Static Analysis Violation Reporter for Atlassian JIRA

In this section:

This workflow artifact shows how to use the Atlassian JIRA importer to create and manage JIRA bugs and/or tasks for problems (e.g., static analysis violations or unit test failures) found during test and analysis.

When new problems are reported, the importer can create issues or tasks in a JIRA bug tracking system and assign these to a user.

In this documentation, we refer to issues and tasks as distinct entities:

- An issue tracks a single reported problem and can update the associated result as its status changes.
- A task is a single entry for a group of associated problems; one task is created per assignee on each run. Tasks are not updated as the related problems change. If a problem reoccurs, a new task will be created.

Requirements

- Atlassian JIRA 6.1.x or higher with REST client API available.
- Extension Designer 5.4.0 and DTP 5.4.0.
- Access to a JIRA user account with permissions to create and alter issues for the project (usually an administrator). Consult your JIRA administrator for credentials

Upgrading

The API driving the Static Analysis Violation Reporter for JIRA was updated in version 2.2. Disable previous versions of this artifact and install this version as a new workflow.

Added Components

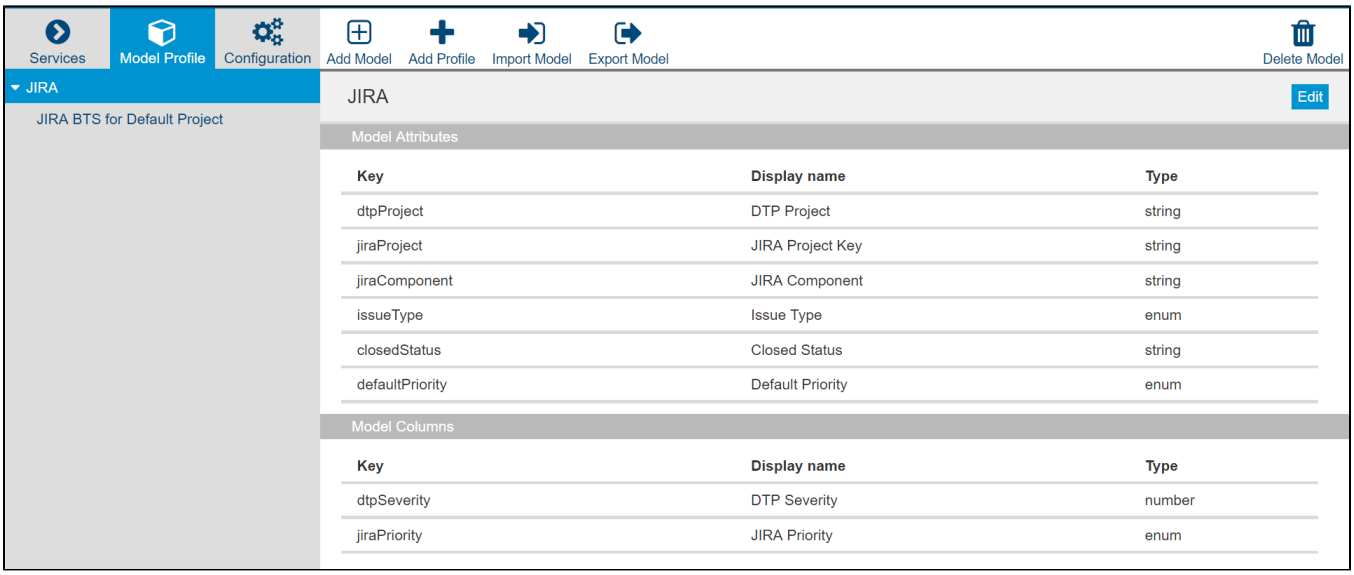
No new nodes are installed with the JIRA Importer artifact. The only additional files in the system are the documentation and the example flow and profile.

Model Configuration

The JIRA Importer flow uses a Data Model, which can be used to customize the JIRA Importer to meet your goals. The model and sample profile are added automatically when you install and deploy the artifact. Also see Working with Model Profiles for general information about configuring the model.

Profile Attributes

In the Model Profile tab, click on the model to view the attributes, which define how to add issues to JIRA.



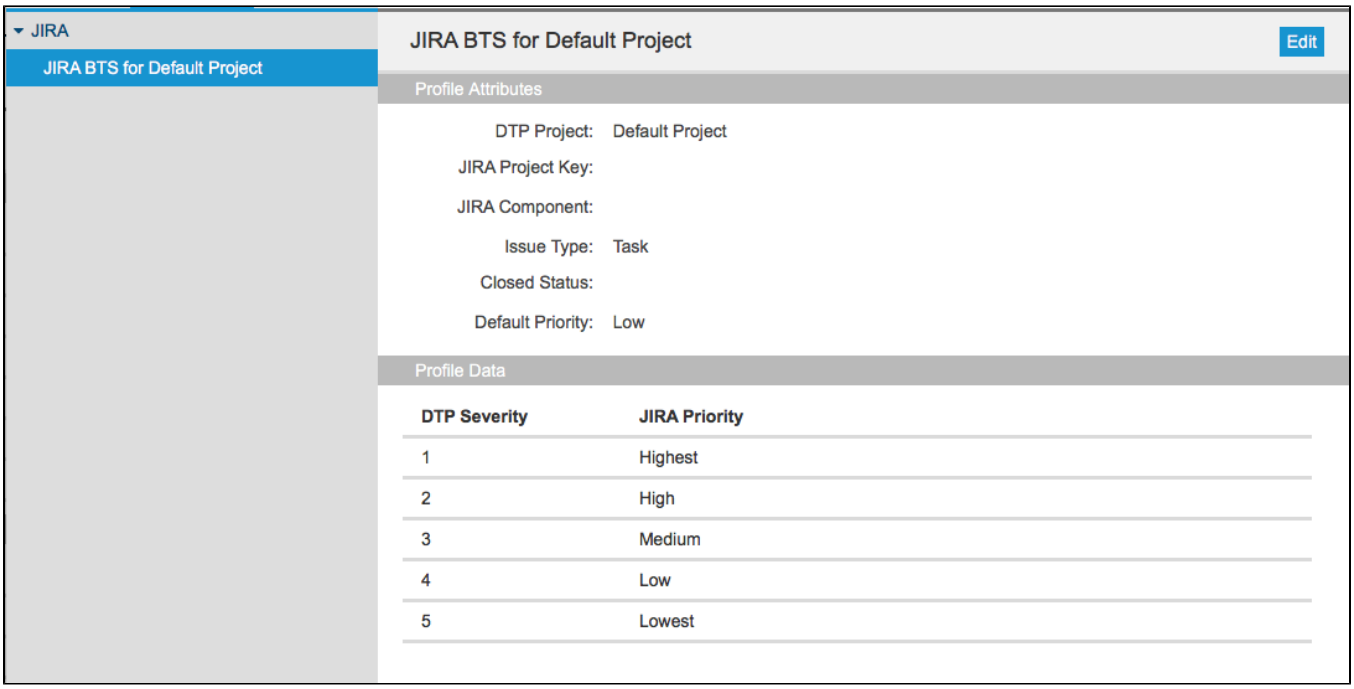
Click **Edit** to to configure the model. The following attributes should be configured:

Model name	The model should be named JIRA. This is the default name of the model.
Profile name	The profile name must match the name specified in the Profile Search node when you configure the flow. See Flow Configuration.
DTP Project	The DTP Project that the profile belongs to. If you do not want to associate the flow with specific DTP projects, you can delete the search parameter from the Model Settings node. See Flow Configuration.
JIRA Project Key	JIRA Project key (use SP for Sample Project). Normally, the key is defined along with a JIRA ID/issue number. For example, if the project key is Sample Project and the JIRA issue number is 1234, you would enter SP-1234.
JIRA Component	Name of the JIRA component for the issue.
Issue Type	Bug and Task are currently supported.
Closed Status	The flow attempts to transition a Bug to this status when the corresponding violation is fixed.
Default Priority	The priority set for all new Tasks, as well as new Bugs with violations that don't match the profile model.

All predefined values like Issue Type or Status are case sensitive. Please be sure to specify the correct case.

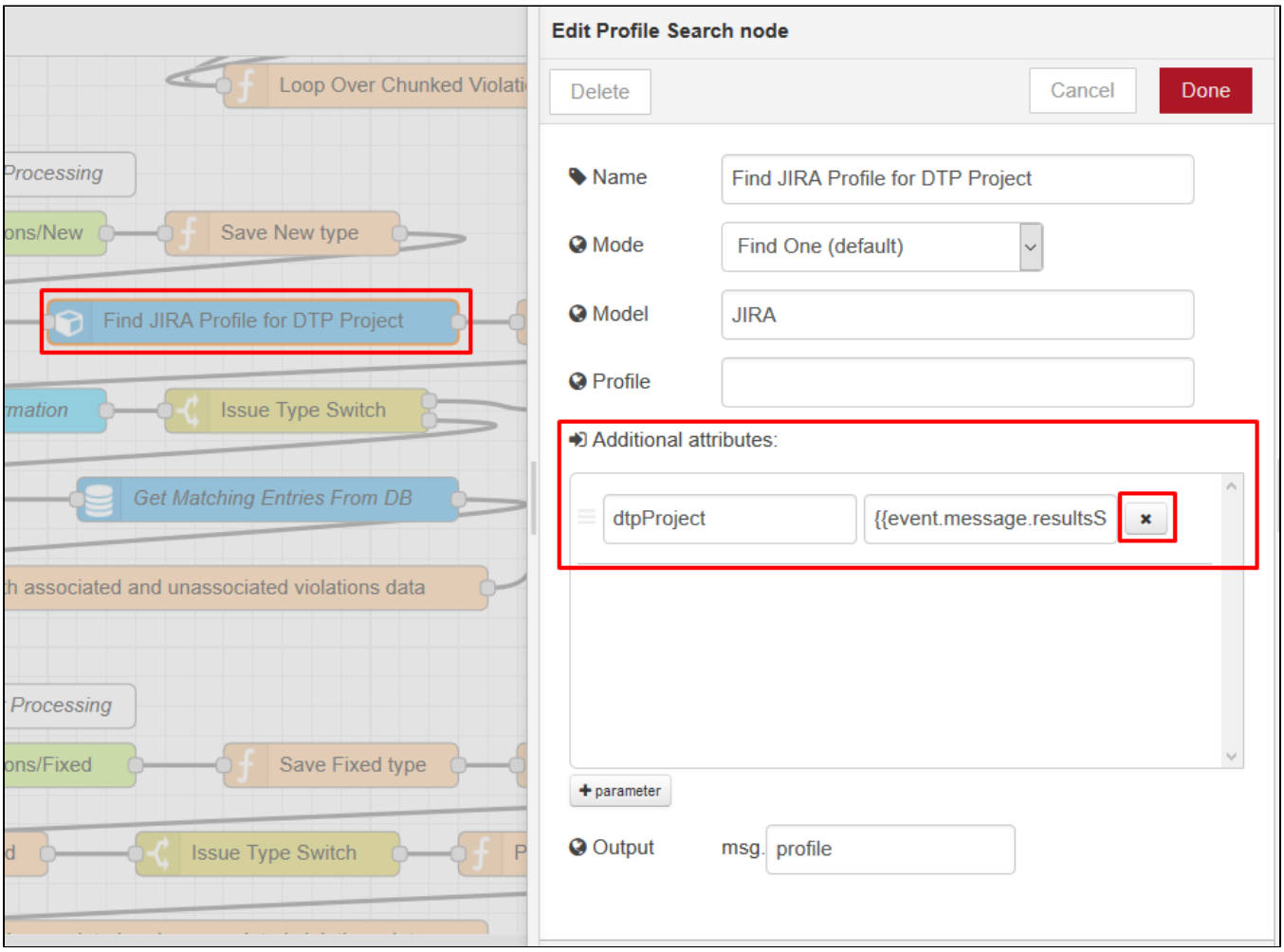
Profile Data

Profile data maps DTP static analysis violation severities to JIRA priorities. The sample profile data is provided for DTP's Default Project. This profile will import default severity and priority data for the project. You will need to provide data for project in DTP that you want to associate with JIRA.



Configuring One Model For All JIRA Projects

The sample flow expects a model to be set for each DTP Project. If you would rather have a single model that can be used across all DTP projects and report to one JIRA project, configure a model with an empty DTP Project and remove the DTP Project parameter from the Find JIRA Profile for DTP Project node.

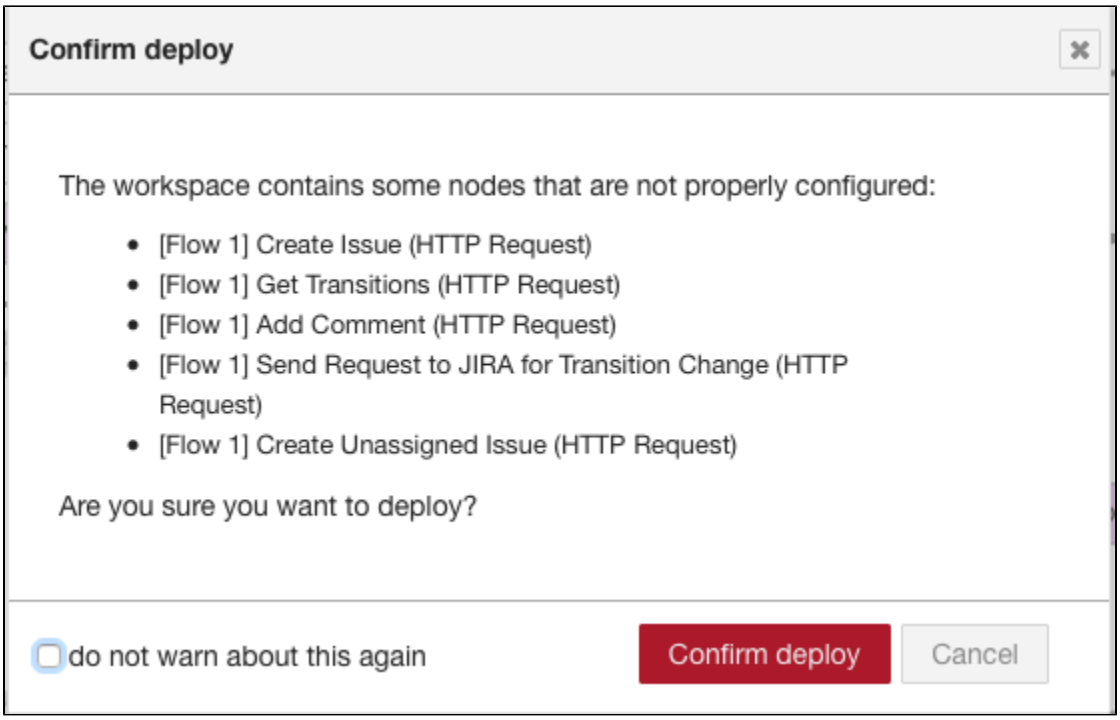


This will return a single JIRA profile that contains all necessary options for all projects.

Flow Configuration

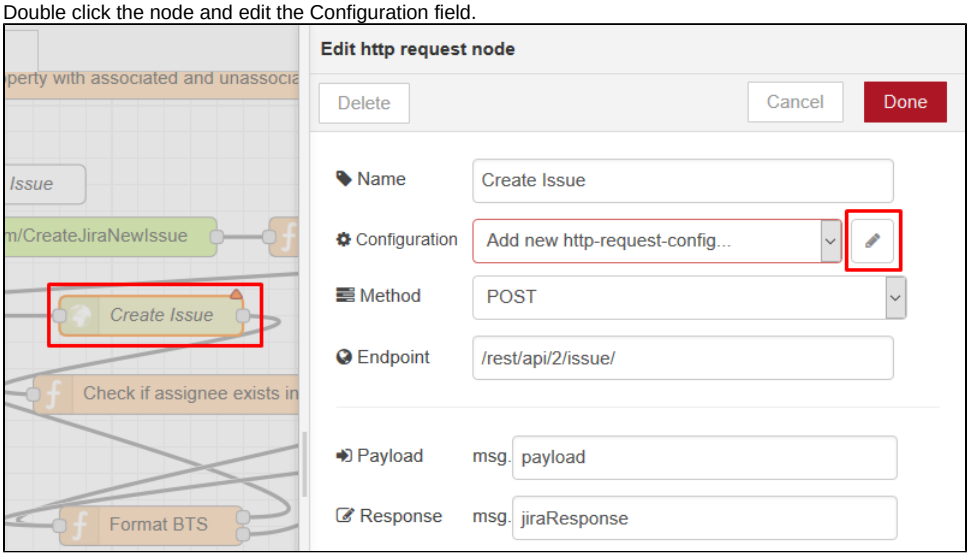
Server and Credentials

After the flow is imported, all HTTP Request nodes need to be modified to point to the correct JIRA server with valid credentials. Be sure to properly configure all the HTTP Request nodes with a JIRA user that has permissions to add and modify issues for the project. You can deploy the unconfigured flow and Extension Designer will list the HTTP Request nodes that need to be configured:

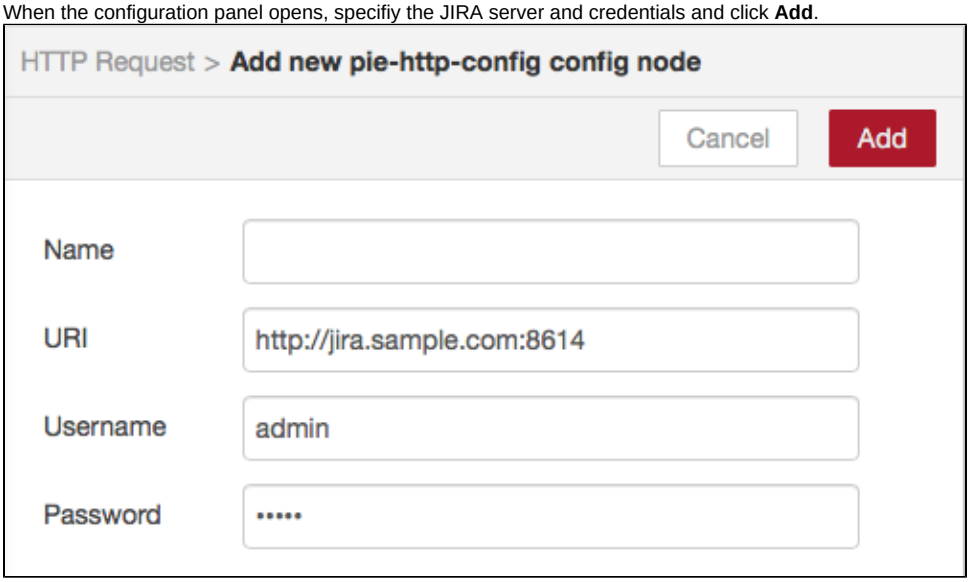


To configure the HTTP Request nodes:

1. Double click the node and edit the Configuration field.



2. When the configuration panel opens, specifiy the JIRA server and credentials and click **Add**.

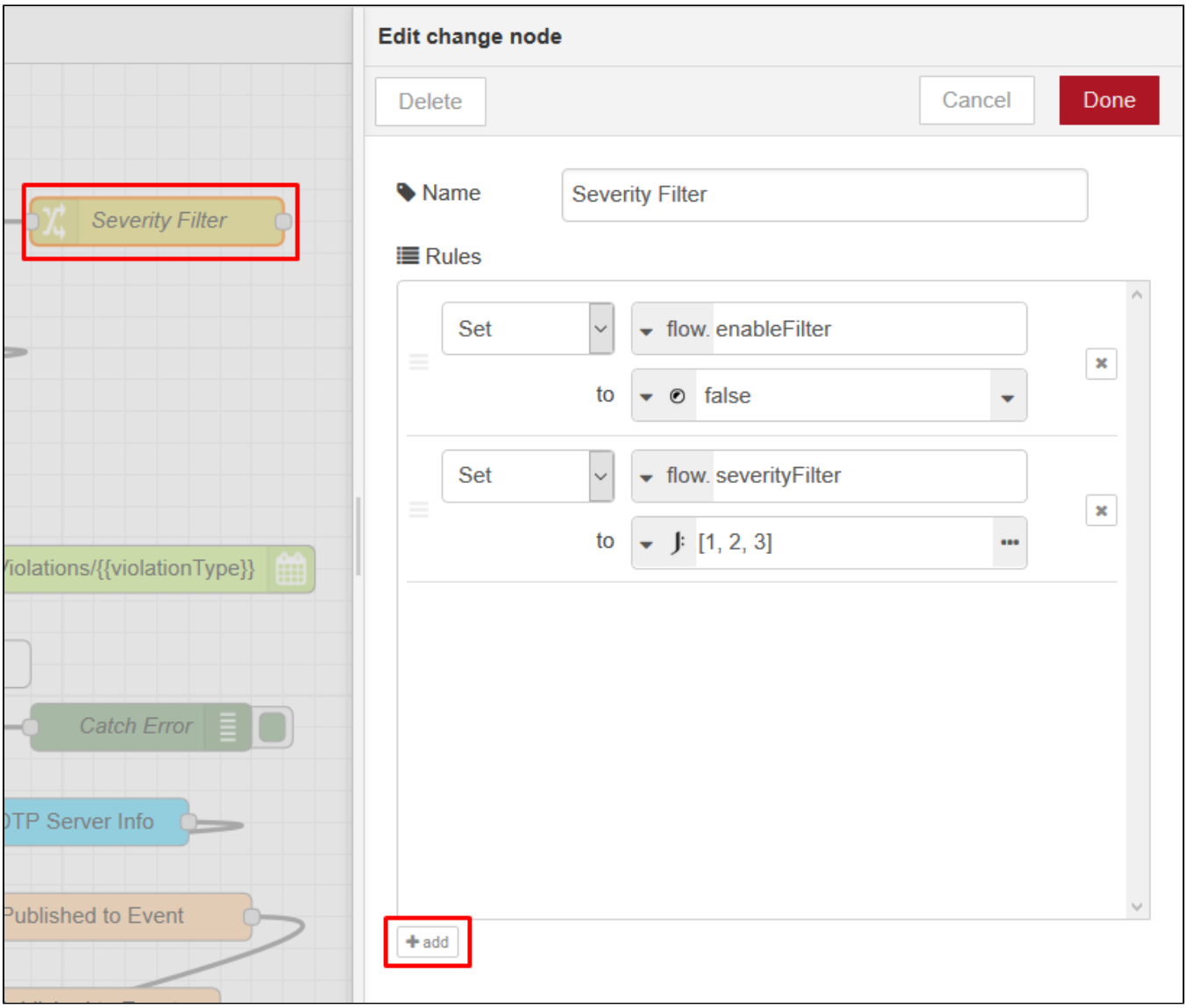


> **Create a Dedicated JIRA User for Extension Designer**
>
> If JIRA detects that a user is making requests from multiple machines, it may lock the user out. We recommend creating a dedicated user for Extension Designer to avoid being locked out by JIRA.

3. Click **Done** when panel closes and repeat for the other HTTP Request nodes.

Filtering Violations by Severity

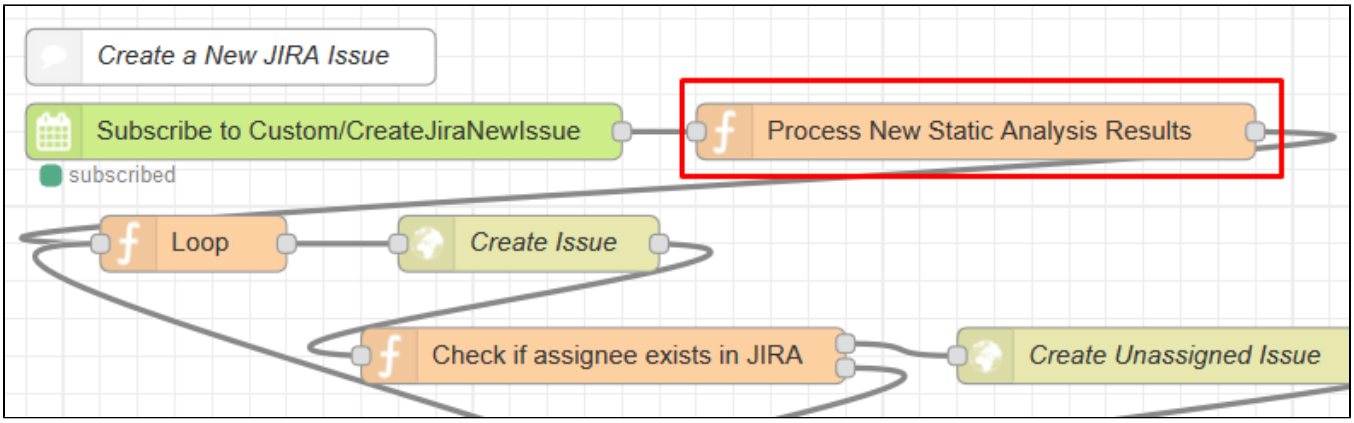
You can configure the Severity Filter node to process specific violation severities.



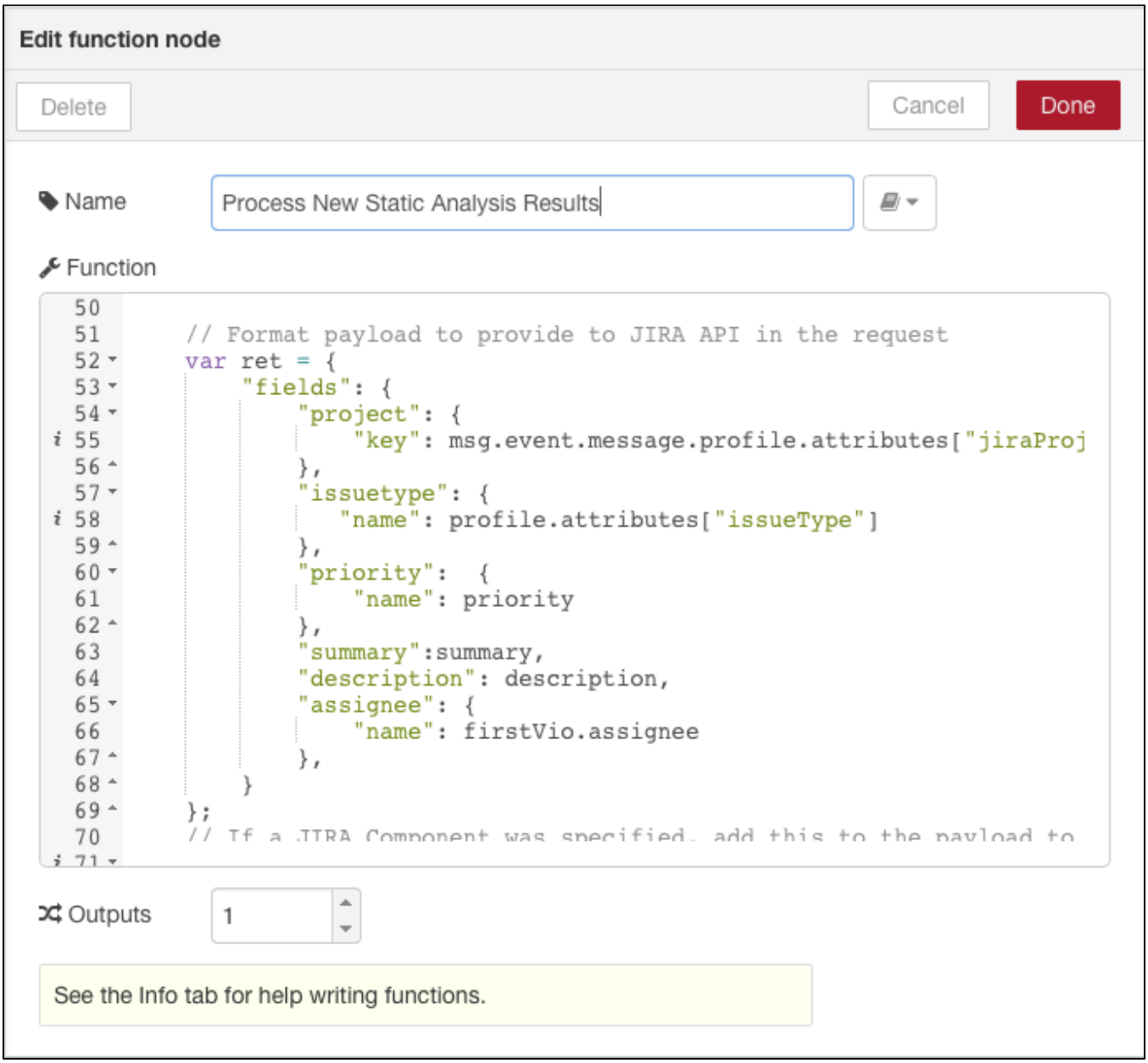
Configure patterns for filtering violations in the Rules field. Click **+add** to configure additional rules as necessary.

Configuring New JIRA Fields

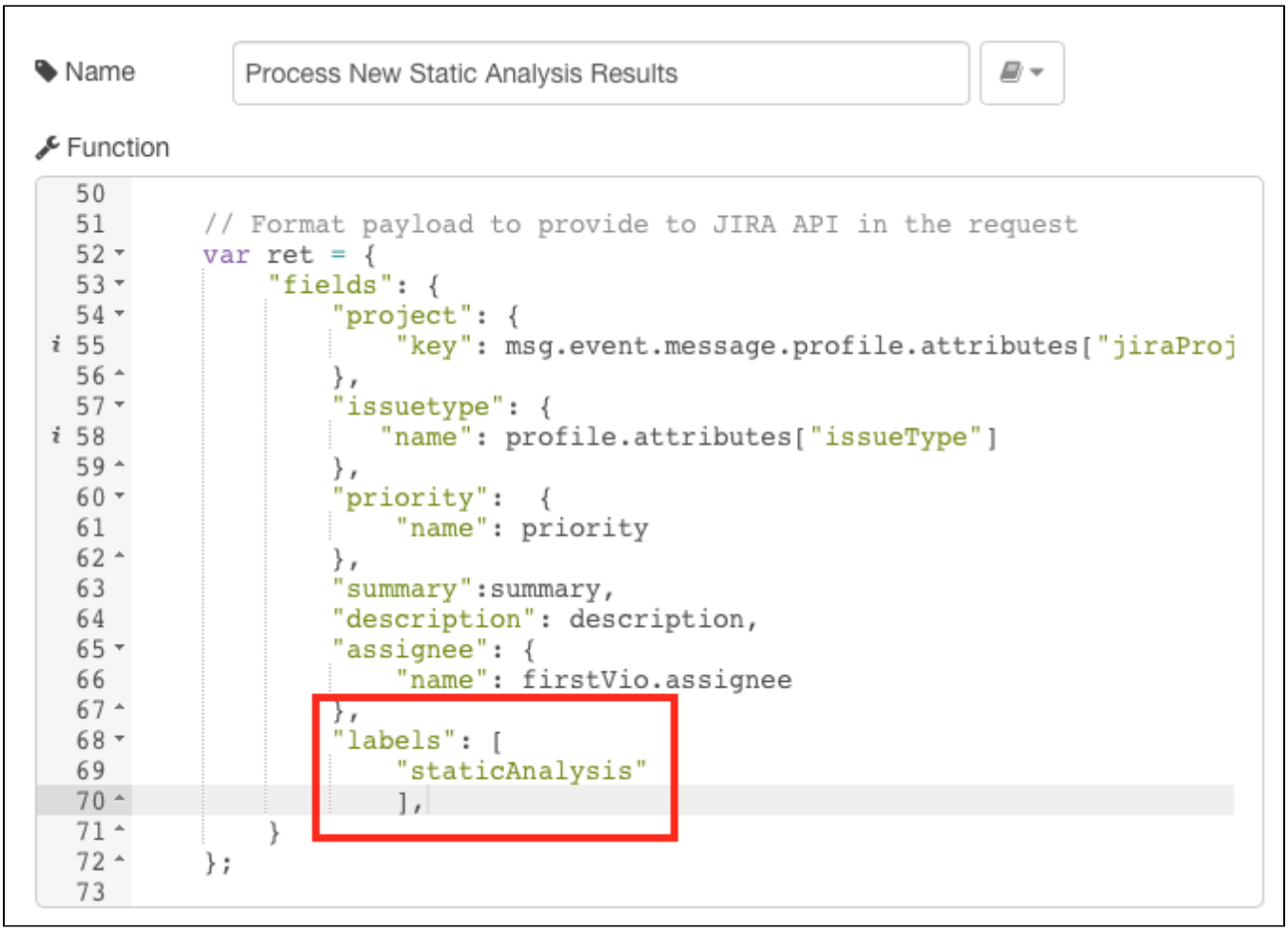
You can configure the JIRA summary or description, as well as add additional fields when creating JIRA issues, by configuring the Process New Static Analysis Results node in the Create JIRA New Issue section.



Double-click the node and modify the `ret` object.



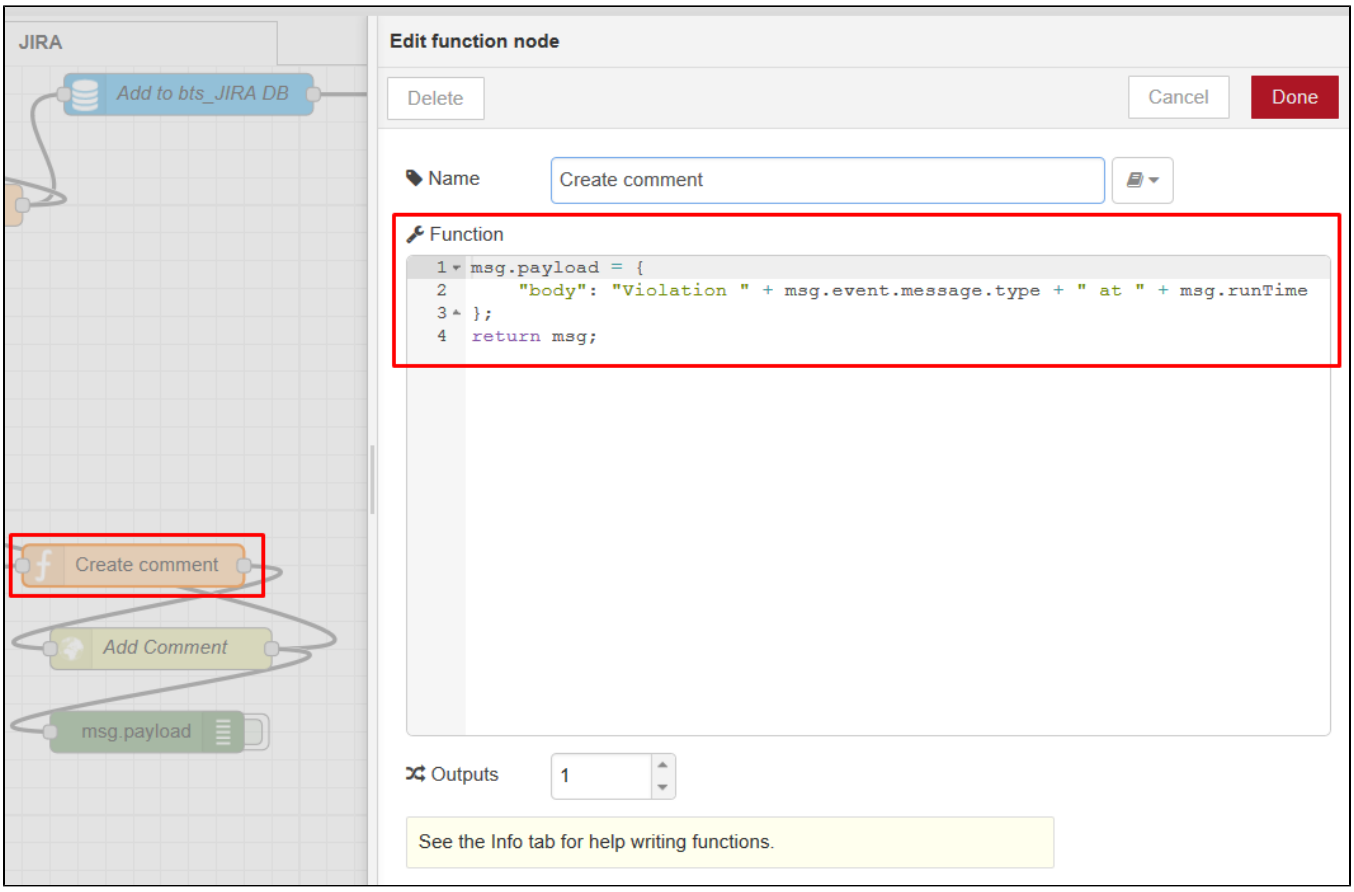
The JIRA Importer flow use the JIRA REST API to create and modify issues in the JIRA bug tracking system. The values in this object correspond to the expected payload of the JIRA POST /rest/api/2/issue endpoint. For example, if you want to add “staticAnalysis” as a label for newly-created JIRA issues, you would add the following to the `ret` object.



Subsequent JIRA issues that are created will have the “staticAnalysis” label.

Configuring Issue Comments in JIRA Bugs

To change what comment is added when a JIRA bug is fixed, modify the msg object in the Create Comment node.



When violations are fixed the corresponding JIRA bugs will include the appropriate message.