

Redmine

Redmine is an open source project management web application. It allows us to organize our project documentation, log important data about changes to projects, organize developer teams for specific projects, and create issue reports and feature suggestions. Redmine is fantastic because it allows non-developers to play a role in the developmental process by suggesting improvements on current projects. Also, Redmine is readily accessible so the source code for a project can easily be viewed from any of the firewalls in our office without having to ssh into our development server.

Contents
1 Accessing Redmine
2 Issue Reporting
3 Project Monitoring
3.1 Overview
3.2 Issues
3.3 Repository

Resnet Redmine

Home

Welcome to the ResNet Redmine!

We use Redmine for two major purposes: Bug / support tracking, and pr

Issues

We track all bugs for our services, servers, and workstations throughout

You may also use Redmine to request a feature (ex: add a program to a

Documentation

Redmine is our central documentation hub for all of our projects. Please : remediation news.

Redmine is a pivotal tool used in the ResNet office for tracking bugs found in the projects we develop. It also allows techs to suggest features to add to current projects.

Accessing Redmine

A link to Redmine is available on the ResNet wiki. Alternatively, Redmine can be accessed by entering the URL <https://rrc.neu.edu/redmine/>.

After entering Redmine, techs will be prompted to input their ResNet credentials (username: first_initial.last_name). If there is an issue logging

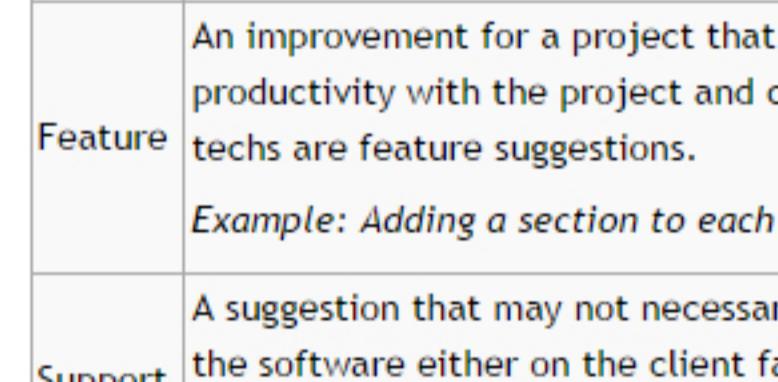
into Redmine, contact a full time staff member; all techs should have access to Redmine in order to report bugs and suggest additional features.

Techs can check if they have successfully logged in by checking the top-right corner of the banner. When signed in, the text in the top right should read "Logged in as USERNAME" followed by a link to account settings and a signout option.

The screenshot shows the ResNet Redmine homepage. On the left is a sidebar with links like Staff Roster, ResNet Projects, Warranty Information, Recent changes, ResNet/NEU Internal Links, ServiceNow!, Dashboard, Timesheet, Online Reg Tool, Redmine, Driver Agent, Drop Tools, NURes-devices, External Links, and Toolbox. The main content area has a banner for 'ResNet News and No' with a note about FTS shifts. Below the banner is a section titled 'Welcome New Trainee' with a note about new training classes.

Issue Reporting

The ResNet office is always changing, new operating systems are released every few years, the internal layout gets altered every so often, office policies rise and fall; in order to keep up with such a dynamic workplace, all techs must be able to report issues in need of attention to developers as efficiently as possible.

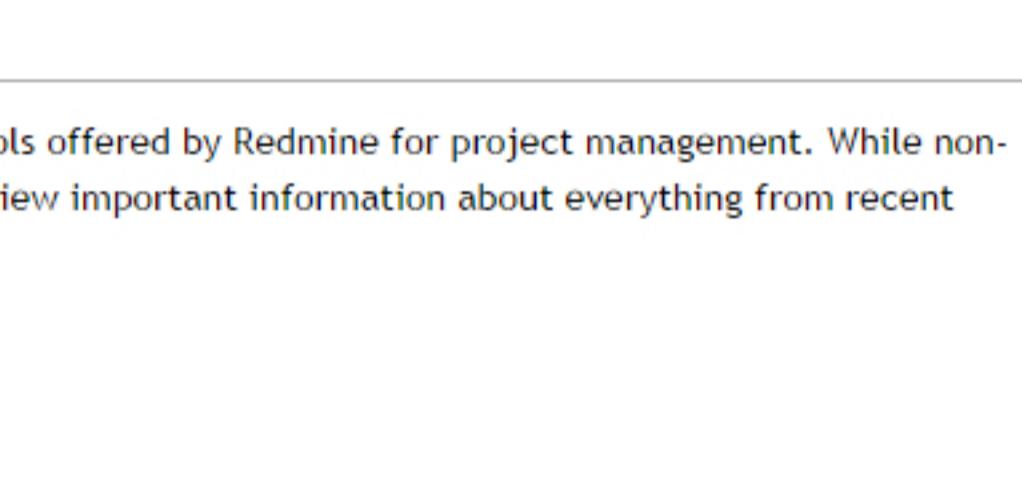


To report an issue, the first thing a tech must do is determine which project the issue is related to. Clicking on the "Projects" link in the Redmine banner takes the user to a page containing a list of all the projects currently in development at ResNet.

After selecting the proper project, a report can be filed by clicking the "New Issue" tab. This will lead to a submission form that will prompt for a tracker label for the issue report, a title, and a description of the issue.

After the tracker, subject, and description components of the form have been completed, the form can either be submitted or previewed. Previewing the report will produce an image below the form depicting how it will look once submitted. There is no need to alter the status, priority, or due date of the issue. The "Assignee" field should also be left blank because a full time staff member must approve any issue reports before they can be worked on.

Submitting the issue form will notify the manager of the project in question, the issue will then either be approved and assigned to a developer or rejected.

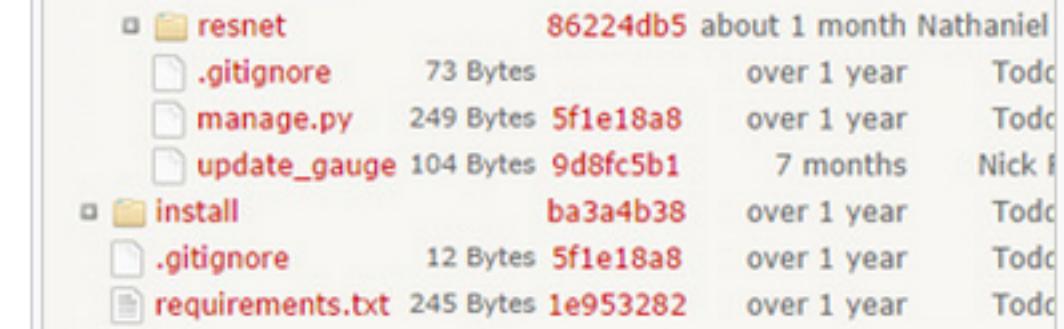


Below is a table containing a description of each type of tracker available in the issue report.

Type	Description
Bug	An issue or malfunction that is preventing the project from being productive or usable. These issues generally receive the highest priority because having them means the project isn't meeting the basic specifications given to it. <i>Example: A webapp reporting incorrect information or crashing.</i>
Feature	An improvement for a project that is otherwise completely functional. Features generally are meant to improve productivity with the project and can vary in how necessary they actually are. The majority of issue reports produced by techs are feature suggestions. <i>Example: Adding a section to each case saying whether or not an A+ certification is needed to work on the machine.</i>
Support	A suggestion that may not necessarily help improve the functionality of a project but perhaps improves the aesthetics of the software either on the client facing side or in the backend. <i>Example: Having a competition to see who can design the best CSS template for a webapp.</i>
Change	An alteration to an existing feature to improve the software's aesthetics or comprehensibility. This is similar to support but offers suggestions on existing, functioning features instead of offering up a new idea. <i>Example: Changing the color scheme of the ResNet Dashboard.</i>
Request	A miscellaneous category for requests that do not fit entirely with any of the previous categories of trackers. Most issue reports do not generally end up here but it still exists for the few instances where it is needed. <i>Example: Displaying additional data in the ResNet Dashboard banner.</i>

Project Monitoring

Bug reporting and feature suggestion are just a few of the numerous tools offered by Redmine for project management. While non-developers only have access to reporting tools, developers are able to view important information about everything from recent changes to the various issue requests available for assignment.



Overview

From the "Overview" tab, two pieces of information are apparent: the number of issues, sorted by category, and the developers and manager for the project. The issue tracking section gives a quick view of how many bugs, feature suggestions, etc are still open and how many have been reported ever and also offers a calendar view of when these issues were resolved or created. The members section provides a list of developers who have worked on the project. This is important because these developers are the best resources for questions regarding the code's functionality or organization.

Issues

The "Issues" tab provides a much more detailed look into what issues have been reported. Organized in a table, this tab contains important information like who each issue is assigned to and when each report was created. Any developer for a project can assign themselves to an issue once it has been approved by the project manager so it saves time if there is a quick way to see which issues have assignees and which don't. There are also options to sort by issues of specific categories or assignees.

#	Tracker	Status	Priority	Subject	Assignee
1172	Feature	New	Normal	Dropdown to show what kind of tech should work on case	
1165	Bug	New	Normal	Tech assignment is next to "None" in parentheses	Nathaniel Dempkowski
1162	Feature	New	Normal	Icon to show when customer replies to ticket	
1160	Feature	Assigned	Low	Custom Locations	Nathaniel Dempkowski
1159	Feature	Assigned	Normal	Tweak the orange (not updated in three days) code	Nathaniel Dempkowski
1157	Feature	New	Normal	Take the numbers from the top right on the dashboard and log/graph the data	Nathaniel Dempkowski
1154	Feature	Assigned	Normal	Add A+ techs field to current techs list	Nathaniel Dempkowski
1148	Feature	Assigned	Normal	"Queue" location	Nathaniel Dempkowski

A list of issue reports

Top: Current master branch, Bottom: revision history.

Above the revision history is a view of the current master branch for the project. This view shows the latest push that has affected each file and the date these files were last modified. By default, clicking on a file leads to a page showing the latest changes to the file. In order to view the actual contents of the file, a developer needs to either download the file or click the "view" option which opens up a read-only version of the document in the browser.

For security reasons, it is not possible to edit the source code of a project from within Redmine. To make changes, a developer needs to ssh into the ResNet development server and access the project files from there. The view of the repository from Redmine, however, is useful in situations where only a quick glance of the code is needed to determine how some component of a project functions.



git-dashboard / django / dashboard / tests.py @ master

History | View | Annotate | Download (383 Bytes)

1
2 This file demonstrates writing tests using the unittest module.
3 when you run "manage.py test".
4
5 Replace this with more appropriate tests for your application.
6
7
8 from django.test import TestCase
9
10
11 class SimpleTest(TestCase):
12 def test_basic_addition(self):
13 """
14 Tests that 1 + 1 always equals 2.
15 """
16 self.assertEqual(1 + 1, 2)

Viewing a file from Redmine.

Categories: Malware Remediation | ResNet Policies

This page was last modified on 24 October 2012, at 18:23.

This page has been accessed 257 times.

Privacy policy About ResNetWiki Disclaimers Mobile view

