JEE Intro

I know you have been waiting a long time to understand what is there in the world of **J2EE** and why we must learn it.

You may be wondering that you know core Java, not only that, you now know how to take a core java program and connect it to a database using the concepts of JDBC.

Is this not enough for me to develop real-world applications? My friend yes, it is enough, but there are certain limitations concerning these features

What are the limitations?

Let us assume using these features you develop an application and right now this application that you've developed is on a certain computer. The computer where you typed the code, stored the code, compiled the code, is on the same computer, you have to execute it and you execute it using the console or the command prompt. Through the console, you type in your commands, and then the program begins execution and a program which you access via a console is only called a **Console Based Application** or another word for it is **Standalone Application**.

Now, you may be wondering, why is it called a standalone?

It is called a standalone because it is such an application which can only be accessed from the computer in which the application is present. For example, Let's assume you have written a Java program and it is present on your computer. If you want to execute your Java program, you need access to your computer, no one else can access it.

Let us assume there's another computer. Can your friend from his computer access the application present on your computer? Not possible. But some of you are thinking that there is something called as your **remote computer access application** such as your **TeamViewer**, **AnyDesk** and there are many such applications where your friend can install it and then through his computer, he can access your computer, but then again I want you to understand it is still standalone only because first of all, he is not accessing his computer's application through his computer he is accessing your computer and while he is accessing your computer, only he can access it. Even if you want you cannot access it because it is a standalone application. If he is moving his mouse and you move your mouse there will be a conflict when he is typing and you also type there will be a conflict.

So, an application is called a standalone application when,

- The software is present in the computer.
- It can only be accessed by a single person at any given point in time.
- It can only be accessed on the same computer and nowhere else.

And many beautiful standalone applications have made our life that much more productive and that much more efficient. To give you some examples. A few of my most famous standalone applications come from the **Adobe suite of applications** such as your **Photoshop, Premiere Pro, After Effects, Illustrator** using which I'm able to create such amazing content for you guys.

Another set of applications that I use daily which is standalone is **Microsoft's Office suite** of applications, which includes your **Word, PowerPoint, Excel**.

Now, these applications have to be installed on my computer, and only through my computer, I can access them which means by nature they are standalone. But as days passed, when the internet began a revolution in the world, when the entire world started to get interconnected because of the invention of the internet. Now, software companies realized that if in the internet world, they must make it large, a different kind of application must be born, a different breed of applications must be created.

Now, what is this application?

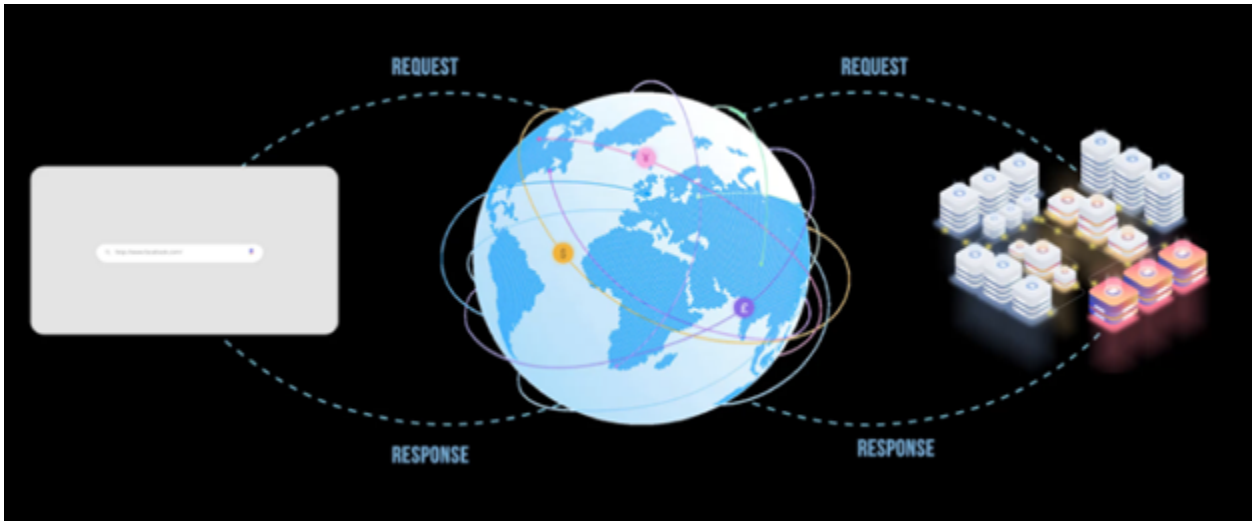
These applications are called **web applications.** Now, the concept of a web application is very simple, there is a computer called the **Server,** and in the server computer is where the code is present, your software is present, your application is present. Of course, if you want, you can open up the terminal in the server and you can execute it there, no doubt about it, you can execute it. But the beauty is that through the internet, this server can now be connected to 1000s of computers somewhere else in the world.

Now let us focus on a single computer.

Now, this computer wants to access the application present within the server. Now how is it going to do that? Well, I am not going to install any remote access application like any desk, or TeamViewer. But, I am now going to install a software called the **browser**. And you know, there are a variety of options for browsers, different companies come with different browsers.

I'm going to install a browser and through the browser, using a concept called a **URL[*Uniform Resource Locator*]**. I am now going to request to:

- Find this computer on the internet.
- And once the request reaches this particular computer,
- This computer must now respond to the request by sending a response through the internet back to the browser.



And whatever response is sent. The browser must render that response to display that response.

For example, if it is a Facebook application or a Facebook server that I'm requesting, the login page of Facebook will be visible.

However, these big names like Zomato, BookMyShow, etc., that you hear about are web applications, which anyone from anywhere in the world can access provided they have an internet connection and a browser. But the beauty is you may ask how is this different from me taking remote access to my friend's computer and accessing it?

It's very different because remote access if you would have taken at any given point of time only one person can request to access the application. But the beauty is through this **request-response protocol**, which we call **HTTP[*Hypertext Transfer Protocol*]** and by making use of advanced software tools such as the browser, lakhs of computers, at the same time can request the server and the server can respond to them. That is why when you are watching a movie on Netflix, so can I, my neighbor, your friend, your mom, your dad, millions of people in the world can access it at the same time. Why all that when you guys are using the learning management system here at *TapAcademy* when you are executing your software or your programs, there are 1000s of other students also executing at the same time.

How is it possible?

Because even that is a web application and you did not install it on your computer.

You are connecting to a server that is executing and sending you back the output operable to think. So, web applications are the need of the world today. And it has been the need of the world for the last close to two decades now.

So clearly coming back. If you look at Java, Java and JDBC put together can only and only create standalone applications for me but for me, the need of the world is not standalone applications, it is web applications, but web applications cannot be created using these concepts. This means there, Java's capabilities must increase, such that using Java code, I can create web applications.

Now, how, you may ask me?

Well for that, developers sat, worked overtime, and added new features into Java, such as a **Servlet**, **JSP**. And using these things, they said, you can now work on web applications using Java, you can now work on the ***request-response protocol*** using Java. And all these concepts which allow you to do those two together are what people refer to as the second version of Java.

And **EE** stands for web applications and web applications are also known as **Enterprise Applications**. Enterprise is just another word for business, you want to create such applications which help you to do business online, they're called enterprise applications and hence, they call it Enterprise Edition. Some people, in short, remove the **2** and call it **JEE**, which means naturally, if you and I want to become an all-rounder kind of Java programmer who is capable of not only working on standalone but also web applications, we must now invest sufficient time in understanding **J2EE (*or in short* JEE)**

These are the things which we must first learn:

- Web Server
- Web Container
- Servlets
- JSP
- Modern View Controller
- Hibernate
- Spring