Handover document for mermaid chart jira cloud plugin

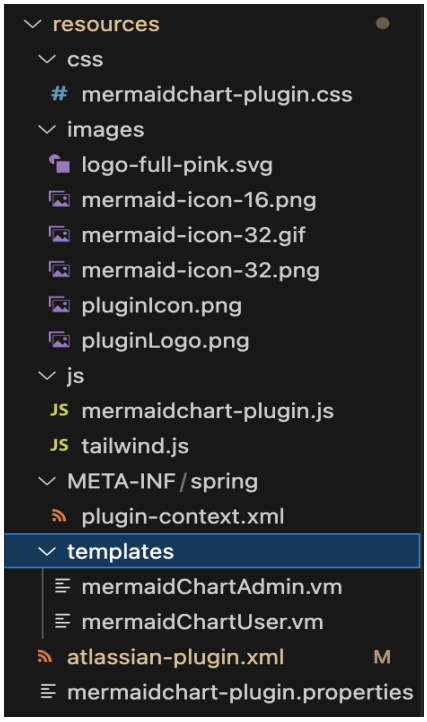
Mermaid chart jira plugin for cloud purely based on java spring with atlassian scanner based architecture and velocity templating engine. There is a plugin descriptor file that informs the jira cloud about the plugin and all of the resources that the plugin is going to use. From the root folder, under src->main->resources folder, there is a plugin descriptor file named atlassian-plugin.xml having main atlassian-plugin configuration. In this file, following information is defined like

- Plugin-info
- Internationalization resource
- Web-resources
- Web-item
- Servlet
- Rest resource

Also, to provide internationalization properties, the file named “mermaidchart-plugin.properties” is placed in the same folder as the plugin descriptor file.

Resources

The resources folder structure is as below:

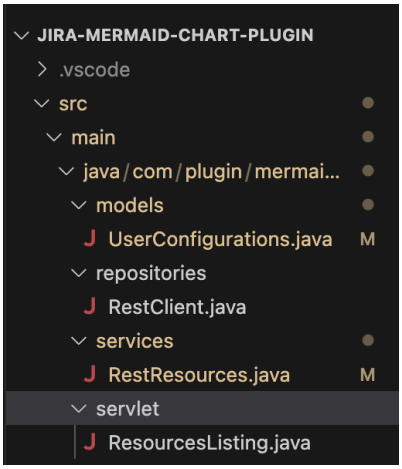


In the same resources folder, there are few other folders.

- “templates” folder consists of velocity templates for user and admin as well.
- “META-INF/spring” folder contains spring beans configuration
- “js” folder contains the main controller js file for plugin and tailwind.css file
- “images” folder have all the images used in plugin
- “css” folder contains a css styling file for velocity template files

Java

After resources folder, let’s have a look on java folder which have folder structure as follows:

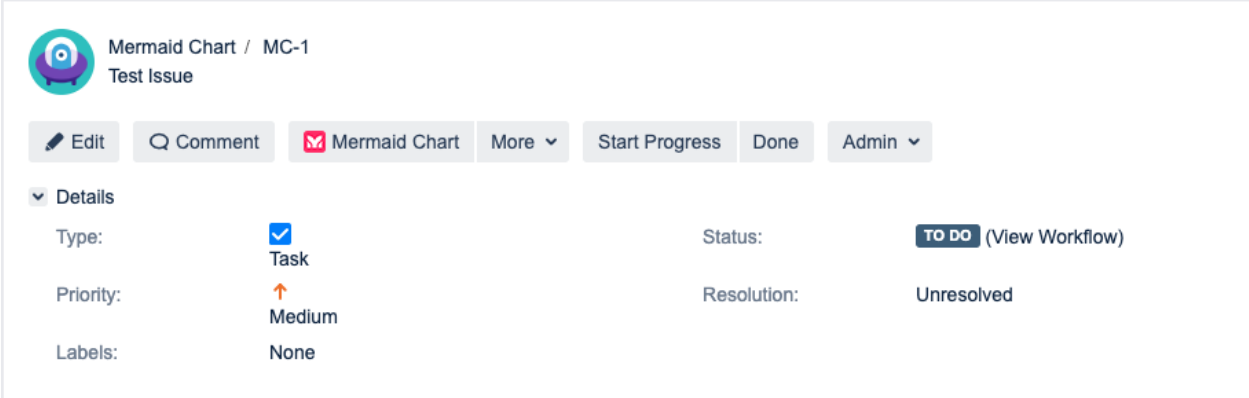


1. In the “models” folder there is a “userConfigurations.java” file that contains two parameters, baseURL and securityToken, to save and fetch them from the atlassian plugin data console.
2. In the “repositories” folder there is a “RestClient.java” file that contains a rest client, the HTTP client, to get data as well as resources from mermaid chart.
3. In the “services” folder, there is a “RestResources.java” file that contains all the rest endpoints that are used by the main controller js file in the resources folder to perform **crud** operations.

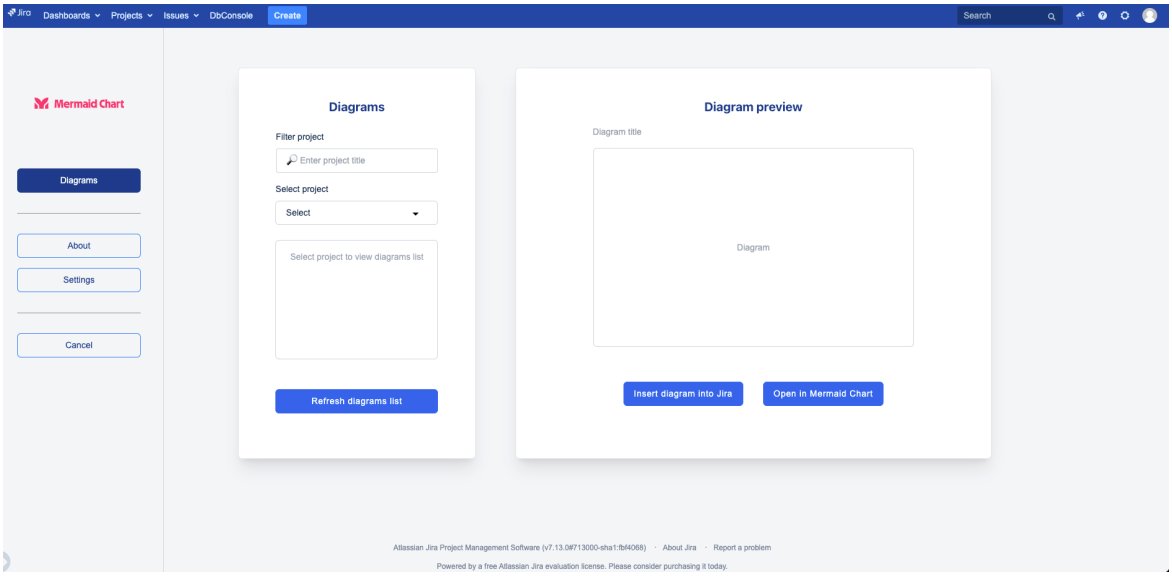
4. “servlet” the last folder contains a file named “resourcesListing.java” that extends the HTTPServlet class to initiate the plugin with required information from the plugin descriptor as well as from the plugin data console.

Core functionality of mermaid chart jira cloud plugin

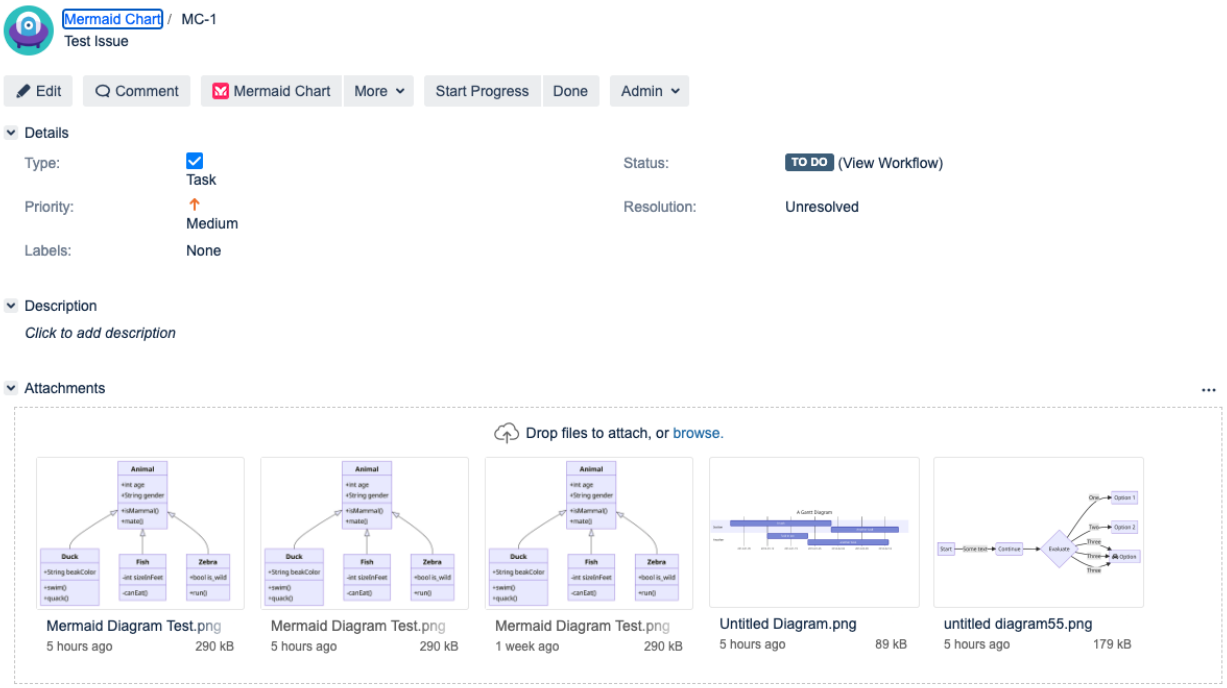
Mermaid Chart’s Jira plugin allows you to import your diagrams into a Jira ticket and provides a link to edit your diagrams on the Mermaid Chart platform. So, first of all, after the installation of plugin in your jira instance, you can see “**Mermaid Chart**” on top actions of the issue screen as



You can see in the image above that beside the edit and comment action, mermaid chart action is shown, which redirects you towards the mermaid chart plugin. Actually, when clicking on this action, the plugin descriptor triggers the action towards java HTTP servlet with some issue parameters like **issue-id**, **project-id**, and **user-id**. **Issue-id** and **project-id** are used to track the redirection record back towards the issue, and **user-id** is used to store the user configuration against that provided **user-id**. So, **Servlet** will respond to the action by triggering the plugin to start with the provided data as well as configurations from the atlassian plugin data console. As a result, you will see following screen:

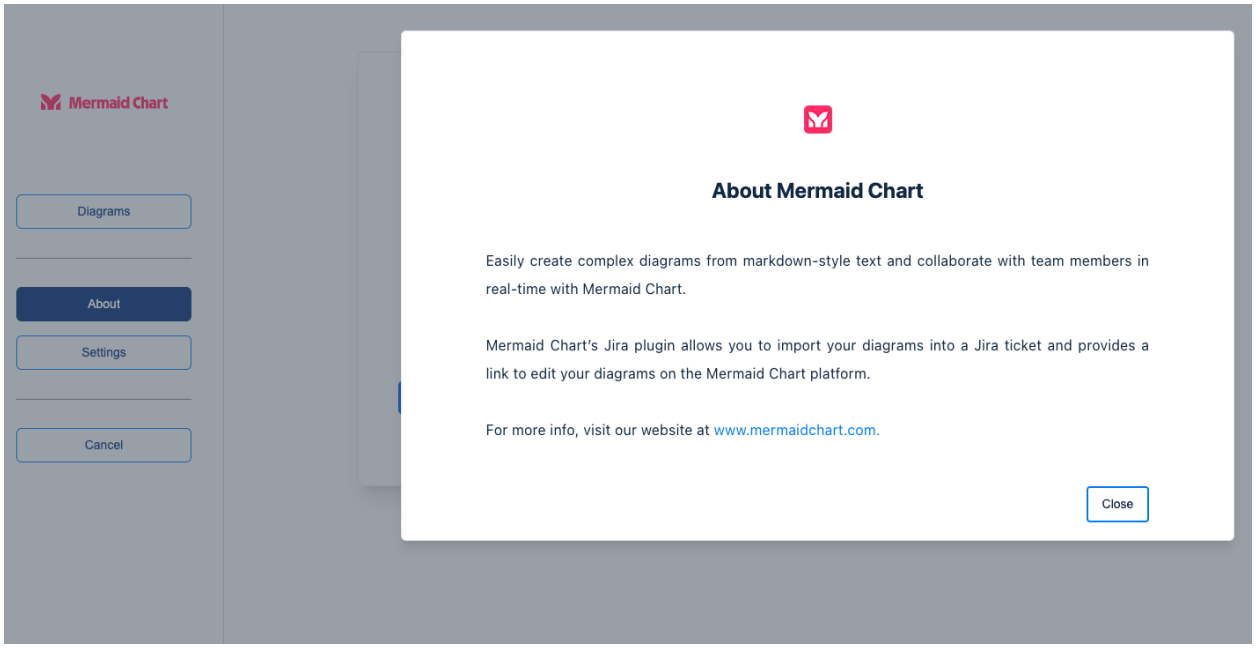


At the very first time, when the user opens the plugin, the user will automatically be redirected towards the settings screen as it is mandatory to configure settings first **PS:** This functionality is mandatory for both **User(s)** and **Admin.** After configuring the settings, the user can continue. Above screen represents the main resources screen, where in the first box, there is the **diagram** selection mechanism, from which you can filter **projects** and also select the project. After selecting the project, the diagrams against that selected project will become available in the diagrams listing section. When clicking on a diagram title in the diagram listing section, a request will automatically trigger to fetch diagram from the mermaid chart pre-rendering service and will be displayed in the diagram preview section. At the end of the box, there is a **refresh diagrams list** button which refreshes the diagrams listing area with the latest diagrams available in the **mermaid chart**. In addition with this box, there is a **Diagrams preview** box which mainly aims to provide the functionality of viewing the diagram. At the end of this box, there are two buttons. **Insert diagram to jira** is used to insert the diagram that is currently selected into the **jira issue**, and the **Open in Mermaid Chart** button is used to open this selected diagram into the mermaid chart site (editor). One more thing is that when the user clicks on **Insert Diagram into Jira** button, the diagram will automatically upload in the attachment section of the issue. Issue having attachment from mermaid chart plugin will become available as follows:

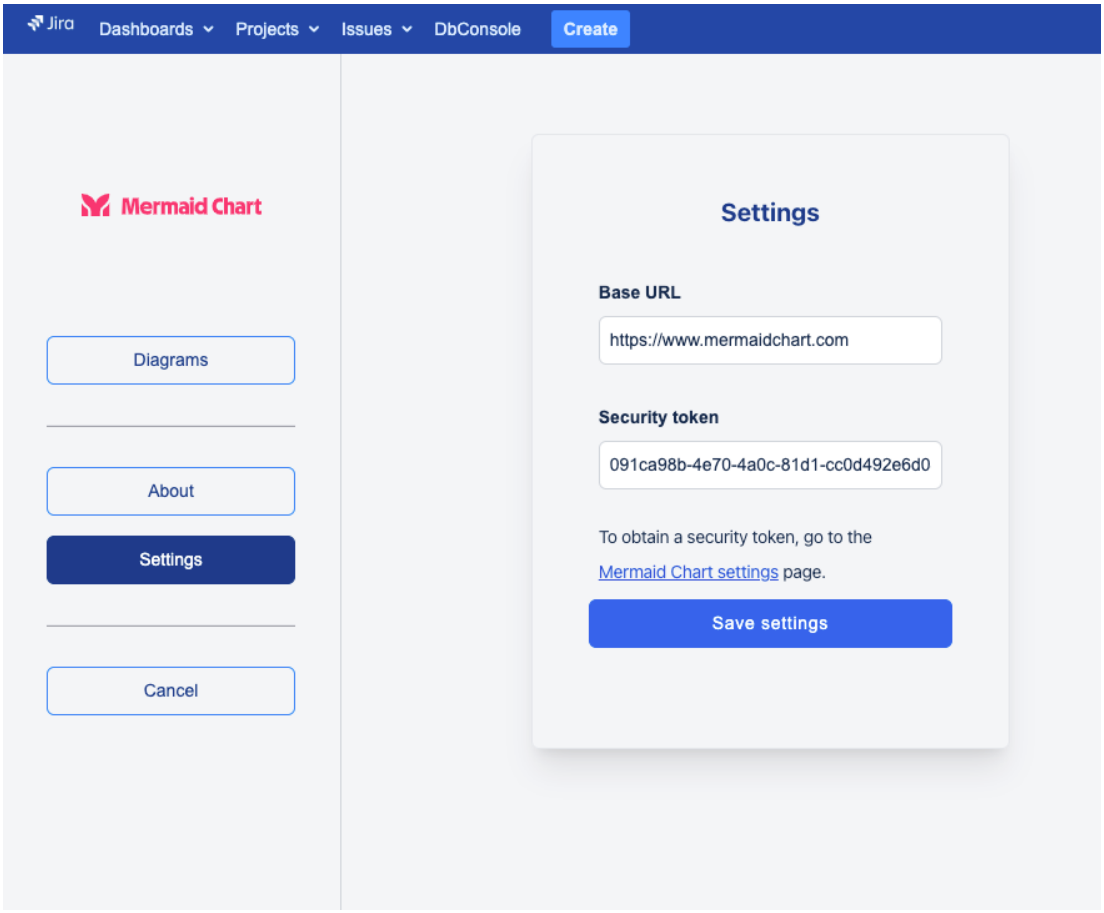


So, this page is mainly used to fetch projects as well as diagrams from the mermaid chart, also providing the functionality to **filter projects**, **refresh diagrams**, **insert diagrams to jira**, and **open diagrams in Mermaid Chart**.

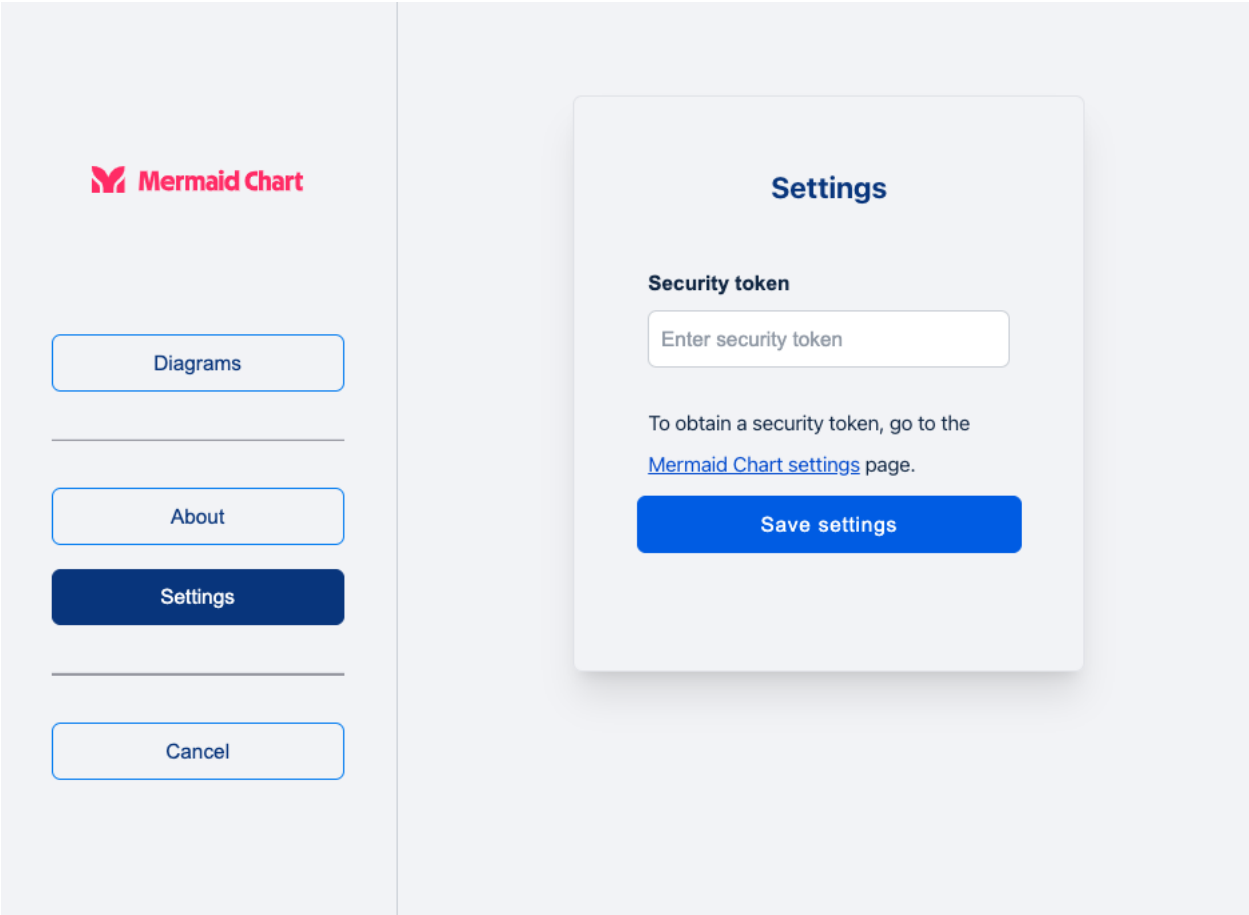
The next option in the sidebar is **About**. This tab shows the information about the plugin. Below screen appears as a result on clicking the **About tab:**



The next option in sidebar is the **Settings**. As the name suggests, this tab is used to configure settings for plugin like **baseURL** (only for admin), and **Security Token**. When logged in as an **Admin**, and wants to configure the settings, then on clicking the **Settings tab**, the following screen will appear:



The scenario will almost be the same for a **User**, but the **UI** will be a little bit different for a **User**. As we know that the user does not have permission to configure **baseURL** for plugin. So, **Users** can only configure security token for their own account in the plugin. When a user wants to configure settings, then on clicking the **settings tab**, the following screen will be appeared for the user:



On clicking the **save settings** button, a REST resource will be called that will save the configuration into the atlassian plugin data console for future use.

And finally the last tab is **Cancel** which is used to cancel the current process and redirect back to the jira issue screen.