

Marionette.TemplateCache

Объект `TemplateCache` предоставляет кэш для загрузки шаблонов из блоков типа `<script type = "text/template"></script>`, находящихся в HTML. Это увеличивает скорость обработки и внедрения шаблона при последующих обращениях к нему.

Базовое использование

Чтобы начать использовать `TemplateCache`, просто прямо вызовите его ``get`-метод`. Внутри механизма `TemplateCache`, экземпляры `TemplateCache`-а **будут созданы и сохранены**, и вручную создавать их не придется. Метод ``get`` **возвратит откомпилированный шаблон** - то-есть превращенный в js-функцию.

```
var template = Backbone.Marionette.TemplateCache.get("#my-template");
// используем шаблон
template({param1:'value1', paramN:'valueN'});
```

Последующие обращения к одному и тому же шаблону будут возвращать уже эту откомпилированную функцию.

Удаление элементов из кеша

Можно по желанию удалять один или более элементов из кеша, с помощью метода ``clear``. Удаление шаблона из кеша повлечет его повторную загрузку из DOM при следующем к нему обращении (и очевидно, что функция ``loadTemplate`` может быть переопределена, как будет показано ниже).

Функция `clear()` без параметров удалит из кеша все имеющиеся шаблоны:

```
Backbone.Marionette.TemplateCache.get("#my-template");
Backbone.Marionette.TemplateCache.get("#this-template");
Backbone.Marionette.TemplateCache.get("#that-template");
// удаляем все
Backbone.Marionette.TemplateCache.clear()
```

Задавая один параметр или более, можно очищать кеш от произвольного количества элементов:

```
Backbone.Marionette.TemplateCache.get("#my-template");
Backbone.Marionette.TemplateCache.get("#this-template");
Backbone.Marionette.TemplateCache.get("#that-template");
// удаляем 2 экземпляра
Backbone.Marionette.TemplateCache.clear("#my-template", "#this-template")
```

Кастомизация доступа к шаблонам

Если вы хотите использовать альтернативный движок шаблонов, и при этом иметь все те преимущества, которые предоставляет механизм их кеширования, или вам нужно кастомизировать способ, которым шаблоны будут найдены и сохранены, вы должны переопределить объект `TemplateCache`. По умолчанию, для извлечения шаблона из DOM, `TemplateCache` обращается к `id` атрибуту, а в качестве компилирующей html функции используется функция `template` из `underscore.js`.

Переопределение способа отыскания шаблона

По умолчанию, `TemplateCache` находит шаблон в DOM [используя jQuery](#), но ничто не мешает действовать иначе:

```
Backbone.Marionette.TemplateCache.prototype.loadTemplate = function(templateId){
  // например, написать свою функцию создания шаблона по templateId и использовать ее здесь
  var myTemplate = myTemplateFunc(templateId);
  // возвращаем результат
  return myTemplate;
}
```

Переопределение способа компиляции шаблона

По умолчанию, в процессе загрузки шаблона последовательно отработывают `loadTemplate` и `compileTemplate`. Это значит, что переопределяя метод `compileTemplate`, нужно позаботиться о том, чтобы он возвращал функцию, которая принимает некоторый объект с параметрами и значениями и возвращает строку, содержащую отформатированный HTML.

```
Backbone.Marionette.TemplateCache.prototype.compileTemplate = function(rawTemplate) {
  // применяем Handlebars.js для компиляции
  return Handlebars.compile(rawTemplate);
}
```