Table of Contents

xbrowse framework.....1

Introduction.....1

xbrowse-based applications.....1

Creating a new xbrowse-based UI.....1

Web API.....2

Standalone demo.....2

All downloads.....3

Change log.....3

Release notes.....3

Incompatible changes.....3

0.21.0 to 0.22.0.....3

Notes.....3

Required changes.....3

0.19.0 to 0.20.0.....3

Notes.....3

Required changes.....4

Optional changes.....4

0.17.0 to 0.18.0.....4

Notes.....4

Required changes.....4

0.7.0 to 0.8.0.....4

Notes.....4

Required changes.....4

0.3.0 to 0.4.0.....4

Notes.....5

Required changes.....5

Further Information.....5

xbrowse framework

This page documents the xbrowse framework.

Introduction

xbrowse is small framework for developing large-scale JavaScript applications. It includes a simple client-side MVC implementation and a number of predefined views such as statistics matrix, plots, controls.

The core MVC implementation includes:

- model.state: UI state, i.e. filters as set by the user. A hierarchical store backed by the URL hash.
- model.data: Data as returned by AJAX calls. A simple store to share data between views.
- view: Skeleton view extended in the predefined views. Views update the model.state in response to user input, and bind to model.state events to update the UI and load data via AJAX calls.
- controller: Initialises the model and views and updates the model.state from the URL hash.

The predefined views include:

- view_dates: Binds to the model.state "load" event to load the latest dates from the server and fire the model.state "refresh" event.
- view: Creates the UI container.
- view_sidebar: Creates the sidebar accordion for UI filter controls.
- view_main: Creates the UI tabs. Binds to model.state "refresh" to show/hide tabs.
- view_titlebar: Creates the titlebar with date-lock, refresh and API links. Update model.state in response to user input. Bind to model.data "change" to title and API links.
- view_ctrl_*: Create various UI filter controls (typically in the sidebar accordion). Update model.state in response to user input. Bind to model.state "change" to update controls.
- view_matrix: Creates matrix tab and toolbar. Update model.state in response to user input. Binds to model.state "refresh" to call web API and update matrix.
- view_errors: Creates error samples pane. Binds to model.state "refresh" to call web API and update error samples.
- view_plots: Creates plots tab and toolbar. Update model.state in response to user input. Binds to model.state "refresh" to call web API and update plots.

xbrowse-based applications

xbrowse-based applications include:

- ATLAS DDM Dashboard
- WLCG Transfers Dashboard
- FTS Dashboard
- FAX XRootD Dashboard
- AAA XRootD Dashboard

Creating a new xbrowse-based UI

Any new xbrowse-based application should:

- Define default key/values for model.state. See `lib/model_adaptor.js` file in example.
- Adapt predefined views. See `lib/view_*_adaptor.js` files in example.
- Bootstrap the UI. See `lib/app.js` file in the standalone example.

- Include scripts in single HTML page. See `index.html` file in example.
- Provide web API to serve JSON responses to AJAX calls from views. See ajax options of `lib/view_*_adaptor.js` files in example.
- (optional) Add new application-specific views.

Web API

The various predefined views depend on a web API that responds with JSON to their AJAX calls.

- view_dates:
    - API example: http://dashb-fts-transfers.cern.ch/dashboard/request.py/resolve-dates?interval=240&from_date=&to_d
    - Code example: (OPEN) ResolveDatesAction.py (CERN) ResolveDatesAction.py
- view_matrix:
    - API example: http://dashb-fts-transfers.cern.ch/dashboard/request.py/transfer-matrix.json?from_date=2014-02-19T1
    - Code example: (OPEN) GetTransferMatrixAction.py (CERN) GetTransferMatrixAction.py
- view_plots:
    - API example: http://dashb-fts-transfers.cern.ch/dashboard/request.py/transfer-bins.json?from_date=2014-02-19T11%
    - Code example: (OPEN) GetTransferBinsAction.py (CERN) GetTransferBinsAction.py
- view_errors:
    - API example: http://dashb-fts-transfers.cern.ch/dashboard/request.py/error-samples.json?from_date=2014-02-19T11
    - Code example: (OPEN) GetErrorSamplesAction.py (CERN) GetErrorSamplesAction.py

Standalone demo

The zip dashboard-xbrowse-0.22.0-1.tgz contains two standalone demos of an xbrowse-based UI using core xbrowse code and demo UI code.

Please download the zip and have a play with it.

Here is the readme.txt file from the zip:

```
*****
xbrowse standalone demo
*****
This demo consists of the core xbrowse code with demo UI code. It includes
two demos one simple that accesses data from Google News, and one advanced that
accesses data from the FTS Dashboard [1].

To view the simple demo open 'demo-simple/index.html' in your browser.
To view the advanced demo open 'demo-fts/index.html' in your browser.

The xbrowse core (release:0.22.0-1) and demo is extracted from GIT [2] using
xbrowse_demo.sh.

To create an xbrowse-based UI you should only need to modify the UI code, see
'demo-simple/readme.txt for more details on the xbrowse MVC life-cycle and
how to add new views. See also the xbrowse twiki [3].

[1]: http://dashb-fts-transfers.cern.ch/ui/
[2]: (OPEN) http://git.cern.ch/pubweb/cosmic.git/tree/HEAD:/dashboard.xbrowse/templates
     (CERN) https://git.cern.ch/web/cosmic.git/tree/HEAD:/dashboard.xbrowse/templates
[3]: https://twiki.cern.ch/twiki/bin/view/ArdaGrid/XbrowseFramework
```

*****

The bash script xbrowse_demo.sh (found in the zip) was used to generate the standalone demo zip.

All downloads

- **LATEST** dashboard-xbrowse-0.22.0-1.tgz: based on xbrowse 0-22-0 code and demo UI code that accesses data from Google News and FTS Dashboard.
- xbrowse_demo_0-20-0.tgz: based on xbrowse 0-19-0 code and demo UI code that accesses data from Google News and FTS Dashboard.
- xbrowse_demo_0-19-0.tgz: based on xbrowse 0-19-0 code and demo UI code that accesses data from Google News and FTS Dashboard.
- xbrowse_demo_0-18-0.tgz based on xbrowse 0-18-0 code and demo UI code that accesses data from FTS Dashboard.
- xbrowse_demo_0-17-0.tgz based on xbrowse 0-17-0 code and demo UI code that accesses data from FTS Dashboard.
- xbrowse_demo_0-15-2.tgz based on xbrowse 0-15-2 code and demo UI code that accesses data from WLCG Transfers Dashboard.
- xbrowse_demo_0-15-0.tgz based on xbrowse 0-15-0 code and demo UI code that accesses data from WLCG Transfers Dashboard.
- xbrowse_demo_0-4-0.tgz based on xbrowse 0-4-0 code and demo UI code that accesses data from WLCG Transfers Dashboard.
- xbrowse_example_0-3-0_0-1-0.tgz based on xbrowse 0-3-0 code and UI 0-1-0 code from WLCG Transfers Dashboard.

Change log

Release notes

The release notes (OPEN) (CERN) give details of all changes between versions.

Incompatible changes

Some releases include changes that break backward compatibility, or have new features disabled by default. The following gives details of how applications should be modified to work with new releases or to take advantage of new features.

0.21.0 to 0.22.0

Notes

This release fixes bug JIRA:WLCGMON-153.

Required changes

If you have code in application-specific lib/model_adaptor.js that reacts to "change" events in model.post_init then modify it according to the following diff model_adaptor.js (OPEN) model_adaptor.js (CERN).

0.19.0 to 0.20.0

Notes

This release add new feature JIRA:WLCGMON-142. It adds the option to use unified apply/cancel/default buttons for accordion controls. That is by setting a flag, the buttons will not appear in each individual accordion control but just once at the bottom of the accordion. In this way, a user may modify many controls before applying the changes and thus avoid reloading data from the server for intermediate parameter changes.

Required changes

None. This is an optional feature.

Optional changes

To enable unified buttons for accordion controls modify application-specific lib/view_adaptor.js according to the following diff view_adaptor.js.

If you have code in application-specific lib/model_adaptor.js that reacts to the "change:date" event in model.post_init then modify it according to the following diff model_adaptor.js. This more complicated but more intelligent code reacts to changes in the calculated date interval rather than just a value change; e.g. a change from a fixed 4 hour interval to a sliding 4 hour interval will not result in resetting the plot bin size.

All xbrowse provided control views have been modified to work both with or without the unified buttons. However, if you have written your own controls for the accordion then you will have to modify them along the lines shown in this diff demo-simple/lib/view_ctrl_query.js.

0.17.0 to 0.18.0

Notes

This release removes external JavaScript libraries from the xbrowse module and relies instead on CDN sources for JavaScript libraries.

Required changes

Update application-specific index.html and plot.html according to the following diffs index.html, plot.html.

The datatables libraries were also including in the xbrowse module, although never used by any xbrowse views. If you were using these libraries then you will need to source them from a CDN. The following lines in index.html should be sufficient:

```
<link type="text/css" href="http://cdnjs.cloudflare.com/ajax/libs/datatables/1.9.4/css/demo_table
<script type="text/javascript" src="http://cdnjs.cloudflare.com/ajax/libs/datatables/1.9.4/jquery
```

0.7.0 to 0.8.0

Notes

This release add a feature for the user to control the maximum number of cells visible in the matrix tab. The user can modify this depending on his/her systems performance using the MAX CELLS toolbar menu.

Required changes

Update application-specific lib/model_adaptor.js according to the following diff model_adapter.js.

0.3.0 to 0.4.0

Notes

The 0.4.0 release includes an jquery plugin to abstract the common plotting functionality used in tabs using view_plots and application-specific tabs displaying plots.

Required changes

Update application-specific index.html and plot.html according to the following diffs index.html, plot.html.

Further Information

- CHEP 2012: Designing and developing portable large-scale JavaScript web applications within the Experiment Dashboard framework
    - slides, video, paper
- GIT: release notes (OPEN) (CERN), master (OPEN) (CERN), tags (OPEN) (CERN).

-- DavidTuckett - 03-May-2012

This topic: ArdaGrid > XbrowseFramework
Topic revision: r13 - 2014-07-15 - DavidTuckett