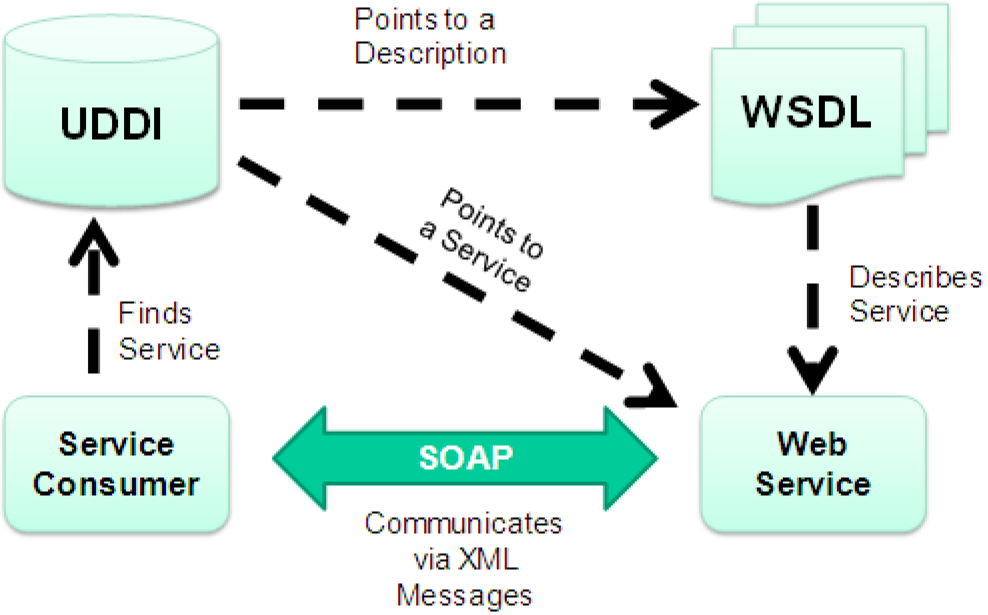
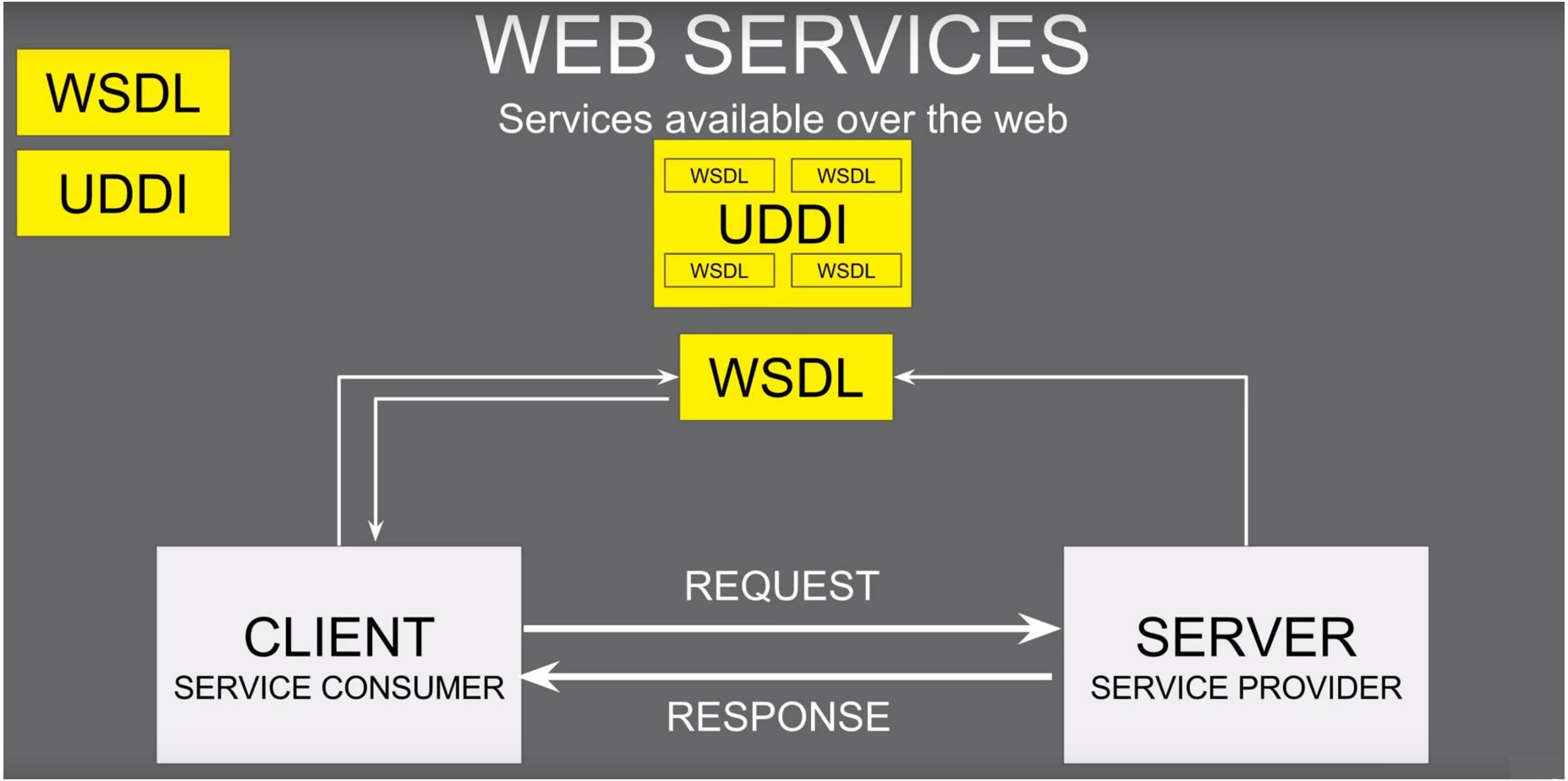
API/Web Services Testing basics

Components of Web Services : WSDL and UDDI

- **What is WSDL?**
- WSDL stands for Web Services Description Language, an XML-based language that describes Web services and how to access and locate them.
- **What is UDDI?**
- UDDI stands for Universal Description, Discovery and Integration. It is an open, Internet-based specification that offers directory service for storing information about web services.

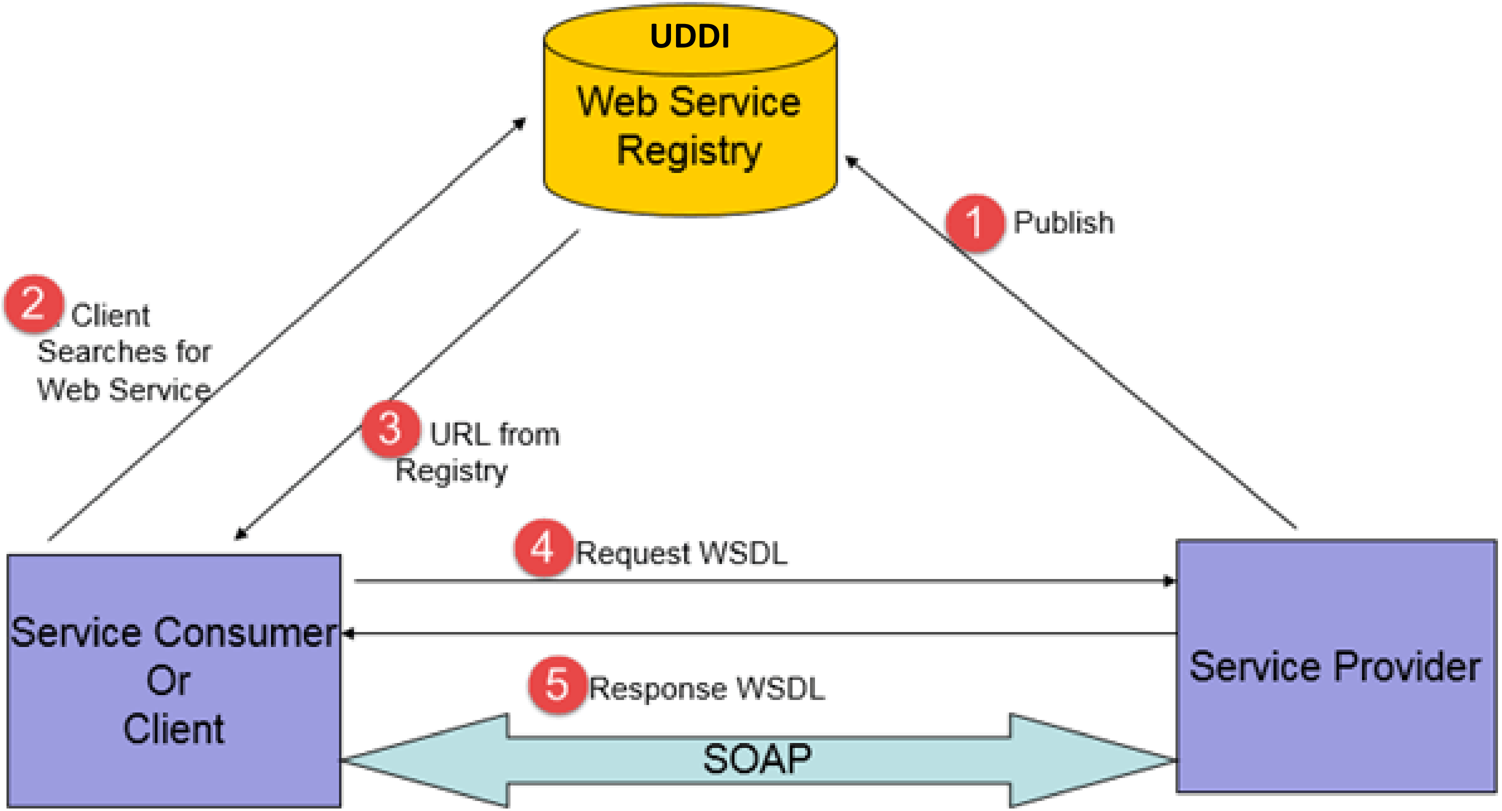




WEB SERVICES

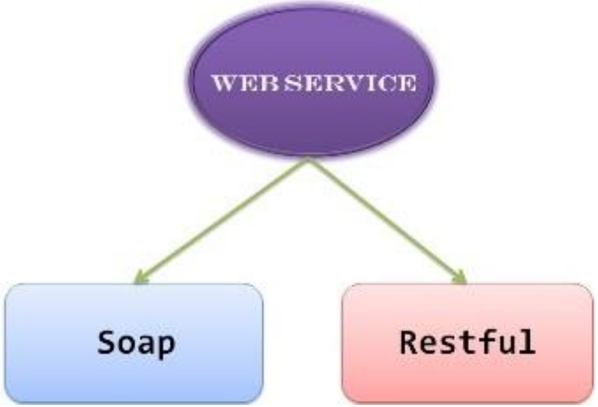
Services available over the web

UDDI (Universal Description, Discovery and Integration)



Types of Web Services

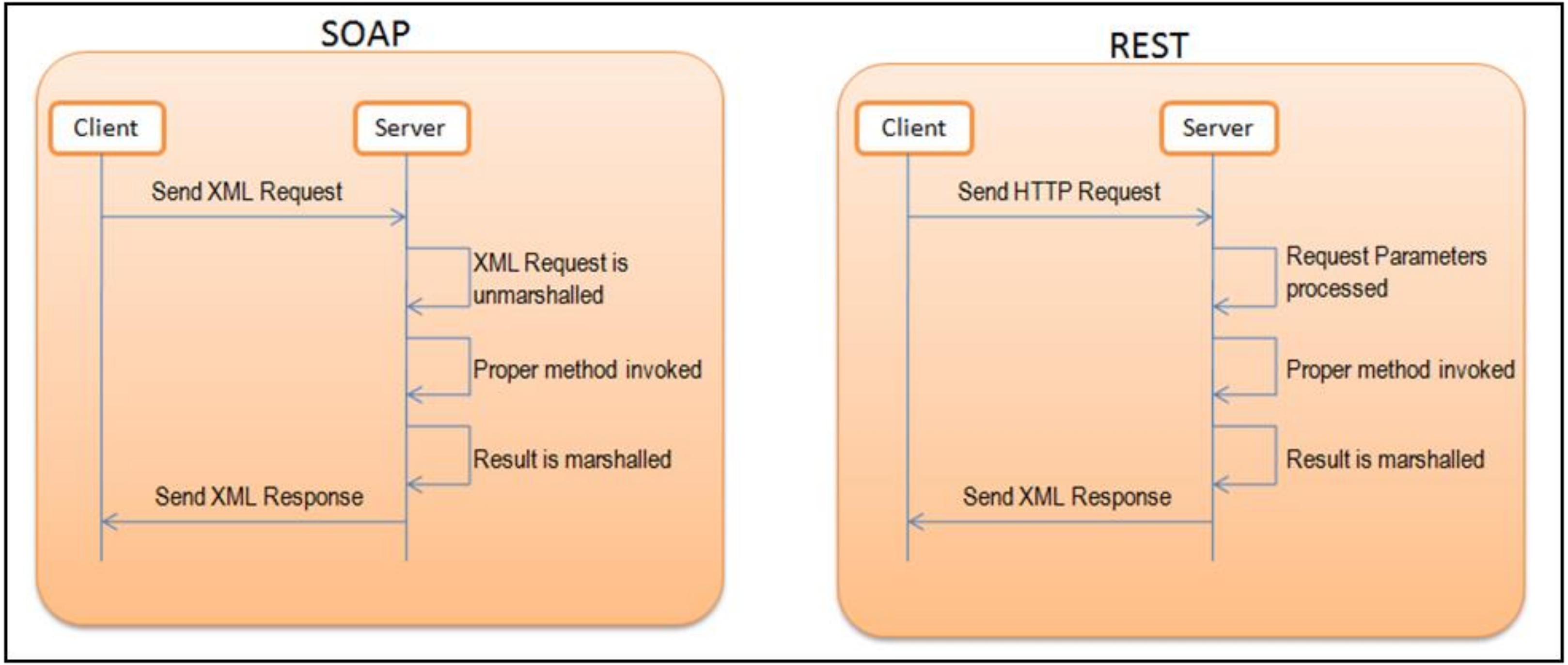
- There are mainly two types of web services.
  - SOAP web services. (Simple Object Access Protocol)
  - RESTful web services. (**Re**presentational **S**tate **T**ransfer)



- **SOAP** (Simple Object Access Protocol) – SOAP is a protocol which was designed before REST and came into the picture. The main idea behind designing SOAP was to ensure that programs built on different platforms and programming languages could exchange data in an easy manner.

- **REST** – This was designed specifically for working with components such as media components, files, or even objects on a particular hardware device. Any web service that is defined on the principles of REST can be called a RestFul web service. A Restful service would use the normal HTTP verbs of GET, POST, PUT and DELETE for working with the required components.

SOAP Vs Rest



SOAP

Client

Server

Send XML Request

XML Request is unmarshalled

Proper method invoked

Result is marshalled

Send XML Response

REST

Client

Server

Send HTTP Request

Request Parameters processed

Proper method invoked

Result is marshalled

Send XML Response

HTTP V/S HTTPS

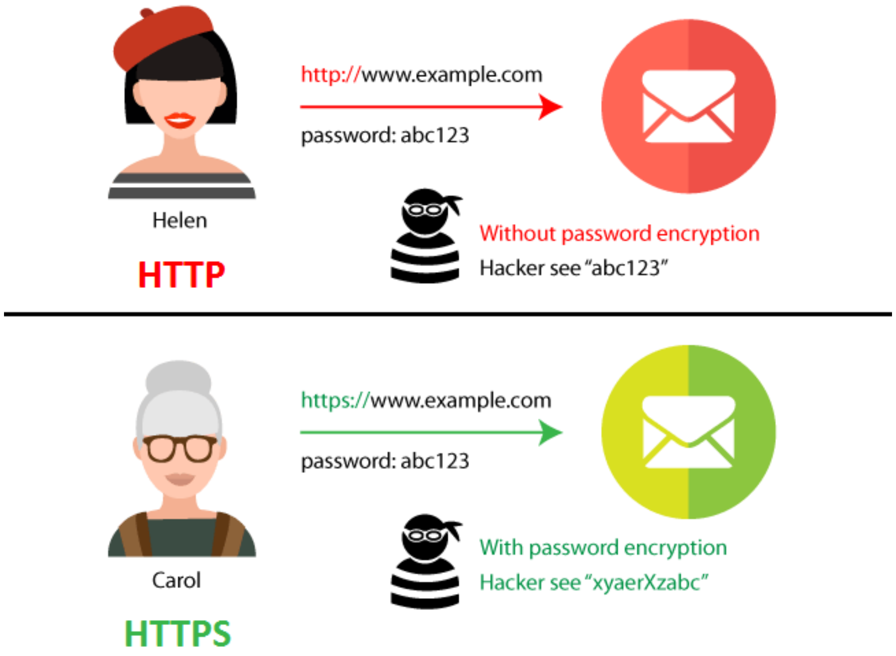
What Is HTTP?

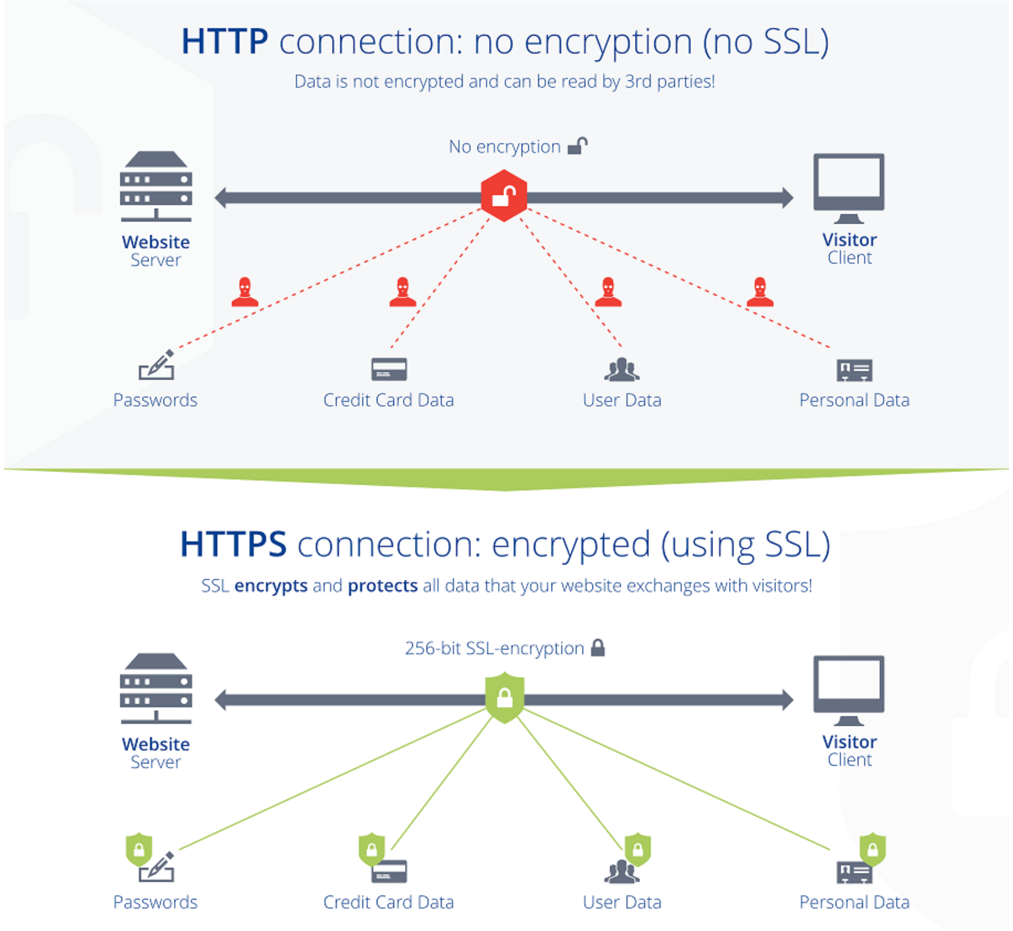
HTTP stands for *Hypertext Transfer Protocol*. At it's most basic, it allows for the communication between different systems. It's most commonly used to **transfer data from a web server to a browser** in order to allow users to view web pages. It's the protocol that was used for basically all early websites.

What Is HTTPS?

HTTPS stands for *Hypertext Transfer Protocol Secure*. The problem with the regular HTTP protocol is that the information that flows from server to browser is not encrypted, which means it can be *easily stolen*. HTTPS protocols remedy this by using an **SSL (secure sockets layer) certificate**, which helps <u>create a secure encrypted connection</u> between the server and the browser, thereby protecting potentially sensitive information from being stolen as its transferred between the server and the browser.

The Main Difference Between HTTP and HTTPS





HTTP connection: no encryption (no SSL)

Data is not encrypted and can be read by 3rd parties!

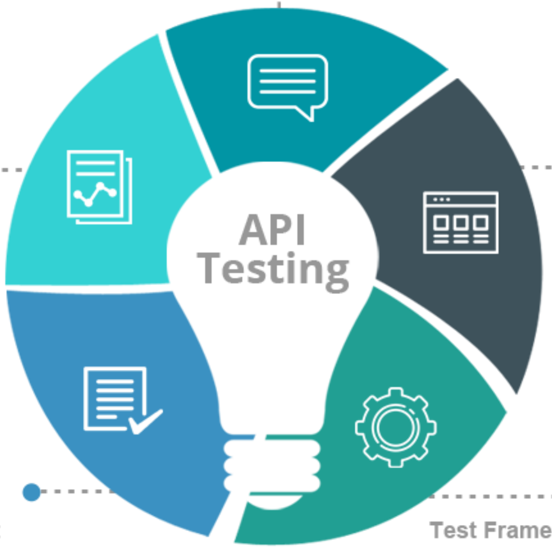
HTTPS connection: encrypted (using SSL)

SSL **encrypts** and **protects** all data that your website exchanges with visitors!

Web Service API testing tools

SoapUI
POSTMAN
Katalon
Assertible
apigee
APACHE JMeter™
REST-assured
TRICENTIS
Karate

Web service/API Testing process



API Specification Review

Review the API specification & use case documentation from test perspective

Test Specification Development

This document details the test conditions and expected results for each test case

Test Framework Development

Use standard open source tools like SoapUI, JMeter or develop a set of static resources

Test Case Development

Coding of test scenarios, creating sanity check test suites, survelliance test suites

Test Execution & Reporting

Execution of the developed test suites & reporting

REST SERVICES

RESTful web services

- REpresentational State Transfer
  - A service represents a resource that can be accessed from the web
  - Commonly makes use of different HTTP request methods to implement CRUD functions:
    - POST – Create new resource
    - GET – Retrieve resource
    - PUT – Update resource
    - DELETE – Delete resource

HTTP Methods

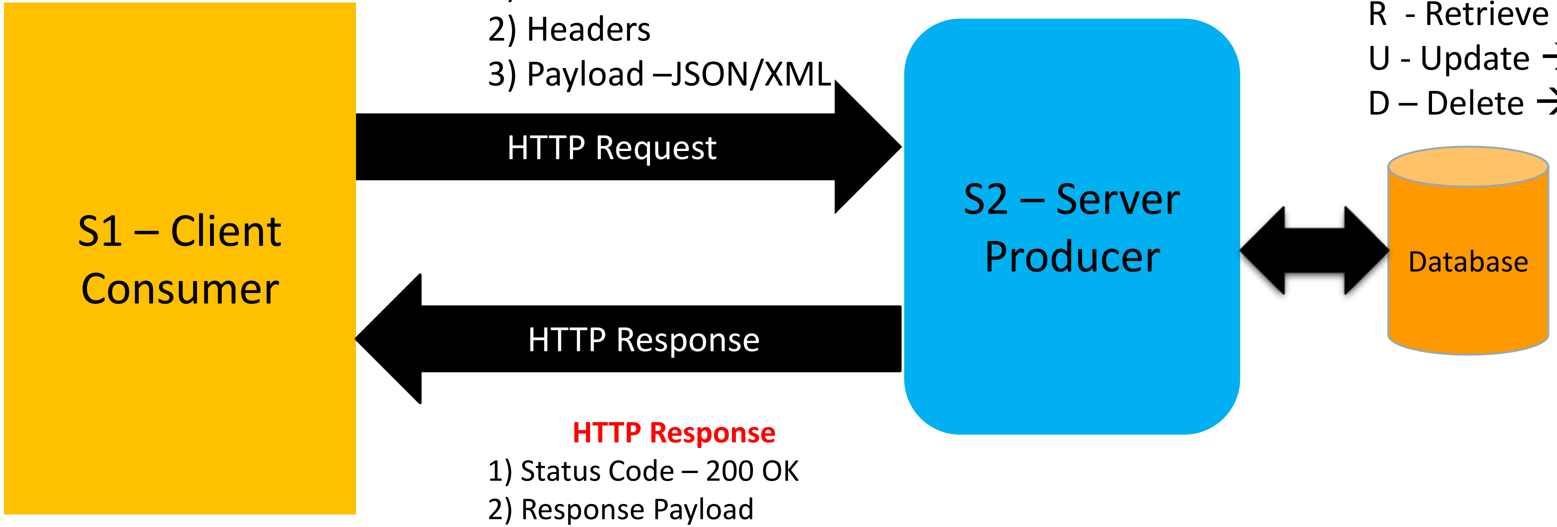
- GET
- POST
- PUT
- DELETE

HTTP Request

1) URI
2) Headers
3) Payload –JSON/XML

HTTP Methods

C - Create → POST
R - Retrieve → GET
U - Update → PUT
D – Delete → DELETE



HTTP Response

1) Status Code – 200 OK
2) Response Payload
3) String message Successful

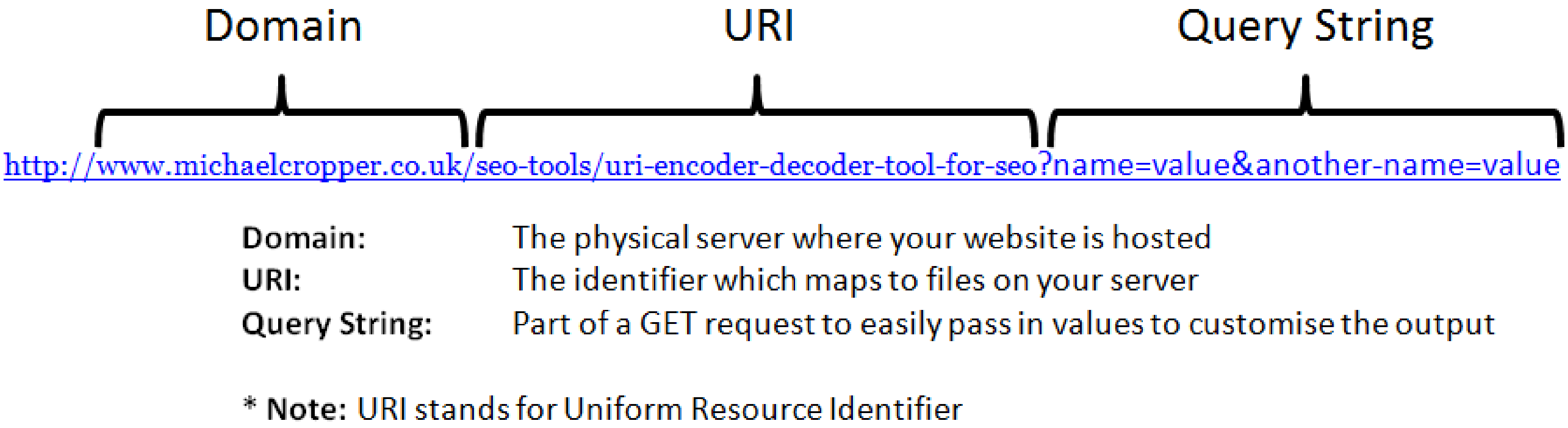
1) Create an account →POST Call
JSON/XML
AccountID=1
AccountName="Tom"
Address="London"
Account Type ="Savings"

2) Get an account by ID/Name →GET Call
URL → http://api/service/accountt/1
Path Parameter
Query Parameter
Headers

3) Update an account →PUT Call
Create + Update
AccountID=1
AccountName=Tome
Address = NewYork
Account Type =Current

3) Delete an account →DELETE Call
AccountID=1
AccountName=Tome
Address
Account Type

What is URI and URL



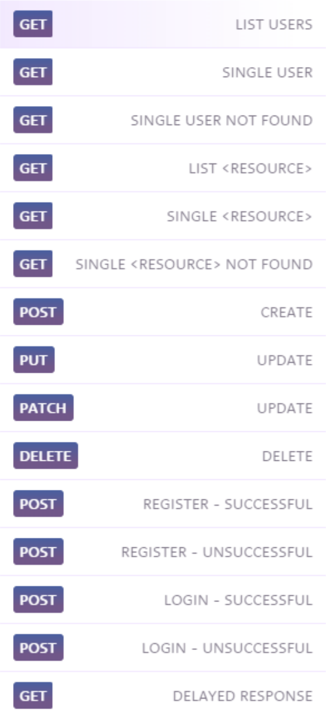
http://www.michaelcropper.co.uk/seo-tools/uri-encoder-decoder-tool-for-seo?name=value&another-name=value

Domain: The physical server where your website is hosted
URI: The identifier which maps to files on your server
Query String: Part of a GET request to easily pass in values to customise the output

* **Note:** URI stands for Uniform Resource Identifier

Examples - Demo

- https://reqres.in/



GET LIST USERS

GET SINGLE USER

GET SINGLE USER NOT FOUND

GET LIST <RESOURCE>

GET SINGLE <RESOURCE>

GET SINGLE <RESOURCE> NOT FOUND

POST CREATE

PUT UPDATE

PATCH UPDATE

DELETE DELETE

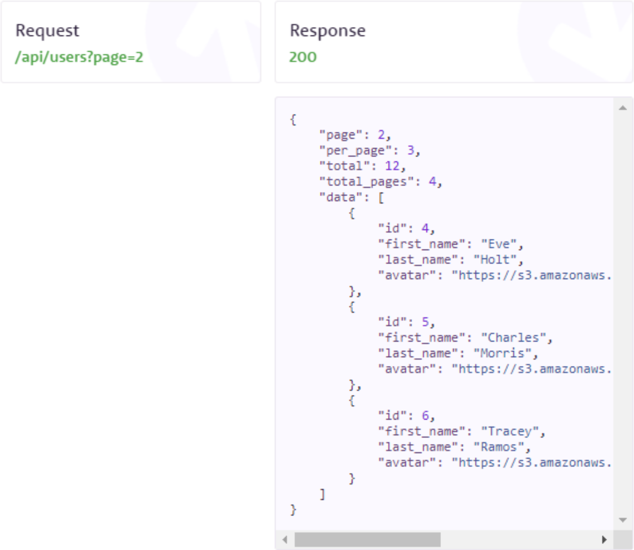
POST REGISTER - SUCCESSFUL

POST REGISTER - UNSUCCESSFUL

POST LOGIN - SUCCESSFUL

POST LOGIN - UNSUCCESSFUL

GET DELAYED RESPONSE



Request
/api/users?page=2

Response
200

```
{
    "page": 2,
    "per_page": 3,
    "total": 12,
    "total_pages": 4,
    "data": [
        {
            "id": 4,
            "first_name": "Eve",
            "last_name": "Holt",
            "avatar": "https://s3.amazonaws.
        },
        {
            "id": 5,
            "first_name": "Charles",
            "last_name": "Morris",
            "avatar": "https://s3.amazonaws.
        },
        {
            "id": 6,
            "first_name": "Tracey",
            "last_name": "Ramos",
            "avatar": "https://s3.amazonaws.
        }
    ]
}
```