



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: 09.03.01 Информатика и Вычислительная техника

**О т ч е т**  
**по лабораторной работе 10**

**Дисциплина: языки интернет-программирования**

**Вариант №15**

Студент гр. ИУ6-33Б

\_\_\_\_\_

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

## Цель

Ознакомиться с технологией xslt. Написать приложение, реализующее различные способы асинхронной доставки данных до клиента от сервера.

## 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

Результаты должны быть представлены в виде двух файлов:

отчет в формате pdf/odt/doc;

архив rails-приложения (zip, tgz, 7z).

## Выполнение

### Backend

Compute\_controller.rb:

```
# frozen_string_literal: true

class ComputeController < ApplicationController
  def input; end

  def is_palindrome(x)
    x = x.to_s
    x == x.reverse
  end

  def view
    n = params[:n].to_i
    @result = (0..n)
      .select { |i| is_palindrome(i * i) }
      .map { |i| { "n": i, "nn": i * i } }

    render xml: @result
  end
end
```

routes.rb:

```
# frozen_string_literal: true

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

  # Reveal health status on /up that returns 200 if the app boots with no exceptions,
  # otherwise 500.
  # Can be used by load balancers and uptime monitors to verify that the app is live.
  get 'up' => 'rails/health#show', as: :rails_health_check

  # Defines the root path route ("/")
  root 'main#index'
```

```
end
```

## Proxy

main\_controller.rb:

```
# frozen_string_literal: true

require 'net/http'
require 'nokogiri'

class MainController < ApplicationController
  include ReverseProxy::Controller

  def initialize
    @xslt = Nokogiri::XSLT(File.read('transformer.xslt'))
  end

  def index
    reverse_proxy('http://localhost:3001/') do |config|
      config.on_success do |_code, response|
        if request.format == 'application/xml'
          response
        else
          res = @xslt.transform(Nokogiri::XML(response.body))
          return render html: res
        end
      end
    end
  end
end
```

transformer.xslt:

```
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <pre>
    Result:
    +-----+-----+
    |           n |           n*n |
    +-----+-----+<xsl:for-each select="objects/object">
```

```

<div class="row">| <div class="column"><xsl:value-of select="n"/></div> | <div
class="column"><xsl:value-of select="nn"/></div> |</div></div>+-----+--
-----+
</pre>
</xsl:template>

</xsl:transform>

```

Index.html:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
  </head>
  <style>
    body {
      font-family: monospace;
    }

    .row {
      display: flex;
    }

    .column {
      width: 118px;
      text-align: right;
    }
  </style>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/x2js/1.2.0/xml2json.js"
integrity="sha512-
JW4yY5Dei/4V3FV0BUZwEhdIrhv5jaUQyABWn9YVdXhOiG4++4Bs6sJAgZFSaMBmVKytwPDOaXic7Qs6yqSMw
g==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="./main.js"></script>
  <body>
    <h3>Lab 10 proxy</h3>
    <div class="form">
      <span>Введите N: </span>
      <input id="input-n">
      <button onclick="make_computation_on_server()">Вычислить на
сервере</button>

```

```

        <button onclick="make_computation_on_client()">Вычислить на
клиенте</button>
    </div>
    <div id="result">
    </div>
</body>
</html>

```

Main.js:

```

const x2js = new X2JS()

const normalize = (str) => str.replaceAll('&lt;', '<').replaceAll('&gt;', '>').replaceAll('&quot;', '"')

const insert_number = (n) => {
    let s = n.toString()
    s += ' '.repeat(14-s.length);
    return s;
};

const make_computation_on_client = async () => {
    const n = document.getElementById("input-n").value;
    const url = "http://127.0.0.1:3000/compute/view.xml?n=" + n
    const response = await fetch(url);
    const result = x2js.xml_str2json(normalize(await response.text()))
    console.log(result.objects.object)
    document.getElementById("result").innerHTML = `
<pre>
Result:
+-----+-----+
|      n      |      n*n      |
+-----+-----+
${result.objects.object.map((x) => {
    return `| \t\t | \t\t |
| ${insert_number(x.n.__text)} | ${insert_number(x.nn.__text)} |`;
}).join('\n')}
+-----+-----+
</pre>`
};

const make_computation_on_server = async () => {

```

```

    const n = document.getElementById("input-n").value;
    const url = "http://127.0.0.1:3000/compute/view?n=" + n
    const response = await fetch(url);
    const res = normalize(await response.text())
    document.getElementById("result").innerHTML = res
  };

```

Routes.rb:

```

# frozen_string_literal: true

Rails.application.routes.draw do
  match '/compute/view' => 'main#index', via: %i[get post]
  # Define your application routes per the DSL in
https://guides.rubyonrails.org/routing.html

  # Reveal health status on /up that returns 200 if the app boots with no exceptions,
  otherwise 500.
  # Can be used by load balancers and uptime monitors to verify that the app is live.
  get 'up' => 'rails/health#show', as: :rails_health_check

  # Defines the root path route ("/")
  root 'main#index'
end

```

rubocop:

```

@vzalygin → /workspaces/ipl.labs/lab10 (master) $ rubocop -a
Inspecting 76 files
.....CCCCC.CCC.....C..C.....C.CCC.....WCCC.CCC.....C..C.....C.CC

Offenses:

backend/app/controllers/compute_controller.rb:3:1: C: Style/Documentation: Missing
top-level documentation comment for class ComputeController.
class ComputeController < ApplicationController
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
backend/app/controllers/compute_controller.rb:6:7: C: Naming/PredicateName: Rename
is_palindrome to palindrome?.
  def is_palindrome(x)
    ^^^^^^^^^^^^^^^^^

```



```

backend/app/controllers/compute_controller.rb:6:21: C: Naming/MethodParameterName:
Method parameter must be at least 3 characters long.
  def is_palindrome(x)
                        ^

backend/app/controllers/main_controller.rb:3:1: C: Style/Documentation: Missing top-
level documentation comment for class MainController.
class MainController < ApplicationController
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

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

backend/app/helpers/compute_helper.rb:3:1: C: Style/Documentation: Missing top-level
documentation comment for module ComputeHelper.
module ComputeHelper
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

backend/app/helpers/main_helper.rb:3:1: C: Style/Documentation: Missing top-level
documentation comment for module MainHelper.
module MainHelper
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

backend/app/mailers/application_mailer.rb:3:1: C: Style/Documentation: Missing top-
level documentation comment for class ApplicationMailer.
class ApplicationMailer < ActionMailer::Base
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

backend/app/models/application_record.rb:3:1: C: Style/Documentation: Missing top-
level documentation comment for class ApplicationRecord.
class ApplicationRecord < ActiveRecord::Base
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

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

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

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

backend/bin/bundle:86:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
  def activate_bundler ...

```

```

#####
backend/bin/bundle:99:121: 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}'`"

#####
#####
backend/config/application.rb:12:3: C: Style/Documentation: Missing top-level
documentation comment for class Lab8::Application.
  class Application < Rails::Application
  #####
backend/config/environments/development.rb:5:1: C: Metrics/BlockLength: Block has too
many lines. [32/25]
Rails.application.configure do ...
#####
backend/test/application_system_test_case.rb:5:1: C: Style/Documentation: Missing
top-level documentation comment for class ApplicationSystemTestCase.
class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
#####
backend/test/controllers/compute_controller_test.rb:5:1: C: Style/Documentation:
Missing top-level documentation comment for class ComputeControllerTest.
class ComputeControllerTest < ActionDispatch::IntegrationTest
#####
backend/test/controllers/main_controller_test.rb:5:1: C: Style/Documentation: Missing
top-level documentation comment for class MainControllerTest.
class MainControllerTest < ActionDispatch::IntegrationTest
#####
backend/test/test_helper.rb:8:3: C: Style/Documentation: Missing top-level
documentation comment for class ActiveSupport::TestCase.
  class TestCase
  #####
proxy/app/controllers/main_controller.rb:6:1: C: Style/Documentation: Missing top-
level documentation comment for class MainController.
class MainController < ApplicationController
#####
proxy/app/controllers/main_controller.rb:9:3: W: Lint/MissingSuper: Call super to
initialize state of the parent class.
  def initialize ...
  #####

```

```

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

proxy/app/helpers/main_helper.rb:3:1: C: Style/Documentation: Missing top-level
documentation comment for module MainHelper.
module MainHelper
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

proxy/app/helpers/proxy_helper.rb:3:1: C: Style/Documentation: Missing top-level
documentation comment for module ProxyHelper.
module ProxyHelper
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

proxy/app/mailers/application_mailer.rb:3:1: C: Style/Documentation: Missing top-
level documentation comment for class ApplicationMailer.
class ApplicationMailer < ActionMailer::Base
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

proxy/app/models/application_record.rb:3:1: C: Style/Documentation: Missing top-level
documentation comment for class ApplicationRecord.
class ApplicationRecord < ActiveRecord::Base
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

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

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

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

proxy/bin/bundle:86:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
  def activate_bundler ...
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

proxy/bin/bundle:99:121: 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}'`"

```

```

proxy/config/application.rb:12:3: C: Style/Documentation: Missing top-level
documentation comment for class Proxy::Application.
  class Application < Rails::Application
    ^^^^^^^^^^^^^^^^^^^
proxy/config/environments/development.rb:5:1: C: Metrics/BlockLength: Block has too
many lines. [26/25]
Rails.application.configure do ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
proxy/test/application_system_test_case.rb:5:1: C: Style/Documentation: Missing top-
level documentation comment for class ApplicationSystemTestCase.
class ApplicationSystemTestCase < ActionDispatch::SystemTestCase
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
proxy/test/controllers/main_controller_test.rb:5:1: C: Style/Documentation: Missing
top-level documentation comment for class MainControllerTest.
class MainControllerTest < ActionDispatch::IntegrationTest
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
proxy/test/test_helper.rb:8:3: C: Style/Documentation: Missing top-level
documentation comment for class ActiveSupport::TestCase.
  class TestCase
    ^^^^^^^^^^^^^^^^^
76 files inspected, 37 offenses detected

```

## Вывод

Проведено ознакомление с технологией xslt. Написано приложение, реализующее разные способы доставки данных до клиента.