Marionette.TemplateCache

The `TemplateCache` provides a cache for retrieving templates from script blocks in your HTML. This will improve the speed of subsequent calls to get a template.

Documentation Index

Basic Usage

To use the `TemplateCache`, call the `get` method on TemplateCache directly. Internally, instances of the TemplateCache class will be created and stored but you do not have to manually create these instances yourself. `get` will return a compiled template function.

```
var template = Marionette.TemplateCache.get("#my-template", {some: options});
// use the template
```

```
template({param1:'value1', paramN:'valueN'});
```

Making multiple calls to get the same template will retrieve the template from the cache on subsequence calls.

Clear Items From cache

You can clear one or more, or all items from the cache using the `clear` method. Clearing a template from the cache will force it to re-load from the DOM (via the `loadTemplate` function which can be overridden, see below) the next time it is retrieved.

If you do not specify any parameters, all items will be cleared from the cache:

```
Marionette.TemplateCache.get("#my-template");
Marionette.TemplateCache.get("#this-template");
Marionette.TemplateCache.get("#that-template");

// clear all templates from the cache
Marionette.TemplateCache.clear()
```

If you specify one or more parameters, these parameters are assumed to be the `templateId` used for loading / caching:

```
Marionette.TemplateCache.get("#my-template");
Marionette.TemplateCache.get("#this-template");
Marionette.TemplateCache.get("#that-template");

// clear 2 of 3 templates from the cache
Marionette.TemplateCache.clear("#my-template", "#this-template")
```

▸ Customizing Template Access

If you want to use an alternate template engine while still taking advantage of the template caching functionality, or want to customize how templates are stored and retrieved, you will need to customize the `TemplateCache object`. The default operation of `TemplateCache`, is to retrieve templates from the DOM based on the containing element's id attribute, and compile the html in that element with the underscore.js `template` function.

▸ Override Template Retrieval

The default template retrieval is to select the template contents from the DOM using jQuery. If you wish to change the way this works, you can override the `loadTemplate` method on the `TemplateCache` object.

```
Marionette.TemplateCache.prototype.loadTemplate = function(templateId, options){
```

v2.4.4


```
  // load your template here, returning the data needed for the compileTemplate
  // function. For example, you have a function that creates templates based on the
  // value of templateId
  var myTemplate = myTemplateFunc(templateId);

  // send the template back
  return myTemplate;
}
```

Override Template Compilation

The default template compilation passes the results from `loadTemplate` to the `compileTemplate` function, which returns an underscore.js compiled template function. When overriding `compileTemplate` remember that it must return a function which takes an object of parameters and values and returns a formatted HTML string.

```
Marionette.TemplateCache.prototype.compileTemplate = function(rawTemplate, options) {
  // use Handlebars.js to compile the template
  return Handlebars.compile(rawTemplate);
}
```