v2.4.4

Marionette.Object

A base class which other classes can extend from.
Object incorporates many backbone conventions and utilities like `initialize` and `Backbone.Events`.

Documentation Index

Initialize

Initialize is called immediately after the Object has been instantiated, and is invoked with the same arguments that the constructor received.

```
var Friend = Marionette.Object.extend({
  initialize: function(options){
    console.log(options.name);
  }
});

new Friend({name: 'John'});
```

Events

`Marionette.Object` extends `Backbone.Events` and includes `triggerMethod`. This makes it easy for Objects to emit events that other objects can listen for with `on` or `listenTo`.

```
var Friend = Marionette.Object.extend({
  graduate: function() {
    this.triggerMethod('announce', 'I graduated!!!');
  }
});

var john = new Friend({name: 'John'});

john.on('announce', function(message) {
  console.log(message); // I graduated!!!
})

john.graduate();
```

mergeOptions

Merge keys from the `options` object directly onto the instance. This is the preferred way to access options
passed into the Object.

More information at [mergeOptions]

getOption

Retrieve an object's attribute either directly from the object, or from the object's this.options, with this.options taking precedence.

More information [getOption].

bindEntityEvents

Helps bind a backbone "entity" to methods on a target object. More information [bindEntityEvents].

Destroying A Object

Objects have a `destroy` method that unbind the events that are directly attached to the instance.

Invoking the `destroy` method will trigger a "before:destroy" event and corresponding `onBeforeDestroy` method call. These calls will be passed any arguments `destroy` was invoked with. Invoking `destroy` will return the object, this can be useful for chaining.

```
// define a object with an onDestroy method
var MyObject = Marionette.Object.extend({

  onBeforeDestroy: function(arg1, arg2){
    // put custom code here, to destroy this object
  }

});

// create a new object instance
var obj = new MyObject();

// add some event handlers
obj.on("before:destroy", function(arg1, arg2){ ... });
obj.listenTo(something, "bar", function(){...});

// destroy the object: unbind all of the
// event handlers, trigger the "destroy" event and
// call the onDestroy method
obj.destroy(arg1, arg2);
```

▸ Basic Use

Selections is a simple Object that manages a selection of things. Because Selections extends from Object, it gets `initialize` and `Events` for free.

```
var Selections = Marionette.Object.extend({

  initialize: function(options){
    this.selections = {};
  },

  select: function(key, item){
    this.triggerMethod("select", key, item);
    this.selections[key] = item;
  },

  deselect: function(key, item) {
    this.triggerMethod("deselect", key, item);
    delete this.selections[key];
  }

});

var selections = new Selections({
  filters: Filters
});
```

```
// use the built in EventBinder
selections.listenTo(selections, "select", function(key, item){
  console.log(item);
});

selections.select('toy', Truck);
```