

Marionette.AppRouter

Служит для уменьшения количества шаблонного кода для обработки событий роутера (route events) и вызова определенного метода на другом объекте.

Позволяет конфигурировать роутеры, напрямую привязывая их к вызовам методов целевого объекта.

AppRouter приложения должен быть сконфигурирован с помощью `appRoutes`. Определение роутера передается на стандартные обработчики роутеров, как это принято в Backbone. Это значит, что вы определяете роутеры как будто бы вы имеете дело с обычным Backbone-маршрутизатором. Однако, вместо того, чтобы помещать функцию, обрабатывающую экземпляр пути, внутри роутера, вам следует добавить ее в соответствующий конструктор (см. ниже)

```
MyRouter = Backbone.Marionette.AppRouter.extend({
  // "someMethod" существует в контроллере в качестве controller.someMethod
  appRoutes: {
    "some/route": "someMethod"
  },

  /* стандартные Backbone-роуты могут дополнены с помощью appRoutes/Controllers, как это показано выше */
  routes : {
    "some/otherRoute" : "someOtherMethod"
  },
  someOtherMethod : function(){
    // делаем что-то
  }
});
```

Назначение контроллера

AppRouter может использовать только один объект контроллера. Вы можете обозначить это прямо в определении роутера:

```
someController = {
  someMethod: function(){ /* ... */ }
};
Backbone.Marionette.AppRouter.extend({
  controller: someController
});
```

... или через параметр, переданный в конструктор:

```
myObj = {
  someMethod: function(){ /* ... */ }
};
new MyRouter({
  controller: myObj
});
```

К объекту, используемому в качестве контроллера, не предъявляется никаких особенных требований, кроме одного: он должен содержать все те методы, которые вы определили в appRouter.

Рекомендуется объединять роутеры и соответствующие контроллеры по функциональности. Нормально, если приложение будет иметь несколько сравнительно небольших роутеров / контроллеров, вместо одного гигантского роутера и контроллера.