ChiliProject - Feature # 767: Remove the artificial separation between Settings and Configuration

Status:	Open	**Priority:**	Normal
Author:	Holger Just	**Category:**	Refactoring
Created:	2011-12-11	**Assignee:**	
Updated:	2011-12-13	**Due date:**	

Remote issue URL:

Affected version:

Description:

Following my efforts to make the Configuration API less bloated and error-prone in #233 I thiught a bit more about the general concept of settings. My conclusion (which differs from my previous statements) is that we should fully merge Settings and Configuration. With Settings being the canonical implementation.

In that general area of Settings, I'd like to cleanup some warts there too which came up after having tried to extend it in plugins.

I started working on the @Setting@ model to accomplish a couple of features:

* Save settings in correctly typed columns in the database and in the code (as the YAML serializer always sucked, esp. for ints and booleans) (_Status: Prototype_)
* Introduce namespaces (_Status: Prototype_)
** This is especially useful for plugins that don't need to live in a single large hash anymore.
* Merge configuration and settings (_Status: Work in Progress_)
** Forcefully splitting up the configuration into web-based settings and file-based configuration is confusing to users
** *Everything* should be settable from the UI (except for the database.yml)
** The code I wrote for #233 will become part of the @Setting@ model
* Allow the admin to enforce certain setting which can not be changed from the UI (_Status: Work in Progress_)
** This allows admins to enforce things like the URL, path settings, or shellout commands
** These settings can subsequently only be changed by editing a YAML file. By default, there aren't restricted settings
** This helps in environments where only partially trusted people have admin rights to handle custom fields, projects, ...

The work will probably not be finished before the first Beta / Feature Freeze of 3.0. So I'd try to keep everything backwards-compatible so that we could introduce it in 3.1. Currently this looks feasible.

Associated revisions

2008-03-05 12:14 pm - Jean-Philippe Lang

Display wiki syntax quick ref link within the jstoolbar (closes #629, #767).
Added named links syntax on quick ref (closes #766, #778).

git-svn-id: http://redmine.rubyforge.org/svn/trunk@1190 e93f8b46-1217-0410-a6f0-8f06a7374b81

History

2011-12-11 03:40 pm - Holger Just

The single most complex part of this issue is probably to come up with a flexible-enough-but-still-usable UI for the email configuration. But once it's there, it tremendously helps to un-confuse users and helps easing up the installation procedure to:

# Download ChiliProject
# Edit @database.yml@ and create database
# @bundle install@
# Start server

The namespacing component helps us to understand and clean up dependencies between components and someday to break them up hopefully.

2011-12-12 08:06 pm - Eric Davis

> Allow the admin to enforce certain setting which can not be changed from the UI (Status: Work in Progress)

This part is the biggest issue for me, since there are no separation between admins.

I'm hesitant to think this feature is a good idea but I'll wait until you have something to review.

2011-12-12 08:21 pm - Holger Just

@Eric: The whole point of this feature is that you as Ãœber-admin (i.e. someone with root on the server, that's the person I meant with "admin" in the above quote) should be able to prevent mere web admins from setting dangerous options. That means, the Ãœber-admin should be able to create a yaml file with enforced settings which can then not be changed from the web UI anymore.

This would work on top of the basic default to all options settable from the web. That way, simple installations can start right of without having to edit config files. On the other hand we are able to support advanced installations where there could be any settings enforceable. That way, you have at least the security of the current @configuration.yml@ approach (probably even more as more settings can be locked) but you gain more user-friendlyness to "casual" admins/users.

2011-12-13 07:53 pm - Eric Davis

Still have the same concerns and will have to see an implementation.

You're just "moving" all the config into settings and adding another layer with a "restricted_settings.yml" instead of "configuration.yml". The way I see it: "configuration.yml" is the system settings already and are the restricted settings while Settings/settings.yml are the web-admin-editable ones.

But... I'm willing to see how the code is first. Pull request is where again? ;)