Marionette.Region

Регионы обеспечивают последовательное управление представлениями в приложении и предоставляют для этого специальные методы "show" и "close".

Задание региона в приложении

Регионы могут быть добавлены в приложение с помощью метода `addRegions`, вызванного на экземпляре приложения.
Данный метод принимает единственный параметр - хеш соответствий: именам регионов в нем соответствуют jQuery-селекторы.
Можно вызывать этот метод столько раз, сколько потребуется, в любой момент работы приложения.

```
MyApp.addRegions({
  mainRegion: "#main-content",
  navigationRegion: "#navigation"
});
```

Как только мы вызовем `addRegions`, регионы станут доступны в объекте приложения для дальнейшей работы.
В примере выше определены два региона: `MyApp.mainRegion` и `MyApp.navigationRegion`.

Для одного и того же региона, определенного дважды, будет справедливо последнее определение.

Инициализация региона с помощью `el`

Вы можете задать в объекте, переданном в качестве параметра в конструктор региона, атрибут `el`:

```
var mgr = new Backbone.Marionette.Region({
  el: "#someElement"
});
```

Базовое использование

Как только регион был определен, можно вызывать его методы `show` и `close` для отображения в регионе представления:

```
var myView = new MyView();
// внедряем и показываем представление
MyApp.mainRegion.show(myView);
// текущее представление для данного региона будет закрыто
MyApp.mainRegion.close();
```

Предыдущее представление, определенное в регионе, будет автоматически закрыто с показом нового представления посредством вызова метода `show`:

```
// показываем первое представление
var myView = new MyView();
MyApp.mainRegion.show(myView);
// заменяем его на другое - метод `close` выполнится автоматически
var anotherView = new AnotherView();
MyApp.mainRegion.show(anotherView);
```

Сброс региона (метод `reset`)

Операция "сброса" региона может быть произведена в любой момент. Эта операция деинсталлирует всякое прежде размещенное в нем представление, и удалит закешированный `el`.

В следующий раз, когда регион будет использован для показа представления, `el` будет запрошен из DOM.

```
myRegion.reset();
```

Это может оказаться полезным в сценариях с повторным использованием региона различными экземплярами приложения, или при юнит-тестировании.

Установка способа прикрепления `el` к DOM

Если необходимо изменить способ, которым представление встраивается в DOM при показе его в регионе, нужно переопределить метод `open` региона.
Метод принимает единственный переметр - экземпляр представления, который должен быть показан.

По умолчанию, реализация метода `open` следующая:

```
Marionette.Region.prototype.open = function(view){
  this.$el.empty().append(view.el);
}
```

В результате содержимое региона (оно находится в $el) заменяется содержимым представления. Но, конечно же, такое поведение может быть переопределено,
например, для реализации различных переходных эффектов и т.д. и т.п.

```
 Marionette.Region.prototype.open = function(view){
  this.$el.hide();
  this.$el.html(view.el);
  this.$el.slideDown("fast");
}
```

В данном примере представление будет "вдвигаться" в регион сверху.

Присоединение существующего представления

Бывают сценарии, где необходимо добавить экземпляр представления к региону, без его отображения, и без замены HTML-контента региона.

Есть два способа достичь этого:

- определить `currentView` в конструкторе региона
- вызвать `attachView` для экземпляра региона

Установка `currentView` при инициализации региона

```
var myView = new MyView({
  el: $("#existing-view-stuff")
});
var region = new Backbone.Marionette.Region({
  el: "#content",
  currentView: myView
});
```

Вызов `attachView` для экземпляра региона

```
MyApp.addRegions({
  someRegion: "#content"
});
var myView = new MyView({
  el: $("#existing-view-stuff")
});
MyApp.someRegion.attachView(myView);
```

События и коллбэки

Регион может триггерить следующие события при показе и закрытии представлений:

- "show" (и метод `onShow`) - Вызывается на экземпляре представления, когда представление отрендерено и показано.
- "show" (и метод `onShow`) - Вызывается на экземпляре региона, когда представление отрендерено и показано.
- "close" (и метод `onClose`) - Вызывается, когда представление закрыто.

Таким образом, у нас есть возможность связывать с данными событиями вызов соответствующего кода.

```
MyApp.mainRegion.on("show", function(view){
  // что-то делаем с представлением или проводим дополнительные операции над регионом посредством
указания ключевого слова `this`
});
MyApp.mainRegion.on("close", function(view){
  // . . . . .
});
MyRegion = Backbone.Marionette.Region.extend({
  // ...
  onShow: function(view){
    // представление `view` уже показано, можно что-то делать
  }
});
MyView = Marionette.ItemView.extend({
  onShow: function(){
    // вызывается, когда представление показано
  }
});
```

Функции обратного вызова и события представления для регионов

Когда представление отобразится в регионе, регион вызовет `onShow` метод на этом представлении. Заодно в представлении породится событие "show":

```
MyView = Backbone.View.extend({
  onShow: function(){
    // представление показано, делаем что-то
  }
});
view = new MyView();
view.on("show", function(){
  // представление показано, делаем что-то
});
MyApp.mainRegion.show(view);
```

Пользовательские типы регионов

Можно определить пользовательский регион-объект, произведя его от `Region`, для добавления новой функциональности.

Присоединение региона пользовательского типа

Раз уж вы определили новый регион-объект, вы можете использовать регион данного типа в приложении:

```
var FooterRegion = Backbone.Marionette.Region.extend({
  el: "#footer"
});
MyApp.addRegions({
  footerRegion: FooterRegion
});
```

Вы можете также указать селектор для региона, определив для конфигурации объектный литерал:

```
var FooterRegion = Backbone.Marionette.Region.extend({
  el: "#footer"
});
MyApp.addRegions({
  footerRegion: {
    selector: "#footer",
    regionType: FooterRegion
  }
});
```

Заметим, что регион, чтобы он появился на странице, должен быть связан с DOM-элементом. Если вы не определяете селектор в экземпляре региона, при добавлении его к экземпляру Application или Layout посредством "addRegions", то внутри самого экземпляра региона должно быть указано значение для атрибута `el` при создании его через вызов конструктора.

Создание экземпляра для объекта региона, определенного пользователем

Бывают случаи, когда нужно добавить регион в приложение после того, как оно сконфигурировано и запущено.
Тогда необходимо наследовать новый объект от объекта `Region`, и затем использовать получившийся в результате конструктор:

```
var SomeRegion = Backbone.Marionette.Region.extend({
  el: "#some-div",
  initialize: function(options){
   // код инициации
  }
});
MyApp.someRegion = new SomeRegion();
MyApp.someRegion.show(someView);
```

Как показано в этом примере, можно добавить опциональную функцию `initialize` в Region-объект, которая получает в качестве аргумента параметр `options`, который приходит в нее из конструктора, аналогично Backbone.View.