YUI Library: Storage Utility

2009-9-8 **v2.8**

Simple Use Case: Requesting a Storage Engine

```
// this will fetch the first available engine
var store = YAHOO.util.StorageManager.get();
store.subscribe(store.CE_READY, function() {
  store.setItem('key', 'value');
  store.setItem('testNumber', 1234567890);
  store.setItem('testBoolean', true);
  alert(store.getItem('key'));
});
```

In this code block, a Storage instance is requested. No specific storage backend is specified, so Storage will try its various engines in an arbitrary order until it finds one that is supported by the current browser. The `CE_READY` custom event is used to identify the moment when the Storage instance is fully ready for use, and when that event fires a series of values are stored in the Storage instance.

Constructor: YAHOO.util.StorageManager

`YAHOO.util.StorageManager([str` *`engine type`*`, str` *`location`*`, obj` *`configuration`*`])`

Arguments:

(1) **engine type:** (Optional) One of three strings, by convention referred to via constants: `YAHOO.util.StorageEngineSWF.ENGINE_NAME`, `YAHOO.util.StorageEngineHTML5.ENGINE_NAME`, or `YAHOO.util.StorageEngineGears.ENGINE_NAME`.

(2) **location:** (Optional) String value, either `session` or `local`; `session` retains data only through the life of the browser session, whereas `local` persists data beyond the session.

(3) **configuration:** (Optional) Configuration object containing configurations specific to this instance. See Configuration Options section for details on what you can put in this object.

Dependencies

The Storage Utility employs the common YUI Core components (Yahoo, Dom and Event) and the Cookie Utility. Unless you are specifically forcing the use of HTML5 or Gears, the Swf Utiltity must be included as well. (Because Flash is currently much more commonly supported than HTML5 Storage and Gears, almost all implementations should allow support for Flash storage.

Including the SWFStore .swf File

You must host the file swfstore.swf (available in `build/swfstore` in the YUI 2.x distribution; it is expected to be in the same directory as the HTML file hosting the page that instantiates Storage. If it is not, specify the location manually:

```
YAHOO.util.StorageEngineSWF.SWFURL = '/pathto/swfstore.swf';
```

Key Storage Utility Configuration Options

Field	Type	Description
force	boolean	Forces Storage to choose the storage mechanism specified in the first argument of the constructor (rather than trying multiple engines if the first one fails).
order	array	Array of engine types specifies the order in which the engines should be tried (some combination of `YAHOO.util.StorageEngineSWF.ENGINE_NAME`, `YAHOO.util.StorageEngineHTML5.ENGINE_NAME`, and `YAHOO.util.StorageEngineGears.ENGINE_NAME`).
engine	obj	Configuration object to pass to the storage engine; see Storage Engine. The engine configuration options are: 1. **swfURL:** string representing the path to your same-domain copy of swfstore.swf; default is /swfstore.swf. 2. **containerID:** string containing the HTML ID value of the element into which the the SWF will be inserted; if no element is specified, the body element will be used. 3. **attributes:** object containing configuration properties for the SWF Utility.

Configuration options should be set in the third argument of the constructor:

```
var store = new YAHOO.util.StorageManager.get(YAHOO.util.StorageEngineSWF.ENGINE_NAME, YAHOO.util.StorageManager.LOCATION_LOCAL, {force: false, order: [YAHOO.util.StorageEngineGears, YAHOO.util.StorageEngineSWF, YAHOO.util.StorageEngineHTML5]});
```

Custom Events in the Storage Utility

Event	Description/Fields:
CE_READY	Event fires when the storage engine is ready. Always defer usage of the Storage instance until CE_READY has fired; the SWFStore implementation is asynchronous, so this deferral is required to avoid errors.

Subscribe: `store.subscribe(store.CE_READY, function(o){});`.

Solutions:

Simple use case:

```
store = YAHOO.util.StorageManager.get(
      YAHOO.util.StorageEngineGears.ENGINE_NAME,
      YAHOO.util.StorageManager.LOCATION_LOCAL,
      {
        order: [
          YAHOO.util.StorageEngineGears,
          YAHOO.util.StorageEngineSWF,
          YAHOO.util.StorageEngineHTML5
          ],
        force: false
      }
     );

store.subscribe(storageEngine.CE_READY, function(e) {
  //use Storage engine here
});
```

YAHOO.util.Storage Properties

length number the current number of keys

YAHOO.util.Storage Methods

clear() clears all existing key/value pairs
getItem(str *key*) returns the data value for the requested key
getName() returns the name of the storage engine being used (e.g., "gears")
hasKey(str *key*) returns booliean indicating that the supplied key does/doesn't exist
key(n *index*) returns the key at the supplied index
removeItem(str *key*) removes key/value pair from storage
setItem(str *key*, str *value*) sets the value for a given key