**Сопроводительный лист на издание в авторской редакции**

Название работы \_\_\_ RIA – Создание FrontEnd составляющей для RESTfull API. \_ Методические указания к выполнению лабораторной работы № 2 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 – «Информатика и вычислительная техника» 02.03.03 – «Математическое обеспечение и администрирование информационных систем»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Актуальность и соответствующий   
научно-методический уровень подтверждаю \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись научного редактора)

Рукопись сверена и проверена автором \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись автора)

Рекомендуется к изданию \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись заведующего кафедрой)



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

Утверждаю

Ректор университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.Н.Федонин

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

**Специальные разделы web программирования**

**RIA – Создание FrontEnd составляющей для RESTfull API**

**Методические указания**

**к выполнению лабораторной работы №2**

**для студентов очной формы обучения**

**по направлениям подготовки**

**09.03.04 – «Программная инженерия»**

**09.03.01 – «Информатика и вычислительная техника»**

**02.03.03 – «Математическое обеспечение и администрирование информационных систем»**

**Брянск 2019**

**УКД 004.**

RIA – Создание FrontEnd составляющей для RESTfull API. [Текст] + [Электронный ресурс]: Методические указания к лабораторной работе №6 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 – «Информатика и вычислительная техника» 02.03.03 – «Математическое обеспечение и администрирование информационных систем». – Брянск: БГТУ, 2017. – 18 с.

Разработали:

Д.Г.Лагерев

канд. техн. наук, доц.,

Д.А.Касницкий

ассистент

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ

**Методические издания публикуются в авторской редакции**

# Цель работы

Целью выполнения данной работы является изучение основ создания frontend составляющей для RESTfull API и практика работы с REST API сторонних веб-сервисов.

Продолжительность работы – 2 часа.

# REST

REST – архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы.

**Свойства** архитектуры, которые зависят от ограничений, наложенных на REST-системы:

Производительность — взаимодействие компонентов системы может являться доминирующим фактором производительности и эффективности сети с точки зрения пользователя

Масштабируемость для обеспечения большого числа компонентов и взаимодействий компонентов.

**Требования** к архитектуре REST

1. Модель клиент-сервер

2. Отсутствие состояния

3. Кэширование

4. Единообразие интерфейса

5. Слои

6. Код по требованию (необязательное ограничение)

К унифицированным интерфейсам предъявляются следующие четыре ограничительных условия:

1. Идентификация ресурсов.

Все ресурсы идентифицируются в запросах, например, с использованием URI в интернет-системах. Ресурсы концептуально отделены от представлений, которые возвращаются клиентам. Например, сервер может отсылать данные из базы данных в виде HTML, XML или JSON, ни один из которых не является типом хранения внутри сервера.

1. Манипуляция ресурсами через представление.

Если клиент хранит представление ресурса, включая метаданные — он обладает достаточной информацией для модификации или удаления ресурса.

1. «Самоописываемые» сообщения.

Каждое сообщение содержит достаточно информации, чтобы понять каким образом его обрабатывать. К примеру, обработчик сообщения (parser) необходимый для извлечения данных может быть указан в списке MIME-типов.

1. Гипермедиа, как средство изменения состояния приложения (HATEOAS)

Клиенты изменяют состояние системы только через действия, которые динамически определены в гипермедиа на сервере (к примеру, гиперссылки в гипертексте). Исключая простые точки входа в приложение, клиент не может предположить, что доступна какая-то операция над каким-то ресурсом, если не получил информацию об этом в предыдущих запросах к серверу. Не существует универсального формата для предоставления ссылок между ресурсами, RFC 5988 и JSON Hypermedia API Language являются двумя популярными форматами предоставления ссылок в REST HYPERMEDIA сервисах.

Рассмотрим подробнее типы XML и JSON, используемые для передачи данных в системах, построенных по архитектуре REST.

# Форматы JSON и XML

JSON (JavaScript Object Notation – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Формат JSON был разработан Дугласом Крокфордом.

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

* Набор пар ключ: значение. В различных языках это реализовано как объект, запись, структура, словарь, хэш-таблица, список с ключом или ассоциативный массив. Ключом может быть только строка (регистрозависимая: имена с буквами в разных регистрах считаются разными) значением — любая форма.
* Упорядоченный набор значений. Во многих языках это реализовано как массив, вектор, список или последовательность.

В качестве значений в JSON могут быть использованы:

* Объект — это неупорядоченное множество пар ключ:значение, заключённое в фигурные скобки «{ }». Ключ описывается строкой, между ним и значением стоит символ «:». Пары ключ-значение отделяются друг от друга запятыми.
* Массив (одномерный) — это упорядоченное множество значений. Массив заключается в квадратные скобки «[ ]». Значения разделяются запятыми.
* Число.
* Литералы true, false и null.
* Строка — это упорядоченное множество из нуля или более символов юникода, заключённое в двойные кавычки. Символы могут быть указаны с использованием escape-последовательностей, начинающихся с обратной косой черты «\» (поддерживаются варианты \", \\, \/, \t, \n, \r, \f и \b), или записаны шестнадцатеричным кодом в кодировке Unicode в виде \uFFFF.

Следующий пример показывает JSON-представление объекта, описывающего человека. В объекте есть строковые поля имени и фамилии, объект, описывающий адрес, и массив, содержащий список телефонов. Как видно из примера, значение может представлять собой вложенную структуру.

{

"firstName": "Иван",

"lastName": "Иванов",

"address": {

"streetAddress": "ул. Институтская, д. 13",

"city": "Брянск",

"postalCode": 123456

},

"phoneNumbers": [

"8-800-555-35-35"

]

}

**XML** (eXtensible Markup Language) – расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы могут входить в состав других элементов.

Несколько основных правил создания верно сформированного документа:

* Каждый элемент XML должен содержать начальный и конечный тэг (либо пустой тэг типа <TAG />, который может нести информацию посредством своих атрибутов).
* Любой вложенный элемент должен быть полностью определён внутри элемента, в состав которого он входит.
* Документ должен иметь только один элемент верхнего уровня.
* Имена элементов чувствительны к регистру.

Пример XML-файла:

**<?xml version="1.0"?>**  
**<!-- Comment -->**  
**<PRODUCTS>**  
**<PRODUCT>**  
**<TITLE> Product #1 </TITLE>**  
**<PRICE> 10.00 </PRICE>**  
**</PRODUCT>**  
**<PRODUCT>**  
**<TITLE> Product #2 </TITLE>**  
**<PRICE> 20.00 </PRICE>**  
**</PRODUCT>**  
**</PRODUCTS>**

Типовой код для GET-запроса при помощи XMLHttpRequest:

var xhr = new XMLHttpRequest();

xhr.open('GET', '/my/url', true);

xhr.send();

xhr.onreadystatechange = function() {

if (this.readyState != 4) return;

// по окончании запроса доступны:

// status, statusText

// responseText, responseXML (при content-type: text/xml)

if (this.status != 200) {

// обработать ошибку

alert( 'ошибка: ' + (this.status ? this.statusText : 'запрос не удался') );

return;

}

// получить результат из this.responseText или this.responseXML

}

# Индивидуальные задания

## Общие комментарии:

Во всех вариантах загрузка данных должна производиться не при загрузке страницы, а при определенном действии пользователя. (ввод, нажатие кнопки «узнать», «получить», «сгенерировать» и т.д. При этом данные должны обновляться без перезагрузки страницы.

P.S. Если кто-то для ввода данных будет использовать <form> помните о его свойстве перезагружать страницы при клике на кнопку в его составе, перехватывайте действия браузера по умолчанию.

## Требования:

* Явное использование HTTP-методов.
* Передача данных в XML, JavaScript Object Notation (JSON) или в обоих форматах.
* Явное использование AJAX
* Отсутствие JQuery.

## Приветствуется:

* Разработка более сложного интерфейса, нежели простая таблица.
* Проработка вопросов кэширования, сохранения данных в браузере и т.д.
* Подробное описание вариантов приводится ниже.
* Теорию работы с запросами можно почитать здесь: https://learn.javascript.ru/ajax-xmlhttprequest

## Подробный разбор вариантов

1. Reqres. Работа со списком юзеров.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка пользоваталей
* Запрос с сервера reqres.in на выгрузку всех пользователей
* Отображение пользователей в интерфейсе
* Регистрация нового пользователя
* Авторизация нового пользователя
* Ссылка <https://reqres.in/>

1. Typicode. Работа с TODO list

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка TODO
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех пунктов списка
* Отображение списка в интерфейсе
* Фильтр по completed (выполнено, не выполнено, все)
* Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа со списком пользователей.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка пользоваталей
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех пользователей
* Отображение пользователей в интерфейсе
* Сортировка таблицы пользователей по параметрам, выбранных студентом
* Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа с альбомами

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения альбомов + превью (первая фотография)
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех альбомов с превью
* Отображение альбомов в интерфейсе
* Сортировка результата по параметрам, выбранных студентом
* Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа с альбомами

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения альбомов + превью (четыре последние фотографии)
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех альбомов с превью
* Отображение альбомов в интерфейсе
* Сортировка результата по параметрам, выбранных студентом
* Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа с отображением постов.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения постов и дерева комментариев к ним
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех постов + комментариев
* Отображение постов в интерфейсе
* Сортировка постов и комментариев по параметрам, выбранных студентом
* Ссылка <http://jsonplaceholder.typicode.com/>

1. Openweathermap. Выгрузка по ID города

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде в нескольких городах.
* Реализация нескольких отдельных запросов с ID выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Openweathermap. Выгрузка по координатам

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде в нескольких городах.
* Реализация нескольких отдельных запросов с координатами выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Openweathermap. Получение погоды в случайной точке Евразии

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде.
* Генерация координат, входящих в густонаселенные части Евразии (например, центр Европы).
* Реализация одного или нескольких (в случаи возвращения ошибки) запросов с полученными координатами. Учесть ограничение запросов к серверу.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Openweathermap. Выгрузка нескольких городов по ID

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде.
* Реализация общего запроса с ID выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Yandex Static Api и Openweathermap. Карта погоды по прямоугольному участку

Разработать приложение, которое будет выполнять следующие функции:

* Отображение карты с выбранным участком.
* Реализация запроса с координатами прямоугольника.
* Отображение полученных данных в интерфейсе (таблице).
* Ссылки:
  + <https://openweathermap.org/> <https://yandex.ru/dev/maps/staticapi/doc/1.x/dg/concepts/input_params.html/>

1. Yandex Static Api и Openweather. Загрузка карты с центром в произвольном городе и актуальной температурой (указанной на карте).

Разработать приложение, которое будет выполнять следующие функции:

* Отображение карты с выбранным участком.
* Реализация запроса погоды по конкретному городу.
* Отображение полученных данных на карте произвольным образом.
* Ссылки:
  + <https://openweathermap.org/>
  + <https://yandex.ru/dev/maps/staticapi/doc/1.x/dg/concepts/input_params.html/>

1. Yandex Static Api и Openweather. Загрузка карты с центром в Москве и актуальной температурой (указанной с помощью цвета метки).

Разработать приложение, которое будет выполнять следующие функции:

* Отображение карты с выбранным участком.
* Реализация запроса погоды по конкретному городу.
* Отображение полученных данных на карте с помощью метки (красная – тепло, синяя – холодно, например).
* Ссылки:
  + <https://openweathermap.org/>
  + <https://yandex.ru/dev/maps/staticapi/doc/1.x/dg/concepts/input_params.html/>

1. Openweathermap. Выгрузка по ID городу

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде в нескольких городах.
* Реализация нескольких отдельных запросов с ID выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Yandex Static Api и Openweathermap. Карта погоды по прямоугольному участку

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде.
* Отображение карты с выбранным участком.
* Реализация запроса с координатами прямоугольника.
* Отображение полученных данных в интерфейсе (таблице).

Ссылка: https://openweathermap.org/

* 1. <https://yandex.ru/dev/maps/staticapi/doc/1.x/dg/concepts/input_params.html/>

1. Reqres. Работа со списком юзеров.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка пользоваталей
* Запрос с сервера reqres.in на выгрузку всех пользователей
* Отображение пользователей в интерфейсе
* Регистрация нового пользователя
* Авторизация нового пользователя

Ссылка <https://reqres.in/>

1. Typicode. Работа с TODO list

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка TODO
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех пунктов списка
* Отображение списка в интерфейсе
* Фильтр по completed (выполнено, не выполнено, все)

Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа со списком пользователей.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения списка пользоваталей
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех пользователей
* Отображение пользователей в интерфейсе
* Сортировка таблицы пользователей по параметрам, выбранных студентом

Ссылка http://jsonplaceholder.typicode.com/

1. Typicode. Работа с альбомами

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения альбомов + превью (первая фотография)
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех альбомов с превью
* Отображение альбомов в интерфейсе
* Сортировка результата по параметрам, выбранных студентом

Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа с альбомами

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения альбомов + превью (четыре последние фотографии)
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех альбомов с превью
* Отображение альбомов в интерфейсе
* Сортировка результата по параметрам, выбранных студентом

Ссылка <http://jsonplaceholder.typicode.com/>

1. Typicode. Работа с отображением постов.

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения постов и дерева комментариев к ним
* Запрос с сервера jsonplaceholder.typicode.com на выгрузку всех постов + комментариев
* Отображение постов в интерфейсе
* Сортировка постов и комментариев по параметрам, выбранных студентом

Ссылка <http://jsonplaceholder.typicode.com/>

1. Openweathermap. Выгрузка по ID города

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде в нескольких городах.
* Реализация нескольких отдельных запросов с ID выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Openweathermap. Выгрузка по координатам

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде в нескольких городах.
* Реализация нескольких отдельных запросов с координатами выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: https://openweathermap.org/

1. Openweathermap. Получение погоды в случайной точке Евразии

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде.
* Генерация координат, входящих в густонаселенные части Евразии (например, центр Европы).
* Реализация одного или нескольких (в случаи возвращения ошибки) запросов с полученными координатами. Учесть ограничение запросов к серверу.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Openweathermap. Выгрузка нескольких городов по ID

Разработать приложение, которое будет выполнять следующие функции:

* Создание интерфейса для корректного отображения данных о погоде.
* Реализация общего запроса с ID выбранных городов.
* Отображение полученных данных в интерфейсе.
* Ссылка: <https://openweathermap.org/>

1. Yandex Static Api и Openweathermap. Карта погоды по прямоугольному участку

Разработать приложение, которое будет выполнять следующие функции:

* Отображение карты с выбранным участком.
* Реализация запроса с координатами прямоугольника.
* Отображение полученных данных в интерфейсе (таблице).
* Ссылки:
  + <https://openweathermap.org/>
  + <https://tech.yandex.ru/maps/doc/staticapi/1.x/dg/concepts/input_params-docpage/>

# Контрольные вопросы

1. Что такое XML?
2. Какие проблемы решает XML по сравнению с классическим HTML?Какие кнопки управления исполнением кода вам известны?
3. Перечислите основные правила валидности XML
4. Что такое JSON? Что представляют из себя данные в формате JSON?

# Список рекомендуемой литературы

1. Браун Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. – Мск, Альфа-книга, 2017 - 363с.
2. Современный учебник Javascript. –, 2017.- Режим доступа: <https://learn.javascript.ru>
3. Н. Л. Подколодный, А. В. Семенычев, Д. А. Рассказов, В. Г. Боровский, Е. А. Ананько, Е. В. Игнатьева, Н. Н. Подколодная, О. А. Подколодная, Н. А. Колчанов Распределённая система RESTful-web-сервисов для реконструкции и анализа генных сетей. Вавиловский журнал генетики и селекции, т. 16, N 4/1, 2012

RIA – Создание FrontEnd составляющей для RESTfull API. [Текст] + [Электронный ресурс]: Методические указания к лабораторной работе №6 для студентов очной формы обучения по направлениям подготовки 09.03.04 – «Программная инженерия», 09.03.01 – «Информатика и вычислительная техника» 02.03.03 – «Математическое обеспечение и администрирование информационных систем». – Брянск: БГТУ, 2017. – 18 с.

ЛАГЕРЕВ ДМИТРИЙ ГРИГОРЬЕВИЧ

КАСНИЦКИЙ ДМИТРИЙ АЛЕКСЕЕВИЧ

Научный редактор Д.И. Булатицкий

Компьютерный набор Д.А. Касницкий

Иллюстрации Д.А. Касницкий

Подписано в печать \_\_\_\_\_\_\_\_\_ г. Формат 60х84 1/16 Бумага офсетная.

Офсетная печать. Усл.печ.л. \_\_\_ Уч.-изд.л. \_\_\_ Тираж 1 экз. Бесплатно

Брянский государственный технический университет

Кафедра «Информатика и программное обеспечение», тел. 56-09-84

241035, Брянск, бульвар 50 лет Октября, 7 БГТУ