Socialwiki Documentation

Jeremy Dunsmore

August 9, 2013

Contents

0.1   peer

A peer has 6 attributes:

- id the userid of the user the peer represents

- trust a number based on following relations

- likesim a percentage of how similar the peers likes are to the user's

- followsim a percentage of how similar the peers follows are to the user's

- popularity percent popularity

- weight an associative array of the weights of each of the 4 scores

0.2   node

A node is used to describe a page in a tree.

0.2.1   attributes

- id: the nodes id

- content: the title and author of the page

- hidden: used in JavaScript to determine if node is to be hidden

- column: used in JavaScript to determine which column the node is placed in

- children: an array of the ids of the nodes children

- parent: the id of the nodes parent -1 if it has no parent

- level: the depth of a node. ex root is 1 root's children 2 etc...

- added used in JavaScript to determine if the node has been added to the tree

- priority: used to determine the order the socialwiki_tree nodes

0.2.2 methods

- ___construct: takes a page sets column, added, hidden and content then sets priority to the page's votes

- set_content: gets the page's author, title and creates the HTML for a link to the page.

- add_child: takes a child id and appends it to the children array.

- display: display's the content in an HTML box.

0.3 socialwiki_tree

The socialwiki_tree is used to show the relations between pages on the history and search pages. The socialwiki_tree has 1 attribute an associative array of socialwiki_nodes.

0.3.1 methods

- build_tree: given an array of pages builds a tree by adding all the pages as nodes and calling add_children to add the child arrays to all the nodes.

- add_node: given a page adds the page as a node to the nodes array with the id being l+pageid

- add_children: goes through the nodes array and builds the children arrays for each node

- sort: organizes the array by ordering leaves in order of priority then organizes the nodes in to a tree placing the root node first followed by its children

- add_levels: goes through the array after it has been sorted and adds a level to each node root gets 1 root's child 2 etc.

- repos_children: this function gets called when sorting the array and when a parent is moved it places all of the children directly after the parent in the array.

- add_parent: recursively adds nodes to the array positioning them based on child location and priority.

- find_ leaves: returns an array of nodes with no children

- find_index: finds sibling locations and determines, based on priority, where the index of the array the node should be placed at.

- display: calls sort then display the tree

0.4 Peerscore Algorithm

The like and follow similarity for a peer are calculated using:

$$\frac{|likes(p1) \cap likes(user)|}{|likes(p1) \cup likes(user)|}$$

The trust of a peer is 1 over the following distance of a peer.

Peer popularity is calculated by:

$$\frac{num\ following}{num\ peers}$$

The peerscore is then calculated using:

$$\sum score_i \times weight_i$$

0.5 pagescore

The page score is calculated as $\left(\sum peerscore\right) + \frac{time\ created}{current\ time}$

0.6 New Database Tables

0.6.1 socialwiki_follows

The follows table has 4 columns:

- id

- userfromid: the userid of the user doing the following

- usertoid: the userid of the user being followed

- subwikiid: the subwikiid for the activity

0.6.2   socialwiki_likes

The likes table has 4 columns:

- id

- userid: the userid of the user doing the liking

- pageid: the pageid of the page being liked

- subwikiid: the subwikiid for the activity

0.6.3   socialwiki

A style column was added to the socialwiki table to track which style the module is to use.

Chapter 1

files and folders

1.1   backup folder

this folder contains all the php files required to backup a socialwiki activity currently only backup for moodle 2 is supported.

1.2   db folder

This folder has all the files that describe the database.

1.2.1   access.php

This file contains all of the permissions for different users for example if a user can view a page or not.

1.2.2   install.xml

This file contains all of the XML to create the database for socialwiki.

1.2.3   upgrade.php

This file tells moodle to update the database when a change has been made to the database in a previous version. For more on how upgrade.php works `http://docs.moodle.org/dev/Upgrade_API`

1.3　lang/en

1.3.1　socialwiki.php

This file contains all the language strings used in the activity

1.4　parser

This folder contains all of the class definitions that are used to parse the wiki pages.

1.5　pix

This folder contains most of the pictures used in the activity

1.6　tree_jslib

This folder contains the JavaScript and css for the trees used to view history and search results.

1.7　admin.php

This file is used for displaying the administration tab.

1.8　ajax.php

This file is used to process ajax calls from search.js

1.9　classic_style.css and modern_style.css

These files are used to format the output of the two different styles classic and modern.

1.10 styles.css

This file contains the css that is used in all styles

1.11 comments.php

This file is used for the comments page.

1.12 create.php

This file is used to create a new wiki page

1.13 diff.php

This file is used to compare two different pages from the same tree.

1.14 edit.php

This file is used to edit a page

1.15 follow.php

This file is used to add or remove a follow.

1.16 help.php

This file is used to create the help tab

1.17 history.js

This file contains all of the JavaScript used for the history page.

1.18   history.php

This file is used to create the history page.

1.19   home.php

This file is used to create the home page.

1.20   lib.php

This file contains functions that are called by the moodle system

1.21   locallib.php

This is one of the more important files, it contains all the function definitions that are used within the activity.

1.22   manage.php

This file is used to create the manage page.

1.23   pagelib.php

This is one of the most important files in this module as it contains all of the class definitions for the pages used in the other php files.

1.24   renderer.php

This file is used to create the HTML for certain functions.

1.25   search.php

This file is used for the search page.

1.26   search.js

This file contains all of the JavaScript for the search page. Creates both the tree for tree view and the weight sliders for tree view and list view.

1.27   toolbar.js

This file contains all the JavaScript for the toolbar used in the modern style.

1.28   version.php

This file is used to track the current version of socialwiki as well as the moodle version required to install the plug-in. The current moodle version required is 2.5.

1.29   view.php

This page is used to view a wiki page and is the first file called when the user enters the activity. However the file just redirects to the home page when this happens.

1.30   viewuserpages.php

This page is used to create the user profile page that gets called when a user clicks on the author of a page.

Chapter 2

users

2.1 Admin User and Teachers

As an admin user when you first create a socialwiki you must give the wiki
a name, a description and give a title to the first page. You can also select
the format of the wiki (HTML, creole or nwiki) as well as the style for the
wiki (classic or modern). Once a socialwiki is created a teacher for the course
must create the first page if they wish for the wiki to have a first page other
wise every time a teacher views the socialwiki they will be prompted to create
the first page. This however doesn't affect the general users for the course
they will see the home page and will still be able to create and view pages
they just won't have a link to the first page. Once a Socialwiki has been
created the administrative user have a few extra options that general users
don't. They can go to the socialwiki administration block and change the
style of the wiki or they can go to the administration tab where they have
the ability to delete pages.

2.2 Enrolled Users

When an enrolled user enters a socialwiki they are taken to the home page
where the can view: updated pages from the last time they logged in, pages
that they created, a list of orphaned pages, a list of all pages, a list of pages
recommended for them and a list of all the teacher's pages. In the navigation
block there are also links that allow them to view the help page or create a
new page. When a user views a page they can see how many likes a page

has, they can like the page or follow the user who created the page. They can also click on the author and see pages the author has liked and see the authors peer scores. When a user searches for pages they can view a tree that shows the history and family for the pages or they can view a list of pages or a list of pages ordered by which pages have the most likes. The history page allows a user to view the history of a page as well as compare two pages to see the difference between them. A user can also edit a page. When a user edits a page they create a new page with any changes they made.