Report Configurations

Each **Reports Scheme Entity** represents a **Report Configuration**.

Report Scheme Entities are created by capturing the parameters used directly from the Reports screen.

On this page:

- Capturing Report Configurations
- Configuring Report Configurations
- Sharing Report Configurations
- Contextualizing Report Configurations
- Velocity Context

Capturing Report Configurations

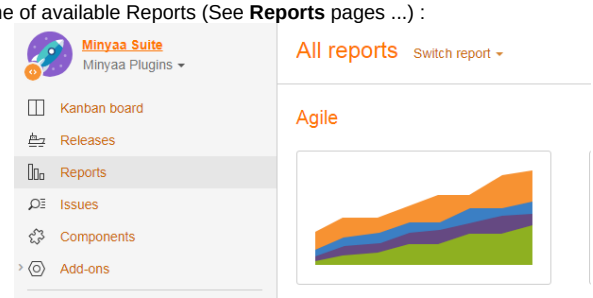
In each Report screen (Result of the action **ConfigureReport.jspa**), ***Minyaa Predefined Report*** appends an icon after the Report's Title, to allow to Capture the Report's configuration.



> **Info ...**
>
> *This Icon is available only for allowed users. To be granted to catch Report Configuration, the user must have permission **Catch Report Configurations** for at least one project. (Implemented only for JIRA 5.x and over)*

Process as follow ...

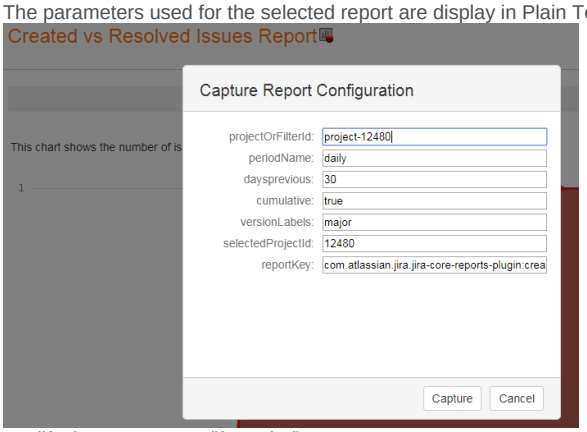
1. Generate one of available Reports (See **Reports** pages ...) :



- ***JIRA*** comes with default set of Reports
- ***Minyaa*** comes with its own set of Reports (Minyaa Time Reports,)
- or any Reports provided by the JIRA Community

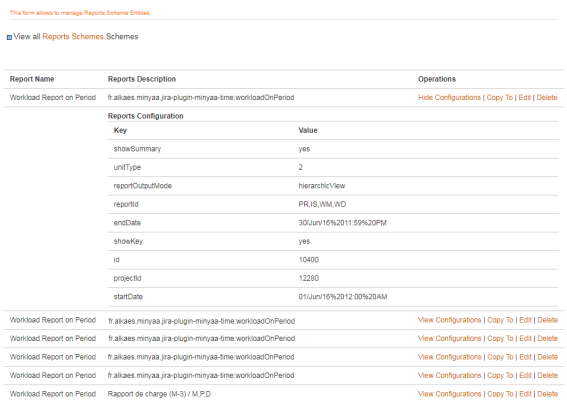
2. When the report result is displayed and fit with your needs :

- Click on Capture Icon ... 
- The parameters used for the selected report are display in Plain Text...



- Modify the parameters (if needed)
- an click on **Capture** button ...

For each captured **Report Configuration** are saved into the Main Reports Scheme...



Configuring Report Configurations

Now, a set of Captured Report Configurations is available in the Main Reports Scheme.

It is time to create your own **Reports Scheme**.

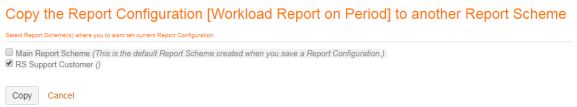
Process as follow ...

Create a new Reports Scheme ...

See Add and Edit Reports Scheme section.

Append **Report Configuration** to your new **Reports Scheme** ...

1. Go to View Reports Scheme Entities page for the **Main Report Scheme**
2. For each wanted Captured Report Configuration :
    - Click on **Copy**



    - Select target Report Scheme(s) ... (normally the just your new Reports Scheme)
    - End by clicking on **Copy**
3. All copied **Report Configurations** will be entirely cloned in your Reports Scheme.

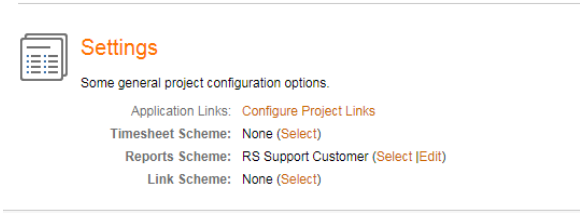
Complete your Report Scheme definition

- In order to give more sense to your **Report Configurations**, you will have modify their description (Name is not editable) following the applied parameters...
    - *"Worklog for a User U1 for the a Project P1",*
    - *"Recently Created Issues Report (All Issue on last 30 Days)"*
    - *...*

Sharing Report Configurations

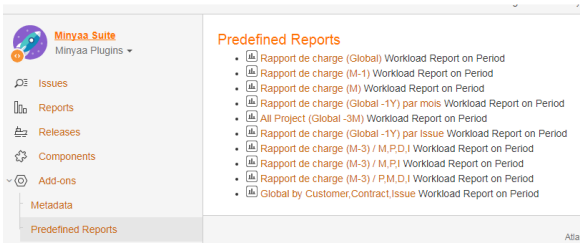
Now that your Reports Scheme is configured with a set of Report Configurations, you are able to share it by associating the scheme with concerned Projects.

Go to Project Administration page



- Click on **Select** of the line **Report Scheme** in **Plugin Scheme** section
- Select the wanted **Report Scheme** to use for this Project ...

2. Now, a new menu ***Predefined Reports*** is available in the Side bar ...



> **Warning ...**
>
> All report are generated with the initial context used when the Configuration has been captured.
> Minyaa Reports allows you to Contextualize your Scheme ...

Contextualizing Report Configurations

To contextualize your reports, **Minyaa** uses different mechasnism depending on the Parameter Types ...

Types	Used mechasnims
Date, Datetime	Relative Date Operator implemented for Minyaa Time's' Reports
Project/Filter	Velocity templating Project dedicated,
User or any others ...	A generic Velocity templating.

Process as follow ...

1. Go to Edit screen of your **Report Scheme Entity** (Report Configurations)
2. Update Report Configurations's values following each parameter types *(where is has sense!)* ...

- In case of Date or Datetime parameter types ...
  Use the Relatives Dates.
- In case of Project/Filter parameter types
  Use Velocity Project variables provided. The Contextualizer will heck that prefix **project-** is respected. See below ...
- In case of User or Project parameter types
  Use Velocity variables provided. See below ...

3. Update Report Description in order to match with the contextualized paraemeters ...

- *"My Worklog for the Current Project",*
- *"Recently Created Issues Report (Current Project on last 30 Days)"*
- *...*

Velocity Context

The Velocity Context will contains ..

Keyword	Object or Value
user	Current logged in User Object
loggedUserName	Current User Name
selectedProjectId	Current Select Project Id
project	Current Select Project Object
locale	Locale of logged in User
...	Many other provided by the **JiraVelocityUtils** class. Content will depend on JIRA Version ... See Atlassian Documentation : • For JIRA 4.x... JIRA Velocity Context • For newer JIRA : Contents of the Velocity context

Below a capture of Velocity Context for JIRA 4.3.1. To complete ...

Keyword	Object or Value
i18n	com.atlassian.jira.web.bean.BackingI18n
dateutils	com.atlassian.core.util.DateUtils
outlookdate	com.atlassian.jira.web.util.OutlookDate
authcontext	com.atlassian.jira.security.JiraAuthenticationContextImpl
permissionCheck	com.atlassian.jira.web.bean.PermissionCheckBean
req	com.atlassian.jira.util.JiraWebUtils$1
currentCalendar	com.atlassian.jira.util.JiraVelocityUtils$LazyCalendar
jiraUserUtils	com.atlassian.jira.user.util.UserUtilImpl
customFieldManager	com.atlassian.jira.issue.managers.DefaultCustomFieldManager
textutils	com.opensymphony.util.TextUtils
jirautils	com.atlassian.jira.util.JiraUtils
projectManager	com.atlassian.jira.project.CachingProjectManager
applicationProperties	com.atlassian.jira.config.properties.ApplicationPropertiesImpl
buildutils	com.atlassian.jira.util.BuildUtils
constantsManager	com.atlassian.jira.config.DefaultConstantsManager
mailPluginsHelper	com.atlassian.jira.mail.JiraMailPluginsHelperImpl
userformat	com.atlassian.jira.plugin.profile.DefaultUserFormatManager
webResourceManager	com.atlassian.plugin.webresource.WebResourceManagerImpl
jirakeyutils	com.atlassian.jira.util.JiraKeyUtils
userutils	com.atlassian.core.user.UserUtils
velocityhelper	com.atlassian.jira.util.JiraVelocityHelper
urlModeAbsolute	ABSOLUTE
externalLinkUtil	com.atlassian.jira.web.util.ExternalLinkUtilImpl
urlcodec	com.atlassian.jira.util.JiraUrlCodec
currentMillis	
xmlutils	com.atlassian.core.util.XMLUtils
atl_token	

requestContext	com.atlassian.jira.util.velocity.SimpleVelocityRequestContext
keyboardShortcutManager	com.atlassian.jira.plugin.keyboardshortcut.CachingKeyboardShortcutManager
map	com.atlassian.core.util.map.EasyMap
baseurl	