МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

Отчет   
по лабораторной работе №5  
на тему: «Проектирование SOA-архитектуры приложения»  
по дисциплине «Проектирование и архитектура программных систем»

Выполнили: Марочкин М.А. Шифр: 170584   
 Шорин В.Д. Шифр: 171406  
 Щекотихин С.Е. Шифр: 170590  
ИПАИТ  
Направление: 09.03.04 «Программная инженерия»  
Группа: 71-ПГ  
Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата «\_\_\_\_» \_\_\_\_\_\_ 2019 г.

Орел, 2019 г.

**Задание на лабораторную работу.**

Разработать клиентское приложение, взаимодействующее с сервисом сети интернет на основе SOAP-протокола. В качестве сервера можно использовать любой сервис сети интернет, реализующий протокол SOAP для использования его функциональности (сервисы РЖД, Аэрофлота, сервисы погоды, банковские сервисы и т.п.).

**Контрольные вопросы.**

*1. Опишите принципы построения сервис-ориентированной архитектуры.*

Основными принципами архитектурного стиля SOA являются:

1) Сервисы автономны. Обслуживание, разработка, развертывание и контроль версий каждого сервиса происходит независимо от других.

2) Сервисы могут быть распределены. Сервисы могут размещаться в любом месте сети, локально или удаленно, если сеть поддерживает необходимые протоколы связи.

3) Сервисы слабо связаны. Каждый сервис совершенно не зависит от остальных и может быть заменен или обновлен без влияния на приложения, его использующие, при условии предоставления совместимого интерфейса.

4) Сервисы совместно используют схему и контракт, но не класс. При обмене данными сервисы совместно используют контракты и схемы, но не внутренние классы.

5) Совместимость основана на политике. Политика, в данном случае, означает описание характеристик, таких как транспорт, протокол и безопасность.

*2. Опишите структуру протокола SOAP для взаимодействия с web-сервисами.*

SOAP — это стандарт для отсылки и получения сообщений по Internet. Спецификация SOAP определяет XML-«конверт» для передачи сообщений, метод для кодирования программных структур данных в формате XML, а также средства связи по протоколу HTTP.

SOAP-сообщения бывают двух типов: запрос (Request) и ответ (Response). Запрос вызывает метод удаленного объекта, ответ возвращает

результат выполнения данного метода. Спецификация SOAP определяет формат кодирования, который, в свою очередь, задает способ представления данных в XML-формате.

*3. Опишите основные роли участников сервис-ориентированной архитектуры.*

Сервисы слабо связаны, потому что используют основанные на стандартах интерфейсы, которые могут быть вызваны, опубликованы и обнаружены. Основная задача сервисов в SOA – предоставление схемы и взаимодействия с приложением посредством сообщений через интерфейсы, областью действия которых является приложение, а не компонент или объект. SOA-архитектура может обеспечить упаковку бизнес-процессов в сервисы, поддерживающие возможность взаимодействия и использующие для передачи информации широкий диапазон протоколов и форматов данных. Клиенты и другие сервисы могут выполнять доступ к локальным сервисам, выполняющимся на том же уровне, или к удаленным сервисам по сети.

*4. Каким образом происходит идентификация web-сервисов?*

С использованием cookie:

1. Пользователь вводит логин и пароль в поля формы и отправляет их на сервер;

2. Сервер сравнивает присланные данные с теми, что хранятся в хранилище и идентифицирует пользователя;

3. На сервере создается объект сессии - чаще всего это хранилище типа “ключ-значение”, в котором могут храниться разные атрибуты, необходимые для работы с пользователем;

4. Объекта сессии с пользователем сервер посылает клиенту сгенерированный идентификатор. Этот идентификатор сессии затем пересылается клиентом - браузером - при каждом запросе к странице.

Вышеописанный процесс предоставляет удобный способ идентификации сессии пользователя для приложений: пользователя не нужно

вводить свои логин и пароль для того чтобы сервер мог удостовериться в том, что перед ним идентифицированный пользователь.

Позже для унификации процесса идентификации сеанса пользователя был создан стандарт JSON Web Token. Это структура, состоящая из 3-х частей:

1. Заголовок. Содержит в себе метаданные маркера (token). Состоит из как минимум двух частей: поля “typ”, определяющего тип подписи (по стандарту здесь должно быть значение “jwt”), и алгоритм шифрования “alg”. Данные хранятся в формате JSON и закодированы алгоритмом Base64;

2. Полезная нагрузка. Тут могут храниться данные, которые обычно содержатся в сессии: товары в корзине, email пользователя и т.д.

3. Цифровая подпись. Эта часть представляет собой цифровую подпись маркера. Она вычисляется отправляющей стороной и использует для проверки целостности сообщения, содержащегося в маркере. Вычисление происходит при помощи алгоритма, описанного в заголовке. Если алгоритм требует секретный пароль, то он так же сообщается в параметрах вызова хэш-функции.

*5. Что такое WSDL описание сервиса?*

WSDL (англ. Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML. Каждый документ WSDL 1.1 можно разбить на следующие логические части: 1) определение типов данных (types) — определение вида отправляемых и получаемых сервисом XML-сообщений; 2) элементы данных (message) — сообщения, используемые web-сервисом; 3) абстрактные операции (portType) — список операций, которые могут быть выполнены с сообщениями; 4) связывание сервисов (binding) — способ, которым сообщение будет доставлено

*6. Опишите преимущества и недостатки сервис-ориентированной архитектуры?*

*Преимущества:*

1. Согласование предметных областей. Повторное использование общих сервисов со стандартными интерфейсами расширяет технологические и бизнес-возможности, а также сокращает стоимость.

2. Абстракция. Сервисы являются автономными, доступ к ним осуществляется по формальному контракту, что обеспечивает слабое связывание и абстракцию.

3. Возможность обнаружения. Сервисы могут предоставлять описания, что позволяет другим приложениям и сервисам обнаруживать их и автоматически определять интерфейс.

4. Возможность взаимодействия. Поскольку протоколы и форматы данных основываются на отраслевых стандартах, поставщик и потребитель сервиса могут создаваться и развертываться на разных платформах.

5. Рационализация. Сервисы обеспечивают определенную функциональность, устраняя необходимость ее дублирования в приложениях.

*Недостатки:*

1. Увеличение накладных расходов. Каждый раз, когда служба взаимодействует с другими службами, полную проверку каждого входного параметра происходит. Это увеличивает время отклика и нагрузку машины, и тем самым снижает общую производительность.

2. Комплексное управление сервисом. Служба должна гарантировать, что сообщения были доставлены своевременно. Но как сохранить сервисы обмена сообщениями для выполнения задач, количество этих сообщений может пойти на миллионы, даже для одного приложения. Это представляет большой вызов, чтобы управлять таким огромным населением услуг.

3. Высокая инвестиционная стоимость. Реализация SOA требует большие первоначальные вложения, технологии, развития и людских ресурсов. SOA-это не рекомендуется для следующих типов приложений.

**Листинг программы.**

using System;using System.IO;using System.Net;using System.Xml;using Newtonsoft.Json;

using Newtonsoft.Json.Linq;

namespace Client {

internal class Program {

static public void SendGetRequest(string url, string param = "") {

var httpRequest = (HttpWebRequest) WebRequest.Create(url + param);

httpRequest.Method = "GET";

var httpResponse = httpRequest.GetResponse();

var responseStream = httpResponse.GetResponseStream();

var reader = new StreamReader(responseStream);

var response = reader.ReadToEnd();

Console.WriteLine(response);

}

public static void Main(string[] args) {

SendGetRequest("https://speller.yandex.net/services/spellservice?WSDL");

SendGetRequest("https://speller.yandex.net/services/spellservice/checkText?", "text=сабака");

}

}

}