



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

О Т Ч Е Т

по лабораторной работе № 10

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

Дисциплина: Языки Интернет-программирования

Студент

ИУ6-35Б

(Группа)

27.10.2023

(Подпись,
дата)

И.А. Буханцев

(И.О. Фамилия)

Преподаватель

Е.Ю. Гаврилова

(Подпись,
дата)

(И.О. Фамилия)

Москва, 2023

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

Цель – Получить практические навыки формирования данных в формате XML и их визуализации с помощью клиентских и серверных средств с использованием XSLT-преобразований.

twins-proxy/test xslt.rb

#frozen_string_literal: true

require 'nokogiri'

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

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

puts xslt.transform(doc)

twins-proxy/some_file.xml

<?xml version="1.0" encoding="ISO-8859-1"?>

<?xml-stylesheet type="text/xsl" href="some_transformer.xslt"?>

<catalog>

<cd>

<id>1</id>

<item>10</item>

</cd>

<cd>

<id>2</id>

<item>12</item>

</cd>

</catalog>

twins-proxy/public/some_transformer.xslt

<?xml version="1.0" encoding="ISO-8859-1"?>

<xsl:stylesheet

version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

```

<xsl:template match="/">
  <xsl:if test="output/input">
    <div><xsl:value-of select="output/input"/></div>
  </xsl:if>

  <table border="1">
    <tr bgcolor="#9933ff">
      <th>1</th>
      <th>2</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="id"/></td>
        <td><xsl:value-of select="item"/></td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
</xsl:stylesheet>

```

twins-proxy/app/views/proxy_view/view.xml.erb

```

<?xml version="1.0" encoding="UTF-8"?>
<output>
  <%= @output %>
</output>

```

twins-proxy/app/javascript/src/client_converter.js

```
function client_side_process(data) {  
    console.log('client_side_process', data);  
    const result = document.getElementById("result");  
    let str = "";  
  
    try {  
        str = new XMLSerializer().serializeToString(data.documentElement);  
    } catch (e) {  
        str = data;  
    }  
  
    result.innerHTML = "<hr/>Результат: " + str +  
        "<hr/><p id='date'>" + Date() + "</p>";  
}
```

// Сохраняем состояние приложения

```
function saveState(_state = null) {  
    let server_radio = $("input:radio[id=server_radio]:checked").val();  
    let n = document.getElementById("n").value;  
  
    let state = "";  
    if (!server_radio) {  
        state = '0';  
    } else {  
        state = '1'  
    }  
  
    if (_state) {  
        state = _state;  
    }  
}
```

```
localStorage.setItem('server_radio', state);
localStorage.setItem('input', n);

console.log('State saved', state, n)
}

// Получаем состояние приложения
function getState() {
    return localStorage.getItem('server_radio');
}

// Восстанавливаем состояние приложения
function restoreState() {
    setFormDataRemote();
    setCheckboxState();
    setInputVal();
}

// Устанавливаем параметр `data-remote` для формы
function setFormDataRemote() {
    let calc_form = $('#calc_form');
    let state = getState();
    console.log('data-remote before:', calc_form.attr('data-remote'));

    if (state === '1') {
        console.log('Radio server');
        $(calc_form).attr('data-remote', false);
    } else {
```

```
console.log('Radio client');  
$(calc_form).attr('data-remote', true);  
}
```

```
console.log('data-remote after:', calc_form.attr('data-remote'));  
}
```

// Устанавливаем состояние активного чекбокса

```
function setCheckboxState() {  
    let state = getState();  
  
    if (state === '1') {  
        $("#server_radio").attr('checked', true)  
    } else {  
        $("#client_radio").attr('checked', true)  
    }  
}
```

// Задаем значение поля ввода из локального хранилища

```
function setInputVal() {  
    document.getElementById("n").value = localStorage.getItem('input');  
}
```

// Сохраняем состояние приложения по-умолчанию

```
function setDefaultState(state='1') {  
    let localState = getState();  
  
    if (!localState) {  
        saveState(state); // устанавливаем чекбокс на сервер, если не стоит по  
умолчанию  
    }  
}
```



```
    }  
}
```

// Меняем action в зависимости от нажатой кнопки

```
$(document).on("click", 'input[id="xslt"]', function () {  
    $("#calc_form").attr('action', '/twins_proxy/view.html');  
});
```

```
$(document).on("click", 'input[id="xml"]', function () {  
    $("#calc_form").attr('action', '/twins_proxy/view.xml');  
});
```

```
$(document).ready(function () {  
    setDefaultState();  
    restoreState();  
  
    console.log('Bind');  
    $("#calc_form").bind("ajax:success",  
        function (xhr, data, status) {  
            console.log('ajax:success', $("#calc_form").attr('data-remote'))  
            // console.log('ajax:success', xhr, data, status);  
            client_side_process(data);  
        })  
    })
```

// Перезагружаем страницу в случае смена чекбокса для сброса кэша

```
$(document).on("change", 'input[type="radio"]', function () {  
    saveState();  
    setFormDataRemote();
```

```
// Костыль
location.reload();
});
```

twins-proxy/config/routes.rb

```
Rails.application.routes.draw do
  get 'twins_proxy/input'
  get 'twins_proxy/view'
  # Define your application routes per the DSL in
  https://guides.rubyonrails.org/routing.html

  root 'twins_proxy#input'
end
```

twins-proxy/app/views/proxy_view/input.html.erb

```
<h1>TwinsProxy#input</h1>

<div>
  <p>
    <a href="<%= url_for(only_path: false) + '.html.erb' %>">Мы находимся по
    адресу: <%= url_for(only_path: false) + '.html.erb' %></a>
  </p>

  <div>
    <input type="radio" id="server_radio" name="selector" value="1" />
    <label for="server_radio">Серверный обработчик</label>
```

```

    <input type="radio" id="client_radio" name="selector" value="2" />
    <label for="client_radio">Клиентский обработчки</label>
</div>

<br>

<form action="" method="get" id="calc_form" accept-charset="UTF-8" data-
remote="false">
  <label for="n">
    <input type="number" id="n" name="n" min="0" minlength="1"
oninput="this.value = Math.abs(this.value);" required />
  </label>
  <input id="xslt" name="commit" type="submit" value="XML+XSLT" />
  <input id="xml" name="commit" type="submit" value="XML" />
</form>

<div id="result"></div>
</div>

```

twins-proxy/app/views/proxy_view/view.html.erb

```

<h1>TwinsProxy#view</h1>

<div>
  <p>
    <a href="<%= url_for(only_path: false) + '.html.erb' %>">Мы находимся по
    адресу: <%= url_for(only_path: false) + '.html.erb' %></a>
  </p>

```

```
<%= @output.html_safe %>
<br/>
<%= link_to "Рассчитать заново", :twins_proxy_input %>
</div>
```

twins-proxy/app/controllers/twins_proxy_controller.rb

```
require 'net/http'
```

```
require 'nokogiri'
```

```
class TwinsProxyController < ApplicationController
```

```
  BASE_API_URL = 'http://127.0.0.1:3000/twins_api/view'.freeze # Путь до
  файла с возможностью преобразования
```

```
  XSLT_TRANSFORM = "#{Rails.root}/public/some_transformer.xslt".freeze #
  Путь до xslt файла
```

```
  def input; end
```

```
  def view
```

```
    print 'Params:', params, "\n"
```

```
    response = make_query BASE_API_URL, '.xml'
```

```
    respond_to do |format|
```

```
      # http://127.0.0.1:3001/twins_proxy/view.html?n=10
```

```
      format.html do
```

```
        print 'Render HTML ', params[:commit], "\n"
```

```

    @output = xslt_transform(response).to_html
end

# http://127.0.0.1:3001/twins_proxy/view.xml?n=10
format.xml do
    print 'Render XML ', params[:commit], "\n"

    @output = insert_browser_xslt(response).to_xml
end

# http://127.0.0.1:3001/twins_proxy/view.rss?n=10
format.rss { render xml: insert_browser_xslt(response).to_xml }
end
end

def make_query(server_url, file_type = "")
    # server_url - URL для получения ответа от приложения 1 (API)
    # file_type - тип файла, по умолчанию .html

    query_str = server_url.to_s + file_type
    query_str << "?n=#{ @input }" if (@input = params[:n]&.split(' ').join('+'))
    pp 'query_str:', query_str

    uri = URI(query_str)

    res = Net::HTTP.get_response(uri)

    if file_type != '.xml'
        # Форматируем html вывод

```

```
str1_markerstring = '<span>' # маркер начала xml
str2_markerstring = '</span>' # маркер конца xml
else
str1_markerstring = '<output>' # маркер начала xml
str2_markerstring = '</output>' # маркер конца xml
end
```

```
output = res.body[/#{str1_markerstring}(.*?)#{str2_markerstring}/m, 1]
output.gsub('&lt;', '<').gsub('&gt;', '>')
end
```

```
def xslt_transform(data, transform: XSLT_TRANSFORM)
# Функция преобразования
```

```
print data, transform, "\n"
```

```
doc = Nokogiri::XML(data)
xslt = Nokogiri::XSLT(File.read(transform))
xslt.transform(doc)
end
```

Чтобы преобразование XSLT на клиенте работало, надо вставить ссылку на XSLT.

Делается это с помощью nokogiri через ProcessingInstruction (потому что ссылка

на XSLT называется в XML processing instruction).

```
def insert_browser_xslt(data, transform: XSLT_TRANSFORM)
doc = Nokogiri::XML(data)
xslt = Nokogiri::XML::ProcessingInstruction.new(doc,
'xml-stylesheet',
```

```
      "type=\"text/xsl\" href=\"#{transform}\"")
    doc.root.add_previous_sibling(xslt)
    # Возвращаем doc, так как предыдущая операция возвращает не XML-
    документ.
    doc
  end
end
```

app/config/importmap.rb

Pin npm packages by running ./bin/importmap

```
pin 'application', preload: true
pin '@hotwired/turbo-rails', to: 'turbo.min.js', preload: true
pin '@hotwired/stimulus', to: 'stimulus.min.js', preload: true
pin '@hotwired/stimulus-loading', to: 'stimulus-loading.js', preload: true
pin_all_from 'app/javascript/controllers', under: 'controllers'

pin_all_from 'app/javascript/src', under: 'src'
pin 'jquery', to: 'jquery.min.js', preload: true
pin 'jquery_ujs', to: 'jquery_ujs.js', preload: true
```

app/config/initializers/assets.rb

Be sure to restart your server when you modify this file.

Version of your assets, change this if you want to expire all your assets.

```
Rails.application.config.assets.version = "1.0"
```

```
Rails.application.config.assets.precompile += %w( jquery.min.js jquery_ujs.js )
```

```
# Add additional assets to the asset load path.  
# Rails.application.config.assets.paths << Emoji.images_path  
  
# Precompile additional assets.  
# application.js, application.css, and all non-JS/CSS in the app/assets  
# folder are already added.  
# Rails.application.config.assets.precompile += %w( admin.js admin.css )
```

twins-proxy/test/controllers/twins_proxy_controller_test.rb

```
require 'test_helper'
```

```
require 'net/http'
```

```
class TwinsProxyControllerTest < ActionDispatch::IntegrationTest
```

```
  BASE_API_URL = 'http://127.0.0.1:3000/twins_api/view'.freeze # Путь до  
файла с возможностью преобразования
```

```
  test 'should get input' do  
    get twins_proxy_input_url  
    assert_response :success  
  end
```

```
  test 'should get view' do  
    get twins_proxy_view_url  
    assert_response :success  
  end
```

```
  test 'check different result' do
```



```
target = "<table border=\"1\">\n<tr  
bgcolor=\"#9933ff\">\n<th>1</th>\n<th>2</th>\n</tr>\n<tr>\n<td>3</td>\n<td>5  
</td>\n</tr>\n</table>\n"
```

assert_equal result, target

end

test 'check xml proxy result' **do**

get "#{twins_proxy_view_url}.xml", params: { n: 3 }

target = "<?xml version=\"1.0\"?>\n<?xml-stylesheet type=\"text/xsl\" href=\"C:/Users/neizvestnyj/Desktop/BMSTU/Semestr_3/IPL/LW/LW10/Project/twins-proxy/public/some_transformer.xslt\"?>\n<catalog>\n <cd>\n <id>3</id>\n <item>5</item>\n </cd>\n</catalog>\n"

assert_equal assigns[:**output**], target

end

test 'check rss' **do**

get "#{twins_proxy_view_url}.rss", params: { n: 3 }

target = "<?xml version=\"1.0\"?>\n<?xml-stylesheet type=\"text/xsl\" href=\"C:/Users/neizvestnyj/Desktop/BMSTU/Semestr_3/IPL/LW/LW10/Project/twins-proxy/public/some_transformer.xslt\"?>\n<catalog>\n <cd>\n <id>3</id>\n <item>5</item>\n </cd>\n</catalog>\n"

assert_equal @**response.body.clone**, target

end

test 'should get different responds for different rss requests' **do**

get "#{twins_proxy_view_url}.rss", params: { n: 3 }

response1 = @**response.body.clone**

get "#{twins_proxy_view_url}.rss", params: { n: 10 }

response2 = @**response.body.clone**

```
    assert_not_equal response1, response2
  end
end
```

twins-api/app/views/twins_api/view.xml.erb

```
<?xml version="1.0" encoding="UTF-8"?>
<output>
  <%= @result[1] %>
</output>
```

twins-api/app/controllers/twins_api_controller.rb

```
class TwinsApiController < ApplicationController

  def view
    n = params[:n].to_i

    # Возвращаем результат
    @result = twins(n)
  end

  def is_prime(num)
    return false if num <= 1

    Math.sqrt(num).to_i.downto(2).each { |i| return false if (num % i).zero? }
    true
  end
```

```

def twins(n)
  pairs = {}
  (n..2 * n).each do |first|
    (first..2 * n).each do |second|
      if (second - first == 2) && is_prime(first) && is_prime(second)
        pairs[first] = second
      end
    end
  end
end

[pairs, to_table(pairs)]
end

def to_table(pairs = @result)
  # Создание xml таблицы

  @table = 'Unknown!'

  unless pairs.empty?
    rows = ""
    pairs.each do |key, value|
      rows += "<cd><id>#{key}</id><item>#{value}</item></cd>"
    end
    @table = "<catalog>#{rows}</catalog>"
  end

  puts @table
  @table
end
end

```



Рисунок 1 – страница ввода



Рисунок 2 – результат работы сервера + xslt программы

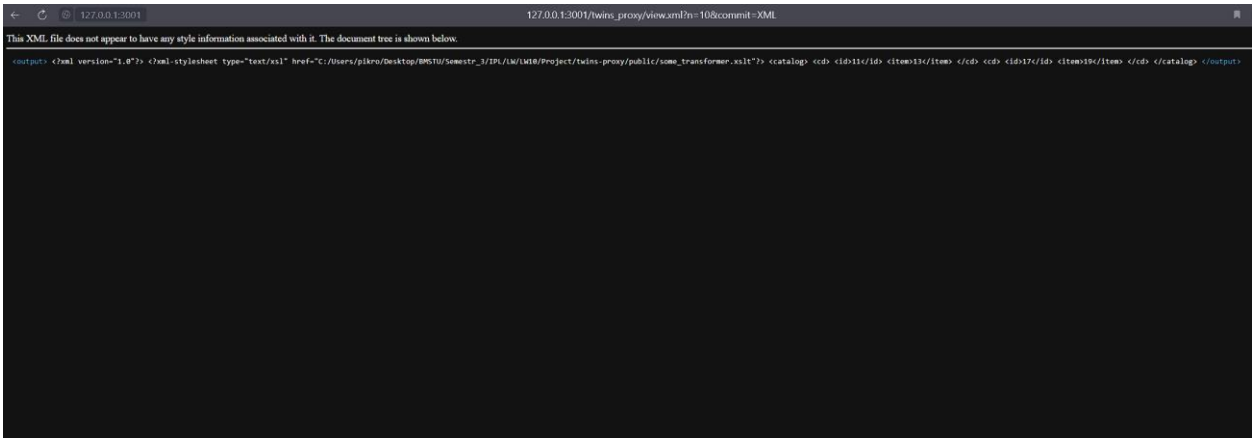


Рисунок 3 – вывод сервером xml

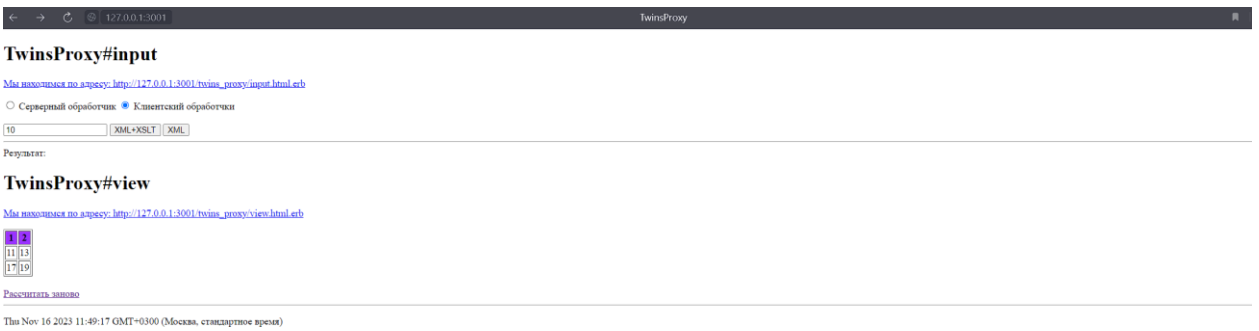


Рисунок 5 – клиентский обработчик с xslt

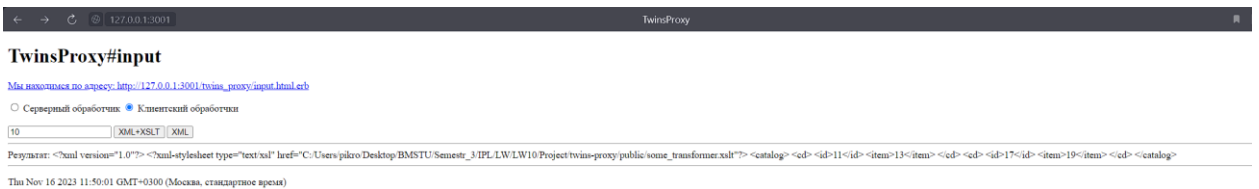


Рисунок 6 – вывод xml

```

Running 8 tests in a single process (parallelization threshold is 50)
Run options: --seed 14339

# Running:

Params:{"n"=>"3", "controller"=>"twins_proxy", "action"=>"view"}
"query_str:"
"http://127.0.0.1:3000/twins_api/view.xml?n=3"
Render HTML

    <catalog><cd><id>3</id><item>5</item></cd></catalog>
C:/Users/pikro/Desktop/BMSTU/Semestr_3/IPL/LW/LW10/Project/twins-proxy/public/some_transformer.xslt
..Params:{"n"=>"3", "controller"=>"twins_proxy", "action"=>"view", "format"=>"xml"}
"query_str:"
"http://127.0.0.1:3000/twins_api/view.xml?n=3"
Render XML
.Params:{"n"=>"3", "controller"=>"twins_proxy", "action"=>"view", "format"=>"rss"}
"query_str:"
"http://127.0.0.1:3000/twins_api/view.xml?n=3"
Params:{"n"=>"10", "controller"=>"twins_proxy", "action"=>"view", "format"=>"rss"}
"query_str:"
"http://127.0.0.1:3000/twins_api/view.xml?n=10"
.Params:{"controller"=>"twins_proxy", "action"=>"view"}
"query_str:"
"http://127.0.0.1:3000/twins_api/view.xml"
Render HTML

    Unknown!
C:/Users/pikro/Desktop/BMSTU/Semestr_3/IPL/LW/LW10/Project/twins-proxy/public/some_transformer.xslt
.

Finished in 1.352132s, 5.9166 runs/s, 5.9166 assertions/s.
8 runs, 8 assertions, 0 failures, 0 errors, 0 skips

```

Рисунок 7 – тесты

Вывод – Получили практические навыки формирования данных в формате XML и их визуализации с помощью клиентских и серверных средств с использованием XSLT-преобразований.