



Представл ения в Rails

eRuby, action view, jbuilder,
partials



Язык шаблонов eRuby

Embedded Ruby – язык для работы с шаблонами текстов. Позволяет добавить код, написанный на Ruby в текстовый шаблон и выполнить этот код внутри шаблона.

```
1 <%# Commented line %>
2 <h1>Today is <%= Time.now.strftime('%A') %></h1>
3 <div>
4   <p>List of items:</p>
5   <% if @items.empty? %>
6     <p>There is no items.</p>
7   <% else %>
8     <% @items.each do |item| %>
9       <p><%= item[:name] %> | <%= item[:price] %></p>
10    <% end %>
11  <% end %>
12 </div>
```

```
1
2 <h1>Today is Sunday</h1>
3 <div>
4   <p>List of items:</p>
5
6   <p>milk | 71</p>
7
8   <p>chocolate | 85</p>
9
10
11 </div>
12
```

```
1 require 'erb'
2
3 erb_template = File.read('./template.html.erb')
4
5 @items = [
6   { name: 'milk', price: 71 },
7   { name: 'chocolate', price: 85 }
8 ]
9
10 renderer = ERB.new(erb_template)
11
12 File.write('./template.html', renderer.result)
```

Контекст объектов в ERB

```
1  require 'erb'
2
3  class ItemsController
4    def index
5      @items = [
6        { name: 'milk', price: 71 },
7        { name: 'chocolate', price: 85 }
8      ]
9
10     render :index
11   end
12
13   def show
14     @item = { name: 'milk', price: 71 }
15
16     render :show
17   end
18
19   private
20
21   def render(file_name)
22     erb_template = File.read("./#{file_name}.html.erb")
23     renderer = ERB.new(erb_template)
24
25     # Rendering without binding doesn't use variables of current obj
26     # File.write("./#{file_name}.html", renderer.result)
27     File.write("./#{file_name}.html", renderer.result(binding))
28   end
29 end
30
31 ItemsController.new.index
32 ItemsController.new.show
```

```
1  <%=# Commented line %>
2  <h1>Today is <%= Time.now.strftime('%A') %></h1>
3  <div>
4    <p>List of items:</p>
5    <% if @item.nil? %>
6      <p>There is no item.</p>
7    <% else %>
8      <p><%= @item[:name] %> | <%= @item[:price] %></p>
9    <% end %>
10 </div>
```

```
1  |
2  <h1>Today is Sunday</h1>
3  <div>
4    <p>List of items:</p>
5    |
6    <p>milk | 71</p>
7    |
8  </div>
```

Action View представления

```

1 require 'action_view'
2
3 view = ActionView::Base
4   .with_empty_template_cache
5   .with_view_paths(['/Users/alex/Documents/views'], {}, nil)
6
7 items = [
8   { name: 'milk', price: 71 },
9   { name: 'bread', price: 40 },
10  { name: 'sugar', price: 100 }
11 ]
12
13 view.assign(items: items)
14
15 puts "\nTemplate view:\n"
16 puts view.render(template: :index)
17
18 puts "\nInline view:\n"
19 puts view.render(inline: '<p><%= @items.first[:name] %></p>')
20
21 puts "\nPlain text:\n"
22 puts view.render(plain: 'simple text')
23
24 puts "\nHtml formatting:\n"
25 puts view.render(html: '<html format & text>')

```

```
Template view: 71 },
<div>', price: 40 },
su<h1>List of items:</h1>
    <div>
        <p>milk</p>
items: <p>71</p>
    </div>
ulate <div> \n"
ender(templates.index)
    <p>bread</p>
    <p>40</p>
line view"
    </div>
ender(template="@items
    <p>sugar</p>
    <p>100</p>
    </div> 'simple text')
</div>
formatting:\n"
Inline view: '<html format &
<p>milk</p>

Plain text:
simple text

Html formatting:
<html format & text&gt;
```

Представления вида XML и JSON

```
1 xml.rss('version' => '2.0') do
2   xml.title('List of items:')
3
4   if @items.empty?
5     xml.error('There is no items')
6   else
7     @items.each do |item|
8       xml.item('item:type' => 'product') do
9         xml.name(item[:name])
10        xml.tag!('item:price', item[:price])
11      end
12    end
13  end
14 end
```

XML template:

```
<rss version="2.0">
  <title>List of items:</title>
  <item item:type="product">
    <name>milk</name>
    <item:price>71</item:price>
  </item>
  <item item:type="product">
    <name>bread</name>
    <item:price>40</item:price>
  </item>
  <item item:type="product">
    <name>sugar</name>
    <item:price>100</item:price>
  </item>
</rss>
```

```
1 json.items do
2   if @items.empty?
3     json.error('There is no items')
4   else
5     json.array! @items do |item|
6       json.name(item[:name])
7       json.price(item[:price])
8     end
9   end
10 end
```

```
1 require 'jbuilder'
2
3 @items = [
4   { name: 'milk', price: 71 },
5   { name: 'bread', price: 40 },
6   { name: 'sugar', price: 100 }
7 ]
8
9 jbuilder = Jbuilder.new do |json|
10   json.items do
11     if @items.empty?
12       json.error('There is no items')
13     else
14       json.array! @items do |item|
15         json.name(item[:name])
16         json.price(item[:price])
17       end
18     end
19   end
20 end
21
22 puts jbuilder.target!
```

```
{"items":[{"name":"milk","price":71}, {"name":"bread","price":40}, {"name":"sugar","price":100}]}
```

Части представления - partials

```
1 <%= render('shared/header') %>
2 <div>
3   <h1>List of items:</h1>
4   <% if @items.empty? %>
5     <p>There is no items.</p>
6   <% else %>
7     <% @items.each do |item| %>
8       <div>
9         <p><%= item[:name] %></p>
10        <p><%= item[:price] %></p>
11      </div>
12    <% end %>
13  <% end %>
14 </div>
15 <%= render('shared/footer') %>
```

```
_header.html.erb x index.html.e
1 <h1>Items list</h1>
2 <p>This is list of items</p>
```

```
_footer.html.erb x _header.html.erb
1 <p>(c) 2022</p>
2 <p>Current time: <%= Time.now %></p>
```

```
<h1>Items list</h1>
<p>This is list of items</p>
<div>
  <h1>List of items:</h1>
  <div>
    <p>milk</p>
    <p>71</p>
  </div>
  <div>
    <p>bread</p>
    <p>40</p>
  </div>
  <div>
    <p>sugar</p>
    <p>100</p>
  </div>
</div>
<p>(c) 2022</p>
<p>Current time: 2022-09-05 12:43:37 +0400</p>
```

Partials для коллекций объектов

```
7 <% @items.each do |item| %>
8   <%= render(partial: 'shared/item', locals: { item: item }) %>
9 <% end %>
```

```
7 <% @items.each do |item| %>
8   <%= render('shared/item', item: item) %>
9 <% end %>
```

```
7 <% @items.each do |item| %>
8   <%= render(partial: 'shared/item', object: item) %>
9 <% end %>
```

```
6 <% else %>
7   <%= render(partial: 'shared/item', collection: @items) %>
8   <## render(@items) - for ActiveRecord objects %>
9 <% end %>
```

```
<h1>Items list</h1>
<p>This is list of items</p>

<div>
  <h1>List of items:</h1>
  <div>
    <p>milk</p>
    <p>71</p>
  </div>
  <div>
    <p>bread</p>
    <p>40</p>
  </div>
  <div>
    <p>sugar</p>
    <p>100</p>
  </div>
</div>
<p>(c) 2022</p>
<p>Current time: 2022-09-05 12:54:44 +0400
```

Полезные ссылки

<https://github.com/rails/rails/tree/9ff7c9c0231b147486dfdd38325932c95604f1ce/actionview> – Библиотека Action View в GitHub, как часть Rails;

<http://rusrails.ru/action-view-overview> – Обзор Action View.

Конец! Спасибо!