



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

## ОТЧЕТ

по лабораторной работе № 10

Название:

**Формирование и отображение XML в HTML  
средствами сервера и клиента.**

Дисциплина: Языки интернет-программирования

Студент

ИУ6-35Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

К. А. Спасскова

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Я. С. Петрова

(И.О. Фамилия)

Москва, 2022

## Вариант 19

**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 xslt.
- 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

Клиент

Контроллер:

```
# frozen_string_literal: true

# class of controller

class ElementsController < ApplicationController

  include ElementsHelper

  before_action :check_data, only: :result

  before_action :set_values, only: :result


  def index; end
```

```

def result

  url_with_params = URL_SERV +
"?length=#{@length}&str_arr=#{@str.split.join('+)}"

  @resp_xml = Nokogiri::XML(URI.open(url_with_params))

  show_result

end

end

```

Хелпер:

```

# frozen_string_literal: true

# helper

module ElementsHelper

  URL_SERV = 'http://127.0.0.1:3000/serv/result.xml'

  XSLT_PATH = 'public/output.xslt'

  def check_data

    if /^[^d^s]/.match(params[:str_arr]) then redirect_to home_path,
notice: "Sequence isn't correct"

    else

      arr = params[:str_arr].split.map!(&:to_i)

      redirect_to home_path, notice: "Length isn't correct" if
arr.length != params[:length].to_i

    end

    flash[:notice] = 'Mode must be selected' if params[:mode].nil?

  end

end

```

```

def set_values

  @str = params[:str_arr]

  @length = params[:length].to_i

  @mode = params[:mode]

end

def show_result

  case @mode

  when 'xml'

    render xml: @resp_xml

  when 'html'

    render inline: xml_transform

  end

end

def xml_transform

  xslt = Nokogiri::XSLT(File.read(XSLT_PATH))

  xslt.transform(@resp_xml)

end

end

```

Пути:

```
# frozen_string_literal: true
```

```

Rails.application.routes.draw do

  root 'elements#index', as: 'home'

  get 'elements/result'

  get 'elements/index'

  post 'elements/result', to: 'elements#result'

  # Define your application routes per the DSL in
https://guides.rubyonrails.org/routing.html

  # Defines the root path route ("/")
  # root "articles#index"

end

```

## Тесты:

```

# frozen_string_literal: true

require 'rails_helper'

RSpec.describe ElementsController, type: :controller do
  it 'returns success' do
    get :index
    expect(response.status).to eq(200)
  end

  it 'show in html' do
    get :result, params: { str_arr: '5 5 6 0 1 4', length: '6', mode:
'html' }
    expect(response.content_type).to eq('text/html; charset=utf-8')
  end

  it 'show in xml' do
    get :result, params: { str_arr: '5 5 6 0 1', length: '5', mode:
'xml' }
    expect(response.content_type).to eq('application/xml;
charset=utf-8')
  end
end

```

```
end
end
```

```
● ksusha@ksusha-notebook:~/uni/labs_web/lab10_client$ rspec spec
...

Finished in 0.52577 seconds (files took 1.21 seconds to load)
3 examples, 0 failures

○ ksusha@ksusha-notebook:~/uni/labs_web/lab10_client$
```

Rubocop:

```
kusha@kusha-notebook:~/uni/labs_web/labai_client$ rubocop
Inspecting 38 files
....CCG..C....C.....

Offenses:

app/controllers/elements_controller.rb:14:35: C: Security/Open: The use of URI.open is a serious security risk.
  @resp_xml = Nokogiri::XML(URI.open(url_with_params))
                              ^^^^^^

app/helpers/application_helper.rb:3:1: C: Style/Documentation: Missing top-level documentation comment for module ApplicationHelper.
module ApplicationHelper
~~~~~

app/helpers/elements_helper.rb:8:3: C: Metrics/AbcSize: Assignment Branch Condition size for check_data is too high. [42, 20, 6> 20.98/17]
  def check_data ...
    ~~~~~

bin/bundle:24:3: C: Metrics/CyclomaticComplexity: Cyclomatic complexity for cli_arg_version is too high. [9/7]
  def cli_arg_version ...
    ~~~~~

bin/bundle:24:3: C: Metrics/MethodLength: Method has too many lines. [11/10]
  def cli_arg_version ...
    ~~~~~

bin/bundle:24:3: C: Metrics/PerceivedComplexity: Perceived complexity for cli_arg_version is too high. [9/8]
  def cli_arg_version ...
    ~~~~~

bin/bundle:91:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
  def activate_bundler ...
    ~~~~~

bin/bundle:104-121: C: Layout/LineLength: Line is too long. [198/120]
  warn "Activating bundler (#{bundler_requirement}) failed:\n#{gem_error.message}\n\nTo install the version of bundler this project requires, run 'gem install bundler -v '#{bundler_r
  requirement}':".
                                     ~~~~~

config/application.rb:12:3: C: Style/Documentation: Missing top-level documentation comment for class LabaiClient::Application.
class Application < Rails::Application
~~~~~

38 files inspected, 9 offenses detected
```

Сервер

Контроллер:

```
# frozen_string_literal: true

# class of controller

class ServController < ApplicationController

  include ServHelper

  before_action :set_params, only: :result

  before_action :segments_of_powers, only: :result

  before_action :largest_segment, only: :result


  def index; end
end
```



```

def result

  @res = { before_processing: @arr, length: @length, after: @res_arr,
largest_segment: @max_subarr }

  respond_to do |format|

    format.html

    format.xml { render xml: @res }

  end

end

end

end

```

Хелпер:

```

# frozen_string_literal: true

# hepler

module ServHelper

  def set_params

    @str = params[:str_arr]

    @length = params[:length].to_i

    @arr = @str.split.map!(&:to_i)

    @res_arr = []

  end

  def power_of_5?(num)

    tmp = 0

```

```

return false if num.zero?

while tmp.zero?

  tmp = num % 5

  num /= 5

end

if tmp.eql?(1) && num.zero? then true

else

  false

end

end

def segments_of_powers

  buf = []

  @arr.each do |elem|

    if power_of_5?(elem) then buf << elem

    elsif !buf.length.zero?

      @res_arr << buf

      buf = []

    end

  end

  @res_arr << buf unless buf.length.zero?

  @res_arr

end

```

```

def largest_segment

  return if @res_arr.length.zero?

  @max_subarr = @res_arr[0]

  @res_arr.each do |elem|

    @max_subarr = elem if @max_subarr.length < elem.length

  end

end

end
end

```

Пути:

```

# frozen_string_literal: true

Rails.application.routes.draw do

  get 'serv/index'

  post 'serv/result', to: 'serv#result'

  get 'serv/result'

  root 'serv#index'

end

```

Тесты:

```

# frozen_string_literal: true

require 'rails_helper'

```

```
# testing requests
RSpec.describe ServController, type: :controller do
  it 'returns http success' do
    get :index
    expect(response.status).to eq(200)
  end

  it 'returns http success' do
    get :result, params: { str_arr: '5', length: '1' }
    expect(response.status).to eq(200)
  end

  it 'returns correct result' do
    get :result, params: { str_arr: '1 5 6 0 5 625 625 0 8', length: '9' }
  }
  expect(assigns(:res)[:after]).to eq [[1, 5], [5, 625, 625]]
end

  it 'returns different result' do
    get :result, params: { str_arr: '10 1 6 5 3 5 9', length: '7' }
    expect(assigns(:res)[:after]).to eq [[1], [5], [5]]
  end

  it 'show in xml' do
    get :result, params: { str_arr: '5 5 6 0 1 4', length: '6', format:
:xml }
    expect(response.content_type).to eq('application/xml;
charset=utf-8')
  end

  it 'show in html' do
    get :result, params: { str_arr: '5 5 6 0 1 4', length: '6', format:
:html }
    expect(response.content_type).to eq('text/html; charset=utf-8')
  end
end
```

```
suggestExtensions: false
● ksusha@ksusha-notebook:~/uni/labs_web/lab10_serv$ rspec spec
.....

Finished in 0.11194 seconds (files took 1.52 seconds to load)
6 examples, 0 failures

○ ksusha@ksusha-notebook:~/uni/labs_web/lab10_serv$
```

Rubocop:

```
Inspecting 35 files
.....C....C.....C.....
Offenses:

app/helpers/application_helper.rb:3:1: C: Style/Documentation: Missing top-level documentation comment for module ApplicationHelper.
module ApplicationHelper
~~~~~
bin/bundle:24:3: C: Metrics/CyclomaticComplexity: Cyclomatic complexity for cli_arg_version is too high. [9/7]
  def cli_arg_version ...
  ~~~~~
bin/bundle:24:3: C: Metrics/MethodLength: Method has too many lines. [11/10]
  def cli_arg_version ...
  ~~~~~
bin/bundle:24:3: C: Metrics/PerceivedComplexity: Perceived complexity for cli_arg_version is too high. [9/8]
  def cli_arg_version ...
  ~~~~~
bin/bundle:91:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
  def activate_bundler ...
  ~~~~~
bin/bundle:104:121: C: Layout/LineLength: Line is too long. [198/120]
  warn "Activating bundler (#{bundler_requirement}) failed:\n#{gem_error.message}\n\nTo install the version of bundler this project requires, run
  `gem install bundler -v '#{bundler_requirement}'`"
  ~~~~~
config/application.rb:12:3: C: Style/Documentation: Missing top-level documentation comment for class Laba10Serv::Application.
class Application < Rails::Application
~~~~~

35 files inspected, 7 offenses detected
```

## Результат работы программы

Lab10Client

localhost:3001

## Adding length of sequence and sequence

Input sequence of positive integers separated by space

Length

Sequence

☐ XML ☐ HTML

Show

localhost:3001/elements/

localhost:3001/elements/result?length=7&str\_arr=1+2+2+4+7+5+5&mode=xm1&commit=Show

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

```
<?xml version="1.0"?>
<before-processing type="array">
  <before-processing type="integer">1</before-processing>
  <before-processing type="integer">2</before-processing>
  <before-processing type="integer">2</before-processing>
  <before-processing type="integer">4</before-processing>
  <before-processing type="integer">7</before-processing>
  <before-processing type="integer">5</before-processing>
  <before-processing type="integer">5</before-processing>
</before-processing>
<length type="integer">7</length>
<after type="array">
  <after type="integer">1</after>
  <after type="integer">2</after>
  <after type="integer">2</after>
  <after type="integer">4</after>
  <after type="integer">7</after>
  <after type="integer">5</after>
  <after type="integer">5</after>
</after>
<largest-segment type="array">
  <largest-segment type="integer">5</largest-segment>
  <largest-segment type="integer">5</largest-segment>
</largest-segment>
</hash>
```

localhost:3001/elements/

localhost:3001/elements/result?length=7&str\_arr=1+2+2+4+7+5+5&mode=html&commit=Show

## Result

1224755715555

## Вывод

В результате выполнения лабораторной работы №10 были освены навыки обработки XML. Ознакомлены с технологией XSLT.