XJCO3011: Web Services and Web Data

Instructor:

Dr. Guilin Zhao

Computer Science, SWJTU-LEEDS JOINT SCHOOL, Spring 2023

Welcome to XJCO3011!

- Today's lecture
  - Course Administration
    - Instructor and Teaching Assistant
    - Assessment
    - Course Plan
  - Module Overview-Introduction
    - RESTful Web Services
    - Search Engines
    - Linked Data and the Semantic Web
  - HTTP – The Workhorse of the Web

Teaching Staff

- Instructor: Dr. Guilin Zhao, X31415, Phone: 18981960065,

  Email: guilinzhao@swjtu.edu.cn or sjugz@leeds.ac.uk

- Time & Mode: Lecture: Monday, 9:50am – 11:25am, 10 weeks;

  Lab: Tuesday, 9:50am – 12:15pm, 7 weeks;

- Office Hours: Wednesday/Thursday, 9:30 am – 10:45 am

  or other time by appointment via email;

- TAs: Lilan Peng (PhD student),

  Phone: 15102816154, Email: llpeng@my.swjtu.edu.cn

  Lin Chen (Master student),

  Phone: 13219045985, Email: 1123364568@qq.com

Grading Policy

- Coursework: 60%
  - Coursework 1: 30%, release/due date: To be announced
  - Coursework 2: 30%, release/due date: To be announced
- Examination: 40%
  - Type: Online Timed Limited Assessment (OTLA)
  - Date: To be announced
  - Time: To be announced

Course Plan – Lecture

Week	Date	Session Number	Topic	Coursework Milestones
1	20 Feb	1&2	Introduction & Principles of HTTP	
2	27 Feb	3&4	Principles of RESTful APIs & The Semantic Gap	
3	6 Mar	5&6	Hypermedia & Django	
4	13 Mar	7&8	More Django & RESTful APIs Design	
5	20 Mar	9&10	Python anywhere & Web Crawling	
6	27 Mar	11&12	Parsing and Tokenization & Link Analysis	
7	3 April	13&14	Indexing & Query Processing	
8	10 April	15&16	Introduction to Linked Data & RDF	
9	17April	17&18	More RDF & Querying Linked Data with SPARQL	
10	24 April	19&20	SPARQL Examples & Closure	
11	28 April			

Course Plan – Lab

Week	Date	Topic
2	28 Feb	HelloWorld
3	7 Mar	Model Implementation
4	14 Mar	Templates View URL Testing
5	21 Mar	Database
6	28 Mar	BlogAPP
7	4 April	Forms
8	11 April	Custom Model

Outline

- ✓ Course Administration
  - Instructor and Teaching Assistant
  - Assessment
  - Course Plan
- Module Overview-Introduction
  - RESTful Web Services
  - Search Engines
  - Linked Data and the Semantic Web
- HTTP – The Workhorse of the Web

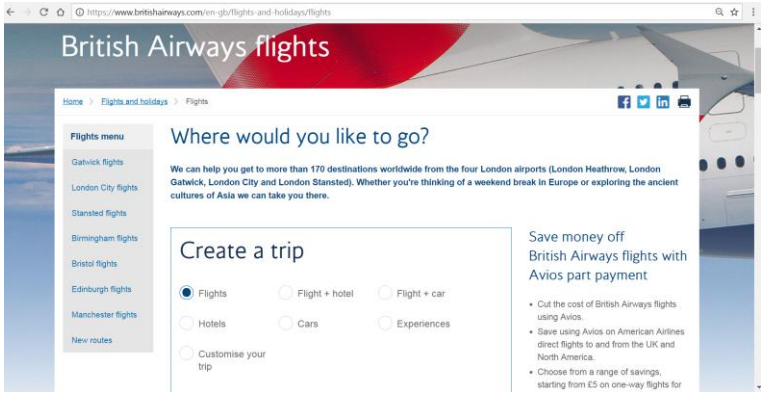
Overview

In this module, we will study the following important and highly related topics:

- RESTful Web Services (8 Units)
- Search Engines (5 Units)
- Linked Data and the Semantic Web (5 Units)

Web Services

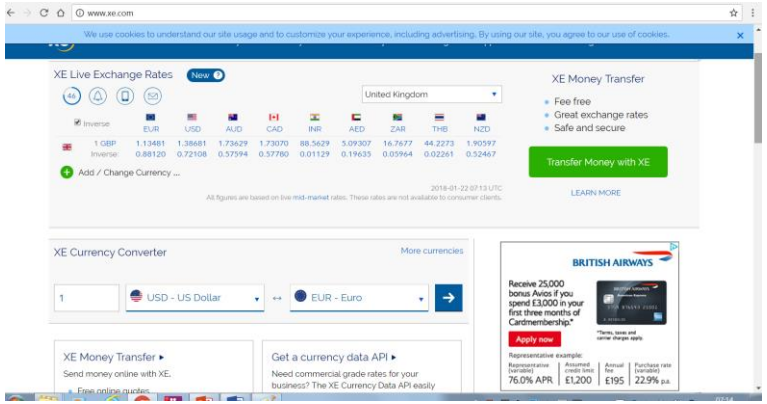
- Modern software systems often need to exchange data with each other over the internet. For example, an airline website may require live currency exchange rates to compute prices in any desired currency. This data can best be provided by a financial services website.
- A web service is a method of communication that allows two software systems to exchange data over the internet.
- The software system that requests data is called a service requester, whereas the software system that would process the request and provides the data is called a service provider.



Airline website

Request currency rates

Response with currency rates



Financial services website

Mashups

- Web services allows us to create mashups.
- A mashup is a software (e.g. web application) that uses content from more than one source to create a single new service displayed in a single graphical interface.
- For example, a user could combine the addresses and photographs of library branches with a Google map to create a map mashup.

Web Services Paradigms

Today, we can distinguish two main paradigms for web services:

- SOAP-based services.
- RESTful APIs.

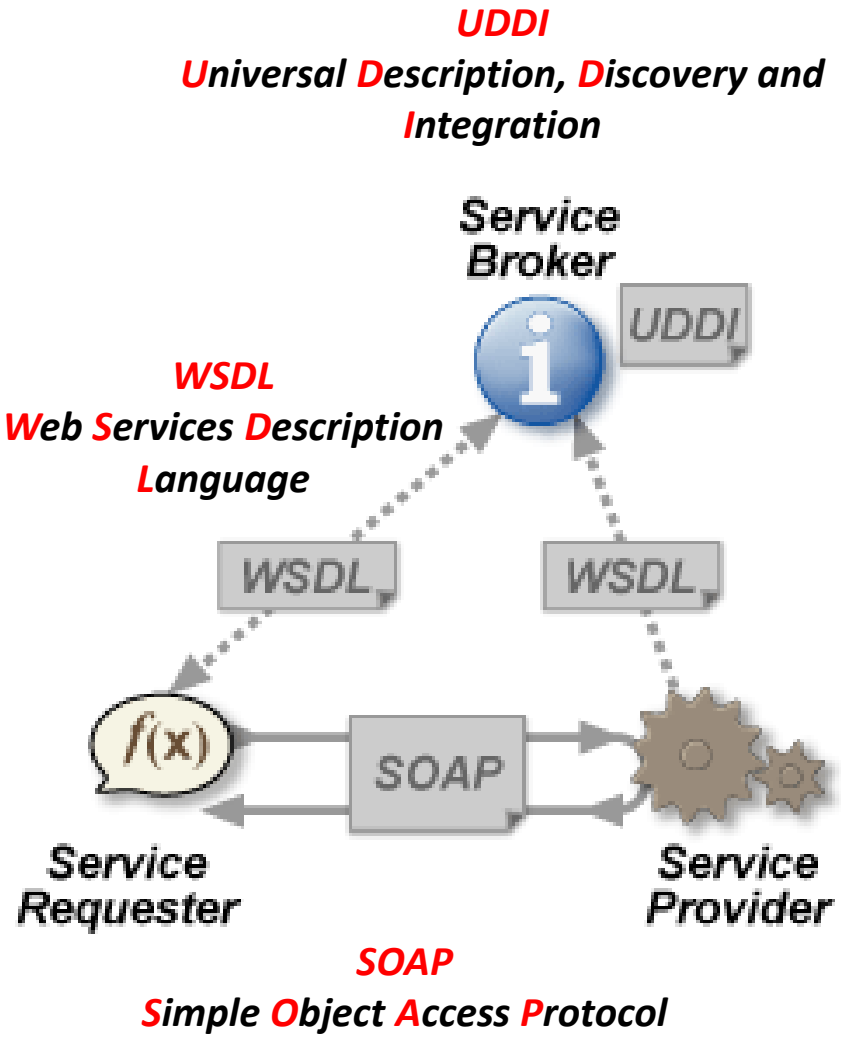
Web Services: The Challenge

- Different software may use different programming languages, platforms, and operating systems, hence there is a need for a method of data exchange that doesn't depend upon a particular platform.
- Fortunately, many common data formats, such as XML and JSON, are recognised by the vast majority of software and hence can be used for data exchange in web services.
- In addition, existing web technology such as HTTP, originally designed for human-to-machine communication, can be utilized for machine-to-machine communication, i.e. for transferring machine-readable file formats such as XML and JSON.

SOAP

The term web service originally described methods for integrating web-based applications using a number of open standards, namely: XML, SOAP, WSDL and UDDI, over an Internet Protocol backbone:

- XML is the data format used to contain the data and provide metadata around it.
- SOAP is used to transfer the data.
- WSDL is used for describing the services available.
- and UDDI lists what services are available.
- Don't worry if the above jargon sounds too complex because it is. Fortunately, we are NOT going to use SOAP in this module (you will learn why in a few slides).



SOAP is Complex

In SOAP based services , rules for communication between different systems need to be defined, such as:

- How one system can request data from another system.
- Which specific parameters are needed in the data request.
- What would be the structure of the data produced.
- What error messages to display when a certain rule of communication is not observed.
- All of these rules of communication are defined in a file called WSDL (Web Services Description Language)
- A directory called UDDI (Universal Description, Discovery and Integration) defines which software system should be contacted for which type of data.
- Once the client finds out which other system it should contact, it would then contact that system using a special protocol called SOAP (Simple Object Access Protocol).
- The service provider would first validate the data request by referring to the WSDL file, and then process the request and send the data under the SOAP protocol.

RESTful Web Services (APIs)

- A RESTful Web API is a development in web services where emphasis has been moving to simpler representational state transfer (REST) based communications.
- Unlike SOAP-based Web services, restful APIs do not require the XML-based web service protocols (SOAP and WSDL) to support their interfaces.
- There is no official standard for RESTful Web APIs. This is because REST is an architectural style, while SOAP is a protocol.
- Although REST is not a standard in itself, RESTful implementations make use of standards, such as HTTP, URI, JSON, and XML.
- Unfortunately, many developers also describe their APIs as being RESTful, even though these APIs actually don't fulfil all of the architectural constraints described later.

REST

- Representational state transfer (REST) is a way of providing interoperability between computer systems.
- In a RESTful system, the server does not maintain state information about the client. Each request is served on its own merits regardless of any previous requests.
- The client is responsible for maintaining its own state transitions based on the representations sent by the server (hence the name of this method).
- REST-compliant services allow requesting systems to access and manipulate resources using a uniform and predefined set of stateless operations.
- In RESTful APIs, web resources (first defined for the World Wide Web as documents or files identified by URLs) have a more generic definition encompassing every entity that can be identified, named, or addressed (such as an object in a data base).
- In a RESTful service, requests made to a resource's URI will trigger a response that may be in XML, HTML, JSON or any other multimedia format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources.
- Web APIs use HTTP as the underlying protocol for operations including those predefined by the CRUD (Create, Read, Update, and Delete) HTTP methods: POST, GET, PUT, and DELETE, as we will learn in Unit 2.

RESTful Architectural Constraints

There are a number of guiding constraints that define a RESTful system. By operating within these constraints, the service gains desirable non-functional properties, such as performance, scalability, simplicity, modifiability, visibility, portability, and reliability. If a service violates any of the required constraints, it cannot be considered RESTful. This is mainly achieved through 3 main principles:

1. Client-server architecture (leading to the separation of concerns). Separating the user interface concerns from the data storage concerns improves the portability of the user interface across multiple platforms
2. Statelessness. The client–server communication is constrained by no client context being stored on the server between requests. Each request from any client contains all the information necessary to service the request, and session state is held in the client.
3. Uniform interface. It simplifies and decouples the architecture, which enables each part to evolve independently.

SOAP vs REST

- SOAP is a protocol.
- SOAP defines standards to be strictly followed.
- SOAP requires more bandwidth and resources than REST.
- SOAP defines its own security.
- SOAP permits the XML data format only.
- SOAP can support stateful implementations.
- SOAP based reads cannot be cached.
- **SOAP is COMPLEX.**

- REST is an architectural style
- REST does not define strict standards like SOAP.
- REST requires less bandwidth and resources than SOAP.
- RESTful web services inherits security measures from the underlying transport layer.
- REST permits different data formats such as plain text, HTML, XML, JSON etc.
- REST follows the stateless model
- REST has better performance and scalability.
- REST reads can be cached.
- JSON usually parses much faster than XML
- **REST is SIMPLE.**

Battle of the Services SOAP against REST

- For some time during the last decade, there were heated arguments between proponents of SOAP and REST.
- Today, the battle is almost over, and REST has won.
- It is estimated that more than 70% of services available today use the REST architecture.
- Major providers of cloud services, such as Amazon and Google, have now adopted the RESTful model, and they are phasing out their support for SOAP-based interfaces.
- SOAP is still preferred when higher levels of security are required.

SOAP

RESTful

Overview

In this module, we will study the following important and highly related topics:

- ✓ RESTful Web Services (8 Units)
- Search Engines (5 Units)
- Linked Data and the Semantic Web (5 Units)

Search Engines

- A web search engine is a software system designed to search for information on the World Wide Web.
- Search engines extract information from a wide variety of data sources such as HTML web pages, pdf files, MS Word docs, images, videos, and many other types of files.
- Some search engines can also mine data from databases
- Unlike web directories, which are maintained by human editors, search engines obtain real-time data by continuously running a web crawler.
- However, search engines cannot discover everything on the web. Web content that cannot be discovered by search engines is called the deep web.

The Deep Web

- The deep web (also called the invisible or hidden web) is that part of the World Wide Web whose contents cannot be indexed by web search engines.
- The content of the deep web is hidden behind HTML forms, web mail, online banking services, paid services, or services protected by a paywall, such as video on demand, some online magazines and newspapers, and many more.
- Content of the deep web can only be located and accessed by a direct URL or IP address, and may require password or other security protection beyond the public website page.

Size of the Deep Web

- It is difficult to estimate the size of the deep web.
- However, some estimates suggest that it is several orders of magnitude larger than the surface web.
- An iceberg analogy depicts the proportions of the surface web to the deep web.

How Search Engines Work

A search engine maintains the following processes in near real time:

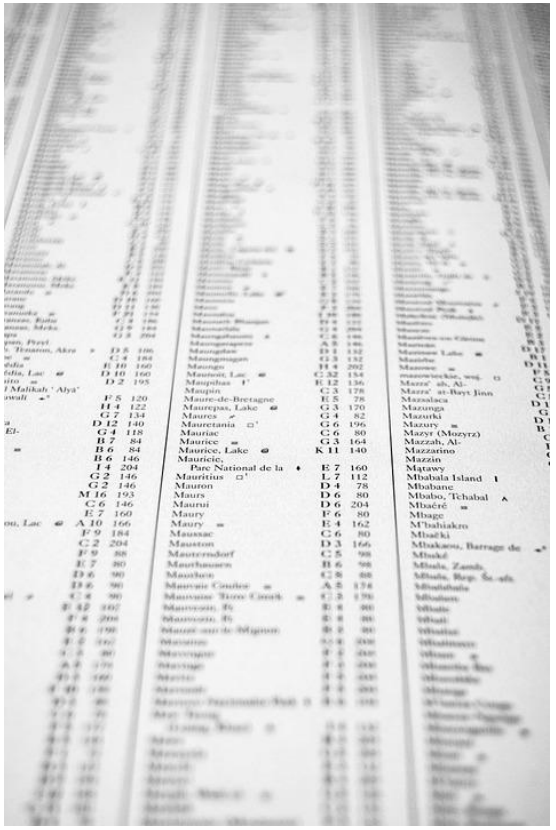
- Web crawling
- Indexing
- Searching

Web Crawling

- Search engines get their information by crawling from one web page to another (like a spider).
- The spider follows the hyperlinks in each page to discover other web pages.
- Yet, a web spider cannot actually crawl the entire reachable web, due to infinite websites, spider traps, link rot, and other factors,
- Crawlers instead apply a crawl policy to determine when the crawling of a site should be deemed sufficient.
- Some sites are crawled exhaustively, while others are crawled only partially.

Indexing

- Indexing means associating words found on web pages to their URLs and HTML-based fields.
- The information to be indexed depends on many factors, such as the titles, page content, JavaScript, Cascading Style Sheets (CSS), headings, or metadata in HTML meta tags.
- The associations are made in a public database, made available for web search queries.
- The index helps find information relating to the query as quickly as possible.



Searching

- When a user enters a query into a search engine, the index already has the URLs containing the keywords, and these are instantly obtained from the index.
- The real processing load is in generating the list of search results of web pages, i.e. every page in the list must be weighted according to information in the indices.
- Then search results requires the lookup, reconstruction, and markup of the snippets showing the context of the keywords matched.

Overview

In this module, we will study the following important and highly related topics:

- ✓ RESTful Web Services (8 Units)
- ✓ Search Engines (5 Units)
- Linked Data and the Semantic Web (5 Units)

Structure of the Traditional World Wide Web

- The WWW is a collection of linked documents that are full of data.
- Many of these documents have little, if any, structure imposed on the data (mostly images and free text).
- The data is available in so many formats such as HTML, XML, PDF, TIFF, CSV, Excel spreadsheets, embedded tables in Word documents, and many forms of plain text.
- This kind of data has a limitation: it's formatted for human consumption.
- It often requires a specialized utility to read it.
- It's not easy for automated processes to access, search, or reuse this data.
- Further processing by people is generally required for this data to be incorporated into new projects or allow someone to base decisions on it.
- With the exception of some very simple cases, only humans can analyse the semantic relationships between data in various web pages.

The Linked Data Web

- Linked Data refers to a set of techniques for publishing and connecting structured data on the Web
- It adheres to standards set by the World Wide Web Consortium (W3C).
- The two images below show how the same information about Star War's character *Darth Vader* can be represented in a human readable form (i.e. natural language) or a machine readable format (called RDF Turtle).

One of the persons is "Anakin," known as "Darth Vader" and "Anakin Skywalker." He has a wife named Padme Amidala.

Unstructured Data. Good for Humans

```
@base <http://rosemary.umw.edu/~marsha/starwars/foaf.ttl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rel: <http://purl.org/vocab/relationship>.
@prefix stars: <http://www.starwars.com/explore/encyclopedia/characters/> .
 <me> a foaf:Person;
        foaf:family_name "Skywalker";
        foaf:givenname "Anakin";
        foaf:nick "Darth Vader";
        rel:Spouse_Of <stars:padmeamidala/> .
```

Structured data. Good for Automated Agents

Django

The Practical Sessions

- In the first coursework, we will be using the Django framework.
- Django is a free open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern.
- Created by Adrian Holovaty and Simon Willison in 2003, and named after the French jazz guitarist Django Reinhardt.
- Django can be used on its own to make a RESTful API, however, the Django REST Framework is an extension to the Django framework that has plenty of support for RESTful APIs.

Click to listen to music by Django Reinhardt

Reading List

- Croft W B, Metzler D, and Strohman T (2015), Search Engines - Information Retreival in Practice, Previously Published by Pearson, now available as a free e-book from http://ciir.cs.umass.edu/downloads/SEIRiP.pdf
- George N, Mastering Django: Core (The Django Book) (2016), available as a free online book here https://djangobook.com/the-django-book/
- Hillar G C, Building RESTful Python Web Services (2016), Packt Publishing, Chapters 1-4
- Richardson L and Amundsen M (2013), RESTful Web APIs, O'Reilly Media (available online through Safari Books Online)
- Totty B et al, HTTP The Definitive Guide (2009), O'Reilly Media (available online through Safari Books Online)
- Wood D et al (2013), Linked Data - Structured data on the Web, Manning Publications, Chapters 1-2