YUI Library: YUI Loader Utility [beta]

2007-12-4c **v2.4**

Simple Use Case: YAHOO.util.YUILoader

Markup:

```
<script src="yuiloader-beta.js"></script>
```

Script:

```
//instantiate Loader:
loader = new YAHOO.util.YUI Loader();

//identify the components you want to load:
loader.require("colorpicker", "treeview");

//configure the Loader instance
loader.loadOptional = true;

//Load files using the insert() method. Insert() takes an optional
//configuration object, and in this case we are setting up an onSuccess
//callback. Your callback will be executed once all required files are
//loaded.
loader.insert({ onSuccess: function() {
    //this is your callback function; you can use
    //this space to call all of your instantiation
    //logic for the components you just loaded.
}});
```

Sets up a YUI Loader instance, configures it to load Color Picker and TreeView, and then executes the load.

Constructor: YAHOO.util.YUILoader

`YAHOO.util.YUILoader(obj` *`config`*`)`

Arguments:

(1) **Configuration object:** When instantiating YUI Loader, you can pass all configurations in as an object argument or configure the instance after instantiation. See Configuration Options section for common configuration object members.

Using YUI Loader to Load Non-YUI Files

See online docs for full syntax and example using `addModule()`.

YUI Loader's `addModule` method can be used to extend YUI Loader to add non-YUI modules. `addModule` takes an object argument with the following members:

name (s)	String modulename.
type (s)	String moduletype (eg, "js" or "css").
path (s)	Path to source file, including file name; will be prefixed with instance's `base` path setting.
fullpath (s)	Full URI to module file. Supersedes `path`.
varName (s)	If module is JavaScript, a variable name (as string) that will be defined by the loaded script; alternatively, use `YAHOO.register`.
requires (arr)	Array of required dependencies, with each member a string module name.
optional (arr)	Array of optional dependencies, with each member a string module name.
skinnable (b)	Does this component have a skin CSS file (in the standard skin-CSS directory?

Key YUI Loader Configuration Options

See online docs for complete list of YUI Loader configuration options.

Option (type)	Default	Description
allowRollup (b)	true	Allow aggregate files (e.g., utilities.js) where appropriate? (improves performance)
base (s)	current build dir on yui.yahooapis.com	Base directory for YUI build.
filter (s | o)	null	Filter that can be applied to YUILoader filenames prior to loading. Use `'DEBUG'` for debug versions of YUI files
loadOptional (b)	false	Load all optional dependencies for the required components?
onFailure (f)	null	Callback function fired if an insert operation fails.
onProgress	null	Callback function fired each time a resource loads successfully.
onSuccess (f)	null	Callback function to run when loading of all required components and dependencies is complete.
require (arr)	true	Array of required YUI components, each of which is a string representing the modulename for the component.
skin (o)	by default, the YUI Sam Skin is applied to skinned components	Object which allows you to specify a global `defaultSkin` and per-component `overrides`. See User's Guide for full syntax.
varName (s)	null for `insert()`; "YAHOO" for `sandbox()`	Only needed for non-YUI scripts. This is the variable defined by the exterrnal script whose presence indicates load-completion; for `sandbox()`, this is the root variable for the loaded library.

YUI Loader options can be set in the constructor's second argument (eg, `{base: '../../'}`) or at runtime on a YUILoader instance (eg, `oLoader.base = '../../'`).

Solutions

Using YUI Loader to create a sandboxed/private YUI:

```
var loader = new YAHOO.util.YUI Loader();
loader.sandbox({
    require: ["treeview"], // what to load
    base: '../../build/',  // relative path to library files
    loadOptional: true,    // pull in optional components

    // Executed once the sandbox is successfully created:
    onSuccess: function(o) {
        var myYAHOO = o.reference; //ref to private YAHOO
        // TreeView in myYAHOO can now be used; note that
        // YAHOO.widget.TreeView may not exist!
        myYAHOO.util.Event.onAvailable("treeEl",function() {
            var tree = new myYAHOO.widget.TreeView("treeEl");
        });
    },
});
```

YAHOO.util.YUILoader: Properties

See also configuration options; all configuraton options can be treated as instance members.

inserted obj list of modules inserted by the YUILoader instance

sorted arr listed of sorted dependencies; available after insert or calculate is called

YAHOO.util.YUILoader: Methods

addModule(o) adds non-YUI module; obj argument specifies all needed metadata for new module

calculate() calculates the list of needed modules based on required components but does not insert them in the page

insert(o) calculates needed modules, inserts them, fires o.onSuccess if that is supplied

Components & Module Names

YUI Loader refers to YUI components by their unique module names — strings by which components are referenced within YUI. Here is the full list of YUI module names:

animation, autocomplete, base, button, calendar, charts, colorpicker, connection, container, container_core, datasource, datatable, dom, dragdrop, editor, element, event, fonts, get, grids, history, imageloader, json, logger, menu, profiler, reset, selector, slider, tabview, treeview, yahoo, yuiloader, yuitest.

Dependencies

YUI Loader does not have any required dependencies; it can be used on pages with no YUI content present or to bring additional YUI components onto pages where YUI is already being used. For some use cases, you may need to include the Yahoo Global Object first; see docs for more on this.