Применение BackboneJS для рефакторинга фронтенда веб-приложения

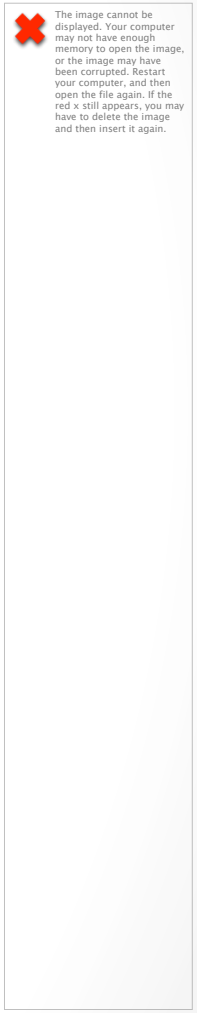
Алексей Авдеев, Go-Promo

Frontend party, Нижний Новгород, 22 ноября 2014

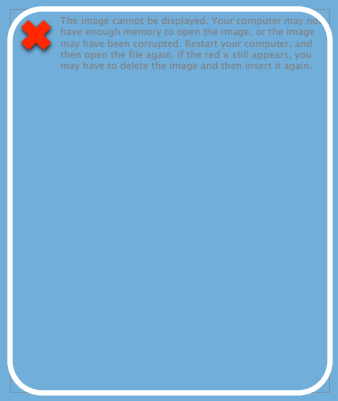
Как бывает

```
1652        file: $player.attr 'src'
1653        image: $player.attr 'poster'
1654        width: 600
1655        height: 350
```

CoffeeScript



Описание проблемы



Проект растёт

- Количество кода увеличивается
- Файл application.js разрастается

Внесение изменений становится опасным

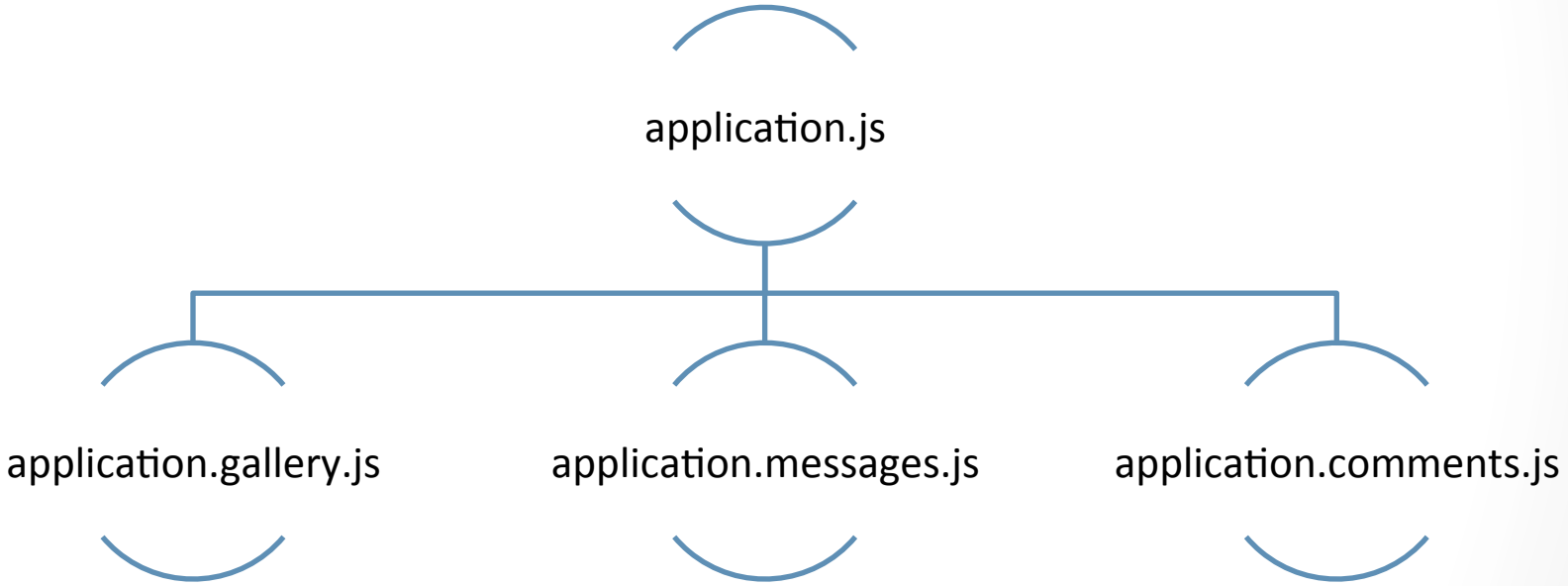
- Общая область видимости переменных
- Конфликты при работе в команде

Рефакторинг

Процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы.

(c) Wikipedia

Первая попытка



Пример

```
#= require jquery.cookie

#= require application.masonry
#= require application.gallery
#= require application.audioPlayer
#= require application.comments
#= require application.messages
#= require application.modalObj
```

CoffeeScript, Ruby on Rails

Преимущества

Меньше кода в каждом файле

Преимущества

Меньше конфликтов при работе в команде

Преимущества

Зачатки модульности

Преимущества

Стало понятнее, куда писать какой код

Недостатки

Название файла – лишь рекомендация

Недостатки

Глобальная область видимости осталась

Недостатки

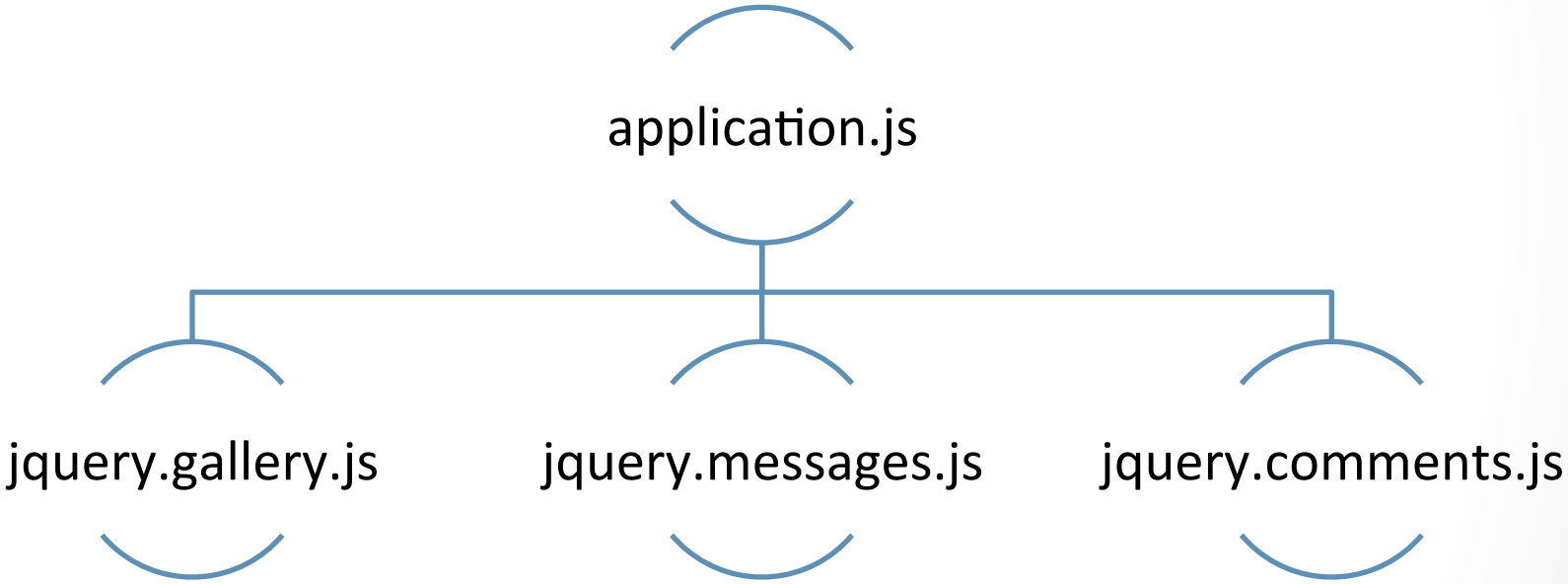
Появились проблемы с зависимостями

Недостатки

Каша в ресурсах лишь увеличилась

Вторая попытка

Что делаем


application.js
jquery.gallery.js
jquery.messages.js
jquery.comments.js

Описываем плагин

```
/**
 * jQuery nmdAuthors
 * http://go-promo.ru
 * Version: 0.3
 * Copyright 2013 Go-Promo
 */


(function($) {
  var PLUGIN_NAME = 'nmdAuthors';

  var methods = {
    init: function(options) {
      var settings = $.extend({
        urlAutocomplete: null
      }, options);

      if (!settings.urlAutocomplete) {
        console.error(PLUGIN_NAME, 'urlAutocomplete', settings.urlAutocomplete);
        return this;
      }

      return this.each(function() {
        var $authors = $(this);
```

JavaScript

Используем плагин

```
$('.js-banner-home-second-right').nmdBanner({
  placementId: app.BANNER_PLACEMENT['home_second_right']
});
```

JavaScript

Преимущества

jQuery-плагины хорошо изолированы

Преимущества

Модульность

Преимущества

Методы, состояния

Недостатки

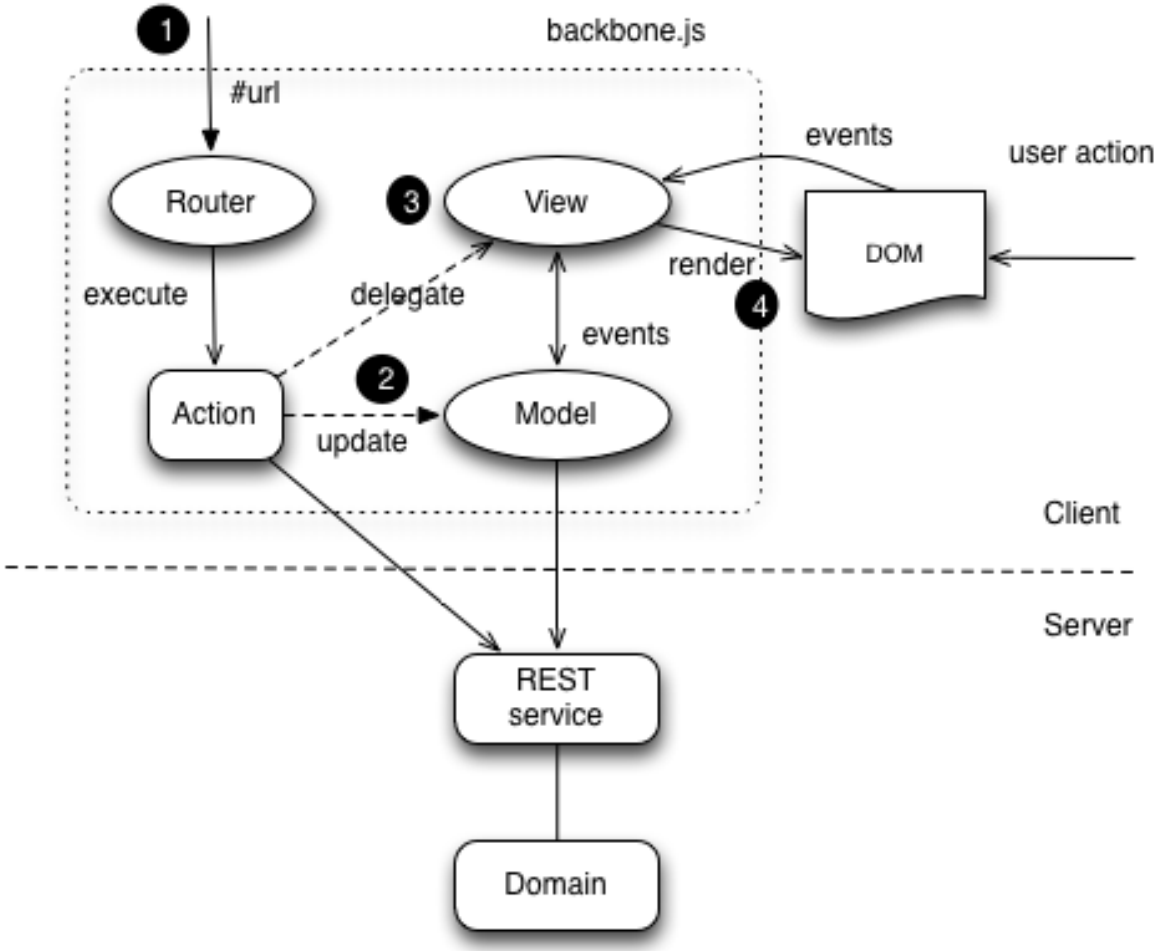
В applications.js всё равно остаётся много кода

Недостатки

Проблемы с зависимостями остаются

Третья попытка

Схема BackboneJS

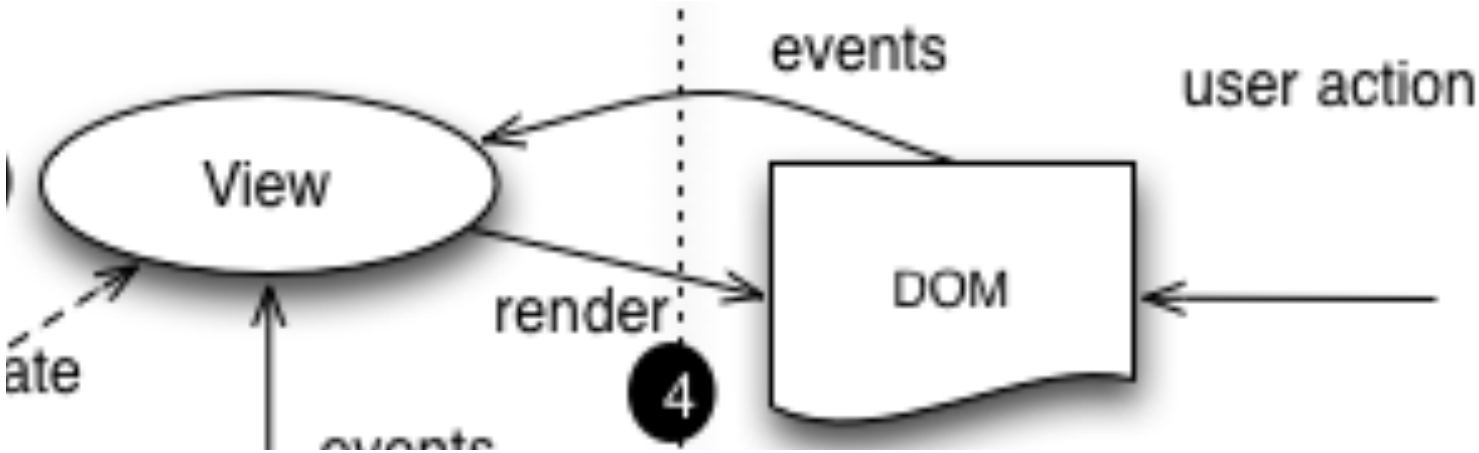


Почему можно рефакторить?

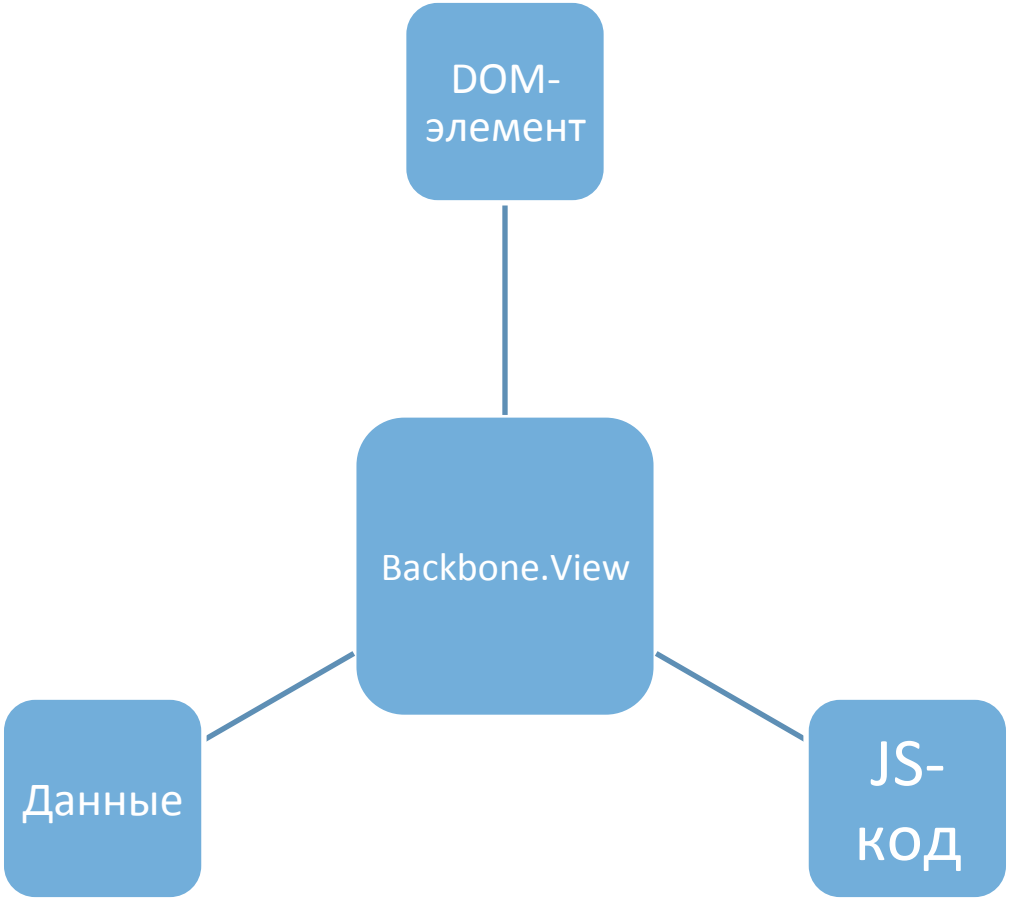
Backbone.js предназначен быть довольно агностичным насчёт многих частых паттернов клиентского кода

(c) backbonejs.ru

Что используем мы



Как будем действовать



Пример Backbone.View

```
GoPromo.Views.DigitalSliderView = Backbone.View.extend({
  el: "#digital-slider",

  events: {
    'click .slider-pager a': 'navigate'
  },

  initialize: function() {
    this.$sliderFeed = this.$('div.slider-feed');
    this.$sliderPager = this.$('div.slider-pager');
  },

  navigate: function(e) {
    e.preventDefault();

    var $li = $(e.target).closest('li');
    var index = $li.index();

    if (!$li.hasClass('active')) {

      this.$sliderPager.find('li').removeClass('active');
      $li.addClass('active');

      var $activeArticle = this.$sliderFeed.find('article').removeClass('active');
```

JavaScript

Что можно вытащить

```
$('section.publication').each(function() {
  var $content = $(this);

  $content.find('.audioPlayer').each(function(i, el) {
    var file = $(el).find('source').attr("src");
    $(el).attr('id', "audio_player_" + i);
    initializeAudioPlayer({
      file: file,
      id: "audio_player_" + i
    });
  });

  var sboxInner = $content.find('#horiz_container');
  var sboxInnerWidth = 0;
  $.each($(sboxInner).children('li'), function() {
    sboxInnerWidth += parseInt($(this).outerWidth()) + 12;
```

JavaScript

Что получается

```
1  class LifenewsRu.Views.Publication extends Backbone.View
2    el: 'section.publication'
3
4    initialize: ->
5      @renderAudio()
6      @renderGalleria()
7      @renderImageCaption()
8      @updateCounters()
9      @renderVideo()
10     @initPopularScroll()
11
12   renderAudio: ->
13     @$(".audioPlayer").each (i, el) ->
14       file = $(el).find("source").attr("src")
15       $(el).attr "id", "audio_player_#{i}"
16       initializeAudioPlayer
```

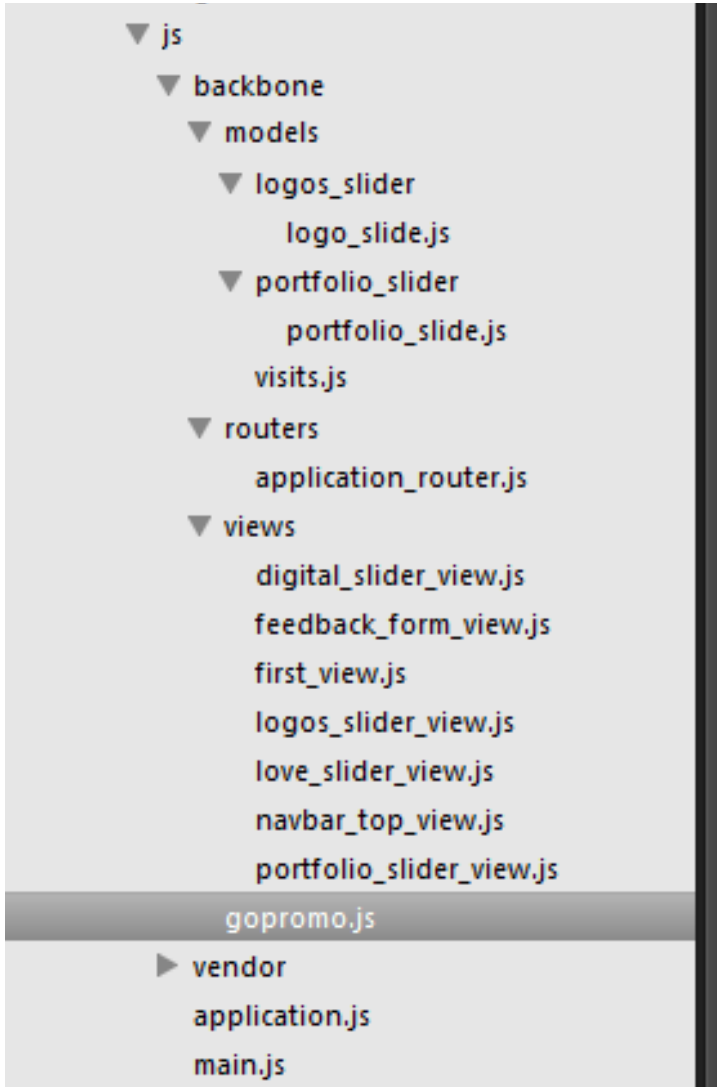
CofeeScript

Как рефакторить

Как рефакторить

Как рефакторить

Пример иерархии



Преимущества

Backbone.View хорошо изолированы

Преимущества

Поддерживаются сторонние решения

Преимущества

Лёгкость рефакторинга

Преимущества

Явная архитектура

Преимущества

Приятная поддержка кода

Преимущества

Лёгкая расширяемость

Преимущества

Архитектура разрешает зависимости

Недостатки

Надо знать Backbone.JS

Недостатки

Подключаем дополнительные файлы (underscore.js, backbone.js)

Недостатки

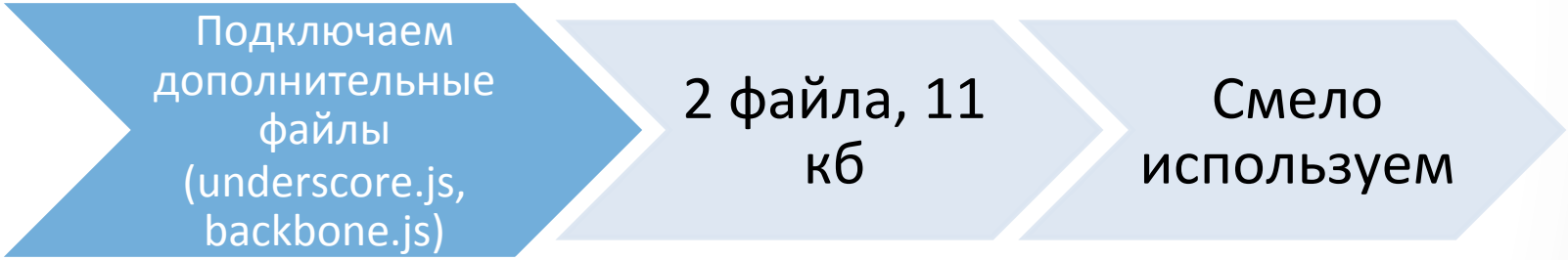
Много JS-файлов на странице

Решаем недостатки

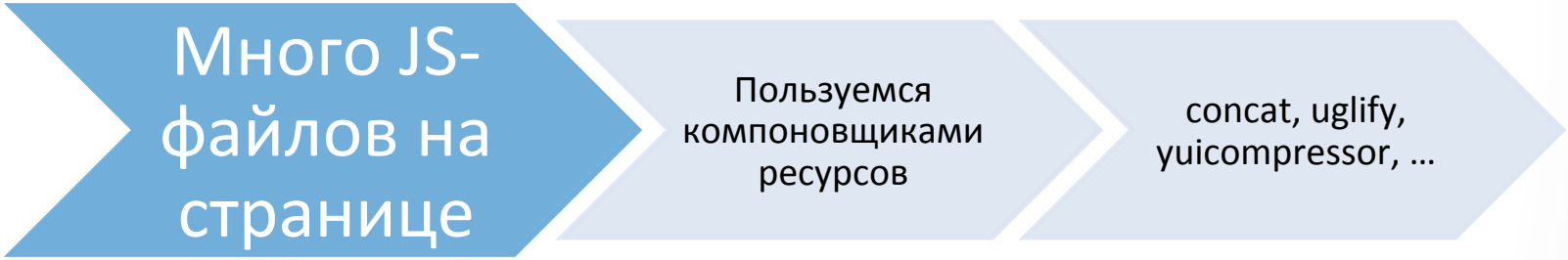
Надо знать Backbone.JS

Много документации, в т.ч. на русском

Решаем недостатки



Решаем недостатки



Пример

```
 </div>
 <script src="js/vendor/jquery-2.1.0.min.js"></script>
 <script src="js/vendor/underscore-min.js"></script>
 <script src="js/vendor/backbone-min.js"></script>
 <script src="js/vendor/jquery.form.min.js"></script>
 <script src="js/main.js"></script>

 <script src="js/backbone/gopromo.js"></script>
 <script src="js/backbone/routers/application_router.js"></script>
 <script src="js/backbone/models/portfolio_slider/portfolio_slide.js"></script>
 <script src="js/backbone/views/portfolio_slider_view.js"></script>
 <script src="js/backbone/views/logos_slider_view.js"></script>
 <script src="js/backbone/views/feedback_form_view.js"></script>
 <script src="js/backbone/views/first_view.js"></script>
 <script src="js/backbone/views/navbar_top_view.js"></script>
/body>
tml>
```

HTML

Ссылки

- **BackboneJS** (backbonejs.org)
- **CoffeeScript** (coffeescript.org)
- **JQuery Plugin** (learn.jquery.com/plugins)
- **UnderscoreJS** (underscorejs.org)
- **Grunt** (gruntjs.com)

Спасибо за внимание. Вопросы.

Алексей Авдеев

Нижний Новгород, Go-Promo, go-promo.ru