Java & JEE Training

Day 26 – Servlets – A deeper look

MindsMapped Consulting

Java & JEE Training

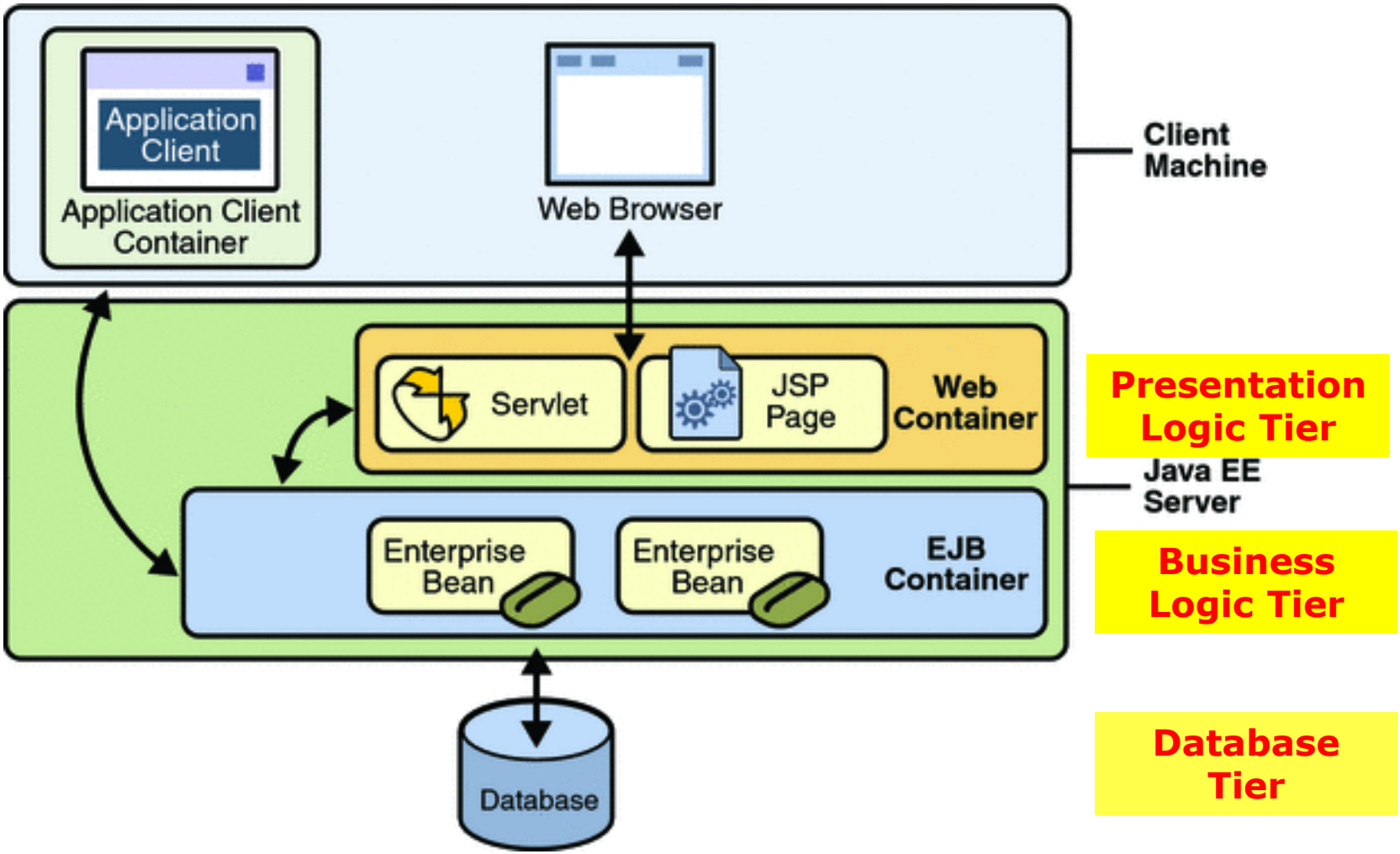
Introduction to JEE

MindsMapped Consulting

Review of previous session…

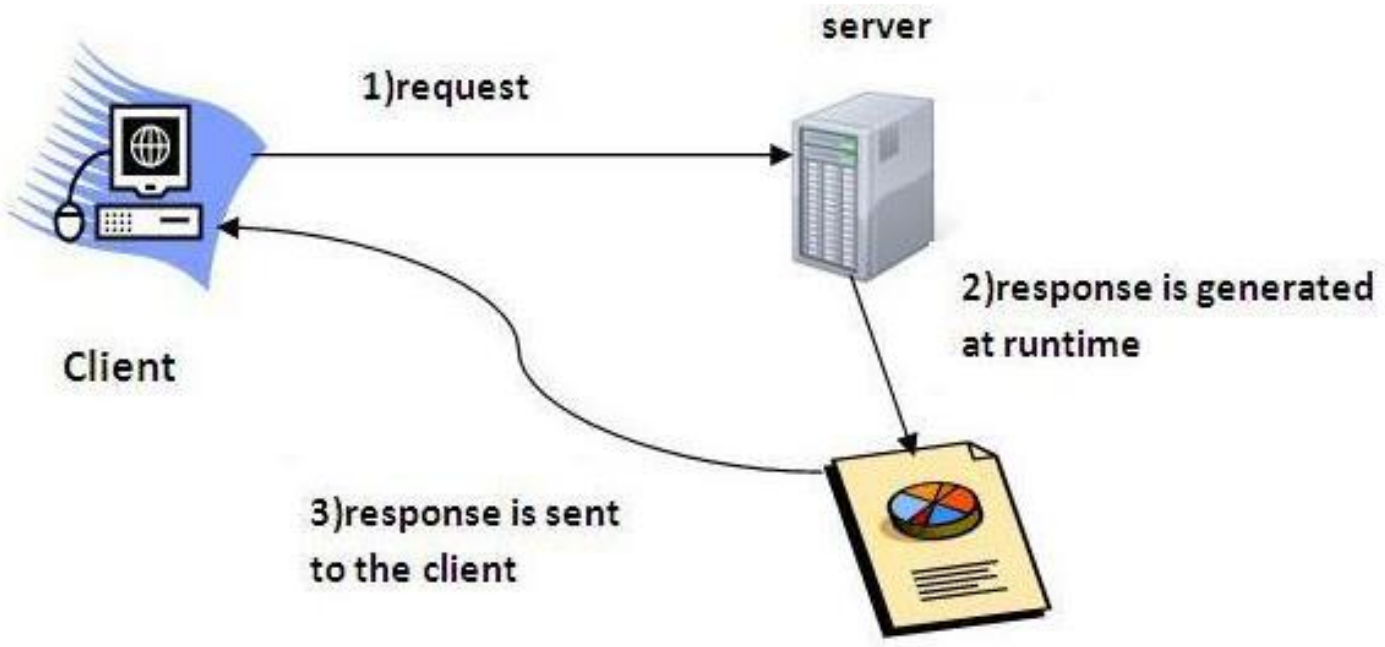
….

Enterprise Java – Multi-tier architecture



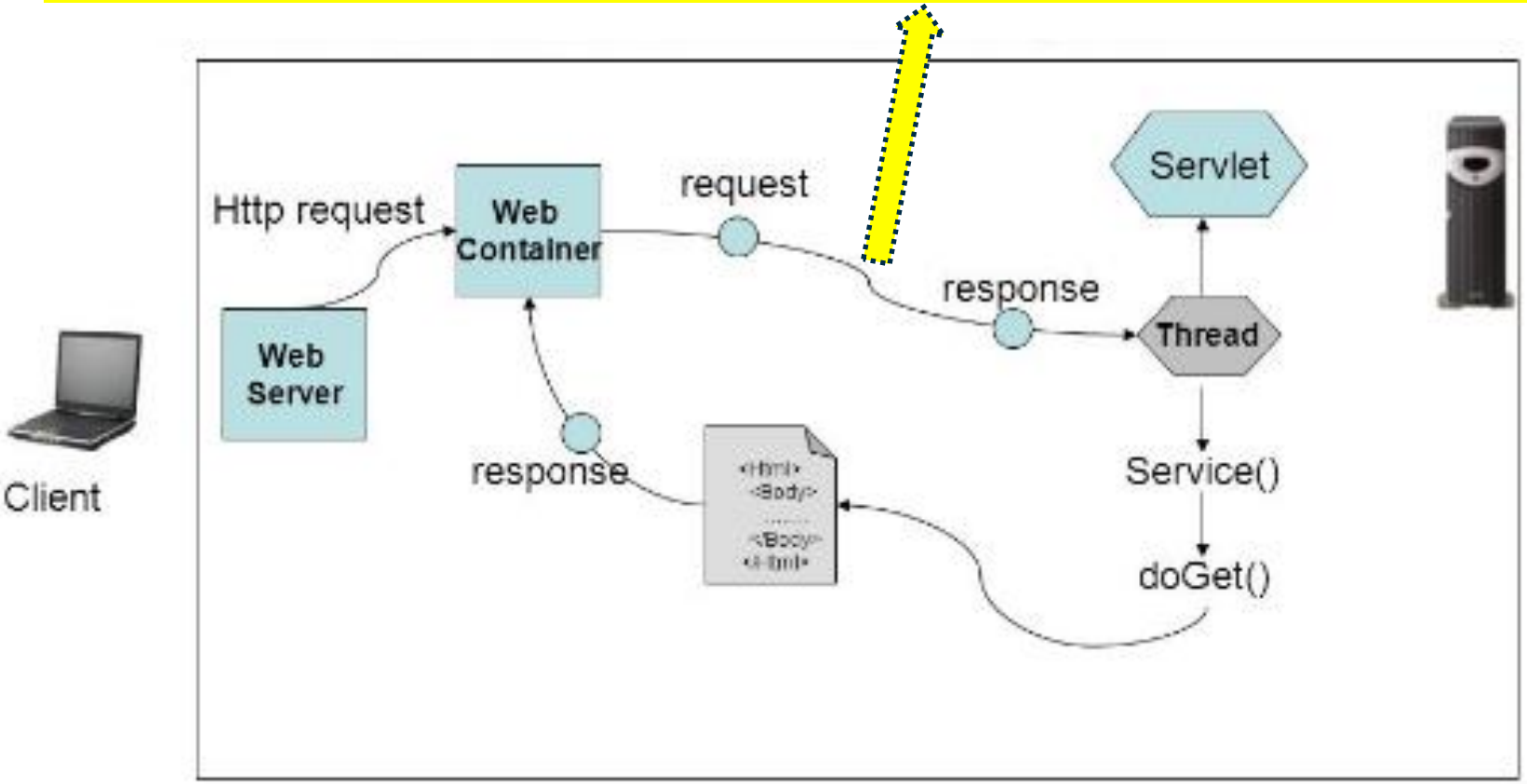
What is a Servlet?

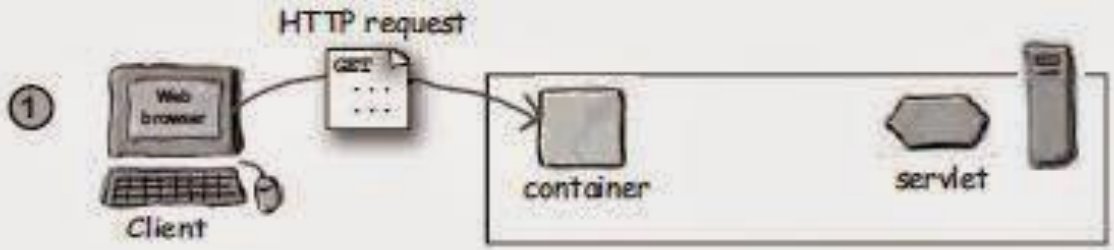
- A technology used to create web application.
- An API that provides many interfaces and classes including documentations.
- An interface that must be implemented for creating any servlet.



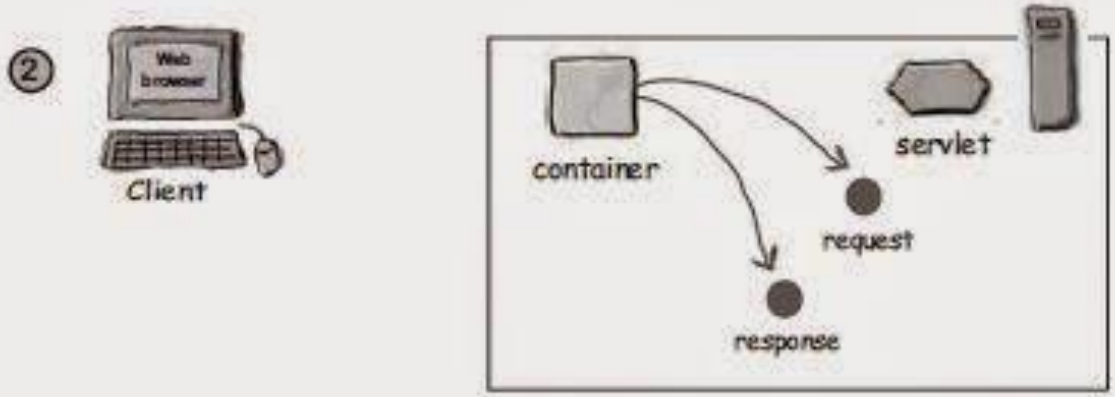
How does a Servlet work?

doGet(HttpServletRequest request, HttpServletResponse response)





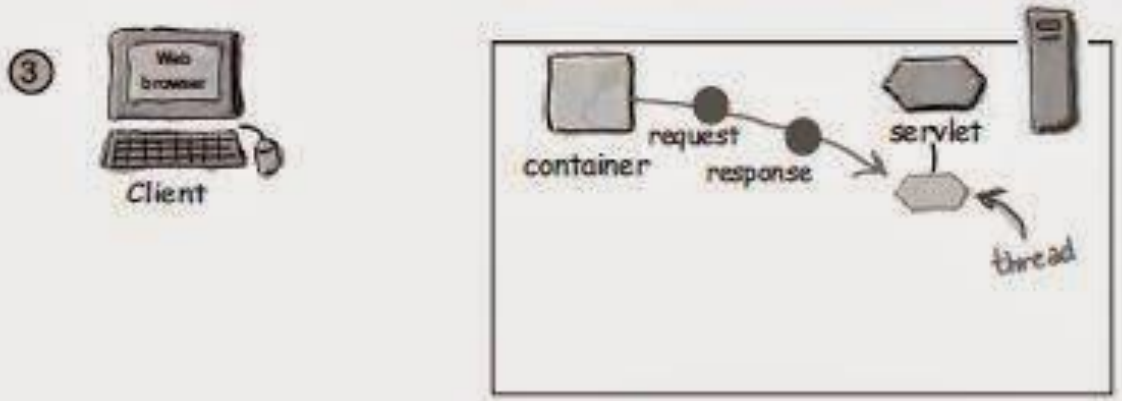
User clicks a link that has a URL to a servlet instead of a static page.



The container "sees" that the request is for a servlet, so the container creates two objects:

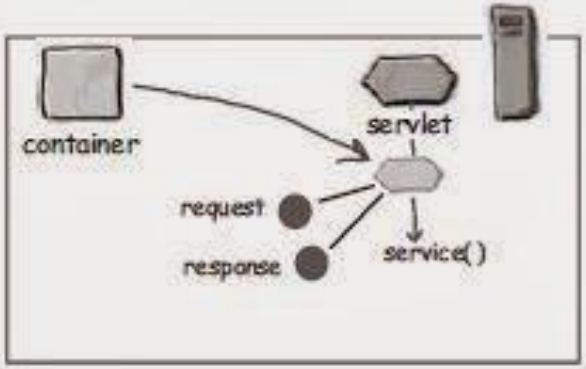
1) HttpServletResponse

2) HttpServletRequest



The container finds the correct servlet based on the URL in the request, creates or allocates a thread for that request, and passes the request and response objects to the servlet thread.

④

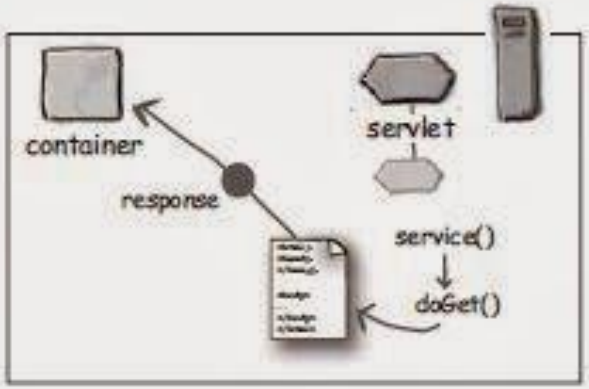


The container calls the servlet's service() method. Depending on the type of request, the service() method calls either the doGet() or doPost() method.

For this example, we'll assume the request was an HTTP GET.

---

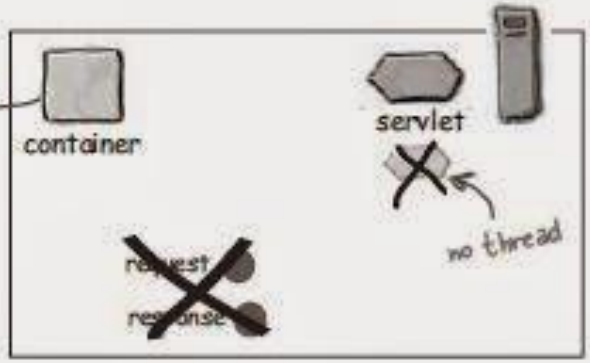
⑤



The doGet() method generates the dynamic page and stuffs the page into the response object. Remember, the container still has a reference to the response object!

---

⑥



The thread completes, the container converts the response object into an HTTP response, sends it back to the client, then deletes the request and response objects.

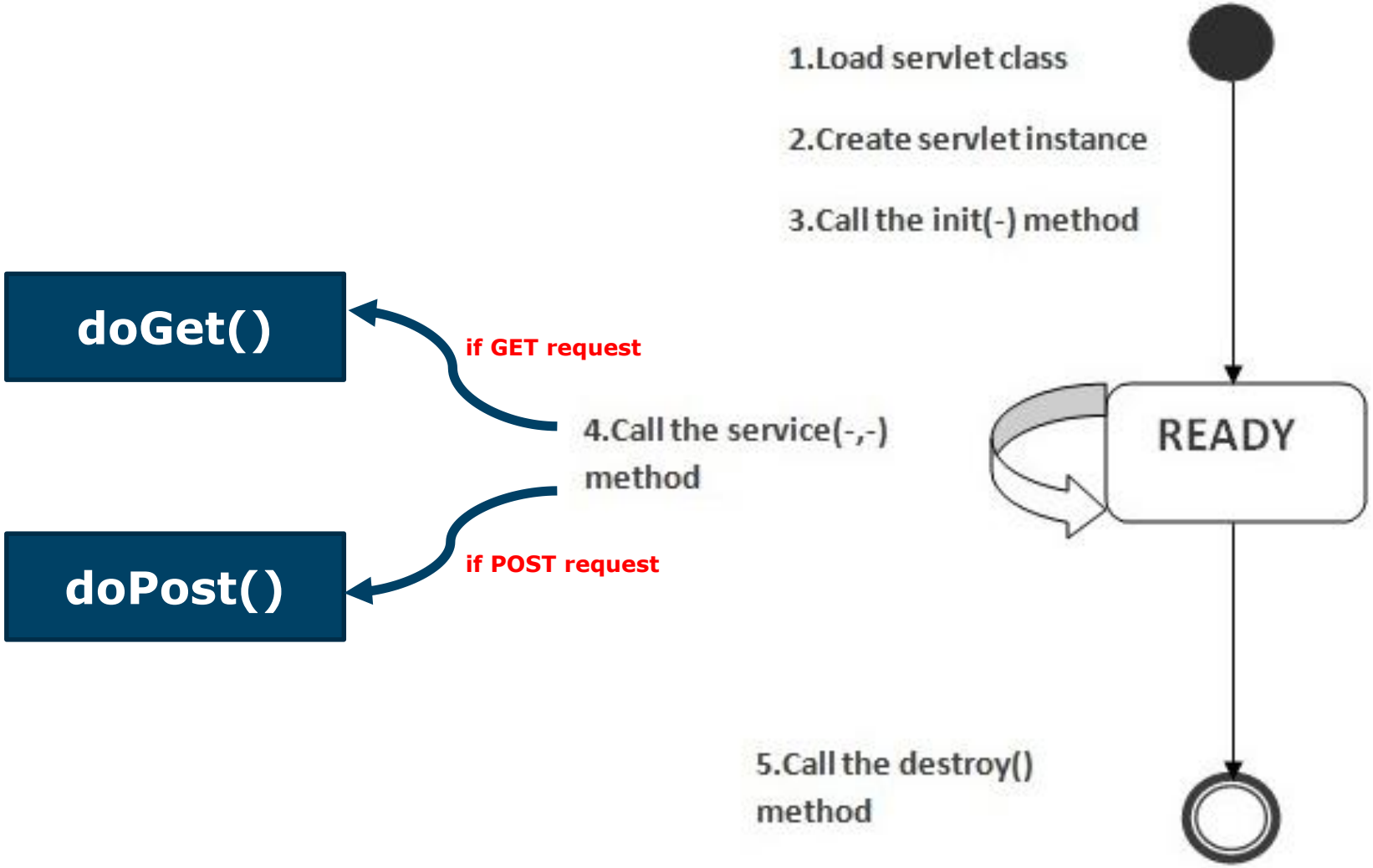
GET vs POST? When to use which?

GET	POST
1) In case of Get request, only **limited amount of data** can be sent because data is sent in header.	In case of post request, **large amount of data** can be sent because data is sent in body.
2) Get request is **not secured** because data is exposed in URL bar.	Post request is **secured** because data is not exposed in URL bar.
3) Get request **can be bookmarked.**	Post request **cannot be bookmarked.**
4) Get request is **idempotent** . It means second request will be ignored until response of first request is delivered	Post request is **non-idempotent.**
5) Get request is **more efficient** and used more than Post.	Post request is **less efficient** and used less than get.

Servlet API

- The **javax.servlet** package contains many interfaces and classes that are used by the servlet or web container. These are not specific to any protocol.
- The **javax.servlet.http** package contains interfaces and classes that are responsible for http requests only.

Life Cycle of a Servlet



Servlets: Let’s look at it in more depth

MindsMapped Consulting

A Simple Servlet

```
import java.io.*;
import javax.servlet.*;

public class SimpleGenericServlet extends GenericServlet
{       public void service (ServletRequest request,
                             ServletResponse response)
              throws ServletException, IOException
        {
            response.setContentType("text/plain");
            PrintWriter out = response.getWriter();
            out.println("Hello World");
            out.close();
        }
}
```

A Simple Servlet

`service`: The most important method in a servlet,

- Determines what the servlet does.
- Invoked automatically when a request comes in.
- Needs to be overridden.

It takes two arguments:

```
service (ServletRequest request,
         ServletResponse response)
```

`ServletRequest` and `ServletResponse` are interfaces defined by the **`javax.servlet`**

Get information about a request from the `ServletRequest` object `request`.

Set information about a response via the `ServletResponse` object `response`.

GenericServlets uncommon. So we don't study those in details

Environment for developing and testing servlets

- Compile:
  - Need Servlet.jar. Available in Tomcat package
- Setup testing environment
  - Install and start Tomcat web server

General Information about Servlets

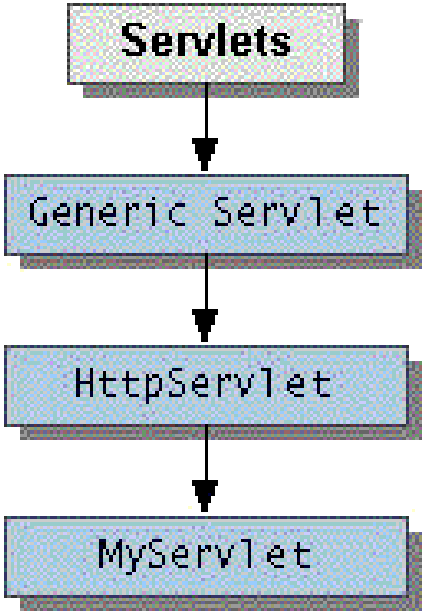
Architecture of package **`javax.servlet`**

`Servlet` interface: declares servlet methods (**`init`**, **`service`**, etc.)

`GenericServlet` implements `Servlet`

`HttpServlet` subclass adds features specific to HTTP

Technically, a servlet is a program that extends either **`GenericServlet`** or **`HttpServlet`**.

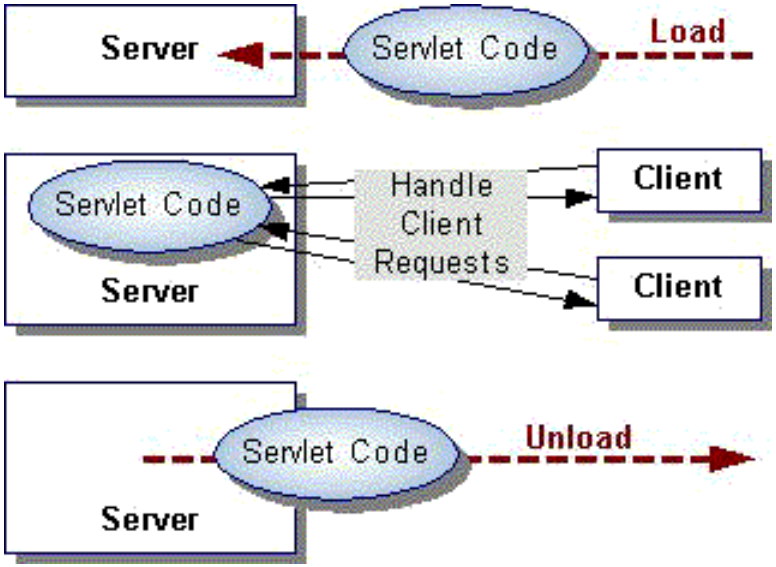


General Information about Servlets

Servlet Life Cycle

Servlets are controlled by servers

1. A server loads and initializes the servlet
   New thread created for each client.
2. The servlet handles zero or more client requests
3. The server terminates the servlet



Servlet Life Cycle

Methods

`public void init():`

Called only once when servlet is being created.

Good place for set up, open Database, etc.

`public void service():`

Called once for each request.

In **`HttpServlet`**, it delegates requests to **`doGet`**, **`doPost`**, etc.

`public void destroy():`

Called when server decides to terminate the servlet.

Release resources.

HTTP Servlets

- HTTP requests include
  - GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS
  - The default is GET.

- The usual ones used by JSP/Servlet technology are GET or POST.

- Other requests carry more meaning while designing RESTful API or RESTful architecture.

HTTP Servlets

Methods	HTTP Requests	Comments
`doGet`	GET, HEAD	Usually overridden
`doPost`	POST	Usually overridden
`doPut`	PUT	Usually not overridden
`doOptions`	OPTIONS	Almost never overridden
`doTrace`	TRACE	Almost never overridden

All methods take two arguments:

- **`HttpServletRequest`**
- **`HttpServletResponse`**.

HTTP Servlets

- **`javax.servlet.http.HttpServlet`**
  - subclass of GenericServlet, hence has service method
  - class has already overridden the **`service`** method to delegate requests to special purpose methods such as **`doGet`** and `doPost`.
  - **Don't override the `service` method when sub classing `HttpServlet`**. Instead, refine the special purpose methods, mostly `doGet` and `doPost`.

HTTP Servlets

- **javax.servlet.http.`HttpServletRequest`** Objects
  - Extends the ServletRequest

  - Provide access to HTTP header data and the parameters of the request.

  - Can get values of individual parameters using the following methods
    - **`getParameterNames`** method provides the names of the parameters
    - **`getParameter`** method returns the value of a named parameter.
    - **`getParameterValues`** method returns an array of all values of a parameter if it has more than one values.

HTTP Servlets

- **`HttpServletResponse`** Objects
  - Provide two ways of returning data to the user:
    - **`getWriter`** method returns a **`PrintWriter`** for sending text data to client
    - **`getOutputStream`** method returns a **`ServletOutputStream`** for sending binary data to client.
    - Need to close the **`Writer`** or **`ServletOutputStream`** after you send the response.
  - HTTP Header Data
    - Must set HTTP header data before you access the **`Writer`** or **`OutputStream`**.
    - **`HttpServletResponse`** interface provides methods to manipulate the header data.
    - For example, the **`setContentType`** method sets the content type. (**This header is often the only one manually set.**)

What is a MIME Type?

- The MIME type is the mechanism to tell the client the variety of document transmitted: the extension of a file name has no meaning on the web.
- It is, therefore, important that the server is correctly set up, so that the correct MIME type is transmitted with each document.
- Browsers often use the MIME-type to determine what default action to do when a resource is fetched.

HTTP Servlets

Commonly used MIME Types:

- text/plain
- text/css
- text/html
- application/pdf
- image/jpeg
- image/png
- audio/mpeg
- video/mp4

Images

Audio/Video

Note: A MIME type is insensitive to the case, but traditionally is written all in lower case.

Handling GET requests: Override the `doGet` method

```
public class BookDetailServlet extends HttpServlet
{   public void doGet (HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {   ...
 // set content-type header before accessing the Writer
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(...); // then write the response
        //Get the identifier of the book from request
        String bookId = request.getParameter("bookId");
        if (bookId != null)
       { out.println( information about the book );}
       out.close();
     ......
```

Handling POST requests: Override the `doPost` method

```
public class ReceiptServlet extends HttpServlet
{   public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {    ...
// set content type header before accessing the Writer
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(...); // then write the response

        out.println("<h3>Thank you for … " +
                    request.getParameter("cardname") +
                    ...);
        out.close();
    }
```

HTTP Servlets

- Handling both GET and POST requests in the same way

```
public class BookDetailServlet extends HttpServlet
{   public void doGet (HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {
         // codes for handling the request
    }
     public void doPost (HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {
            doGet( request, response);
      }
}
```