**Core Java**

**What is Java?**

Java is a high level, robust, object-oriented and a secure and stable programming language but it is not a pure object-oriented language because it supports primitive data types like int, char etc.

**Application of Java**

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. Following are some other usage of Java :

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

### What is a Variable?

When we want to store any information, we store it in an address of the computer. Instead of remembering the complex address where we have stored our information, we name that address.The naming of an address is known as variable. Variable is the name of memory location.

In other words, variable is a name which is used to store a value of any type during program execution.

To declare the variable in Java, we can use following syntax

datatype variableName;

**Java Programming language defines mainly three kind of variables.**

1. Instance Variables
2. Static Variables (Class Variables)
3. Local Variables

## Instance variables in Java

Instance variables are variables that are declare inside a class but outside any method,constructor or block.

class Student

{

String name;

int age;

}

## Static variables in Java

Static are class variables declared with static keyword. Static variables are initialized only once. Static variables are also used in declaring constant along with final keyword.

class Student

{

String name;

int age;

static int instituteCode=1101;

}

## Local variables in Java

Local variables are declared in method, constructor or block. Local variables are initialized when method, constructor or block start and will be destroyed once its end.

float getDiscount(int price)

{

float discount;

discount=price\*(20/100);

return discount;

}

# Data Types in Java

Java language has a rich implementation of data types. Data types specify size and the type of values that can be stored in an identifier.

In java, data types are classified into two catagories :

1. Primitive Data type
2. Non-Primitive Data type

# Type Casting in Java

Casting is a process of changing one type value to another type. In Java, we can cast one type of value to another type. It is known as type casting.

# What is OOPS

OOPS is a programming approach which provides solution to real life problems with the help of algorithms based on real world. It uses real world approach to solve a problem.

**Main Features of OOPS**

* Inheritence
* Polymorphism
* Encapsulation
* Abstraction

## Java Class

In Java everything is encapsulated under classes. Class is the core of Java language. It can be defined as a template that describe the behaviors and states of a particular entity.

A class defines new data type. Once defined this new type can be used to create object of that type.

#### Java class Syntax

class class\_name {

field;

method;

}

### Java Object

Object is an instance of a class while class is a blueprint of an object. An object represents the class and consists of **properties** and **behavior**.

Properties refer to the fields declared with in class and behavior represents to the methods available in the class.

#### Java Object Syntax

className variable\_name = new className();

# Methods in Java

Method in Java is similar to a function defined in other programming languages. Method describes **behavior of an object**. A method is a collection of statements that are grouped together to perform an operation.

return-type methodName(parameter-list)

{

//body of method

}

# Access Modifiers in Java

Access modifiers are keywords in Java that are used to set accessibility. An access modifier restricts the access of a class, constructor, data member and method in another class.

Java language has four access modifier to control access level for classes and its members.

* **Default:** Default has scope only inside the same package
* **Public:** Public has scope that is visible everywhere
* **Protected:** Protected has scope within the package and all sub classes
* **Private:** Private has scope only within the classes

**JUnit**

Top of Form

Bottom of Form

JUnit is a unit testing framework for Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit, that originated with JUnit.

Testing is the process of checking the functionality of an application to ensure it runs as per requirements. Unit testing comes into picture at the developers’ level; it is the testing of single entity (class or method). Unit testing plays a critical role in helping a software company deliver quality products to its customers.

Unit testing can be done in two ways − manual testing and automated testing.

|  |  |
| --- | --- |
| Manual Testing | Automated Testing |
| Executing a test cases manually without any tool support is known as manual testing. | Taking tool support and executing the test cases by using an automation tool is known as automation testing. |
| Time-consuming and tedious − Since test cases are executed by human resources, it is very slow and tedious. | Fast − Automation runs test cases significantly faster than human resources. |
| Huge investment in human resources − As test cases need to be executed manually, more testers are required in manual testing. | Less investment in human resources − Test cases are executed using automation tools, so less number of testers are required in automation testing. |
| Less reliable − Manual testing is less reliable, as it has to account for human errors. | More reliable − Automation tests are precise and reliable. |
| Non-programmable − No programming can be done to write sophisticated tests to fetch hidden information. | Programmable − Testers can program sophisticated tests to bring out hidden information. |

## What is JUnit ?

JUnit is a unit testing framework for Java programming language. It plays a crucial role test-driven development, and is a family of unit testing frameworks collectively known as xUnit.

## JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress

## Features of JUnit

* JUnit is an open source framework, which is used for writing and running tests.
* Provides annotations to identify test methods.
* Provides assertions for testing expected results.
* Provides test runners for running tests.
* JUnit tests allow you to write codes faster, which increases quality.

## What is a Unit Test Case ?

A Unit Test Case is a part of code, which ensures that another part of code (method) works as expected. To achieve the desired results quickly, a test framework is required. JUnit is a perfect unit test framework for Java programming language.

A formal written unit test case is characterized by a known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition.

**JDBC**

**Introduction to JDBC**

java Database Connectivity(JDBC) is an Application Programming Interface(API) used to connect Java application with Database. JDBC is used to interact with various type of Database such as Oracle, MS Access, My SQL and SQL Server. JDBC can also be defined as the platform-independent interface between a relational database and Java programming. It allows java program to execute SQL statement and retrieve result from database.

**JDBC API**

JDBC API is mainly divided into two package. Each when we are using JDBC, we have to import these packages to use classes and interfaces in our application.

1. Java.sql
2. Javax.sql

### java.sql package

This package include classes and interface to perform almost all JDBC operation such as creating and executing SQL Queries.

### The javax.sql package

This package is also known as JDBC extension API. It provides classes and interface to access server-side data.

**Steps to connect a Java Application to Database**

The following 5 steps are the basic steps involve in connecting a Java application with Database using JDBC.

1. Register the Driver
2. Create a Connection
3. Create SQL Statement
4. Execute SQL Statement
5. Closing the connection

# Servlet: Introduction to Web

Web consists of billions of clients and server connected through wires and wireless networks. The web clients make requests to web server. The web server receives the request, finds the resources and return the response to the client. When a server answers a request, it usually sends some type of content to the client. The client uses web browser to send request to the server. The server often sends response to the browser with a set of instructions written in HTML(HyperText Markup Language). All browsers know how to display HTML page to the client.

## Web Application

# A website is a collection of static files(webpages) such as HTML pages, images, graphics etc. A Web application is a web site with dynamic functionality on the server. Google, Facebook, Twitter are examples of web applications.

# Introduction to Servlet

Servlet Technology is used to create web applications. Servlet technology uses Java language to create web applications.

Web applications are helper applications that resides at web server and build dynamic web pages. A dynamic page could be anything like a page that randomly chooses picture to display or even a page that displays the current time.

## Advantages of using Servlets

* Less response time because each request runs in a separate thread.
* Servlets are scalable.
* Servlets are robust and object oriented.
* Servlets are platform independent.

## GenericServlet class

In Servlet, GenericServlet is an abstract class. This class implements the servlet, ServletConfig and Serializable interface. This class provides the implementation of most of the basic servlet methods. The protocol of this class is independent as it can handle any type of request.

**Steps to Create Servlet Application using tomcat server**

To create a Servlet application you need to follow the below mentioned steps. These steps are common for all the Web server. In our example we are using Apache Tomcat server. Apache Tomcat is an open source web server for testing servlets and JSP technology. Download latest version of tomcat server and install it on your machine.

After installing Tomcat Server on your machine follow the below mentioned steps :

1. Create directory structure for your application.
2. Create a Servlet
3. Compile the Servlet
4. Create Deployement Descriptor for your application
5. Start the server and deploy the application

# Introduction to ServletRequest class

True job of a Servlet is to handle client request. Servlet API provides two important interfaces javax.servlet.ServletRequest and javax.servlet.http.HttpServletRequest to encapsulate client request. Implementation of these interfaces provide important information about client request to a servlet.

**WAR File in Servlet**

War stands for Web Application Resource or Web application Archive. A War file have the files of a web project. It can have files like servlet, xml, jsp, image, html, css etc.

# Managing Session in Servlets

Session Management is a mechanism used by the Web container to store session information for a particular user. There are four different techniques used by Servlet application for session management. They are as follows:

1. Cookies
2. Hidden form field
3. URL Rewriting
4. HttpSession

# Introduction to JSP

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain then a Servlet. JSP pages are opposite of Servlets as a servlet adds [HTML](https://www.studytonight.com/code/html/) code inside Java code, while JSP adds [Java code](https://www.studytonight.com/java/overview-of-java.php) inside HTML using JSP tags. Everything a Servlet can do, a JSP page can also do it.

JSP enables us to write HTML pages containing tags, inside which we can include powerful Java programs. Using JSP, one can easily separate Presentation and Business logic as a web designer can design and update JSP pages creating the presentation layer and java developer can write server side complex computational code without concerning the web design. And both the layers can easily interact over HTTP requests.

### Why JSP is preffered over servlets?

* JSP provides an easier way to code dynamic web pages.
* JSP does not require additional files like, java class files, web.xml etc
* Any change in the JSP code is handled by Web Container(Application server like tomcat), and doesn't require re-compilation.
* JSP pages can be directly accessed, and web.xml mapping is not required like in servlets.

### Advantage of JSP

* Easy to maintain and code.
* High Performance and Scalability.
* JSP is built on Java technology, so it is platform independent.

# JSP Directive Tag

Directive Tag gives special instruction to Web Container at the time of page translation. Directive tags are of three types: page, include and taglib.

# JSP Declaration Tag

We know that at the end a JSP page is translated into Servlet class. So when we declare a variable or method in JSP inside Declaration Tag, it means the declaration is made inside the Servlet class but outside the service(or any other) method. You can declare static member, instance variable and methods inside Declaration Tag.

# JSP Expression Tag

# Expression Tag is used to print out java language expression that is put between the tags. An expression tag can hold any java language expression that can be used as an argument to the out.print() method

# Implicit Objects in JSP

JSP provide access to some implicit object which represent some commonly used objects for servlets that JSP page developers might need to use. For example you can retrieve HTML form parameter data by using request variable, which represent the HttpServletRequest object.

.

**Mockito**

Top of Form

Bottom of Form

Mockito is a mocking framework, JAVA-based library that is used for effective unit testing of JAVA applications. Mockito is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing. This tutorial should help you learn how to create unit tests with Mockito as well as how to use its APIs in a simple and intuitive way.

## What is Mocking?

Mocking is a way to test the functionality of a class in isolation. Mocking does not require a database connection or properties file read or file server read to test a functionality. Mock objects do the mocking of the real service. A mock object returns a dummy data corresponding to some dummy input passed to it.

## Mockito

Mockito facilitates creating mock objects seamlessly. It uses Java Reflection in order to create mock objects for a given interface. Mock objects are nothing but proxy for actual implementations.

Consider a case of Stock Service which returns the price details of a stock. During development, the actual stock service cannot be used to get real-time data. So we need a dummy implementation of the stock service. Mockito can do the same very easily, as its name suggests.

## Benefits of Mockito

* No Handwriting − No need to write mock objects on your own.
* Refactoring Safe − Renaming interface method names or reordering parameters will not break the test code as Mocks are created at runtime.
* Return value support − Supports return values.
* Exception support − Supports exceptions.
* Order check support − Supports check on order of method calls.
* Annotation support − Supports creating mocks using annotation.

**Maven**

**Introduction to Maven**

Maven is a simple build automation tool which is basically used with java projects. Maven is also defined as a comprehensive project management tool. It aims to provide the developers a complete and detailed build life cycle framework of an application.

Maven also ease up the task of developers in checking a build status, generating reports (basically javadocs) and setting up the automated build process and monitor the same.

The ease of source code compilation, distribution, documentation, collaboration with different teams and other vital tasks are seamless by using maven.

Maven aims to describe 2 important things :

1. How a software is built.
2. The dependencies, plug-ins & profiles that the project is associated in a standalone or a distributed environment.

The maven can also be used in building & managing the projects written using C#, ruby and other programming languages as well.

### Advantages of Using Maven over Ant

* Managing dependencies.
* Convention over configuration - configuration is very minimal
* Multiple/Repeated builds can be achieved.
* Focus on automation.
* Plugin management.
* Testing - ability to run JUnit and other integration test suites.
* Making the development process transparent.
* Provision to check the status of each build.
* Avoiding inconsistent setups.

**What is POM?**

The easiest way to describe a POM in a maven project is, it is nothing but the core element of any maven project. Basically any maven project consists of one configurable file called pom.xml, which stands for the abbreviation Project Object Model. This pom.xml will always be located in the root directory of any maven project. This file represents the very basic and fundamental unit in maven.

The pom.xml basically contains the information related to the project which is built or to be built in. It contains all the necessary information about the configuration details, dependencies included and plug-ins included in the project. In simple, it contains the details of the build life cycle of a project.

Below are some of the configurations that can be handled in the pom.xml file :

* Dependencies used in the projects (Jar files)
* Plugins used (report plugin)
* Project version
* Developers involved in the project

## Super POM

All the maven projects pom.xml files always extends the super pom.xml file. This super pom.xml basically defines a set of default configurations which is shared and used by all the maven projects. This super pom.xml is not required to be written by the developers. This will come as a default one with the maven installation.

**What is Repositories in Maven?**

For maven to download the required artifacts of the build and dependencies (jar files) and other plugins which are configured as part of any project, there should be a common place where all such artifacts are placed. This common shared area is called as Repository in maven.

In maven, repositories are classified into 3 main categories as shown below :

1. Local Repository
2. Remote Repository
3. Central Repository

**Maven Plugins**

Plugin in maven is the one of the vital feature that is basically used to reuse the common build logic across different projects. Plugins are the ones through which all the tasks like compiling code, testing them with the junits, creating jar/war/ear files and documentation of the projects are carried out. Most of the work in maven is done using plugins, since the dependencies (jar files) are added only to the classpath while executing tasks.

So, maven is nothing but a plugin execution framework in which every tasks are accomplished by the usage of plugins.

Below are the basic list of tasks that are handled by plugins in maven :

* Creating jar/war/ear files.
* Code compilation
* Unit testing of the code.
* Project documentation

**GIT**

Git is a version control system that helps us to better manage our projects. A version control system is a software that tracks changes made to a project and stores all the modifications made to a project by storing different versions of it. Additional information like who made the changes and at what time are also tracked. Git is not the only version control system out there but it is by far the most popular and most widely used one. Any project managed using Git is called a Git Repository.

**What is Git used for?**

* A major use case of Git or any version control system is to compare the current version of a project to previous versions and eliminate complicated errors.
* Another reason why Git is widely used is for collaborating on the same project. This is done with the help of hosting services for Git Repositories like GitHub or BitBucket. We can create a remote repository on sites like GitHub and then push or upload our local Git repository on these sites and allow others to pull or download from this remote repository.
* Git makes it very easy to roll back changes in our projects by simply reverting to previous versions.

**How does Git work?**

* Git manages the different versions by maintaining a series of snapshots of the project. Whenever we want to commit or store a change permanently a new snapshot of our current version is taken and stored by Git.
* Some version control systems store the differences in the versions instead of storing the actual version but this proves to be inefficient as every time we revert to a version we will need to reconstruct that version of the project using all the previous differences.
* Each version of the project that is committed points to the version that was committed before it and refers to it as its parent. This makes reverting to previous versions easier.

### Repository

The place where Git permanently stores or "commits" these changes as different versions of our project. All the snapshots of the project versions are stored here along with some additional information.

## Pros and Cons

Even though Git seems like a perfect solution for many problems but it comes with its own advantages and disadvantages. Let's take a look at some of them:

### Advantages

* Promotes Open Source Softwares.
* Git works on a Distributed Model which means that all changes and updates to the repository will be visible to anyone who clones it. However, the owner has the control to make the repository public or private.
* Git works seamlessly on almost all the major platforms and is faster than other version control systems.
* Git provides excellent branching and merging features.

### Disadvantages

* Git provides mediocre support for binary files like images and multiple changes to such files make Git a bit cumbersome to use.
* Git has a steep learning curve and one needs to put in the time and effort to master it.
* Git is somewhat slower on the Windows platform as compared to other platforms.

## What is the .gitignore file?

* We would often have many files in our repository which may be autogenerated by our software or we may even have some personal files whose content we don’t want to share.
* The .gitignore file is a **text file** in which we can mention names or patterns to identify the files or folders that we want Git to ignore.
* By ignoring we mean that Git should not track any changes made to these files.
* We can create a single .gitignore file in our repository or multiple .gitignore files in different subdirectories of our project.
* We can also create a **global .gitignore file** that will ignore the mentioned files in all the Git repositories on that system.
* However, we have to commit the .gitignore file to make sure that if someone else clones this repository then that person also has the same ignore rules.

# Git Pull Request

When working in a team and collaborating on a project, we will be pushing changes to a central remote repository hosted on platforms like GitHub or Bitbucket. Pull request is a way of informing other team members about the changes that you have made. It is a way of requesting other team members to pull your changes and analyze them. Let's learn more about how pull requests work.

## What are Pull Requests?

* A pull request is used to inform other team members and collaborators of the changes that you have pushed to the remote repository.
* When a pull request is created, all the developers can view and discuss your changes and can add new follow-up commits if needed. It is a convenient way of getting your code reviewed by the project manager.
* We can also push incomplete branches and start a pull request to ask for help from other developers.
* Once the changes are verified and approved by the team leader or the project manager, they can be merged with the master branch. This is why they are also known as **Merge Requests**.
* A pull request serves a three-fold purpose. First, it informs other developers that you have pushed changes and started a pull request. Second, it provides a platform to discuss the proposed changes, and third, it allows developers to add **follow-up commits** if they feel like something is missing.

## How does a Pull Request work?

As discussed above, a pull request is made by a team member when he has pushed his/her code to the remote and wants to get it reviewed by others. A pull request can be made in two different scenarios. One, when the developer is a team member and working on a shared repository, and second when the developer wants to suggest changes to the authors of some other public repository. Let's discuss how to create a pull request in each of the above cases.

### Pull Request for a Shared Repository

When working and collaborating on a shared repository, we will first clone the repository to our local system, and create a new branch for our work. We can then directly push changes to the remote and create a pull request to inform other developers. If our work is approved by the project manager then the branch that we created can be merged into the main branch of the repository.

# Git Push

We won't always be working alone on a project on just our system and we often have to share our work with others or let others contribute and collaborate with us on a project. This is where **Git Push**and **Remote Repositories**come into action. Pushing a repository is like uploading it on a platform where other developers can see our project and its version history and even collaborate with us. In this tutorial, we'll see how to **push**our local repository content on a remote repository.

# Git Pull

When working in a team we need to make sure that others have access to our work and we have access to their work. Just like there is a **Git Push** command to share our work with others, we also have a **Git Pull** command to **download**other people's work from a remote repository. In this tutorial, we will learn more about Git Pull and how to use it.

**What is JPA?**

Spring Boot JPA is a Java specification for managing relational data in Java applications. It allows us to access and persist data between Java object/ class and relational database. JPA follows Object-Relation Mapping (ORM). It is a set of interfaces. It also provides a runtime EntityManager API for processing queries and transactions on the objects against the database.

## Why should we use JPA?

JPA is simpler, cleaner, and less labor-intensive than JDBC, SQL, and hand-written mapping. JPA is suitable for non-performance oriented complex applications. The main advantage of JPA over JDBC is that, in JPA, data is represented by objects and classes while in JDBC data is represented by tables and records. It uses POJO to represent persistent data that simplifies database programming. There are some other advantages of JPA:

**JPA Features**

There are following features of JPA:

* It is a powerful repository and custom object-mapping abstraction.
* It supports for cross-store persistence. It means an entity can be partially stored in MySQL and Neo4j (Graph Database Management System).
* It dynamically generates queries from queries methods name.
* The domain base classes provide basic properties.

## JPA Architecture

JPA is a source to store business entities as relational entities. It shows how to define a POJO as an entity and how to manage entities with relation.

The following figure describes the class-level architecture of JPA that describes the core classes and interfaces of JPA that is defined in the **javax persistence** package. The JPA architecture contains the following units:

## Object-Relation Mapping (ORM)

In ORM, the mapping of Java objects to database tables, and vice-versa is called **Object-Relational Mapping.** The ORM mapping works as a bridge between a **relational database** (tables and records) and **Java application** (classes and objects).

**Difference between JPA and Hibernate**

JPA: JPA is a Java specification that is used to access, manage, and persist data between Java object and relational database. It is a standard approach for ORM.

Hibernate: It is a lightweight, open-source ORM tool that is used to store Java objects in the relational database system. It is a provider of JPA. It follows a common approach provided by JPA.

The following table describes the differences between JPA and Hibernate.

|  |  |
| --- | --- |
| JPA | Hibernate |
| JPA is a Java specification for mapping relation data in Java application. | Hibernate is an ORM framework that deals with data persistence. |
| JPA does not provide any implementation classes. | It provides implementation classes. |
| It uses platform-independent query language called JPQL (Java Persistence Query Language). | It uses its own query language called HQL (Hibernate Query Language). |
| It is defined in javax.persistence package. | It is defined in org.hibernate package. |
| It is implemented in various ORM tools like Hibernate, EclipseLink, etc. | Hibernate is the provider of JPA. |
| JPA uses EntityManager for handling the persistence of data. | In Hibernate uses Session for handling the persistence of data. |

## Advantages of Hibernate Framework

### ) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

### 2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.

### 3) Database Independent Query

**What is the CRUD operation?**

The CRUD stands for Create, Read/Retrieve, Update, and Delete. These are the four basic functions of the persistence storage.

The CRUD operation can be defined as user interface conventions that allow view, search, and modify information through computer-based forms and reports. CRUD is data-oriented and the standardized use of HTTP action verbs. HTTP has a few important verbs.

* POST: Creates a new resource
* GET: Reads a resource
* PUT: Updates an existing resource
* DELETE: Deletes a resource

CRUD operations are at the foundation of the most dynamic websites. Therefore, we should differentiate **CRUD** from the **HTTP** **action verbs**.

Suppose, if we want to **create** a new record, we should use HTTP action verb **POST**. To **update** a record, we should use the **PUT** verb. Similarly, if we want to **delete** a record, we should use the **DELETE**verb. Through CRUD operations, users and administrators have the right to retrieve, create, edit, and delete records online.

**Web Services**

**What are Web Services?**

Web Services are client and server applications that communicate over the World Wide Web's (WWW) Hypertext Transfer Protocol (HTTP). As described by the World Wide Web Consortium (W3C), web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks.

Web Services can be looked as a *code on demand*. Just like we call functions and methods, web services can be looked upon as calling a function or method over the internet using some sort of protocols and some agreements.

Types of Web Services: SOAP and REST

There are two types of web services:

1. SOAP Web Services
2. REST Web Services

## SOAP Web Services

SOAP is an XML-based protocol. The biggest advantage of using the SOAP Web Service is its own security. SOAP stands for Simple Object Access Protocol.

SOAP provides an envelope to send a web services messages over the Internet, using the HTTP protocol. The messages are generally in XML format.

## REST Web Services

The REST stands for Representational State Transfer. REST is not a set of standards or rules, rather it is a style of software architecture. The applications which follow this architecture are referred to as RESTful

Unlike SOAP which targets the *actions*, REST concerns more on the resources. REST locates the resources by using URL and it depends on the type of transport protocol(with HTTP - GET, POST, PUT, DELETE,...) for the actions to be performed on the resources. The REST service locates the resource based on the URL and performs the action based on the transport action verb. It is more of architectural style and conventions based.

## Basic Features of the HTTP

As we have already learnt HTTP is a protocol which allows us to send files back and forth on the web, which involves a client and a server. HTTP is text based, which makes it easier to monitor.

The basic features of HTTP are:

1. HTTP is connectionless.
2. HTTP is media independent, which means any type of data can be sent through the http.
3. HTTP is stateless, neither the server nor the client keeps a track of the last request.

# Understanding the REST API Response

As the REST API's response is consumed by some application and not the browser, so we don't have to worry about styling it to make it look good.

In case of API response, it can be simple XML or JSON or any other media type. Also, at times, one REST API is being consumed by different applications. Out of which, one application might need the response in the form of XML and other might need the response in the form of JSON. We can develop such REST API, programming them to send the response according to the input header of the HTTP request.

There is a Media-Type attribute in the header which can be used in such cases and the response can be sent accordingly. We will learn how to do this in coming tutorials but for now lets concentrate on the design aspects of the REST API

# HATEOAS - Important Concept For REST API

HATEOAS stands for Hypertext As The Engine Of Application State. Now what does it mean and why this is important. It's a well agreed belief that a good written software is the one which has a good documentation. But there is little deviation in this belief when it comes to the Rest API. A Rest API is said to be perfect if it doesn't need any documentation at all. It should be so crystal clear to the developer who is consuming the service that he/she should not even refer to the documentation. It should be designed like a website, like once we go to the Home page we can navigate to the different resources with the help of hyperlinks.

Now the question is how to achieve this? The simple answer is providing as much related links as possible about the resource in the response. Lets take the case of our registration application.

# How RESTful is your API - Richardson Maturity Model

Since REST is just a specification or we can say a way of implementing web services, with no hard coded documentation. So how we know the API which we have developed is fully RESTful. For this, there is Richardson Maturity Model, which is usually referred to analyse how RESTful the API is.

The Richardson Model is a way to grade your API according to the constraints of the REST. The more the API follows these constraints the more RESTful the API is.

The Richardson Maturity Model has 4 levels numbered from 0 to 3. Level 0 is not RESTful while Level 4 means your API is fully RESTful.

## Level 0

This will use only one URI to expose the whole API and generally uses only one HTTP method. If we take example of our registration API we see that that every resource has its own URI. The student resource has a different URI, the course resource has a different URI and the registration resource has a different URI.

SOAP based web services are the example of this level. This level of API doesn't leverage the full concepts of the HTTP specification. For example: Every request will be POST request, so if someone wants to delete a student record they will have send the action also in the request.

## Level 1

In this level every resource has its own URI. This level uses multiple URI where every URI is an entry point to a specific resource. Still this level uses only POST method for operation.

## Level 2

This level says that your API should use protocol properties as much as possible. You must not use single POST method for all the operations, but make use of GET when you are requesting a resource, use a DELETE method when you want to delete a resource.

The use of proper response codes is also essential. Use 201 Created response code when a post request comes and resource is created. Use 200 Ok response for the get request and so on.

## Level 3

This level suggests to use the concept of HATEOAS. The response should contain the logical links of all the resources which the current resource is related to.

For Example: In case of when the client request for the student information with the roll number 1, then in the response along with the response of the student, a link of all the URI of all the courses which that student has registered should also be sent.

If an API is at the level 3 this will be known as fully RESTful. This Richardson Model should be used while designing the REST API to make sure the web services are fully RESTful.

**Spring Framework**

**Spring Introduction**

Spring is a Java-based application framework that is designed and developed by the Pivotal Software Company.

In this tutorial, we will talk about Spring 5 which is the latest and more improved version of the conventional Spring framework.

Spring is an application framework that is used to create Enterprise Applications. We can create web-based applications easily due to its vast library and tools.

Spring provides an easy and friendly environment to create Java enterprise applications. It is [full of features](https://www.studytonight.com/spring-framework/spring-features) and provides various other sub-projects such as Spring Security, Spring Boot, Spring MVC, Spring Cloud, Spring Data, etc that help to build applications accordingly.

## Spring 5 has been improved over time, in the early days of Java EE and Spring, we pring Modules

The Spring Framework is divided into several modules based on their services. These modules are:

* **Spring Core Container:** It is the core module of the Spring that provides containers like BeanFactory and ApplicationContext.
* **Inversion of Control:** It is also known as dependency injection and used to configure application components and lifecycle management of Java objects.
* **Aspect-Oriented Programming:** This module enables implementing cross-cutting concerns inside the Spring framework such as transaction management, remote access, etc.
* **Data Access:** It helps with working with database systems by using Java Database Connectivity (JDBC) and ORM (Object-Relational Mapping) tools.
* **Model View Controller:** It is also known as the MVC model that helps to create web-based applications and RESTful Web services.
* **Authentication And Authorization:** It is used to configure security processes within the framework by using the Spring Security (a sub-project of Spring).
* **Messaging:** Spring uses a message listener object to convey the message by using JMS (Java Message Service) which is the improvement of JMS API.

deploy applications to an application server but now with the help of Spring Boot we can create applications in a DevOps and cloud-friendly way.

The core and heart of Spring is an IOC container that manages bean objects and allows dependency injection. We will discuss these later in our tutorial.

Spring is built with several components(modules) to work with the web, database, network, etc. Below is the image of Spring Runtime that shows its internal architecture.

# Spring Features

Java Spring Framework is full of features and provides and helps to create Java-based scalable applications. Here, we are discussing some features. Although these are not limited as spring provides dozens of variety of projects such as Spring Data, Spring Cloud, Spring Boot, etc. The following are the Features of the Spring Framework.

* Flexible
* Productive
* Fast
* Secure

# Spring IOC Container

Spring IoC Container is a core part of the Spring framework which is used to manage the application bean. It injects dependencies when a bean is created and **manages the bean life cycle** during execution.

The fundamental tasks of Spring IoC are:

* Instantiating
* Configuring, and
* Assembling Bean

The IOC container gets configuration related information from the Spring configuration file. That can be either **XML** or **Java** files.

The container uses **dependency injection** to manage the components that make up an application.

Spring provides two types of IOC containers:

* BeanFactory
* Application Context

# Spring Bean Scope

Bean Scope refers to the lifecycle of a bean, visibility of a bean, how long does the bean live, how many instances are created, how is the bean shared?

Bean's default scope is a singleton. The spring container will create a single instance of the bean. It is cached in memory. All requests for the bean will return a shared reference of the same bean.

Spring provides @Scope annotation to mark a bean scope.

# Spring Constructor-Based Dependency Injection

In this topic, we are using the constructor-based dependency injection technique to inject values through the constructor but before moving further let's first understand what is Dependency Injection(DI).

**Dependency Injection** is a technique by which an object defines its dependencies. The IOC container then injects these dependencies during bean creation. This process is fundamentally the inverse and known as Inversion of Control as well. Dependency Injection makes our code loosely coupled. It is classified into two major categories Constructor-based dependency injection and Setter-based dependency injection. Here, we will discuss Constructor-based DI with an example.

**Jenkins**

1. What is Jenkins?
2. Jenkins is an open source automation server which enables developers around the world to reliably build, test, and deploy their software. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.
3. What is Continuous Integration (CI)?
4. Continuous integration is a process in which all the development activities (every day activities) are integrated at a given point of time by compiling and building the project and well tested. The basic idea behind the continuous integration is to ensure that there are no compilation issues at the end of the day by numerous check ins made by the developers in a team. Also, this would enable to identify any compilation issues at the early stages of the development process.

In this process, all the developers activities are collaborated and merged at the central system (repository at which all the check ins are made). The main aim again here in this process is to eliminate the "integration problems". Each and every integration is automatically built, deployed and tested thoroughly.

1. Advantages of CI

* Reduced level of risk.
* No longer integration process.
* Lot of bad code smells & bugs can be reduced at the early stages of the development process.
* Frequent deployment process is easier and faster.

## What Jenkins does?

The primary job of the Jenkins is to execute a list of steps configured which is based on a trigger. Below are the list of steps/tasks performed by Jenkins when triggered :

* Perform the code compilation & build the software with ANT, Maven or Gradle.
* Run an internal shell script.
* Archive the resultant builds
* At last start the execution of the integration tests.
* Monitoring the execution of the above tasks.
* Provision to stop the build in case of failure in any of the steps.
* Notifying the user about the success or failure of each build.

# How to Configure Jenkins Build Server

* Download the latest Jenkins.war file.
* Deploy the Jenkins.war file in the local web server of our machine. E.g. Tomcat
* Start the server
* Click on the create new jobs link to configure a project for build automation. Enter the details of the project.
* In the next page, fill all the required details like - description, path of the project's pom.xml and other details and click save.
* A project/job is created in Jenkins
* Click on the Build now option as shown above. This will read the pom.xml and pull out the latest code and executes a build process and generates the jar/war file accordingly.
* Click on the build number link and it opens up the build details. Click on the console output to see the logs of the build.

**Docker**

**Introduction to Containers and Docker**

In order to understand what Docker is and why it is used, we must first know what containers are and what problems do they solve. Containers are completely isolated environments that are set up on top of an existing operating system to provide a virtual separation for the application running inside it from the outside world. And, docker is a software that helps us control the lifecycle of these containers.

Don't worry too much if you are not able to understand it. Let's take a simple everyday life example to help you understand the concept of containers.

Consider a tenant living on rent in a house which he/she shares with the landlord's family or let's say another tenant. The tenant has a separate room, bathroom and kitchen for fulfilling his basic needs. Now if we consider the house to be an operating system and the area available to the tenant as a container then the tenant lives alone in his area, manages it as per his own requirements, has no interference from others but it shares some common resources of the house like electricity, water etc.

Similarly, a container is an isolated environment(user space) set up on top of an operating system, wherein it utilizes the same OS Kernel but has its own processes, services, networking, storage mounts just like a virtual machine, but containers are not a virtual machine(We will learn the difference between VM and containers soon.).

Docker software helps us in managing the lifecycle of these containers which includes setup of containers, monitoring containers, destroying containers, attaching containers to network, etc.

Containerisation is an old concept as there are technologies that enable container setup and have been in use for more than 10 years. Some other ways to create containers are:

1. LXC
2. LXT
3. LXCFS etc.

## What is Docker or Docker Engine?

Docker or Docker Engine is a software which helps us in managing the lifecycle of containers, define how they will be set up, what applications/software/services they will run inside them, their networking requirements, their storage requirements, and if required how to easily destroy a container and start afresh.

Docker uses **docker images** to run processes inside the container. We will learn about docker images in detail later on, for now, consider them as files required to install any service inside a docker container.

## Why use Docker?

Due to increasing demand of microservices and devops, docker has become very popular in the software industry these days as it helps developers and system admins to build and run applications in containers. Here are some of the things which make docker so popular:

# Container vs. Virtual Machine

Docker containers and virtual machines are absolutely different in their architecture but are often confused to be similar. In this tutorial, we will see the difference between a docker container and virtual machines by understanding their architecture and how they work on top of the host operating system.

We have already covered how docker container works, **utilizing the same OS Kernel** with resource limitation done using the **cgroups** and **namespaces** to decide the boundaries of a docker container. Let's dig a bit deeper to understand what all this means.

## Virtual Machine

Unlike a container, a virtual machine **doesn't utilize the host OS Kernel**, on the contrary, it **installs its own guest Operating system**. The virtual machine also **emulates the hardware layer**, which virtually utilizes the host resources through a hypervisor.

**Microservices**

**What are Microservices?**

Definition: According to Sam Newman, "Microservices are the small services that work together."

According to James Lewis and Martin Fowler, "The microservice architectural style is an approach to develop a single application as a suite of small services. Each microservice runs its process and communicates with lightweight mechanisms. These services are built around business capabilities and independently developed by fully automated deployment machinery."

There is a bare minimum of centralized management of these services, which may be written in different programming language and use different data storage technologies.

Points to remember

* These are the services which are exposed by REST.
* These are small well-chosen deployable units.
* The services must be cloud-enabled.

The microservice defines an approach to the architecture that divides an application into a pool of loosely coupled services that implements business requirements. It is next to Service-Oriented Architecture (SOA). The most important feature of the microservice-based architecture is that it can perform continuous delivery of a large and complex application.

Microservice helps in breaking the application and build a logically independent smaller applications. For example, we can build a cloud application with the help of Amazon AWS with minimum efforts.

## principles of Microservices

There are the following principles of Microservices:

* Single Responsibility principle
* Modelled around business domain
* Isolate Failure
* Infrastructure automation
* Deploy independently

### Single Responsibility Principle

The single responsibility principle states that a class or a module in a program should have only one responsibility. Any microservice cannot serve more than one responsibility, at a time.

### Modeled around business domain

Microservice never restrict itself from accepting appropriate technology stack or database. The stack or database is most suitable for solving the business purpose.

# Advantages of Microservices

* Microservices are self-contained, independent deployment module.
* The cost of scaling is comparatively less than the monolithic architecture.
* Microservices are independently manageable services. It can enable more and more services as the need arises. It minimizes the impact on existing service.
* It is possible to change or upgrade each service individually rather than upgrading in the entire application.
* Microservices allows us to develop an application which is organic (an application which latterly upgrades by adding more functions or modules) in nature.
* It enables event streaming technology to enable easy integration in comparison to heavyweight interposes communication.
* Microservices follows the single responsibility principle.
* The demanding service can be deployed on multiple servers to enhance performance.
* Less dependency and easy to test.
* Dynamic scaling.
* Faster release cycle.

## Disadvantages of Microservices

* Microservices has all the associated complexities of the distributed system.
* There is a higher chance of failure during communication between different services.
* Difficult to manage a large number of services.
* The developer needs to solve the problem, such as network latency and load balancing.
* Complex testing over a distributed environment.

# Microservices Monitoring

Monitoring is the control system of the microservices. As the microservices are more complex and harder to understand its performance and troubleshoot the problems. Given the vivid changes to software delivery, it is required to monitor the service. There are **five** principles of monitoring microservices, as follows:

* Monitor container and what's inside them.
* Alert on service performance.
* Monitor services that are elastic and multi-location.
* Monitor APIs.
* Monitor the organizational structure.

These principles allow us to address technological changes associated with the microservices and organizational changes related to them.

## Microservices Monitoring Tool

There are three monitoring tools are as follows:

* Hystrix dashboard
* Eureka admin dashboard
* Spring boot admin dashboard

## Microservice Virtualization

Microservices virtualization is the method to simulate the behavior of specific components in various component-based application like cloud-based application, SOA, and API driven architecture. Service virtualization also reduces cost and save time. By combining service virtualization, an organization can develop the application which can be delivered from various locations and dissimilar environments