# Runtime:

A software or framework or library or some instructions(most probably instructions which you did not specify yet being executed for smooth functionality) which are getting executed while your program is running

Ex: C++ runtime

# Utility software/tool:

A set of tools that help configure, analyse, and **optimize** computer resources to help a set of users to perform tasks efficiently

Ex: file manager, anti-virus

# Interface

It’s the way in which the software object interacts with outside world. Ports (Charging port, HDMI port) of a machine are the interfaces of that machine i.e., it exposes the functionality(charging port functionality is charging, HDMI port functionality is that, it enables media transfer from one machine to another) inside the machine to the outside world

## What is an interface in software

interface means exposing the functionality of a software to other pieces of same/another software

## API

API means application programming interface

A software intermediary that allows two applications to talk to each other

Ex: using PAYTM to book IRCTC tickets, using REDBUS to book the same IRCTC tickets

In the background, the work, you have done on PAYTM/REDBUS application is translated in a way that the IRCTC understands, here the TRANSLATOR is API

## REST API

Stands for Representational State Transfer i.e., a set of rules that developer follows when they create their API’s

REST is a pattern to make API’s, using that API, you can access resources, like image, posts, videos and so on…

They are about

1. Communication (between client and server)
2. **RESTFUL a service that uses REST API’s to communicate**

Benefits of REST API’s

1. Are Simple and standardized
2. Are Scalable and stateless
3. Has high performance (even if the service gets complex) and supports caching

Let’s take an example of an ice cream seller wanting a website, the rest API (endpoint) looks like this

<http://icecream.com/API/flavours>

**API** signifies API portion of the endpoint

**flavours** is the resource, i.e., we are working with flavours resource in this REST API

### The main building blocks of the REST API are

**Request**, that is sent by client (using instruments like web browser) to the server

a REQUEST block contains-

Header- special part which might have API key or some authentication data

Operations- post, get, put and delete operations

ENDPOINT- url

Parameters/body – data or the credentials that you want to send to server



**Response**, that is the response by the server which will be sent to the client

It will have json with a key value pairs

#### What are the operations that can be done with REST API

We can perform CRUD operations

C(reate)- POST

R(ead)- GET

U(pdate) - PUT

D(elete) - DELETE

#### What is in a response block

A response block will have the data that is sent as a response from the server to the client’s request

### An analogy of request and response

Let’s say you have a website where you manage the business of ice creams, and there is a flavours page where you list all (**GET request**) available ice creams and a particular flavour is out of stock, you want to update (**PUT request**) an old flavour with a new flavour (as a substitute), then the request and response objects look like this

A picture containing diagram

Description automatically generated

If you want to create a new flavour and add it to your website, you need to create a **PUT** request, then your objects look like this

## RESTFUL API

### What is web services

Web services provide a common platform that allows, multiple applications built using various programming languages to have the ability to communicate with each other

## Difference between REST and RESTful

REST is a set of constraints and RESTful is an API adhering to those set of constraints

## Types of web services

### Simple object access protocol (SOAP) web services

### RESTful web services

## RESTFUL API

A service which is built on REST architecture is called RESTful service, so the api built using REST architecture is called RESTful API

## Why RESTful API

It allows web applications that are built with various programming languages to communicate with each other

With the help of restful services, different applications can communicate even if the applications reside on different environments , i.e., one web application on linux and other on Windows

# JWT

Java web token

Used for authorization, not authentication.

Authentication – the process or action of proving or showing something to be true, genuine, or valid

ex: making sure the logged in user has given correct username and password

Authorization – having enough permission to do something

Ex: the user who has logged in above example, after logging in if he tries to access a service which needs some certain permissions, then checking whether he is having permissions, then that process is called authorization

## Normal session authorization (on left) versus JWT (on right)

Diagram

Description automatically generated

In jwt token is stored on client (browser) and server will add its own secret and serializes, so that

1. It don’t have to store token
2. As the server alone knows the secret, it is almost impossible to have security issues

# Github actions

An automation tool to automate tasks

Similar to Travis CI, Jenkins, GitLab CI/CD

Runs jobs when code changes

## Common uses

* Run unit tests
* Perform linting
* Handle deployment

## How it works

* Setting up triggers
  + Triggers can be anything that happens to your project
  + There are various trigger options, are available here: <https://docs.github.com/en/actions/using-workflows/events-that-trigger-workflows>

# Bootstrapping