapshots of screens will be placed if something goes wrong with the bot during execution. ➢ **Input Folder** is where the input files that the bot needs for execution of the use case, is saved • **IssueData.csv** provides name, key, summary, description for creating issue in a specific project of a user's Jira Cloud account. • **ProjectData.csv** provides, description, lead, name, assigneeType, projectTypeKey,

2. My MetaBots ➢ JiraCloudMetaBot.mbot	for creating Project in the user's Jira Cloud account. • **FindAllIssuesInProject.csv** provides project key from we need to find all issues. ➢ **My Tasks** folder contains all the Developed Platform Source Code **My MetaBots** folder contains the developed Meta bots needed for the bot execution.

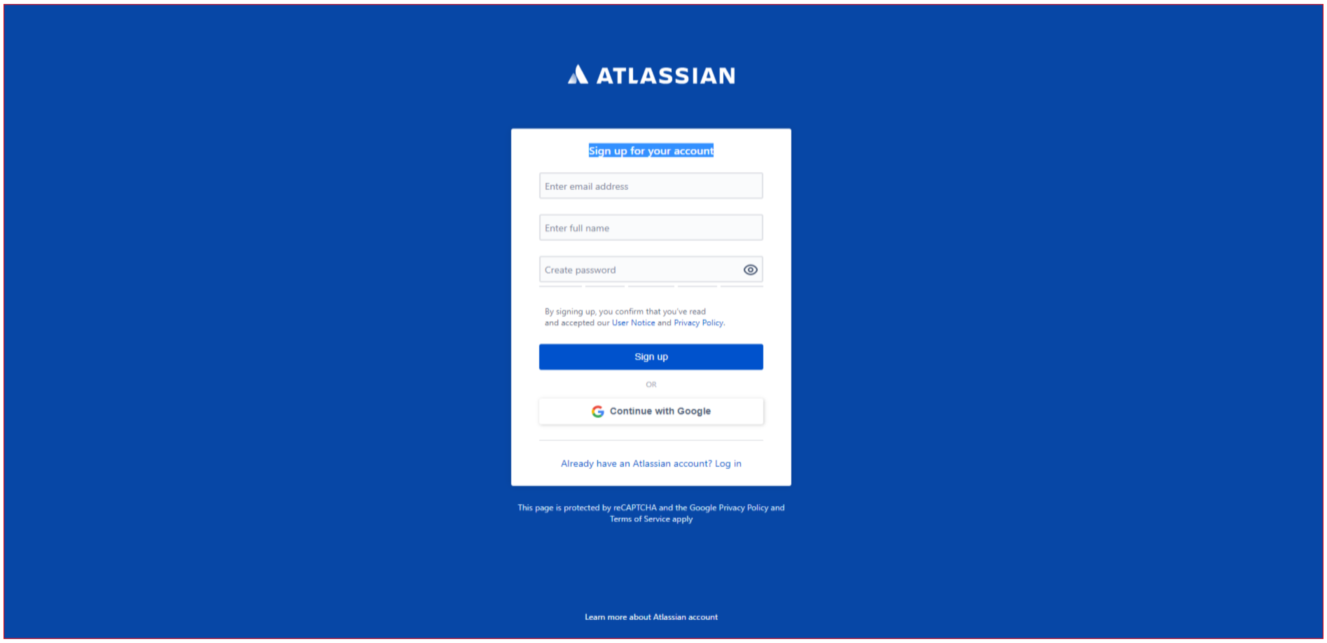
Quick Start

3.1.1 Setup

STEP 1 (Setting up JIRA Account):

Click here

The below page will appear once the above link is clicked -



By signing up, you confirm that you've read and accepted our User Notice and Privacy Policy.

This page is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply

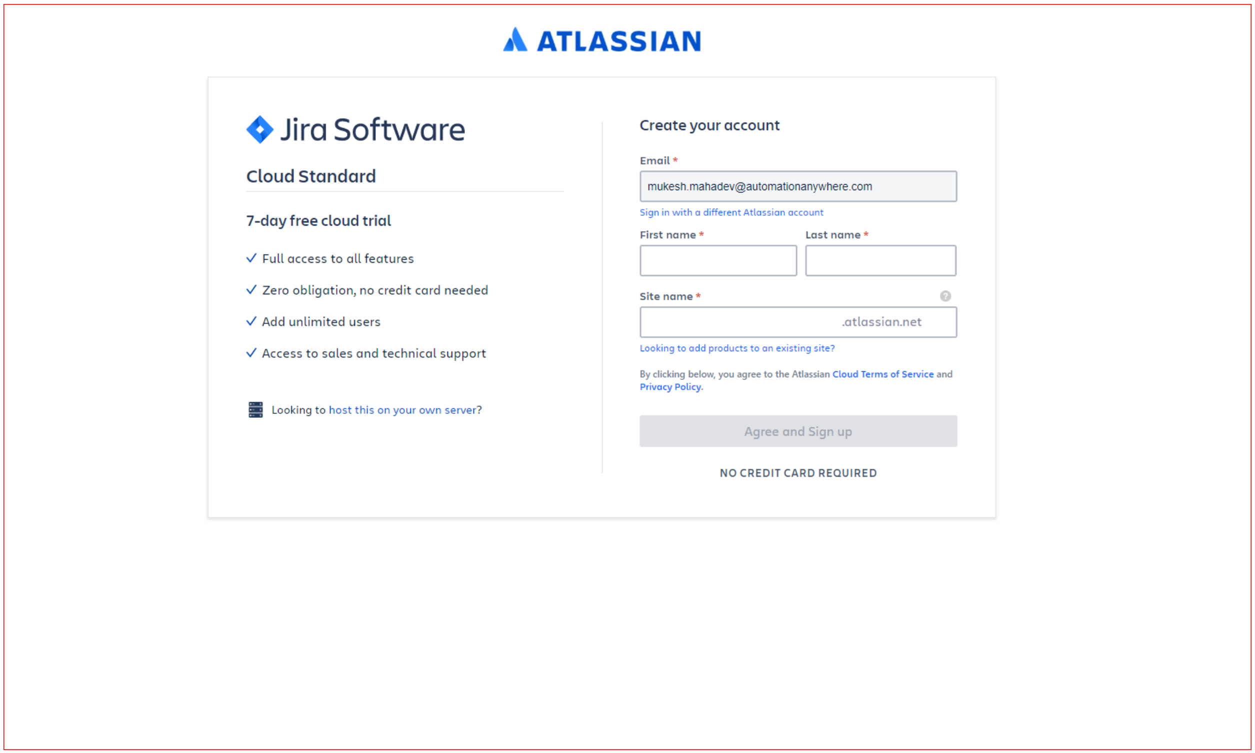
- If you already have an account, click on '**Log in**'.
- For new users, enter email, username and password than click on Sign up.

Next try with JIRA software and create JIRA Url.

JIRA Software It will show all product.

Create your JIRA Cloud account Click Here.

The below page will appear once the above link is clicked



ATLASSIAN

Jira Software

Cloud Standard

7-day free cloud trial

✓ Full access to all features

✓ Zero obligation, no credit card needed

✓ Add unlimited users

✓ Access to sales and technical support

Looking to host this on your own server?

Create your account

Email *

mukesh.mahadev@automationanywhere.com

Sign in with a different Atlassian account

First name *

Last name *

Site name *

.atlassian.net

Looking to add products to an existing site?

By clicking below, you agree to the Atlassian Cloud Terms of Service and Privacy Policy.

NO CREDIT CARD REQUIRED

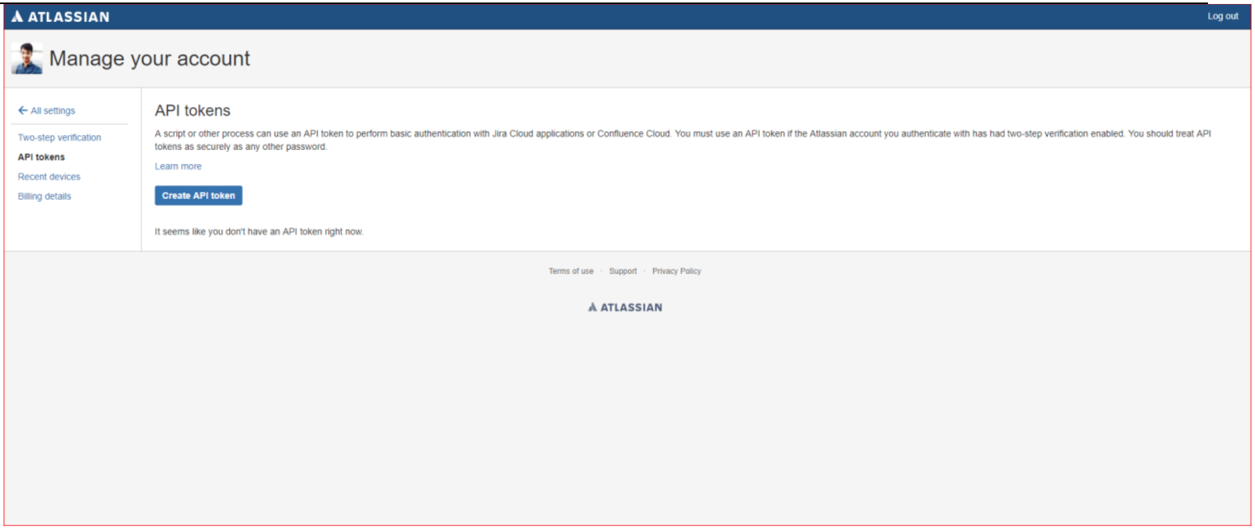
Enter require data and create your site url using just click on "**Agree and Sign up**".

Now you will get your Jira account url as similar like : https://{your site name}.atlassian.net

Step 2 (Generating URL & API Token):

Find or Generate Your API Key

- Click here to navigate to the JIRA manage your account.
- The below page will appear once the above link is clicked



API tokens

A script or other process can use an API token to perform basic authentication with Jira Cloud applications or Confluence Cloud. You must use an API token if the Atlassian account you authenticate with has had two-step verification enabled. You should treat API tokens as securely as any other password.

It seems like you don't have an API token right now.

Click on create API token and enter any suitable label. It will generate API token. Save your API token in the credential vault so that it can be used while using Meta Bot.

3.1.2 Inputting your Access Code

Step by Step Guide for Customers: Inputting Your Access Code click here

3.1.3 Configuration

STEP 1 (Configuring Credential in Control Room's Credential Vault)

Passwords and other sensitive information such as API Token, Username which are included in automation tasks should be stored as credentials centrally in the Credential Vault.

These sensitive data items become secured (stored in CR) and can't be accessed locally which is used by the bots when it executes in the machine.

Below are the step-by-step instructions for what needs to be done to set up the URL and Token for JIRA Cloud in the Credential Vault.

- 'Locker Consumer Role' needs to be assigned to the AA user (bot runner) which ensures that the user have full access to the locker.
- Locker needs to be created with name as "JIRA_Cloud"
- Credential needs to be created with name as "JIRACloud", where the URL and Token , UserName , IssueUrl needs to be saved.

Locker	Credentials	Attribute Name	Value

Name	Name		
JIRA_Cloud	JIRACloud	APIToken	API token to perform basic authentication On Jira-Cloud. For more information Click Here
JIRA_Cloud	JIRACloud	ProjectUrl	Base url with end point Ex: https://{yoursite}.atlassian.net/rest/api/2/project Endpoint : click here
JIRA_Cloud	JIRACloud	UserName	The user-name associated with the api token
JIRA_Cloud	JIRACloud	IssueURL	Base url with end point Ex: https://{yoursite}.atlassian.net/rest/api/2/issue Endpoint : click here
JIRA_Cloud	JIRACloud	SearchIssueURL	Base url with end point Ex: https://{yoursite}.atlassian.net/rest/api/2/search Endpoint : click here

- For more information on how to set up credentials in the Control Room, please visit this link on the Automation Anywhere Product Documentation portal.

STEP 2 (Configuring Parameters with an External File)

External file can be configured at -

AA Application Path > My Tasks > Bot Store > JIRACloudBot-AppPerfectCorporation > Input Folder > IssueData.csv

AA Application Path > My Tasks > Bot Store > JIRACloudBot-AppPerfectCorporation > Input Folder > ProjectData.csv

AA Application Path > My Tasks > Bot Store > JIRACloudBot-AppPerfectCorporation > Input Folder > FindAllIssuesInProject.csv

This is where, the input values for variables are stored which is needed by the bot at the time of execution. (Sensitive variables – like user name, password, and URLs will be an input in the Credential)

Below is a table that summaries what the variable name is, what it does, and an example of the input:

CreateIssue -

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vCSVFilePath	Text	Yes	Absolute Path where all parameters are present for creating issue.	C:\Users\Administrator\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ JIRACloudBot-AppPerfectCorporation \Input Folder\IssueData.csv

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	ExampleOutput
vResponse	Text	Yes	Return string of Issue Url,ID,Key	Created Successfully Issues: Issue URL,ID,Key https:aps.altassian.net/rest/api/2/issue/41,41,EX1

CreateProject –

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vCSVFilePath	Text	Yes	Absolute Path where all parameters are present for creating Project.	C:\Users\Administrator\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\ JIRACloudBot-AppPerfectCorporation \Input Folder\IssueData.csv\ProjectData.csv

OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Return string of Project Url,ID,Key	Created Successfully Projects: Project URL,ID,Key https:aps.altassian.net/rest/api/2/project/41,41,EX1

GetAllProject -

- INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER

  In this all the parameters are coming from Credential Vault.

- OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	All projects name,key,id,projectTypeKey is returned.	ProjectName,key,ID,projectTypeKey Example,EX,12,business

GetAllIssuesInProject

- INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vSearchIssuesInProject	Text	Yes	Unique project key given by the user to distinguish between 2 project.	KP

- OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	All issues present in Project having parametes: IssueName, IssueId, Summary, Key	Issue1, AB12, AnIssue, Bug

GetIssue

- **INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vIssueIDorKey	Text	Yes	Unique Id given by the user to distinguish between 2 issues.	10005

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Complete description of an issue is returned .	Complete Description of issue like Issue-type, Issue-Url, Issue-Summary

GetProjectDetail

- **_INPUT VARIABLES: NEEDS TO BE CONFIGURED BY THE USERS FOR INPUT PARAMETER**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Input
vProjectIDorKey	Text	Yes	Unique Id given by the user to distinguish between 2 projects.	KP

- **OUTPUT VARIABLES: DISPLAYS OUTPUT RESPONSE**

Variable Name	Type	Mandatory (Yes/No)	Purpose	Example Output
vResponse	Text	Yes	Complete Description of a project	Complete Description of project like Project-type, Project-Url, Project-Summary

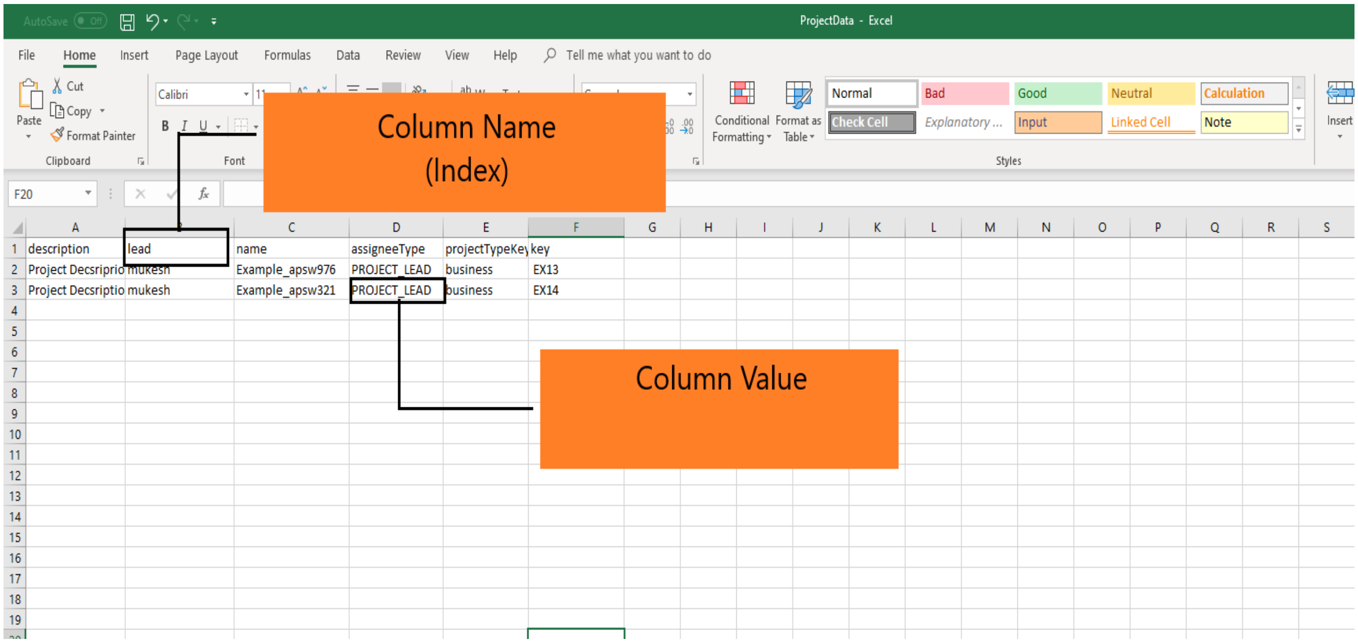
			is returned	

Guidance for configurating Filter conditions :

a. Filter based on Column Value

Column Name is the column name which is present in the first row of CSV file. Please do not delete or alter Column Name.
Column Value is the value which needs to be matched against the mentioned Column name. You can add or delete parameters as per your requirements.

Example



	A	B	C	D	E	F
1	description	lead	name	assigneeType	projectTypeKey	key
2	Project Decsriprio	mukesh	Example_apsw976	PROJECT_LEAD	business	EX13
3	Project Decsriptio	mukesh	Example_apsw321	PROJECT_LEAD	business	EX14

Input CSV Files-

1) IssueData.csv

- **Required Parameters-**

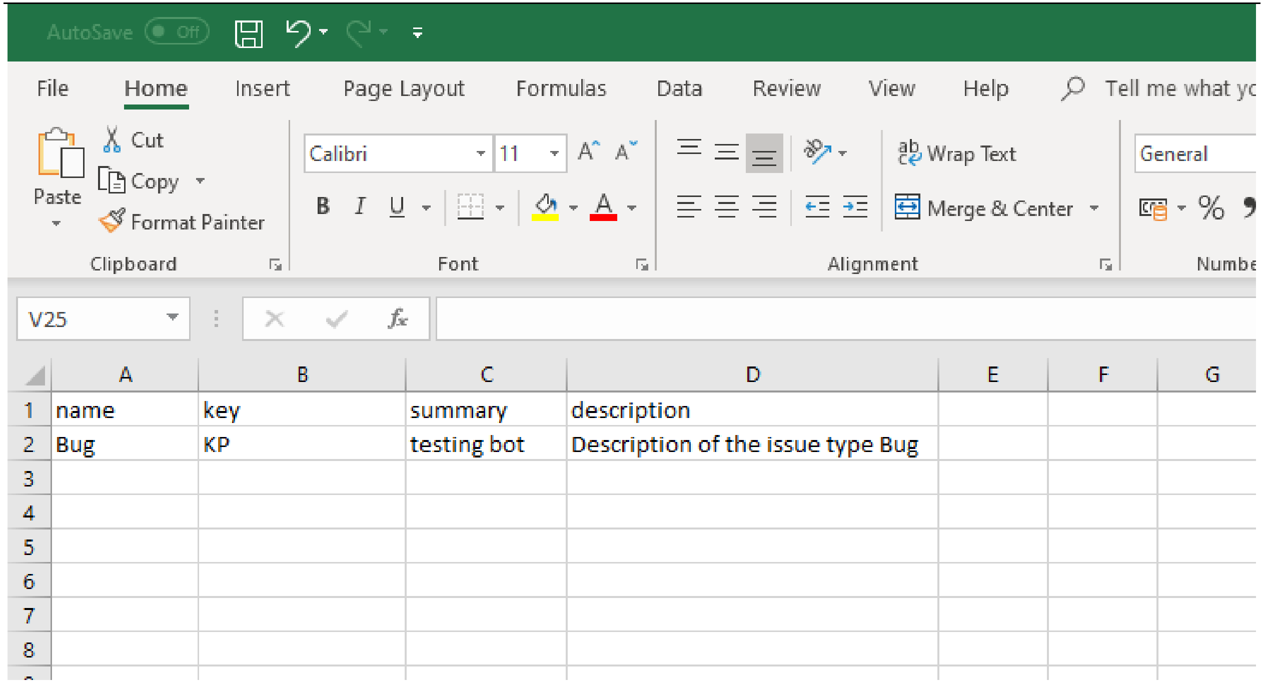
File Parameter Name	Type	Description	Example Input
name	Text	Type of issue which is created	Bug
key	Text	Project Key in which the issue is to be added	KP
summary	Text	A brief summary about the issue	testing bot

- **Optional Parameters -**

(If you don't want to add any value in the parameter leave it empty)

File Parameter Name	Type	Description	Example Input
description	Text	Detailed description about the issue	Description xyz about the issue

Example:



	A	B	C	D	E	F	G
1	name	key	summary	description			
2	Bug	KP	testing bot	Description of the issue type Bug			
3							
4							
5							
6							
7							
8							

2) ProjectData.csv

- **Required Parameters-**

File Parameter Name	Type	Description	Example Input
lead	Text	Name of the Project lead	mukesh
name	Text	Name of the project that has been created	Project_1
assigneeType	Text	The default assignee when creating issues for this project. Valid values: PROJECT_LEAD, UNASSIGNED	PROJECT_LEAD
projectTypeKey	Text	It tells us the application-specific feature set. Valid values: software, service_desk, business	business

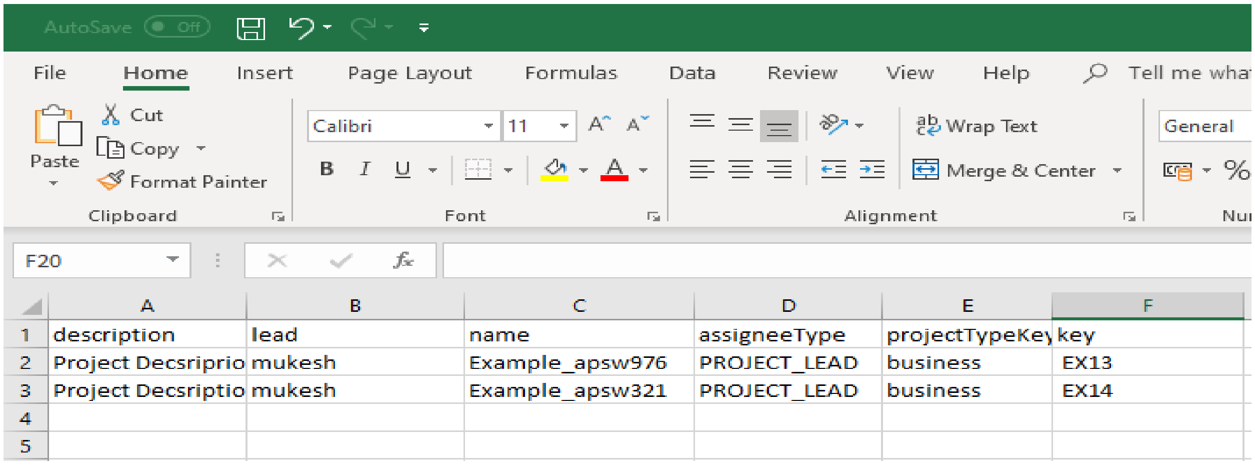
key		Project keys must be unique and start with an uppercase letter followed by one or more uppercase alphanumeric characters. The maximum length is 10 characters	EX12

- **Optional Parameters -**

(If you don't want to add any value in the parameter leave it empty)

File Parameter Name	Type	Description	Example Input
description	Text	Detailed description about the project	Description xyz about the project

Example:



	A	B	C	D	E	F
1	description	lead	name	assigneeType	projectTypeKey	key
2	Project Decsriprio	mukesh	Example_apsw976	PROJECT_LEAD	business	EX13
3	Project Decsriptio	mukesh	Example_apsw321	PROJECT_LEAD	business	EX14
4						
5						

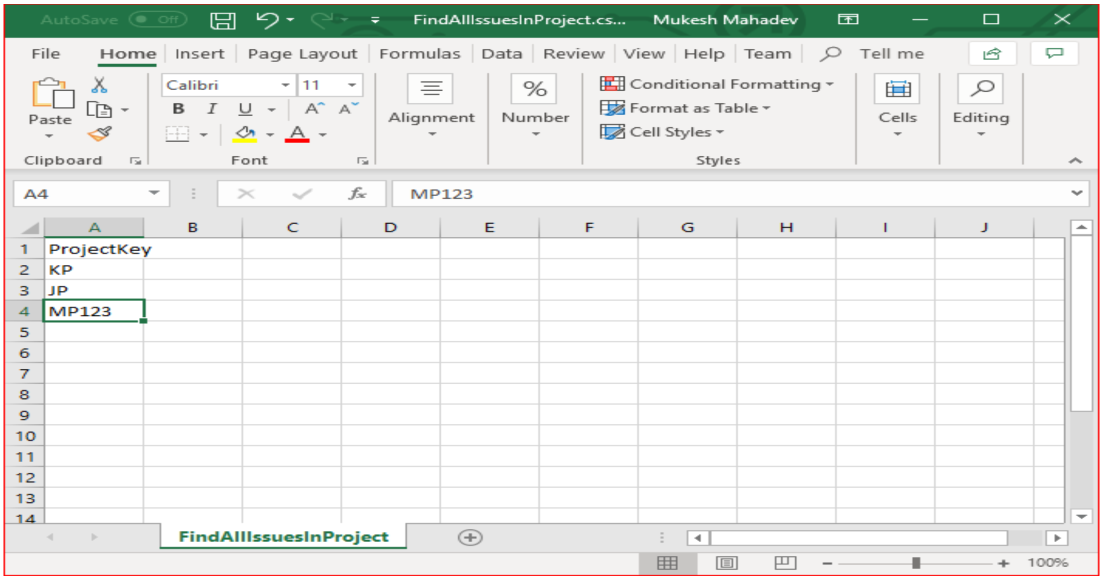
3) FindAllIssuesInProject.csv

- **Required Parameters-**

File Parameter Name	Type	Description	Example Input

ProjectKey	Text	Unique project key given by the user to distinguish between 2 project.	KP

Example:



Formula bar: MP123

	A
1	ProjectKey
2	KP
3	JP
4	MP123

4. Reports

There are no Bot Insight Reports generated for this Bot.

5. Logs

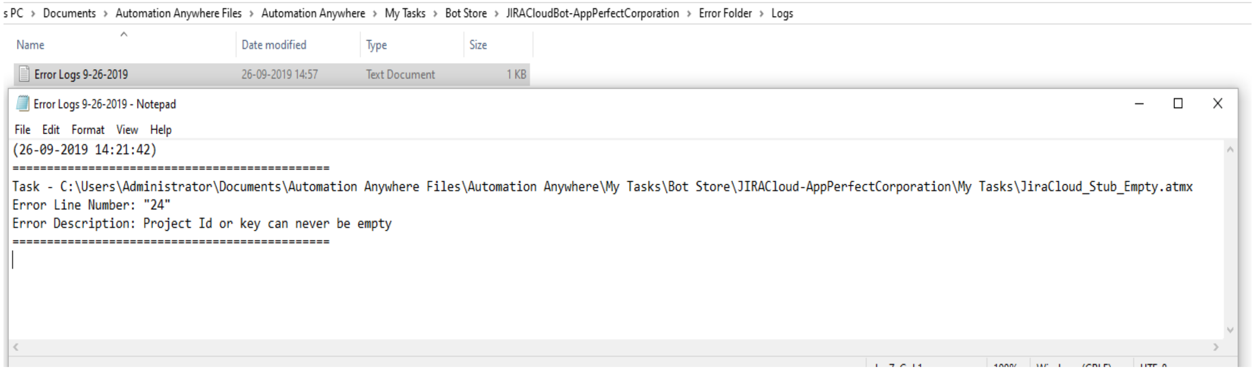
In case of Errors, Error Logs & Screenshots are generated within Error Folder (Highlighted below in yellow)

- My Tasks
  - Bot Store
    - Bot Name
      - Error Folder
        - Logs (Folder)
          - Error Logs Month-Day-Year.txt
        - Snapshots (Folder)
          - Error Snap Month-Day-Year HourMinSec.png

Error Logs will contain the below information -

- Generated Timestamp
- Task Name
- Error Line Number
- Error Description

Example of Error Generated Error LogsFile:



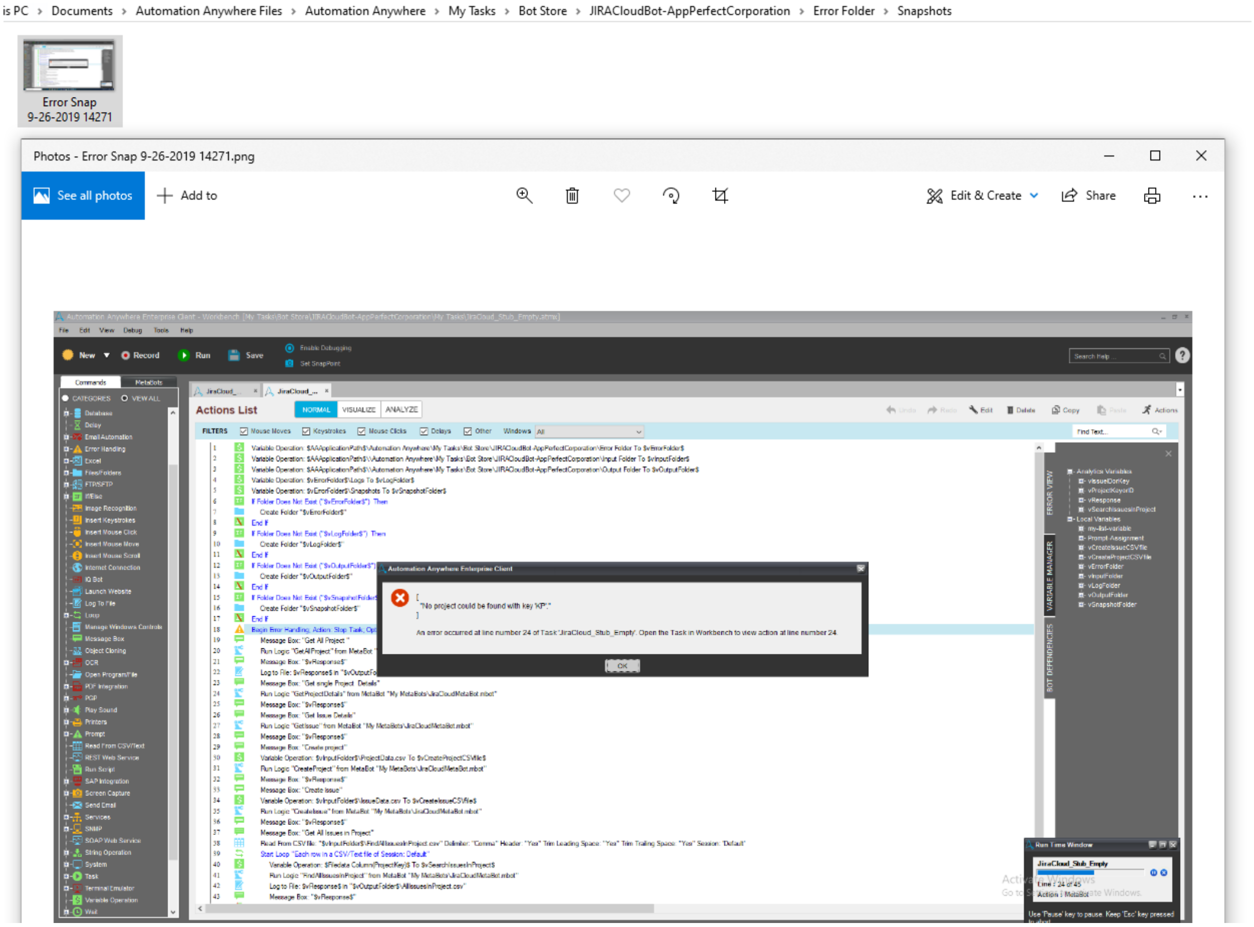
s PC › Documents › Automation Anywhere Files › Automation Anywhere › My Tasks › Bot Store › JIRACloudBot-AppPerfectCorporation › Error Folder › Logs

Name	Date modified	Type	Size
Error Logs 9-26-2019	26-09-2019 14:57	Text Document	1 KB

Error Logs 9-26-2019 - Notepad

```
(26-09-2019 14:21:42)
=============================================
Task - C:\Users\Administrator\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store\JIRACloud-AppPerfectCorporation\My Tasks\JiraCloud_Stub_Empty.atmx
Error Line Number: "24"
Error Description: Project Id or key can never be empty
=============================================
```

Example of Screenshot Generated Error File:



is PC › Documents › Automation Anywhere Files › Automation Anywhere › My Tasks › Bot Store › JIRACloudBot-AppPerfectCorporation › Error Folder › Snapshots

Error Snap 9-26-2019 14271

Photos - Error Snap 9-26-2019 14271.png

```
1  Variable Operation: $AAApplicationPath$\Automation Anywhere\My Tasks\Bot Store\JIRACloudBot-AppPerfectCorporation\Error Folder To $vErrorFolder$
2  Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRACloudBot-AppPerfectCorporation\Input Folder To $vInputFolder$
3  Variable Operation: $AAApplicationPath$\\Automation Anywhere\My Tasks\Bot Store\JIRACloudBot-AppPerfectCorporation\Output Folder To $vOutputFolder$
4  Variable Operation: $vErrorFolder$\Logs To $vLogFolder$
5  Variable Operation: $vErrorFolder$\Snapshots To $vSnapshotFolder$
6  If Folder Does Not Exist ("$vErrorFolder$") Then
7      Create Folder "$vErrorFolder$"
8  End If
9  If Folder Does Not Exist ("$vLogFolder$") Then
10     Create Folder "$vLogFolder$"
11 End If
12 If Folder Does Not Exist ("$vOutputFolder$")
13     Create Folder "$vOutputFolder$"
14 End If
15 If Folder Does Not Exist ("$vSnapshotFolder$
16     Create Folder "$vSnapshotFolder$"
17 End If
18 Begin Error Handling; Action: Stop Task; Opt
19     Message Box: "Get All Project "
20     Run Logic "GetAllProject" from MetaBot "
21     Message Box: "$vResponse$"
22     Log to File: $vResponse$ in "$vOutputFo
23     Message Box: "Get single Project  Details"
24     Run Logic "GetProjectDetails" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
25     Message Box: "$vResponse$"
26     Message Box: "Get Issue Details"
27     Run Logic "GetIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
28     Message Box: "$vResponse$"
29     Message Box: "Create project"
30     Variable Operation: $vInputFolder$\ProjectData.csv To $vCreateProjectCSVfile$
31     Run Logic "CreateProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
32     Message Box: "$vResponse$"
33     Message Box: "Create Issue"
34     Variable Operation: $vInputFolder$\IssueData.csv To $vCreateIssueCSVfile$
35     Run Logic "CreateIssue" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
36     Message Box: "$vResponse$"
37     Message Box: "Get All Issues in Project"
38     Read From CSV file: "$vInputFolder$\FindAllIssuesInProject.csv" Delimiter: "Comma" Header: "Yes" Trim Leading Space: "Yes" Trim Trailing Space: "Yes" Session: 'Default'
39     Start Loop "Each row in a CSV/Text file of Session: Default"
40         Variable Operation: $Filedata Column(ProjectKey)$ To $vSearchIssuesInProject$
41         Run Logic "FindAllIssuesInProject" from MetaBot "My MetaBots\JiraCloudMetaBot.mbot"
42         Log to File: $vResponse$ in "$vOutputFolder$\AllIssuesInProject.csv"
43         Message Box: "$vResponse$"
```

Automation Anywhere Enterprise Client

```
[
  "No project could be found with key 'KP'."
]

An error occurred at line number 24 of Task 'JiraCloud_Stub_Empty'. Open the Task in Workbench to view action at line number 24.
```

<u>Note</u>: There are no Custom Logs generated for this Bot.

6. Troubleshooting & Support

Support

Please visit our Support Portal for any assistance on Bot functionality or Feature.

Automation Anywhere provides a Product Documentation portal which can be accessed for more information about AA's products and guidance on building bots and Digital Workers.

The "Build" section of the portal includes these sections:

- Getting Started - information on building bots recommended practices (including use of the Credential Vault)
- Build Advanced Bots - details on Meta Bots and the approach to integrating code into them
- Build Digital Workers - high level architecture

FAQs

Q: The 'Create Locker' button is not visible in the Credential Vault Page. What do I do?

A: Please check if the role 'Locker Consumer' is assigned to the user and you are using an Enterprise Client version.

Appendix A: Record of Changes

No.	Version Number	Date of Change (DD/MM/YYYY)	Author	Notes
1	1.0	04/09/2019	AppPerfectCorporation	Version 1 Release

Appendix B: Acronyms

No.	Acronym	Description
1	AA	Automation Anywhere
2	CSV	Comma Separated File
3	DW	Digital Worker
4	CR	Credential Vault
5	CRM	Customer Relationship Management
6	API	Application Programming Interface
7	PC	Personal Computer
8	RAM	Random Access Memory
9	OS	Operating System

Appendix C: References

No.	Topic	Reference Link
1	System Requirements - Client	*Click here*
2	System Requirements – Control Room	*Click here*
3	How to create credential & Locker?	*Click here*
4	Credential Overview	*Click here*
5	Apeople Community Site	*Click here*
6	Product Documentation portal	*Click here*