

SOAP & RESTFULL

№ урока: 4 **Курс:** Основы Java EE

Средства обучения: JDK, IntelliJ Idea

Обзор, цель и назначение урока

Создание SOAP и RESTfull сервисов. Работа с JSON.

Изучив материал данного занятия, учащийся сможет:

- Создать первый SOAP и RESTfull веб-сервисы
- Тестировать веб-сервисы
- Создавать клиент к веб-сервисам
- Познакомиться с JSON и как с ним работать в Java

Содержание урока

1. Введения в веб сервисы.
2. Виды веб сервисов
3. Установка Glassfish
4. Тестирование веб-приложения с использованием Glassfish
5. Создание SOAP сервиса.
6. Тестирование SOAP сервиса.
7. Создание клиента к SOAP сервису
8. Общие типы параметров HTTP-запроса
9. Создание RESTfull сервиса и его тестирования
10. Подключения к API и получение JSON ответа
11. Конвертирование Java объектов в JSON и наоборот

Резюме

- **Веб сервис** — идентифицируемая веб-адресом программная система со стандартизированными интерфейсами. Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (SOAP, XML-RPC и т. д.) и соглашениях (REST). Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения. В обиходе *веб-сервисами* называют услуги, оказываемые в Интернете. В этом употреблении термин требует уточнения, идёт ли речь о поиске, веб-почте, хранении документов, файлов, закладок и т. п. Такими веб-сервисами можно пользоваться независимо от компьютера, браузера или места доступа в Интернет.
- Все технологии строятся на спецификациях. Веб сервис — общее понятие, не спецификация. Веб сервис состоит из набора спецификаций (мы выбираем какие именно хотим использовать для достижения нашей цели). У одной спецификации может быть множество реализаций (библиотек) от разных компаний.
- В основе веб-сервисов лежат Internet-стандарты. Эти стандарты определяют протоколы, а не способы их реализации. Такое утверждение является залогом успеха Internet — ни одна компания не может влиять на Internet-стандарты и задавать собственные правила игры. Например, стандарты веб-сервисов разрабатываются совместно такими компаниями, как IBM, Microsoft, Ariba и некоторыми другими, и обсуждаются комитетом World Wide Web Consortium (W3C). Веб-сервисы базируются на трех основных веб-стандартах:

- SOAP (Simple Object Access Protocol) — протокол для отправки сообщений по протоколу HTTP и другим Internet-протоколам;
- WSDL (Web Services Description Language) — язык для описания программных интерфейсов веб-сервисов;
- UDDI (Universal Description, Discovery and Integration) — стандарт для индексации веб-сервисов. Серверы приложений являются хранилищами веб-сервисов и делают их доступными через протоколы HTTP GET, HTTP POST и HTTP SOAP.

Существующие веб-сервисы описываются в WSDL-документах, которые располагаются либо на сервере приложений, либо в специальных XML-хранилищах. WSDL-документ может ссылаться на другие WSDL-документы и документы XSD (XML Schema), в которых описаны типы данных, используемые веб-сервисами. XML-хранилища используются для управления WSDL-документами. Внутри WSDL-документа находится адрес (URL) веб-сервиса. Веб-сервисы описаны и проиндексированы в бизнес-реестре, содержащем адреса (URL) WSDL-документов.

- **HTTP** — протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных). Основой HTTP является технология «клиент-сервер», то есть предполагается существование:

Потребителей (клиентов), которые инициируют соединение и посылают запрос;

Поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

HTTP в настоящее время повсеместно используется во Всемирной паутине для получения информации с веб-сайтов. В 2006 году — в Северной Америке доля HTTP-трафика превысила долю P2P-сетей и составила 46 %, из которых почти половина — это передача потокового видео и звука.

HTTP используется также в качестве «транспорта» для других протоколов прикладного уровня, таких как SOAP, XML-RPC, WebDAV.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т. д. (в частности, для этого используется HTTP-заголовок). Именно благодаря возможности указания способа кодирования сообщения, клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

HTTP — протокол прикладного уровня; аналогичными ему являются FTP и SMTP. Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URI. В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ». Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами (например, «куки» на стороне клиента, «сессии» на стороне сервера). Браузер, посылающий запросы, может отслеживать задержки ответов. Сервер может хранить IP-адреса и заголовки запросов последних клиентов. Однако сам протокол не осведомлён о предыдущих запросах и ответах, в нём не предусмотрена внутренняя поддержка состояния, к нему не предъявляются такие требования.

- **GET** - запрос используется для запроса содержимого указанного ресурса. С помощью метода GET можно также начать какой-либо процесс. В этом случае в тело ответного сообщения следует включить информацию о ходе выполнения процесса.

Клиент может передавать параметры выполнения запроса в URI целевого ресурса после символа «?».

GET/path/resources?param1=value1¶m2=value2 HTTP/1.1

Согласно стандарту HTTP, запросы типа GET считаются идемпотентными

Кроме обычного метода GET различают ещё

Условный GET — содержит заголовки и подобные;

Частный GET — содержит в запросе Range.

Порядок выполнения подобных запросов определён стандартами отдельно.

- POST** - запрос применяется для передачи пользовательских данных заданному ресурсу. Например, в блогах посетители обычно могут вводить свои комментарии к записям в HTML-форму, после чего они передаются серверу методом POST и он помещает их на страницу. При этом передаваемые данные (в примере с блогами — текст комментария) включаются в тело запроса. Аналогично с помощью метода POST обычно загружаются файлы на сервер. В отличие от метода GET, метод POST не считается идемпотентным, то есть многократное повторение одних и тех же запросов POST может возвращать разные результаты (например, после каждой отправки комментария будет появляться очередная копия этого комментария). При результате выполнения 200 (Ok) в тело ответа следует включить сообщение об итоге выполнения запроса. Если был создан ресурс, то серверу следует вернуть ответ 201 (Created) с указанием URI нового ресурса в заголовке Location. Сообщение ответа сервера на выполнение метода POST не кэшируется.
- SOAP** (от англ. Simple Object Access Protocol — простой протокол доступа к объектам) — протокол обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC). Сейчас протокол используется для обмена произвольными сообщениями в формате XML, а не только для вызова процедур. Официальная спецификация последней версии 1.2 протокола никак не расшифровывает название SOAP. SOAP является расширением протокола XML-RPC. SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS и др. Однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно. Чаще всего SOAP используется поверх HTTP. SOAP является одним из стандартов, на которых базируются технологии веб-служб.
- REST** (сокр. от англ. Representational State Transfer — «передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы. В определённых случаях (интернет-магазины, поисковые системы, прочие системы, основанные на данных) это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине. REST является альтернативой RPC. В сети Интернет вызов удалённой процедуры может представлять собой обычный HTTP-запрос (обычно «GET» или «POST»; такой запрос называют «REST-запрос»), а необходимые данные передаются в качестве параметров запроса. Для веб-служб, построенных с учётом REST (то есть не нарушающих накладываемых им ограничений), применяют термин «**RESTful**». В отличие от веб-сервисов (веб-служб) на основе SOAP, не существует «официального» стандарта для RESTful веб-API. Дело в том, что REST является архитектурным стилем, в то время как SOAP является протоколом. Несмотря на то, что REST не является стандартом сам по себе, большинство RESTful-реализаций используют стандарты, такие как HTTP, URL, JSON и XML.
- GlassFish** — сервер приложений с открытым исходным кодом, реализующий спецификации Java EE, изначально разработанный Sun Microsystems. В настоящее время спонсируется корпорацией Oracle. Актуальная версия платформы называется Oracle GlassFish Server. В основу GlassFish легли части кода Java System Application Server компании Sun и ORM TopLink (решение для хранения Java объектов в реляционных БД, предоставленное Oracle). В качестве сервлет-контейнера в нём используется модифицированный Apache Tomcat, дополненный компонентом Grizzly, использующим технологию Java NIO.
- JSON** (англ. *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. Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта ECMA-262 1999 года), формат считается независимым от языка и может использоваться практически с любым языком программирования. Для многих языков существует готовый код для создания и обработки данных в формате JSON. За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, в таком ключе он

уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-сопряжения).

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 (регистрo зависима: 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.

Это универсальные структуры данных: как правило, любой современный язык программирования поддерживает их в той или иной форме. Они легли в основу JSON, так как он используется для обмена данными между различными языками программирования.

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`.

Закрепление материала

- Что такое веб сервисы и зачем они нужны?
- Какие есть виды веб сервисов?
- Что такое SOAP?
- Что такое RESTful?
- В чем основное отличие SOAP и RESTful сервисов?
- Что такое JSON?
- В чем отличие между JSON и XML?
- Зачем нужен Glassfish?

Дополнительное задание

Задание

Реализуйте свои первые простые SOAP и RESTfull сервисы для перевода слов. Сервис знает около 20 слов на английском языке. Вы вводите слово на русском, а сервис дает ответ на английском.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Используя JSON подключиться к НБУ и узнать текущий курс валют (RUB, UAN, USD, EURO).

Задание 3

С помощью скрипта из вложений carsshop.sql создать базу данных carsshop. И создать SOAP-сервис, который с помощью jdbc будет брать dao объектов из таблиц. Создать клиента к этому сервису и получить данные.

Рекомендуемые ресурсы

WSDL

https://www.w3schools.com/xml/xml_wsdl.asp

SOAP

https://www.w3schools.com/xml/xml_soap.asp

SOAP Services

<https://jcp.org/aboutJava/communityprocess/mrel/jsr224/index4.html>