
NodeConductor Assembly Waldur Documentation

Release 2.8.0

OpenNode

Nov 17, 2017

Contents

1	Guide	3
2	API	5
3	Developer	7
4	License	9
5	Indices and tables	11

Plugin for interaction and management of Atlassian JIRA.

1.1 Installation

1. Install Waldur Core
2. Clone Waldur MasterMind repository

```
git clone https://github.com/opennode/waldur-mastermind.git
```

3. Install Waldur MasterMind into virtual environment

```
cd /path/to/waldur-mastermind/  
python setup.py develop
```

1.2 Setup backend Atlassian

1.2.1 NodeConductor Settings configuration

1. Define active backend.

```
# For Service Desk  
WALDUR_SUPPORT.update({  
    'ACTIVE_BACKEND': 'nodeconductor_assembly_waldur.support.backend.  
↪atlassian:ServiceDeskBackend',  
})  
# For JIRA  
WALDUR_SUPPORT.update({  
    'ACTIVE_BACKEND': 'nodeconductor_assembly_waldur.support.backend.  
↪atlassian:JiraBackend',  
})
```

2. Setup connection. Define server URL and user details to connect JIRA or Service Desk to NodeConductor:

```
WALDUR_SUPPORT['CREDENTIALS'].update({
  'server': <server URL>,
  'username': <Atlassian user username>,
  'password': <Atlassian user password>,
})
```

3. Project setup. Define project key.

```
WALDUR_SUPPORT['PROJECT'].update({
  'key': <project key>,
})
```

4. Project issues setup.

4.1. Make sure that selected project supports registered types of issues: `WALDUR_SUPPORT['ISSUE']['types']`.

4.2. Make sure that project issues have fields that corresponds to 'impact_field', 'reporter_field', 'caller_field'. It is possible to override default field names:

```
WALDUR_SUPPORT['ISSUE'].update({
  'impact_field': <issue impact field name in JIRA or Service desk>,
  'reporter_field': <issue reporter field name in JIRA or Service desk>,
  'caller_field': <issue caller field name in JIRA or Service desk>,
})
```

1.2.2 Web hook installation

It's possible to track updates of JIRA issues and apply them to Waldur immediately.

An instruction of JIRA configuration can be found at <https://developer.atlassian.com/jiradev/jira-apis/webhooks>

Step by step guide:

1. Log in to JIRA as administrator
2. Click on a cogwheel in the upper right corner and pick 'System'.
3. Scroll down to the lower left corner and find a "WebHook" option under the Advanced tab.
4. Now click on "Create a Web Hook" You will be presented with a web hook creation view. There are only 3 mandatory fields - Name, Status and Url.
 - 4.1 Name your hook
 - 4.2 Select whether you want to enable it. It can be disabled at any moment from to the same menu.
 - 4.3 Configure a url to send a "POST" request to. For instance: <http://nodeconductor.example.com/api/support-jira-webhook/> It is not needed to add any additional fields to request.

Note: In case of VirtualBox localhost usually is 10.0.2.2. So the complete url will be next: <http://10.0.2.2:8000/api/support-jira-webhook/>
 - 4.4 Add a description.
 - 4.5 Please make sure you've picked 'created, updated and deleted' actions under 'Events' section. No need to to check Comments events, they will be synced by the issue triggers.
 - 4.6 Save configuration.

CHAPTER 2

API

3.1 How to install JIRA

1. Download JIRA Install www.atlassian.com/software/jira/download

1.1 Please make sure the system for installation is prepared and you have selected the right version of JIRA.

3.1.1 For Windows:

2.1. Run an installer and follow the installation wizard

3.1.2 For Linux:

2.1. Place downloaded image on the server

2.2. Make the package executable:

```
chmod a+x atlassian-jira-software-X.X.X-x64.bin
```

2.3. Run the installer:

```
sudo ./atlassian-jira-software-X.X.X-x64.bin
```

3.1.3 Post installation steps

3. After JIRA is installed open the port displayed after installation is finished. Usually it is 8080.

4. Configure JIRA by following an installation guide in your favourite browser.

PS. If you are installing JIRA on a virtual machine please make sure that port forwarding is configured.

CHAPTER 4

License

NodeConductor Assembly Waldur plugin is open-source under MIT license.

CHAPTER 5

Indices and tables

- `genindex`
- `search`