

# Views for Developers

(and data geeks)

Bec White

[white@palantir.net](mailto:white@palantir.net)

DrupalCamp WI, July 2010

# What is Views?

"a tool for making lists of stuff"

Views describes the data in  
Drupal, and then layers  
filtering and display on top.

# Developing with Views:

- Export Views
- Call Views in your code
- Providing data to Views
- Plugins and handlers

# hook\_views\_api()

```
function example_views_api() {  
  return array(  
    'api' => 2,  
    'path' => drupal_get_path(...),  
  );  
}
```

- lives in example.module
- tells Views to look for two includes:
  - example.views.inc
  - example.views\_default.inc

# Exporting Views

(you should already be doing this)

# hook\_views\_default\_views()

```
function example_views_default_views() {
    $views = array();

    $view = new view;
    $view->name = 'example_view';
    $view->description = 'My list of stuff.';
    $view->base_table = 'node';
    $handler = $view->new_display('default', ... );
    $handler->override_option('fields', array( ... ));
    ...
    $views[$view->name] = $view;

    return $views;
}
```

# Calling Views from code

# What do you want from the View?

- **Rendered block of HTML**
- `views_embed_view('view_name', 'default')`
- **Array of View results**
- `views_get_view_result('view_name', 'default')`
- 
- **The View object**
- `views_get_view('view_name')`

# Views as a data model

Views uses `hook_views_data()`  
implementations to describe Drupal  
data

# Views as a query builder

takes the Views data model and  
builds SQL

# Views as a display layer

"squeezes results through the  
theme layer"

# Extending Views

- Describe your data to Views
- `hook_views_data()`
- Provide handlers for your data
- `hook_views_handlers()`
- 
- Manipulate View results for output
- `hook_views_plugins()`

# Handlers vs. Plugins

handlers interpret data; plugins deal  
with results

# Plugins

- Display plugins - where should the results of a View appear?
  - on a page, in a block, attached to another display
  - Views Attach - [http://drupal.org/project/views\\_attach](http://drupal.org/project/views_attach)
- Style plugins - change the "shape" of the output
  - lists, tables, grids
  - calendars, map layers, tag clouds
- Row plugins - change the "shape" of data in an individual row
  - generally "node" or "fields"

# hook\_views\_plugins()

```
function tablegroup_views_plugins() {
  return array(
    'module' => 'tablegroup',
    'style' => array(
      'tablegroup' => array(
        'title' => t('Grouped Table'),
        'help' => t('Displays results as cells in a table...'),
        'handler' => 'tablegroup_plugin_style_tablegroup',
        'theme' => 'tablegroup_view_tablegroup',
        'uses row plugin' => TRUE,
        'uses fields' => TRUE,
        'uses options' => TRUE,
        'type' => 'normal',
      ),
    ),
    'display' => array ( ... ), // looks similar
  );
}
```

# tablegroup\_plugin\_style\_tablegroup.inc

```
class tablegroup_plugin_style_tablegroup extends views_plugin_style
{
  function option_definition() {
    $options = parent::option_definition();
    $options['row_grouping'] = array('default' => NULL);
    return $options;
  }

  function options_form(&$form, &$form_state) {
    parent::options_form($form, $form_state);
    $form['row_grouping'] = array( ... );
  }

  function render() {
    $output = 'some HTML';
    return $output;
  }
}
```

# Handlers

- fields
- filters
- sorts
- arguments
- relationships

and even

- joins
- default arguments

# hook\_views\_data()

- Use it if you have custom data (as database tables)
- OR use the Data module, which does it for you
  - <http://drupal.org/project/data>

Use hook\_views\_data() to describe your data model.

- "Base tables" represent core objects
  - node, user, comment
- other tables may join to base tables -- and add more stuff to what Views knows about a core object
  - CCK adds to nodes (fields, filters, arguments, sorts)
  - Profile module adds to users

# hook\_views\_data() (continued)

```
function node_views_data() {
  $data['node']['title'] = array(
    'title' => t('Title'),
    'field' =>
      array('handler' => 'views_handler_field_node'),
    'sort' =>
      array('handler' => 'views_handler_sort'),
    'filter' =>
      array('handler' => 'views_handler_filter_string'),
    'argument' =>
      array('handler' => 'views_handler_argument_string'),
  );
  return $data;
}
```

# a filter handler

```
class views_handler_filter_string extends views_handler_filter {
    function option_definition() {
        $options = parent::option_definition();
        $options['case'] = array('default' => TRUE);
        return $options;
    }

    function options_form(&$form, &$form_state) {
        parent::options_form($form, $form_state);
        $form['case'] = array( ... );
    }

    function query() {
        $this->ensure_my_table();
        $field = "$this->table_alias.$this->real_field";
        $this->query->add_where($this->options['group'],
                               "$field = '%s'", $this->value);
    }
}
```

**Views documentation is  
accessible via Advanced Help**

or in the 'help' directory of the Views  
module

# Views 3 preview...

"area" plugins\*, pager plugins\*,  
exposed form plugins\*, query  
plugins

# Resources

- <http://drupal.org/project/views>
- [http://drupal.org/project/views\\_attach](http://drupal.org/project/views_attach)
- <http://drupal.org/project/tablegroup>
- <http://drupal.org/project/data>
  
- Views API documentation (sort of):
  - <http://views.doc.logrus.com/>
  - in the module: `views/help`
  
- #drupal-views in IRC