Marionette.Callbacks

Объект `Callbacks` помогает в управлении коллекцией колбэков и заведует их выполнением в асинхронно-безопасной манере.

Имеет всего 2 метода:

- add
- run

Метод `add` добавляет новый коллбэк, который выполнится позднее.

Метод `run` запускает на выполнение все зарегистрированные коллбэки, используя для них переданный в качестве второго аргумента контекст и переданные первым аргументом настройки.

Базовое использование

```
var callbacks = new Backbone.Marionette.Callbacks();
callbacks.add(function(options){
 alert("I'm a callback with " + options.value + "!");
});
callbacks.run({value: "options"}, someContext);
```

Пример выведет предупреждение с текстом "I'm a callback with options!". Контекст выполнения для обоих коллбэков установлен как `someContext` объект, это опциональный параметр, являющийся валидным JavaScript-объектом.

Определение контекста выполнения для каждого коллбэка

Можно также задать контекст выполнения для каждого коллбэка, передавая его вторым параметром:

```
var callbacks = new Backbone.Marionette.Callbacks();
callbacks.add(function(options){
 alert("I'm a callback with " + options.value + "!");
}, myContext);

callbacks.run({value: "options"}, someContext);
```

Поскольку контекст выполнения (`this`) в данном примере изначально установлен как myContext, попытка задать `someContext` в этом качестве будет проигнорирована.

Продвинутое / асинхронное использование

`Callbacks` запускает на выполнение каждый коллбэк в асинхронно-дружественной манере.

Объект `Marionette.Application` использует `Callbacks`, чтобы обеспечить реализацию инициализаторов.

В сценариях, базирующихся на событийной архитектуре, механизм коллбэков, равно как и механизм инициализаторов, может быть задействован для гарантированного выполнения зарегистрированных функций.