**Servlets**

Pre-requisites:

Should have basic knowledge on OOPs concepts and Java Programming in general.

Agenda

* Introduction
  + Introduction of Servlet and web world in general.
* Servlet Architecture
* Servlet lifecycle
  + Different life cycle of a method is.
* Request and Response
* Being a Web Container
  + Discussion about Servlet Engine
* Session management
  + How it takes place in Servlet world.
* Overview of JSP
* JSP Elements
* Demo
  + At the end of the session, I will show how to code, develop and execute servlet.

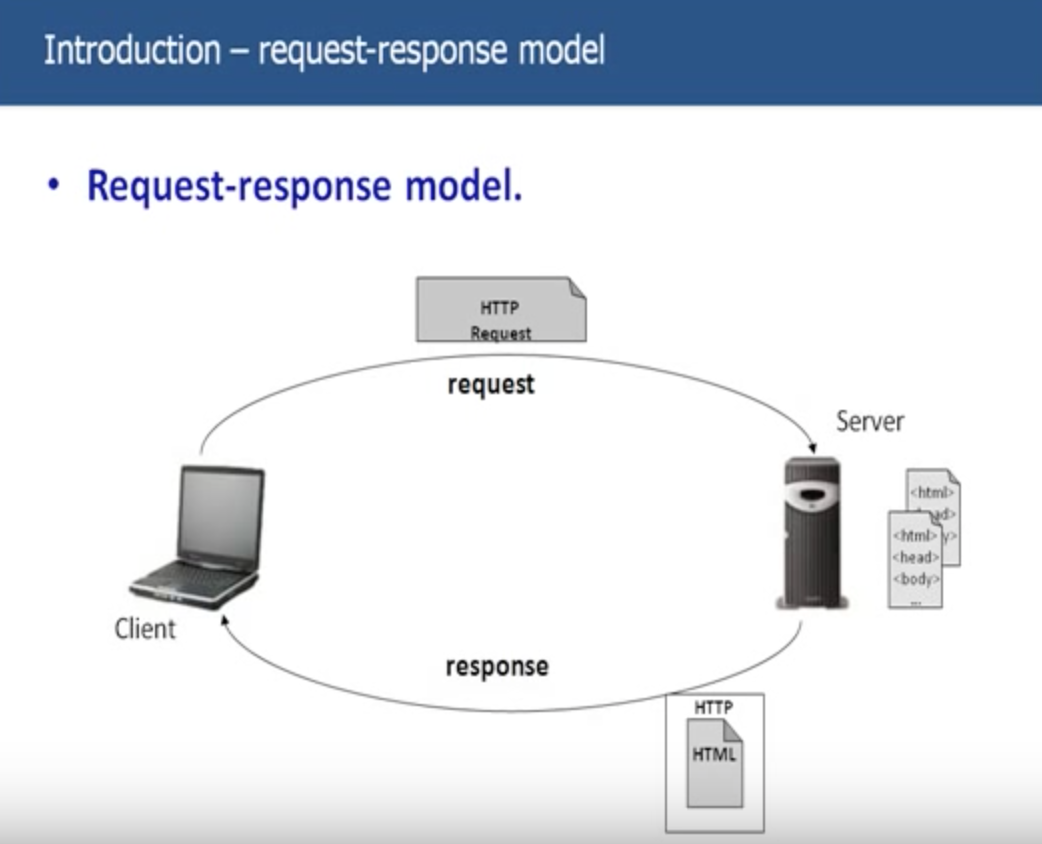
By end of this session you should know:

* what a servlet is?
* where servlets are used?
* How to use a servlet?
* What are the advantages of using a servlet?
* You should also able to develop and execute your own servlet apps.

**Introduction**

**Request-response model.**

Let us discuss how communication takes place in web world.



**Web Server is a server or a machine which can take in a HTTP Request over the web and can serve the client with the response. And this response is nothing but a static page.**

Let’s look into more details about HTTP Request and HTTP Response:

Key elements of request are

* HTTP method:

We will discuss about this HTTP methods and difference between the various methods we have in later part of the session. But for now, you just keep in mind that HTTP method is one of the parameters sent from the client to server when making a request for a resource. And this method basically would be GET and POST most of the methods.

* The page to access(a URL): When you access yahoomail.com [www.yahoomail.com](http://www.yahoomail.com) is a URL and you are sending URL to the server.
* Form Parameters:

Ex: You typed yahoomail.com, request went to the server and then server sends back the response to you. And that response is nothing but the login page for the yahoo mail.

In second page, you see username and password and these are text boxes. After entering the details, click on go. Form Parameters are nothing but username and password which you have typed.

HTTP request contains Username and Password in the Form Parameters.

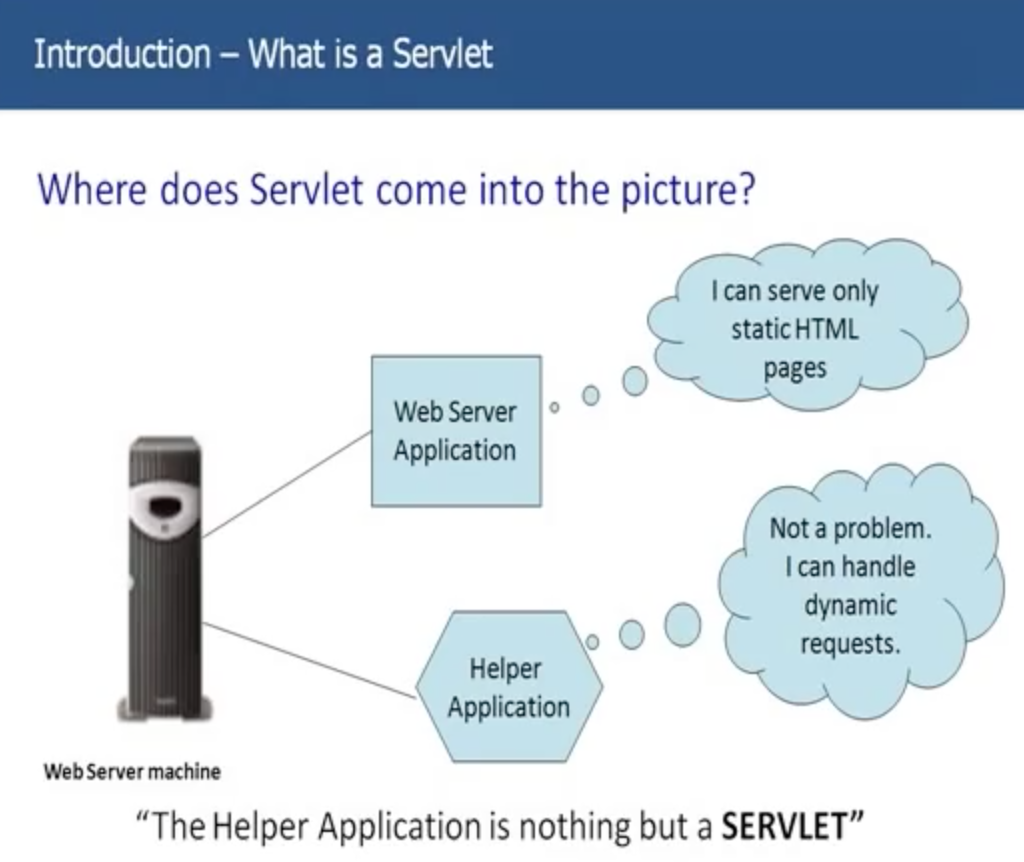
Key elements of response are

* A status code: This status code will tell the client whether the request was successful or not.

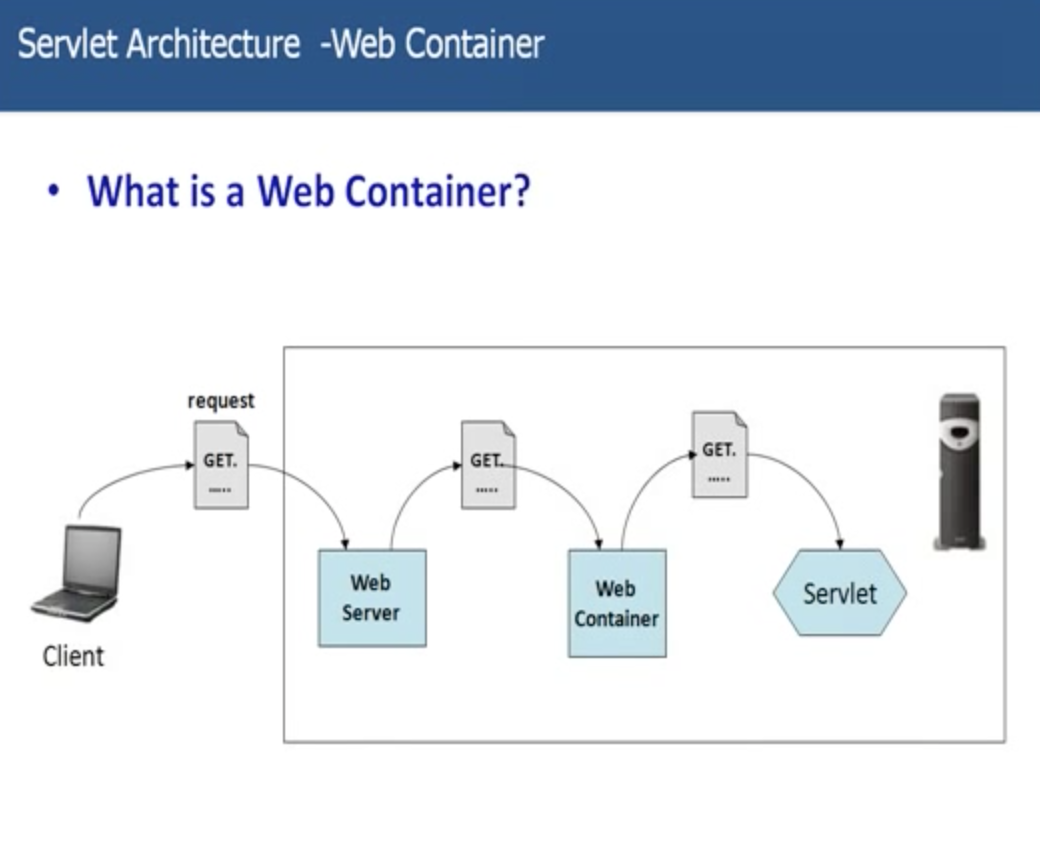
200 – Successful

404 – Page Not found

* Content-type:
* The content: The actual content the server is sending back to client.

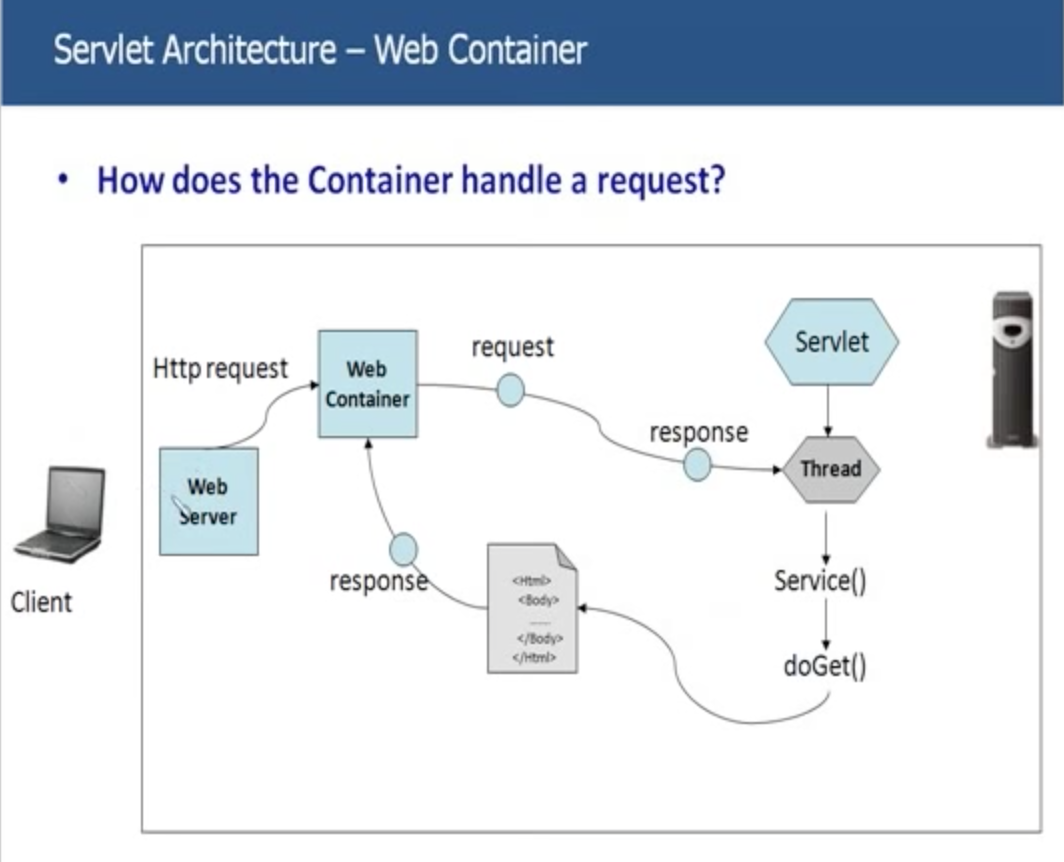


**Let’s see how the flow goes from Client to Server**



**How does the web server communicate to Servlet?**

**How does the Container handle a request?**

****

**What is the role of Web Container?**

Container is nothing but a Servlet engine in which Servlets and JSPs as well live and die. It is responsible for everything that happens to a Servlet.

* Communication Support
* Lifecycle Management:
  + As I told you, Servlet has got some callback methods, its got some methods to initialize, it got methods to destroy the servlet. All these methods were invoked by container. And it knows when to invoke which method. Basically, web container is responsible for creating a Servlet, invoke methods on the Servlet, execute the Servlet and ultimately even destroy the Servlet.
* Multi-threading support:
  + As I told you, for every request comes in for that particular servlet, web container sponsor a thread i.e., separate thread for every request comes in for the Servlet.
* Security:
  + Since, client directly cannot talk to Servlet, it has to go through Web Container. So you can put all kind of security code in that container. So that, only valid request will go to the Servlet. Any other invalid request will not have taken over to the Servlet.
* JSP Support

**how does the web container know which Servlet can serve the request requested by the client?**

Servlet can have 3 names:

* Client known URL name: url-pattern
* Deployer known secret name: servlet-name
* Actual file name: servlet-class

