

Hours & Locations Portal Code Notes

Further explanatory comments can be found in all code files for the public-facing Hours Portal. The primary files and functions used are:

- **dbConnect.php** – Contains the database connection information, using the configuration file (db.conf).
- **conf/db.conf** – Contains the database configuration details.
- **functions.php** – All PHP functions (and queries) for the public site, which includes:
 - *displayTime*: given a time, determine if minutes should be removed, add appropriate trailing am/pm
 - *rv*: check submitted month/year requests for monthly calendar display
 - *sanitize*: basic security check on POST variables for print views
 - *displayLocationsStatus*: retrieve and display locations with their current status (open or closed) in a dl table based on (optional) branch ID
 - *displayCurrentStatus*: retrieve and display the current status (open or closed) in span tags based on a date, time and (required) branch ID
 - *getBranchInfo*: get an array of location information (listed out individually with functions below) from the database based on (optional) name ID
 - *getNameIDs*: get name IDs (login names) based on (optional) location ID
 - *getID*: given name ID, get actual location ID number
 - *getMapCode*: given name ID, get the location's Google map code
 - *getName*: given name ID, get location name
 - *getBuilding*: given name ID and prepend (e.g. "at"), get location building name
 - *getDescription*: given name ID, get location description

- *getAddress*: given name ID, get location address
 - *getPhone*: given name ID, get location phone number
 - *getURL*: given name ID, get location URL address
 - *getAccessURL*: given name ID, get location URL address for disability information
 - *getHoursNotes*: given name ID, get any available location notes (for emergencies/closures)
 - *getWidgetNote*: given name ID, get location widget notes
 - *getRangesByMonth*: get an array of matching date ranges given a month, year and (optional) branch id
 - *displayHoursByMonth*: retrieve and display hours in a dl based on a month, year and (optional) branch id (includes collapsing hours), for use with a monthly calendar view
 - *getAllRanges*: given month and category ID, get applicable range ids, start dates and end dates
 - *getAllHours*: given category ID, limiting location IDs, range IDs, range begin dates and range end dates, get all matching hours for print views
- **index.php** – Starts the session, pulls in three main files (head.inc.php, core.inc.php, footer.inc.php) to build the page.
 - **includes/head.inc.php** – Pulls in header display for the UBC look and feel.
 - **includes/footer.inc.php** – Pulls in footer display for the UBC look and feel.
 - **includes/core.inc.php** – The core display of the Hours Portal page, also pulls in the calendar (calendar.inc.php) and includes the JavaScript to make this work.
 - **includes/calendar.inc.php** – Contains all code for the monthly calendar view in each location's panel (included separately to allow for AJAX).
 - **print.php** – The launching point for the print views.

- **print-hours.php** – Uses variables passed in from print.php to show printable full-page or bookmark-size views of hours and locations.
- **widget-hours.php** – The initial PHP file of the embeddable hours display (with five views of the hours, two weekly and three daily).
- **widget-hours-js.php** – The “widgetized” version of the embeddable hours display (intended for use on other sites, with variables passed in to call particular locations and views), sent as JavaScript.
- **css/hours.css** – Stylesheet for the main site.
- **css/print-hours.css** – Stylesheet for the print-hours.php page (very different styling than above and not RWD).
- **js/hours.js** – Most of the primary JavaScript for the site, including code activating the sliding panel, creating highlighting with the monthly calendar, working with the hash tag, hiding/displaying the scrollbar, controlling the Google map and accounting for new functionality with smaller screen views:
 - *forcePageReload*: force the page to reload when the drop-down menu is used to change the branch (and hash tag)
 - *slideOpen*: open the sliding information panel
 - *contentSwitch*: given a div ID, display the selected content and initiate the scrollbar plugin
 - *slideShut*: close the sliding information panel
 - *switchMap*: switch between table and map views when small screen display is in effect (and not all elements fit on the page at once)
 - *newMarker*: create a new Google map marker
 - *customControl*: create a new Google map custom control
 - *tableMouseover*: what happens with Google map when location table name hovered
 - *tableMouseout*: what happens with Google map when location table name moused out

- **js/jquery.hashchange.min.js** – Plugin to add monitoring of hash change in URL (mainly for small screen view).
- **js/jquery.hoverIntent.min.js** – Plugin to delay hover over the main locations table.
- **js/jquery.jscrollpane.min.js** – Plugin to control display of location panel scrollbar.
- **js/jquery.mousewheel.js** – Plugin working in conjunction with the above to control mouse wheel when hovering over a scrollable panel.
- **js/markerwithlabel_packed.js** – Plugin for Google Maps to allow for easier display of marker labels.