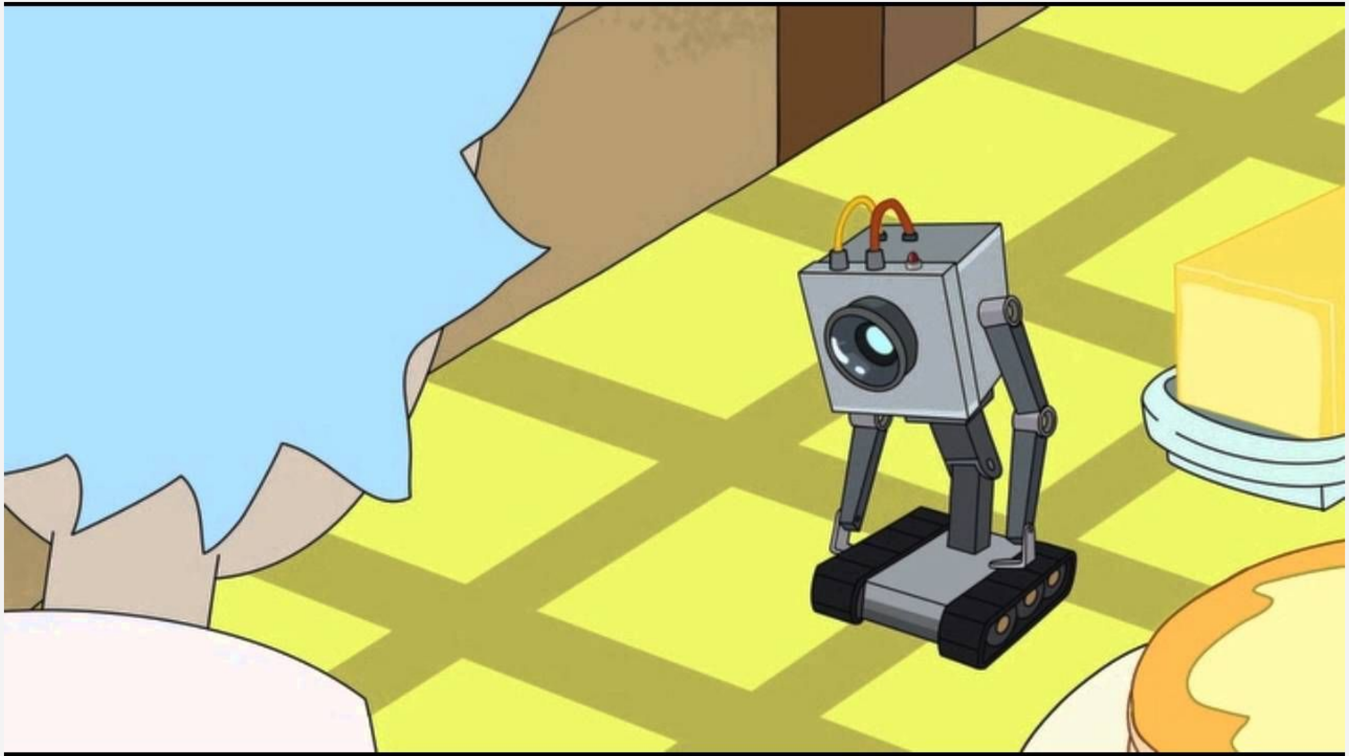
Yalantis

Лекция 11: Views

Курс лекций по основами web-разработки на языке программирования Ruby

Yalantis

Зачем нужны вьюхи?



HTTP GET /users

```
# config/routes.rb
 resources :users
```

app/views/users/index.html.erb

```
class UsersController < ApplicationController
  def index
    @users = User.all
  end
end
```

app/views/layouts/application.html.erb

магия
memesmix.net

Магия
MemesMix.net

1ST MONDAY
MAGIC

Layout

```
class ApplicationController < ActionController::Base
  layout :determine_layout
  def determine_layout
    if is_admin?
      "admin"
    else
      "application"
    end
  ...
end
```

Render

Rendering an action

```
render action: "goal"
```

Rendering text

```
render text: "OK!"
```

Rendering partials

```
render partial: "person", locals: { name: "david" }
```

Rendering a file

```
render file: "/path/to/some/template.erb"
```

Rendering an inline template

```
render inline: "<%= 'hello, ' * 3 + 'again' %>"
```

Yield

```
<%= yield :head %>
```

```
<% content_for :head do %>
```

Yield

```
# app/views/layouts/application.html.erb
<!DOCTYPE html>
<html>
 <head>
   <title>test</title>

   <%= yield :seo_data %>
 </head>

 <body>
   <%= yield %>


   <%= yield :some_scripts %>
 </body>
</html>
```

content_for

```
<% content_for :seo_data do %>
  <meta name="keywords" value="users">
<% end %>

<%= render partial: user_header %>


<% content_for :some_scripts do %>
  <script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY" defer></script>
<% end %>
```

Helpers

```ruby
module CisTypesHelper
 def ci_type_icon(ci)
   content_tag :div, class: 'icon', title: ci.name do
     content_tag :i, nil, class: ci.icon
   end
 end
end
```

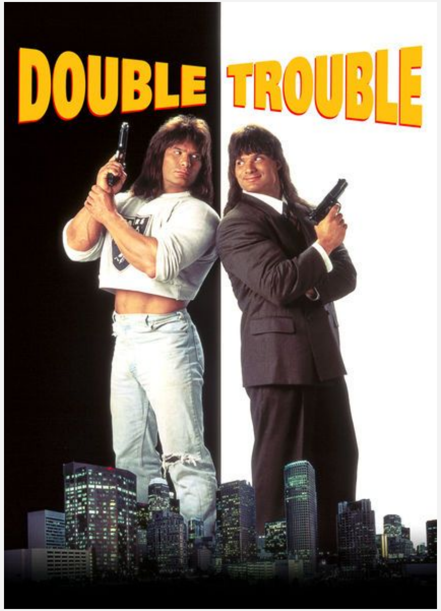
`config.action_controller.include_all_helpers`

Yalantis

ViewHelpers

```
<%= form_for @person do |f| %>
  First name: <%= f.text_field :first_name %><br />
  Last name : <%= f.text_field :last_name %><br />
  Biography : <%= f.text_area :biography %><br />
  Admin?    : <%= f.check_box :admin %><br />
  <%= f.submit %>
<% end %>
```

Проблема двойного рендеринга



```ruby
def show
   if @user.admin?
     render 'users/admin' and return
   end
   render 'users/visitor'
 end
```

Форматы данных

```
respond_to do |format|
     format.html { @cis = @cis.paginate(page: params[:page]) }
     format.csv
   end
```

```
/users.csv
/users?format=csv
```

Custom MIME type

```
# config/initializers/mime_types.rb
Mime::Type.register "image/jpg", :jpg
```

```
respond_to do |format|
     format.html
     format.jpg
   end
```

JSON Format

jbuilder

```
json.(comment, :id, :body, :is_public, :from_agent, :source, :created_at)
```

```
json.partial! 'api/v1/common/user', user: comment.created_by, attribute: :created_by
json.partial! 'api/v1/common/attachments', attachments: comment.attachments
```

```
{
  "id": 2,
  "body": "Comment, woo!",
  "is_public": false,
  "from_agent": true,
  "source": "mobile_app",
  "created_at": "2020-04-30T17:54:12.674Z",
  "created_by": {
    "id": 1049036234,
    "friendly_id": 1,
    "first_name": "Chester",
    "last_name": "Floyd",
    "username": "admin"
  },
  "attachments": []
}
```

Yalantis

Serializers

```
def index
  render json: @users, each_serializer: Api::V1::UserSerializer
end
```

```
class Api::V1::UserSerializer < ActiveModel::Serializer
  attributes :id, :name, :email
end
```

JSON API

```
{
 "data": [
   {
     "id": "1",
     "type": "amenities",
     "attributes": {
       "title": "amenity"
     }
   },
   {
     "id": "2",
     "type": "amenities",
     "attributes": {
       "title": "another amenity"
     }
   }
 ]
}
```

application/vnd.api+json

SOAP XML

```ruby
require 'savon'
client = Savon.client(wsdl: 'wsdl/magic.wsdl', ssl_verify_mode: :none, ssl_version: :TLSv1)

client.call(:example, message: {
  someStuff: 123,
  someAuthStuff: {
    username: "abc",
    password: "test123"
  },
  :attributes! => {
   someAuthStuff: {
     "xsi:type" => "n2:SpecialLogin",
     "xmlns:n2" => "http://magiccompany.com/specialstuff"
   }
 }
})
```

Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xmlns:mag="http://magiccompany.com/supermagic">
   <soapenv:Header/>
   <soapenv:Body>
      <mag:example>
         <someStuff>123</someStuff>
         <someAuthStuff xmlns:n2="http://magiccompany.com/specialstuff"
      xsi:type="n2:SpecialLogin">
           <username>abc</username>
           <password>test123</password>
         </someAuthStuff>
      </mag:example>
   </soapenv:Body>
</soapenv:Envelope>
```

Assets

```
{
 "name": "test",
 "version": "1.0.0",
 "main": "index.js",
 "repository": "https://bitbucket.org/test/test",
 "author": "Dmitriy Bavykin <test@gmail.com>",
 "license": "MIT",
 "dependencies": {
   "signature_pad": "^3.0.0-beta.3"
 }
}
```

Assets Pipeline

```
# app/assets/javascripts/application.js
//= require signature_pad/dist/signature_pad
//= require settings/account/general
//= require settings/cmdb/general
//= require settings/email/notifications
```

```
# app/assets/stylesheets/application.scss
@import "bootstrap-datetimepicker";
@import "bootstrap-multiselect";
@import "font-awesome";
```

```
# app/views/layouts/application.html.erb
<%= javascript_include_tag 'application' %>
```

```
<% if session[:use_high_contrast_theme] %>
     <%= stylesheet_link_tag 'application-high-contrast' %>
   <% else %>
     <%= stylesheet_link_tag 'application' %>
   <% end %>
```

WebPacker

```
app/javascript:
  ├── packs:
  │   # only webpack entry files here
  │   └── application.js
  └── src:
  │   └── application.css
  └── images:
      └── logo.svg
```

```
<%= javascript_pack_tag 'application' %>
<%= stylesheet_pack_tag 'application' %>
```

Asset Hacks

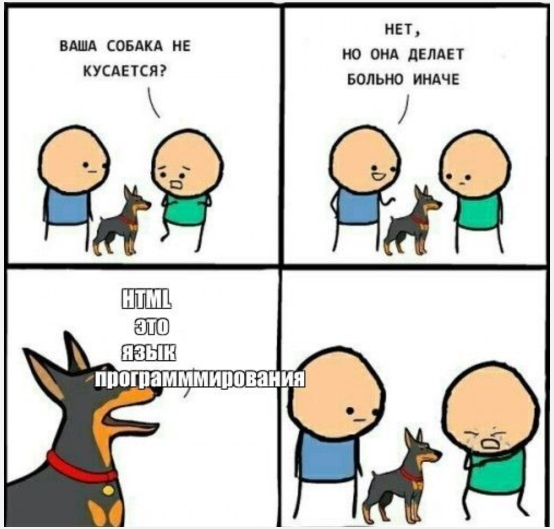
1. Minifiers
2. Linters
3. Turbolinks
4. Compressed assets
5. Cloudfront

HTML

```
<!DOCTYPE html>
<html>
 <head>
   <meta charset="utf-8">
   <meta name="keywords" content="HTML, CSS, JavaScript">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>My test page</title>
   <link rel="stylesheet" href="style.css">
   <style>
     a    {

      }
   </style>
 </head>
 <body>
   <img src="images/firefox-icon.png" alt="My test image" style="display: none;">
   <script defer="defer" src="some.js"></script>
 </body>
</html>
```


ВАША СОБАКА НЕ КУСАЕТСЯ?
НЕТ, НО ОНА ДЕЛАЕТ БОЛЬНО ИНАЧЕ
HTML ЭТО ЯЗЫК программмирования

HAML

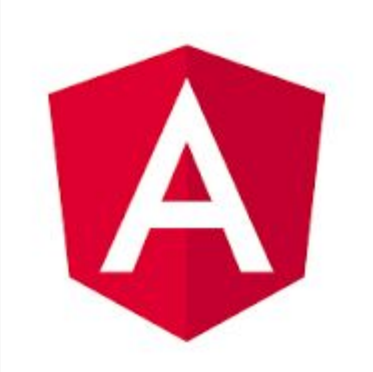
```
#content
  .left.column
    %h2 Welcome to our site!
    %p= print information
  .right.column
    = render partial: "sidebar"
```

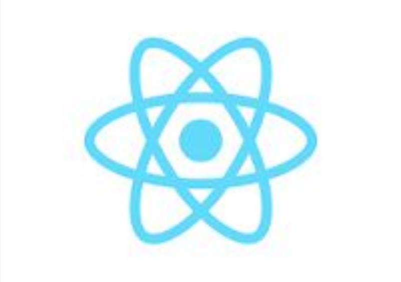
ERB

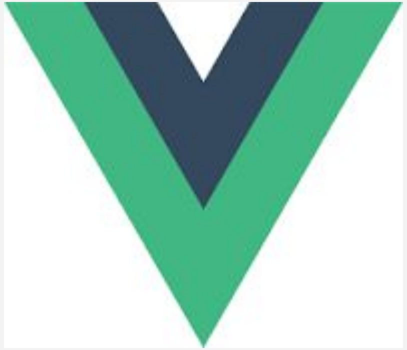
```
<div class="content-footer">
   <%= will_paginate @users %>
</div>


<%= render partial: 'common/footer %>
```

SPA







Асинхронные запросы

```
 <%= form_tag users_path, remote: true do |f| %>
```

```
format.json do
 if @purchase_item.save
   render json: @purchase_item
 else
   render json: {error: @purchase_item.errors.full_messages}, status: :unprocessable_entity
 end
end
```

```
def xhr_redirect_back
 render js: "Turbolinks.visit(location.toString());"
end
```

GraphQL

```
type Query {
  user(id: ID!): User
}

type Mutation {
  addPost(title: String!, userId: ID!): Post
}
```

Query

```
query ($id: Int) { # Define which variables will be used in the query (id)
  Media (id: $id, type: ANIME) { # Insert our variables into the query arguments (id) (type: ANIME is hard-coded in the query)
    id
    title {
      romaji
      english
      native
    }
  }
}
```

```
{
  "data": {
    "Media": {
      "id": 1,
      "title": {
        "romaji": "Cowboy Bebop",
        "english": "Cowboy Bebop",
        "native": "カウボーイビバップ "
      }
    }
  }
}
```

Mutation

```
mutation ($mediaId: Int, $status: MediaListStatus) {
  SaveMediaListEntry (mediaId: $mediaId, status: $status) {
    id
    status
  }
}
```

```
{
  "data": {
    "SaveMediaListEntry": {
      "id": 4,
      "status": "CURRENT"
    }
  }
}
```

Links

1. https://apidock.com/rails/ActionView/Helpers
2. https://anilist.gitbook.io/anilist-apiv2-docs/overview/graphql/getting-started
3. http://rusrails.ru/action-view-overview
4. http://haml.info/
5. https://github.com/Netflix/fast_jsonapi
6. https://jsonapi.org/
7. https://github.com/savonrb/savon
8. https://github.com/turbolinks/turbolinks
9. https://www.nopio.com/blog/rails-with-react-webpacker-gem/