

Dashboard Framework Developer Documentation

Christopher Troup

David Wolfe

Dashboard Framework Developer Documentation

Christopher Troup

David Wolfe

Table of Contents

1. Installation	1
Getting Dashboard	1
Web Installer	1
2. Templating	2
Admin Interface	2
Programming API	2
3. DBRow	3
Programming API	3
init	3
make	3

List of Tables

2.1. Database Schema: Template Table	2
--	---

List of Examples

2.1. Getting A List of Templates	2
3.1. Quick Example	3
3.2. Quick Example	3

Chapter 1. Installation

Getting Dashboard

The best way to get the latest sources for Dashboard Framework is through GIT.

```
git clone git@github.com:minichate/dashboard.git  
  
cd dashboard.git  
  
git submodule update
```

This will download an installation of Dashboard complete with relevant submodules, etc. You should now set up your webserver to serve from this document root. In the following examples, we will assume that you have a virtual host set up at <http://dashboard/>.

Web Installer

Point your browser to <http://dashboard/buildtools/install/install.php>. The first couple steps will verify that you have the necessary components to allow Dashboard to run properly.

- PHP 5.2 or greater
- MySQL 5.0 or greater

It will also verify that the proper permissions have been set up. If any of the checks do not pass, please correct them before continuing onto the next step.

The database set will attempt to create a database and create the necessary tables for a minimal set up. Enter the database name, username, password and hostname that Dashboard should attempt to connect to.

Warning

This executes a **DROP TABLE** command, so it will remove old tables. Do not run the web installer if you have data that should not be deleted.

Once the installer has completed, you should be able to visit <http://dashboard/> and see some basic example text.

Access the administration interface through <http://dashboard/admin>. The default administration username is norex, and the password is D3vP@ss

Chapter 2. Templating

Admin Interface

Site templating in Dashboard is done through a Web GUI. The Templater module is accessible via <http://dashboard/admin/Templater>

Programming API

Table 2.1. Database Schema: Template Table

Column Name	Type	Description
module	text	The module name which the template belongs to
path	text	The name of the template. Example: <code>site.tpl</code> or <code>admin/list.tpl</code>
data	longtext	The actual contents of the template.
timestamp	timestamp	The date and time the template was last edited.
name	text	The human readable name of the template. Example: Site Template
Table name:	templates	

Example 2.1. Getting A List of Templates

```
$templates = Template::toArray(); // this will get a list of all templates
$templates = Template::toArray('CMS'); // list of all templates associated to CMS
```

Chapter 3. DBRow

Programming API

init

Initialize an object type. This function must be called on each class before the class can be used. It takes the class name of the object as its only parameter.

Example 3.1. Quick Example

```
<?php

class Example extends DBRow {
    public function createTable() {
        return parent::createTable ('example_table', __CLASS__);
    }
}
DBRow::init('Example');

?>
```

make

Fetches an object from the database and prepares it for display or editing.

Example 3.2. Quick Example

```
<?php

$user = DBRow::make(1, 'User'); // $user is now populated with the user object
                                   // that has the ID of 1.
echo $user->get('name'); // Prints out the users first name
$user->set('name', 'NewFirstName'); // Sets the first name to 'NewFirstNane'
$user->save(); // Save the changes

?>
```