Java Broker Design - Preferences

This page documents the design, requirements and scope of work for implementation of Web Management Console Preferences.

- Introduction
- Expectations
- Preferences
- Preferences rest interfaces
  - Accessing preferences with REST API
  - Delivering preferences
- Storing preferences
- Preferences model
- Saving tabs configuration if saveTabs preference is turned on

Introduction

Web Management Console users should be able to customize the web management console by setting their preferable time zone, language, update interval, etc.
The user preferences should be stored per user on broker side. A special UI should be added into a web management console to set and save the preferences.

Expectations

In future the preferences should apply to multiple brokers but for the current work it is assumed that preferences are stored per broker.
The preferences should be plug-able and might have several implementations, for example, preferences could be implemented with LDAP or JDBC or file system.
The preferences should be bundled with authentication provider. Thus, allowing to store and provide preferences for the users managed by authentication provider.
On Authentication provider creation it should be possible to select a preference implementation.
The user preferences should be deleted on deletion of the user account from authentication provider if user deletion is managed by the authentication provider. If authentication provider does not support the user deletion operation from managing interfaces, then the preferences should be deleted from a management console using special UI with the list of users having preferences defined.

Preferences

Here is the list of the preferences which can be implemented

Name	Description
Filtering criteria	It should be possible to store the filtering criteria for Log Viewer and other components. On saving the criteria user should be able to specify the name of the criteria. It should be possible to delete and edit the stored criteria.
Time zone	User should be able to change the time zone on web management console. The various date/times displayed in the console should be automatically converted and displayed with regards to the preferred timezone. This should include message timestamps, log timestamps etc.
Save Tabs	If enabled, the tabs will be restored on opening of the console.
Update interval	Currently all opened components are updated every 5 seconds. The time is hard-coded and it is not possible to change. A special preference to set the update interval (or even switch it off and use manual update) could be added.
Web Management Console Style	It would be nice to customize a style for different brokers in order to distinguish easily production environment from non-production, etc.
Language	We can add a support for different languages into Web Management Console. Users should be able to select the preferable language.

Preferences rest interfaces

A special Rest interfaces should be implemented to display, update, add and delete preferences:

- list user names having preferences stored on the broker
- list preferences for a given user
- save preferences
- remove preferences for a given user

The preferences should be stored separately in a special store.

Accessing preferences with REST API

The REST API should provide an access to the preferences data for user and authentication provider

```
/rest/userpreferences/<authentication provider name>/<user name>
```

Also, for the currently logged user the preferences could be accessed using

```
/rest/preferences
```

Only currently logged user can set/change preferences. Users should be granted permissions to view and delete other user preferences.

Delivering preferences

Preferences should be delivered in JSON format. An example of the preferences JSON:

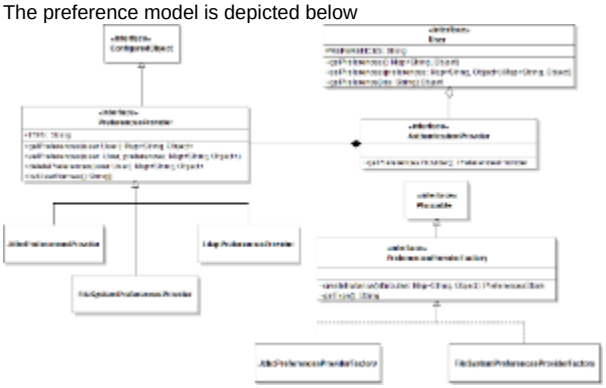
```
{
    "alex": {
        timezone: "Europe/London",
        language: "en",
        background: "#cccccc",
        saveTabs: true,
        updateInterval: 5,
        "logviewer.defaultFilter": {...},
        "logviewer.filters": { "warning": {...}, "alerts": {...}, ... },
        ...
      },
    ...
    "anotherUser": {
    ...
    }
}
```

Storing preferences

The preferences should be represented as key-value pairs.
The preference key value could be a complex name which could represent the hierarchy where each item in hierarchy is separated by '.' character. For instance, logviewer.defaultFilter and logviewer.filters would represent the preferences "defaultFilter" and "filters" for a logviewer widget.
The value could be a serialized json string to represent the complex preference settings or to be a primitive value like int, long, double, String

Preferences model

The preference model is depicted below



With this model the PreferencesProvider belongs to AuthenticationProvider and instantiated by the AuthenticationProvider.
User configured object has the getPreferences()/setPreferences() methods which delegates the call to PreferencesProvider (user.
getAuthenticationProvider().getPreferencesProvider().getPreferences(user))

Saving tabs configuration if saveTabs preference is turned on

The tabs configuration needs to be serialized as json and stored in openTabs preference
We need to identify what exactly information we need to save for the opened tab. It looks like we need to consider storing at lest of the following:

1. type

2. UUID for the configured object
   Potentially we can save the configured objec name for cases when user deleted and re-created the queue