Web Application Processing – Server, Client

Web Application

1. A piece of software which can be accessed from a browser.

2. Multiple users, same version of App.

3. Doesn't Need to be installed.

4. For a web app to operate it needs – web server, Database

Web Server

(serves web content) Physical Device with Network Connection can host multiple Apps

1. Listens
2. On a Port
3. For a Request
4. Transport Protocol (HTTP, HTTPS)
5. Response
6. Resource – HTML, JSON, PDF, Image, Document, etc.

web client receives response from web server.

Browser, Mobile Application

Something that can talk to a server.

Client $\xleftrightarrow{\text{Protocol (HTTP)}}$ Server

Web Server — A network Application listening to some port.

Protocol — Set of agreed upon rules between client and server.

URL (Uniform Resource Allocator)

https://www.amazon.com/orders/123

protocol | addr. of web server | addr. of resource

54.192.0.0/16

Default Port Addr: 80 (HTTP) 443 (HTTPS)

HTTP Requests

GET
POST
PUT
DELETE

GET /orders/123
Host: 54.192.0.0:16
User-Agent: (Client Data) (Browser, Operating Sys info)
Connection: keep-Alive
|Body X

Responses

HTTP/1.1 200 OK
Date:
Server:
Content-length:
" -Type:
Connection: closed

1XX Informational
2XX Success
3XX Redirect
4XX Client Error
5XX Server "

Static Vs Dynamic

HTTP

i) Stateless
ii) Connectionless
iii) Application layer protocol
iv) TCP/IP - based
v) Media Independent

MIME Type

Multipurpose Internet Mail Extension

text/plain
html
csv

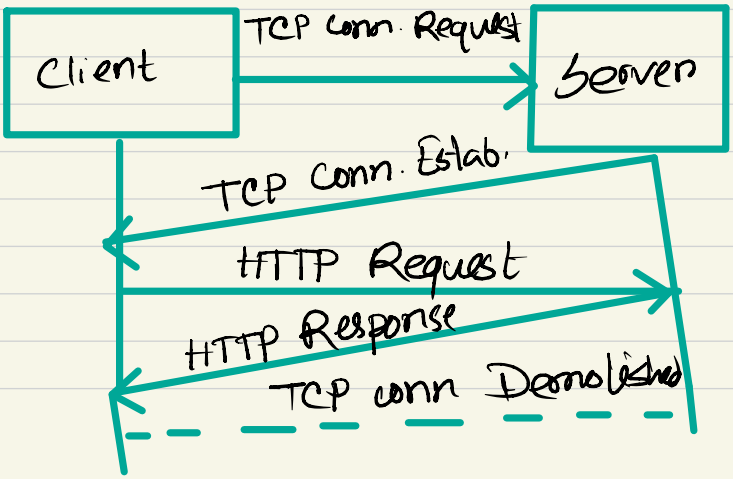
images/jpeg
/png

video/mp4

application/pdf
/zip

32-bits

Source Port	Destination Port	→ 16 bits
TCP/IP Packet format		



HTTP methods

GET
HEAD
POST
PUT
DELETE
CONNECT
OPTIONS
TRACE
PATCH

HTTP 1.0 / 1.1 single / multiple Request, Response.

Status Codes

1xx	101 Protocol Switch
2xx	200 OK
3xx	Request needs redirection
4xx	401 Unauthorized
5xx	500 Server failed

URL Fragments

https://www.swe4537.com/home?theme=dark#projects

URL scheme — host — Resource Path — QueryString — Fragments

What is Javascript?

- HTML is a markup language that we use to structure and give meaning to our web content e.g., defining paragraphs, headings, and multimedia contents, etc.
- CSS is a language of style rules that we use to apply styling to our HTML contents eg., font size, background colors, etc.
- JS is a scripting language that can update and manipulate both HTML and CSS, which, in turn enables you to create dynamically updating content, control multimedia, animate images, and so on.

Scripting languages do not need compiling step.

- Server: JS, Python, PHP
- Client: JS, JQuery, AJAX

Why is it called JavaScript?

- To make websites live by utilizing client side processing power.
- Livescript, then JavaScript
- JS doesn't make websites dynamic. ✗
  Responsive ✓

JavaScript Engine

- software component that executes JS code.
- Intially, Interpreters.
- Built by modern web browser vendors.
- modern engines use Just in-time compilers.

comments // /* */
variables var, let
constants universal, local
Naming conventions
Data Types
Type Conversion

semi-colon, prettier

Number — float, integer, Infinity, Neg. Infinity, NaN
Big Integer
String
Boolean
Null
Undefined
Symbol
Object

primitive – single Val.
Non-primitive – multiple

✓ Infinity (+ve / 0)
✓ Neg Infinity (-ve/0)
✓ Not a Number (data types where mathematical operations aren't possible)

1 / " 1" = 1 (spaces before and after it is possible)
1 / " 1 2 3" = NaN
string in a '+' operation results in a string.
'5' + 2 = 52
2 + '5' = 25

✓ Null – has a value – null
✓ Undefined – is not assigned a value.
✓ BigInt (Number)
12345n

JSON (JavaScript Object Notation)

{ key: value, key.value }
must be string

Typecast

DataType (data)
String (123)
x.toString()

Operators

Unary ++ , -- , + , -
Comparison > , < , >= , <= , !=
Logical && , || , !
Binary + , - , / , % , **
Bit-Wise & , | , ^ XOR , ~ , << LS , >> RS
Equality == , ===
Assignment

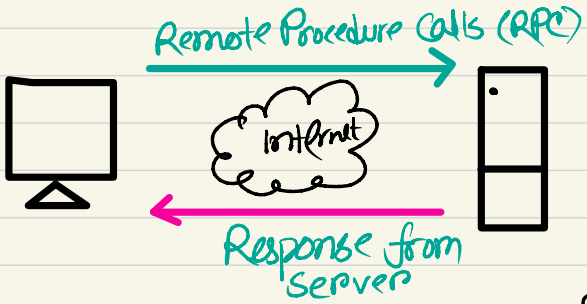
If-else
JS functions

Web Services

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. Other systems interact with the web systems in a manner prescribed by its description using SOAP messages conveyed using HTTP with an XML Serialization.

If an application can be accessed over a network using a combination of protocols like HTTP, XML, SMTP, etc. then it is a web service.

It acts as an abstraction layer separating the user of the code and the programming language specific details of how the application code is actually invoked.



The client would invoke a series of web service calls which would host the actual web service.

RPC A web service can have multiple methods to provide different functionality. These methods are called via RPC.

Data is transferred between the client and server as XML (Extensive Mark-up language, Easy to Understand Understood by many PLs)

Web services use something known as SOAP (Simple Object Access Protocol) for sending XML data between Applications. Data is sent over normal HTTP.

Data is called SOAP message.

Two types of web services:

1. SOAP web services
2. Restful Web services
   Representational State Transfer

SOAP

platform & OS independent messaging protocol.

- XML Document
- Structure follows a specific pattern, not the content.
- Best Part is sent via HTTP (content-type: text/XML
- light-weight data interchange protocol

Root Element
identifies XML as SOAP

<Envelope>
<Header> </Header> [Routing data, client info]
<Body> </Body> [content]
</Envelope>

Authentication

So, for message passing web services use SOAP over HTTP.

WSDL (Web Services Description Language)
Keeps the functionality of a web service.

- XML Document (Not SOAP)
- 3 parts
  - → Message - Data Elements for Each Operation
  - → PortType (Operation, Input-Output)
  - → Bindings - Protocol (transport, encoding)
    - transport → SOAP/http
    - encoding → SOAP

UDDI - Universal Description, Discovery, & Integration
(UDDI) stores the list of webservices.
All clients have access to this.

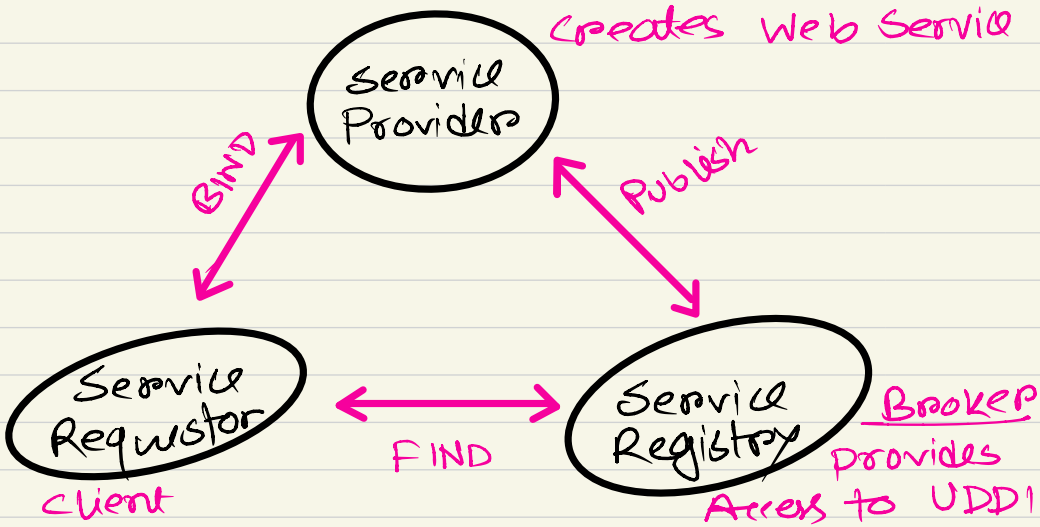
Advantages of Web Services

1. Exposing business functionality on the network.
2. Interoperability among applications.
3. Standardized protocol stack that everybody understands.
4. Reduction in cost of communication

previously used - EDI B2B
Electronic Data Interchange for
Business 2 Business.

Protocol Stack

1. Service Transport (Responsible for message passing)
   HTTP, SMTP, FTP, BEEP
   Box Extensible Exchange Protocol.
2. XML Messaging (Encode messages to a common format)
3. Service Description (Describes the public Interface WSDL)
4. Service Discovery (UDDI) - for centralizing services into a common registry



Characteristics

OS, network, Platform

XML Based – Eliminates Dependency

Loosely Coupled – If client or server changes, doesn't affect the communication

Synchronous / Asynchronous

Ability to support RPC

Supports Document Exchange → Simple to Complex
'String' 'Book'

Restful Web Services

Everything is a resource. You can access, manipulate.

RPC
SOAP X

URI	HTTP VERB	DESCRIPTION
api/users/1	GET	Gets first user.

Doesn't Require XML, WSDL, SOAP.
URL is enough. Not structured/protocol-based.

Service Exposure

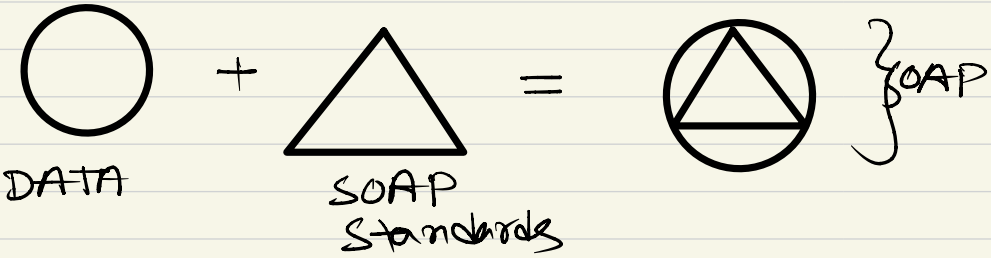
Uses URI to locate resources. Doesn't require a seperate document to transport messages.
HTTP is enough.
less Bandwidth.

Message format

XML
JSON
Any Other.

Conventions

Noun - Endpoint/Resources
Request using HTTP action verb
plural form of resources
HTTP Response code



Differences

1.	SOAP follows strict guidelines, since Protocol.	
2.	" uses XML.	Uses XML, JSON, any.
3.	RPC call,	URL is enough
4.	Difficult to Implement	
5.	More Bandwidth	
6.	Easier to perform transactions.	