

Create a simple rating/review system with a Drupal 6 installation and the latest stable versions of the following modules:

- Fivestar
- Voting API
- CCK

This tutorial was written based on testing performed in the following environment:

- Windows Vista Home Edition
- Apache 2.2.8, PHP 5.2.6, MySQL 5.0.51a
- Drupal 6.10
- CCK 6.x-2.2
- Fivestar 6.x-1.13
- Voting API 6.x-2.0-rc2

This tutorial will use two content types created using CCK - "Product" and "Review". "Product" is the target node type, or in other words, the node type that is being reviewed by users. Users can submit reviews of Products using the "Review" node type. The "Review" node type will contain the fivestar cck fields that users will use to rate the Product nodes. A single rating review system and a multi-axis review system will be demonstrated. The multi-axis review system will consist of reviews that allows users rate a Product in more than one category (e.g. Reliability and Value instead of a single Overall Rating).

### **Single rating review system:**

#### Step 1: Create the Product node type using CCK and create a Product

Go to admin/content/types/add and create a new node type called "Product" (no quotations). IMPORTANT: CHECK THE "ENABLE FIVESTAR RATING" BOX UNDER THE "FIVESTAR RATINGS" FIELDSET. Votes made by the Review content type will not be calculated unless this box is checked. Then select "Hidden" under "Teaser Display" and "Full node display". This will hide the rating widget on the Product nodes (we don't want the widget on the product nodes, we want it on the Review nodes). Save this content type and create a Product node titled "Product A". Note the url of this node as you will need to come back to it later.

#### Step 2: Create the Review node type with a single rating field using CCK and create a Review node

Go back to admin/content/types/add and create a new node type called "Review" (no quotations). Save the node type and then click "manage fields" for the Review node type. Add a new field with the label "Select the product you wish to review", field name "product\_reference", field type "Node reference", and form element "Select list". Under the global settings, make the field required and check "Product" under "Content types that can be referenced". Save the field and add another field with the label "Overall Rating", field name "overall\_rating", field type "Fivestar Rating", and form element "Stars". Under Review settings, uncheck "Allow user to clear value". Under Global Settings, check the "Required" box and under "Voting target", select the node reference field "field\_product\_reference". PHP CODE IS NOT NEEDED -- LEAVE BLANK. VOTING AXIS IS NOT NEEDED FOR SINGLE REVIEW SYSTEM -- LEAVE BLANK. Save the field settings.

Now you can create a Review, enter a title, enter body text (optional), give it a rating, and select "Product A" under "Select the product you wish to review". Submit the review.

#### Step 3: Create custom node template file

Go to your theme folder. Copy node.tpl.php and rename it node-product.tpl.php. Insert the following code in the template file to display the average of the rating:

```
<?php
$current_rating = votingapi_select_results(array('content_id' => $node->nid, 'function' =>
'average'));
print '<div><strong>Rating:</strong>';
print theme('fivestar_static', $current_rating[0]['value'], '5'); //this is for 5 stars,
enter 10 (or whatever) instead as appropriate
print '</div>';
?>
```

The position of this code in the template file will determine where the rating widget(s) is/are displayed on the Product node pages.

After you have inserted the display code into node-product.tpl.php, visit the Product A node. You should now see the rating widget(s). In order for the widget to display an average of ratings, you must submit multiple Reviews. By design, a second rating by a user will overwrite the initial rating that user submitted. In other words, you must submit Reviews by at least two different users to see an average calculated on the Product node pages. After you submit multiple Reviews, you can verify that the average is being calculated and displayed.

### **Multi-axis Rating System:**

#### Step 1: Create the Product node type using CCK and create a Product

Go to admin/content/types/add and create a new node type called "Product" (no quotations). IMPORTANT: CHECK THE "ENABLE FIVESTAR RATING" BOX UNDER THE "FIVESTAR RATINGS" FIELDSET. Votes made by the Review content type will not be calculated unless this box is checked. Then select "Hidden" under "Teaser Display" and "Full node display". This will hide the rating widget on the Product nodes (we don't want the widget on the product nodes, we want it on the Review nodes).

Save this content type and create a Product node titled "Product A". Note the url of this node as you will need to come back to it later.

### Step 2 (Multi-axis Rating System): Create the Review node type with multiple rating fields using CCK and create a Review node

Go back to admin/content/types/add and create a new node type called "Review" (no quotations). Save the node type and then click "manage fields" for the Review node type. Add a new field with the label "Select the product you wish to review", field name "product\_reference", field type "Node reference", and form element "Select list". Under the global settings, make the field required and check "Product" under "Content types that can be referenced". Save the field and add another field with the label "Reliability Rating", field name "reliability", field type "Fivestar Rating", and form element "Stars". Under Review settings, uncheck "Allow user to clear value". Under Global Settings, check the "Required" box and under "Voting target", select the node reference field "field\_product\_reference". PHP CODE IS NOT NEEDED -- LEAVE BLANK. Under voting axis, enter "reliability". Save the field settings. Enter a second rating field with label "Value Rating", field name "value", field type "Fivestar Rating", and form element "Stars". Use the same field settings as the "Reliability Rating" field, but under voting axis, enter "value". Save the field settings.

### Step 3 (Multi-axis Rating System): Create custom node template file

Go to your theme folder. Copy node.tpl.php and rename it node-product.tpl.php. Insert the following code in the template file to display the average of the rating:

```
<?php
$reliability_rating = votingapi_select_results(array('content_id' => $node->nid, 'tag'
=>'reliability', 'function' => 'average'));
print '<div><strong>Reliability Rating:</strong>';
print theme('fivestar_static', $reliability_rating[0]['value'], '5');
print '</div>';
$value_rating = votingapi_select_results(array('content_id' => $node->nid, 'tag' =>'value',
'function' => 'average'));
print '<div><strong>Value Rating:</strong>';
print theme('fivestar_static', $value_rating[0]['value'], '5');
print '</div>';
?>
```

The position of this code in the template file will determine where the rating widget(s) is/are displayed on the Product node pages.

After you have inserted the display code into node-product.tpl.php, visit the Product A node. You should now see the rating widget(s). In order for the widget to display an average of ratings, you must submit multiple Reviews. By design, a second rating by a user will overwrite the initial rating that user submitted. In other words, you must submit Reviews by at least two different users to see an average calculated on the Product node pages. After you submit multiple Reviews, you can verify that the average is being calculated and displayed.

### **Displaying Fivestar Multi-axis Review Ratings with Views 2.x**

This tutorial was using Drupal 6.11 and Fivestar 6.x-1.15.

After setting up a multi-axis review system using the above instructions, the next logical step would be to display the results using a view. Here are the instructions for building such a view.

#### Step 1: Have a multi-axis review system in place with at least 1 reviewable node and 2 reviews submitted for that node.

Following the previously mentioned tutorial, build a review system consisting of a "Product" node type and "Review" node type. "Product" is the node type that will be reviewed by the "Review" node type. "Review" node type will allow users to create reviews of Products and rate the products on multiple axes.

Once the review system is in place, make sure you have at least 1 "Product" node (call it "Product A") and 2 "Review" nodes (call them Review 1 and Review 2) created by 2 different users that rate the "Product A" node.

#### Step 2: Build a view that displays all "Product" nodes and the average of each rating given to each "Product" node. Sort views by highest rating in different categories.

Go to admin/build/views and create a new view. Give it a View name of "Products" and a View type of "Node". Add a Page display and make the title of the page "Products and Product Ratings". Make the Style of the page a "Table" and make the Path of the page "products". Add a field of type "Node: Title". Add a filter of type "Node: Type" and select "Is one of" and check the node type "Product".

Now add a Relationship to the view of the type "Node: Voting results". In the previously mentioned tutorial, the axes that were used were "reliability" and "value". Staying consistent, give the relationship a label of "Reliability Results". Make the Value type "Percent", the Vote tag "reliability", and the Aggregation function "Average". Add a second Relationship of type "Node: Voting results" - label it "Value Results", make the Value type "Percent", Vote tag "value", and Aggregation function "Average".

Now add a field of type "Voting API results: Value". Make the Appearance "Fivestar Stars (display only)", the Relationship "Reliability Results" and the Label "Average Reliability Rating". Add another field of the type "Voting API results: Value". Make the Appearance "Fivestar Stars (display only)", the Relationship "Value Results" and the Label "Average Value Rating".

You should now have a view that displays the Product name and the aggregate averages for each rating axis (Reliability and Value). Now a sort can be added to this view. Add a Sort Criteria of type "Voting API Results: Value". Make the Relationship "Reliability Rating" and sort "Descending" to go from highest values to lowest values. This will sort the Products from highest "Reliability Rating" to lowest "Reliability Rating". This view can then be cloned and edited so that the products are sorted by different rating axes (sort by Value rating for instance).