



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети(ИУ6)

О т ч е т

по лабораторной работе № 10

Дисциплина: Языки интернет-программирования

Название лабораторной работы: Формирование и отображение XML в HTML средствами сервера и клиента.

Студент гр. ИУ6-32Б _____ Д.В.Лабзунова
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____
(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Задание

Модифицировать код ЛР 8 таким образом, чтобы по запросу с указанными параметрами выдавался результат в формате XML (средствами стандартной сериализации ActiveSupport).

- Проверить формирование XML и сохранить в файл для отладки XSLT и второго приложения.
- Написать функциональный тест, проверяющий формат выдаваемых данных при запросе RSS.

Разработать XSLT-программу преобразования полученной XML в HTML.

Добавить в проверяемый XML-файл строку привязки к преобразованию `<?xml-stylesheet type="text/xsl" href="some_transformer.xslt">`. Проверить корректность отображения браузером результата преобразования.

Проверить на автономной Ruby-программе корректность преобразования, используя следующий фрагмент кода:

```
require 'nokogiri'

doc = Nokogiri::XML(File.read('some_file.xml'))

xslt = Nokogiri::XSLT(File.read('some_transformer.xslt'))

puts xslt.transform(doc)
```

Разработать второе приложение, являющееся посредником между клиентом и первым приложением, задачей которого является преобразование XML в HTML или передача в неизменном виде браузеру для отображения браузером. Приложение должно запускаться с указанием номера порта TCP, отличным от номера порта первого приложения (например rails server -p 3001)!

- Подготовить каркас приложения, а также форму формирования запроса, форму отображения результата и соответствующие действия контролера.
- Добавить в контроллер преобразование XML в HTML с помощью ранее разработанного XSLT-файла.
- Подключить запрос XML с первого приложения и проверить работу приложений в связке.
- Написать функциональный тест, проверяющий что при различных входных данных результат генерируемой страницы различен.
- Доработать код контроллера и представлений данного приложения для выдачи браузеру XML-потока в неизменном виде (организовать возможность выбора формата выдачи для пользователя).
- Проверить, что браузер получает XML первого приложения в неизменном виде.
- Доработать код контроллера приложения таким образом, чтобы XML-поток первого приложения получал дополнительную строку, указывающую xsl. Модифицировать форму запроса параметров таким образом, чтобы браузер получал в ответ XML. При этом разместить XSLT-файл в директории public.
- Проверить, что браузер производит преобразование XML->HTML в соответствии с xlt.
- Реализовать функциональные тесты второго приложения. Проверить результаты, формируемые приложением, на соответствие выбранному формату выдачи.

Итоговая форма ввода параметра должна содержать кнопки или селектор, позволяющие проверить два варианта преобразования:

- Серверное `xml+xslt->html`
- Клиентское `xml+xslt→html`

Код:

xml_controller:

```
# frozen_string_literal: true
```

```
# Output controller
```

```
# :reek:TooManyStatements
```

```
# :reek:InstanceVariableAssumption
```

```
class XmlController <
```

```
  before_action :parse_params, only: :index
```

```
  def index
```

```
    result = "
```

```
    sequence = @sequence
```

```
    if sequence == "
```

```

    result = 'you did not write anything:('
  elsif sequence.to_i.zero?
    result = 'please, enter a number'
  else
    res = find(sequence)
    rising_subsequences = res[0]
    if rising_subsequences.length.zero?
      result = 'There are no rising subsequences'
    else
      max = res[1]
      result = rising_subsequences.map { |elem| { subsequence: elem.to_s } }
        .append(max: max.to_s)
    end
  end
end

respond_to do |format|
  res = render xml: result.to_xml
  format.xml { res }
  format.rss { res }
end
end

protected

def parse_params
  @sequence = params[:sequence]
end

# :reek:UncommunicativeVariableName
# :reek:UtilityFunction
def find(sequence)
  rising_subsequences = []
  subsequence = sequence[0]
  max = ''
  count = 1
  n = 0
  (1..sequence.length).each do |i|
    seq_i = sequence[i]
    if seq_i.to_i > sequence[i - 1].to_i
      count += 1
      subsequence += seq_i
    else
      if count > 1
        rising_subsequences[n] = subsequence
        max = subsequence if subsequence.length > max.length
        n += 1
      end
      count = 1
      subsequence = seq_i
    end
  end
  [rising_subsequences, max]
end
end
end

```

xml_controller test:

```
require 'test_helper'
```

```

class XmlControllerTest < ActionController::TestCase
  # Проверяем, какой формат получили от контроллера при запросе в RSS.
  test 'check rss format' do
    # Делаем запрос с параметрами - словарь превратится в lower=10&upper=100&format=rss
    get '/', params: { sequence: 13457321934, format: :rss }
  end
end

```

```

# Проверили, что получили статус ответа 200 OK
assert response :success
# Тип ответа лежит в его заголовках, поэтому смотрим туда.
assert includes @response.headers['Content-Type'], 'application/rss'
end
end

```

proxy_controller:

```

require 'nokogiri'
require 'open-uri'

# Request controller
# :reek:UtilityFunction
# :reek:InstanceVariableAssumption
class ProxyController < ActionController::Base
  before_action :parse_params, only: :output
  before_action :prepare_url, only: :output

  def input; end

  def output
    api_response = open(@url)

    if @side == 'server'
      @result = xslt_transform(api_response).to_html
    elsif @side == 'client-with-xslt'
      render xml: insert_browser_xslt(api_response).to_xml
    else
      render xml: api_response
    end
  end

  private

  BASE_API_URL = 'http://localhost:3000/?format=xml'.freeze
  XSLT_SERVER_TRANSFORM = "#{Dir.pwd}/public/server_transform.xslt".freeze
  XSLT_BROWSER_TRANSFORM = '/browser_transform.xslt'.freeze

  def parse_params
    @sequence = params[:sequence]
    @side = params[:side]
  end

  def prepare_url
    @url = BASE_API_URL + "&sequence=#{@sequence}"
  end

  def xslt_transform(data, transform: XSLT_SERVER_TRANSFORM)
    doc = Nokogiri::XML(data)
    xslt = REXML::XSLT.new( File.read(transform))
    xslt.transform(doc)
  end

  def insert_browser_xslt(data, transform: XSLT_BROWSER_TRANSFORM)
    doc = Nokogiri::XML(data)
    xslt = REXML::XSLT.new(
      File.read(transform),
      'xmlstylesheet',
      'type="text/xsl" href="' + transform + '"'
    )
    doc.root.add_previous_sibling(xslt)
    doc
  end
end

```

end

inout.html.erb:

```
<%= form_tag url_for(controller: :proxy, action: :output), method: 'get' do %>
  <div class="form-group">
    <h3>Введите последовательность</h3>
  </div>
  <div class="form-group">
    <%= label_tag :sequence, 'последовательность' %>
    <%= text_field_tag :sequence, nil, class: 'form-control' %>
  </div>
  <fieldset class="form-group">
    <legend>Выбор стороны рендеринга</legend>
    <div class="form-check">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="side" value="server" id="server-side" checked>
        Рендеринг на стороне сервера этого приложения
      </label>
    </div>
    <div class="form-check">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="side" value="client" id="client-side">
        Отображение сырого XML на стороне браузера
      </label>
    </div>
    <div class="form-check">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="side" value="client-xslt" id="client-side-xslt">
        Рендеринг на стороне браузера с помощью XSLT
      </label>
    </div>
  </fieldset>
  <%= submit_tag 'Отправить', class: 'btn btn-primary' %>
<% end %>
```

output.html.erb

```
<h1>Ответ сервера</h1>
<%= render inline: @result %>
<!--“возьми текст в @result и отрисуй его как HTML”;-->
```

proxy_controller test:

require 'test_helper'

```
class ProxyControllerTest < ActionController::TestCase
  test "should get input" do
    get proxy_input_url
    assert_response :success
  end

  test "should get output" do
    get '/proxy/output', params: { sequence: '123423', side: 'server' }
    assert_response :success
    assert_includes @response.headers['Content-type'], 'text/html'

    get '/proxy/output', params: { sequence: '123423', side: 'client-with-xslt' }
    assert_response :success
    assert_includes @response.headers['Content-type'], 'application/xml'

    get '/proxy/output', params: { sequence: '123423', side: 'client' }
    assert_response :success
    assert_includes @response.headers['Content-type'], 'application/xml'
  end
end
```

end

Результаты выполнения:

Выбор:

Введите последовательность

последовательность

Выбор стороны рендеринга

- ☒ Рендеринг на стороне сервера этого приложения
- ☐ Отображение сырого XML на стороне браузера
- ☐ Рендеринг на стороне браузера с помощью XSLT

Отправить

На стороне сервера:

Ответ сервера

subsequences

12345

12

09

346

Maximum length rising subsequence:

12345

XML на стороне браузера:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<objects type="array">
  <object>
    <subsequence>12345</subsequence>
  </object>
  <object>
    <subsequence>12</subsequence>
  </object>
  <object>
    <subsequence>09</subsequence>
  </object>
  <object>
    <subsequence>346</subsequence>
  </object>
  <object>
    <max>12345</max>
  </object>
</objects>
```

XSLT на стороне браузера:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<objects type="array">
  ▼<object>
    <subsequence>12345</subsequence>
  </object>
  ▼<object>
    <subsequence>12</subsequence>
  </object>
  ▼<object>
    <subsequence>09</subsequence>
  </object>
  ▼<object>
    <subsequence>346</subsequence>
  </object>
  ▼<object>
    <max>12345</max>
  </object>
</objects>
```

Тесты:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-api/test$ rake
(in /home/labzunova/Рабочий стол/labs BMSTU/LR10/xslt-api)
Run options: --seed 32873
```

```
# Running:
```

```
.
```

```
Finished in 0.055599s, 17.9858 runs/s, 53.9574 assertions/s.
1 runs, 3 assertions, 0 failures, 0 errors, 0 skips
```

```
2 runs, 1 assertions, 0 failures, 1 errors, 0 skips
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-proxy/test$ rake
(in /home/labzunova/Рабочий стол/labs BMSTU/LR10/xslt-proxy)
Run options: --seed 64216
```

```
# Running:
```

```
/home/labzunova/Рабочий стол/labs BMSTU/LR10/xslt-proxy/app/controllers/proxy_
el#open is deprecated, call URI.open directly or use URI#open
/home/labzunova/Рабочий стол/labs BMSTU/LR10/xslt-proxy/app/controllers/proxy_
el#open is deprecated, call URI.open directly or use URI#open
/home/labzunova/Рабочий стол/labs BMSTU/LR10/xslt-proxy/app/controllers/proxy_
el#open is deprecated, call URI.open directly or use URI#open
..
```

```
Finished in 0.179481s, 11.1432 runs/s, 55.7162 assertions/s.
2 runs, 10 assertions, 0 failures, 0 errors, 0 skips
```

Проверка rubocop и reek:

Rubocop:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-api/app/controllers$ rubocop
NOTE: Gem.gunzip is deprecated; use Gem::Util.gunzip instead. It will be removed on or after 2018-12-01.
Gem.gunzip called from /usr/lib/ruby/vendor_ruby/unicode/display_width/index.rb:5.
Warning: unrecognized parameter AllCops:NewCops found in /home/labzunova/Рабочий стол/labs BMSTU/LR10/.rubocop.yml
Inspecting 2 files
..

2 files inspected, no offenses detected
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-proxy/app/controllers$ rubocop
NOTE: Gem.gunzip is deprecated; use Gem::Util.gunzip instead. It will be removed on or after 2018-12-01.
Gem.gunzip called from /usr/lib/ruby/vendor_ruby/unicode/display_width/index.rb:5.
Warning: unrecognized parameter AllCops:NewCops found in /home/labzunova/Рабочий стол/labs BMSTU/LR10/.rubocop.yml
Inspecting 2 files
..

2 files inspected, no offenses detected
```

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-api/app/controllers$ reek
Inspecting 2 file(s):
..

0 total warnings
```

Reek:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR10/xslt-proxy/app/controllers$ reek
Inspecting 2 file(s):
..

0 total warnings
```