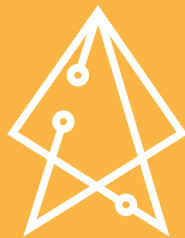


ACADEMIK



Java EE™

Servlets



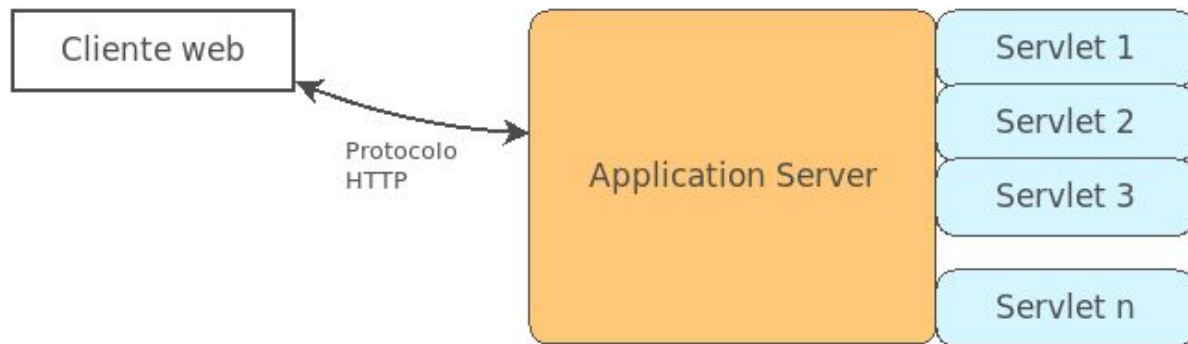
ACADEMIK

¿Cómo funciona internet?

La clave, protocolos de comunicación

HTTP - Hypertext Transfer Protocol

Es el protocolo de comunicación más utilizado a lo largo de todo internet. Este protocolo cuenta con un modelo simple que se basa en URL (Unique Resource Locations), verbos y encabezados.



HTTP - Peticiones

Cada petición HTTP cuenta con un conjunto de encabezados, una URL destino y un verbo (según el verbo, algunas peticiones permiten la incorporación de un cuerpo). Los principales verbos son:

- **GET**, para obtener recursos que provienen del servidor
- **POST**, para crear recursos en el servidor
- **PUT**, para modificar recursos existentes en el servidor
- **DELETE**, para remover, desactivar o eliminar recursos en el servidor

* Una **misma URL** puede aceptar **todos los métodos** HTTP

HTTP - Encabezados

Por su parte los encabezados son utilizados para manejar la metadata de la comunicación, entre los headers más importantes están:

- **Content-type**, establece el formato en el que se espera la información
- **Authorization**, útil para manejar aspectos de autenticación y autorización
- **Accept-Encoding**, para determinar el encoding permitido en la comunicación
- **User-Agent**, para informar al servidor desde qué cliente se está realizando una petición

HTTP - Respuestas

Las respuestas, al igual que las peticiones, también tienen encabezados, pero las respuestas tienen además un encabezado especial que determina el estado de la respuesta. Los estados se clasifican de la siguiente forma:

- **1XX** - Information
- **2XX** - Success
- **3XX** - Redirect
- **4XX** - Client error
- **5XX** - Server error

HTTPS - Estados

1XX - Information	100 Continue 102 Processing	101 Switching protocols
2XX - Success	200 Ok 202 Accepted	201 Created 204 No content
3XX - Redirect	300 Multiple choices 302 Found	304 Not modified
4XX - Client error	400 Bad request 403 Forbidden	401 Unauthorized 404 Not found
5XX - Server error	500 Internal server error 501 Not implemented	

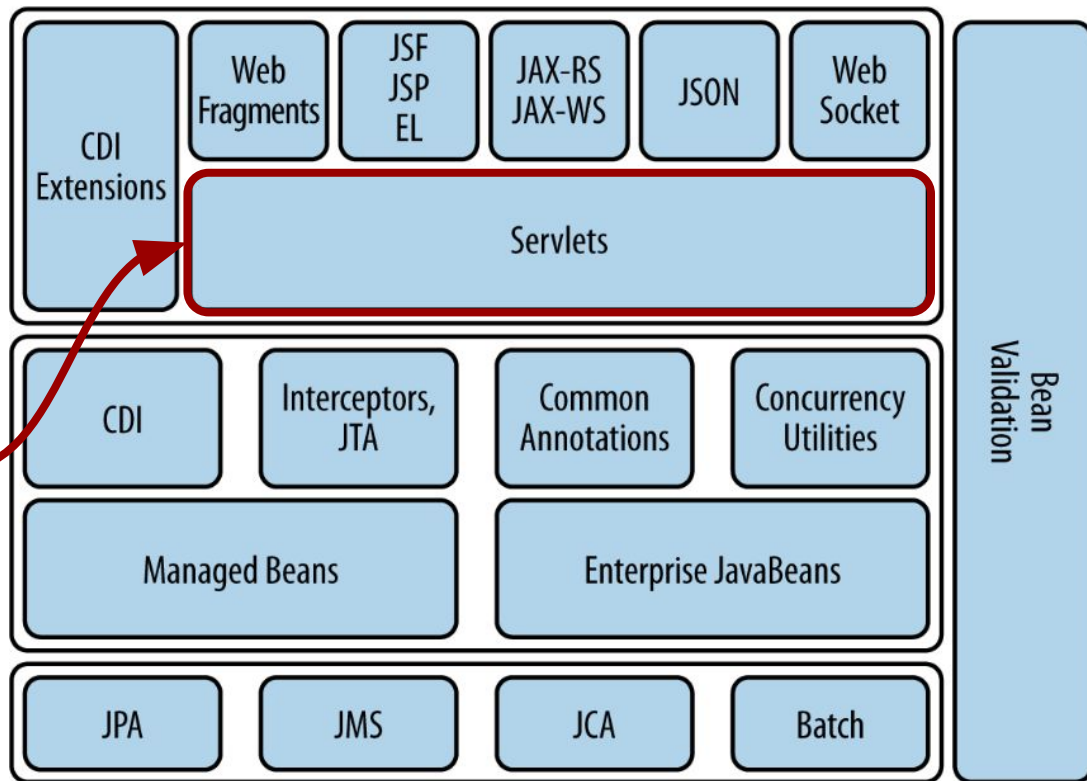


ACADEMIK

Ya conozco Java EE, ¿ahora qué?

Podemos crear nuestro primer proyecto

Arquitectura de Java EE 7



Los servlets son el corazón de la mayoría de componentes de la sección web

Servlets

Definidos en el **JSR 340**, es una clase Java que hereda de la clase **javax.servlet.http.HttpServlet**, esta clase utiliza el patrón de request/response para la comunicación con los clientes desde el contenedor de servlets



Clase HttpServlet

Esta clase es asociada con un path (URL) y cuenta con un método para cada verbo HTTP de la forma doXXX

Métodos HTTP

- doGet()
- doPost()
- doPut()
- delete()

Además de otros métodos para el ciclo de vida de cada servlet

- init()
 - destroy()
-

La clase Servlet

Para que una clase sea reconocida como Servlet, además de heredar de `javax.servlet.http.HttpServlet`, debe estar anotada con `@WebServlet`.

```
@WebServlet("/hello")  
public class HelloServlet extends HttpServlet {  
    // Declare your methods here  
}
```

```
@WebServlet(urlPatterns={"/hello", "/hola", "/bonjour"})  
public class HelloServlet extends HttpServlet {  
    // Declare your methods here  
}
```

Los métodos HTTP

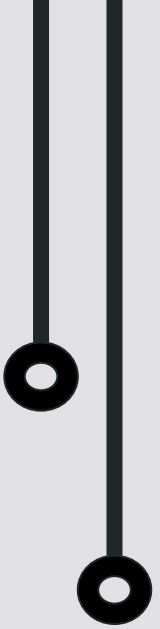
La firma de cada método HTTP es la siguiente:

```
protected void doXXX(  
    HttpServletRequest request,  
    HttpServletResponse response  
) {  
    // Do something  
}
```



Manos a la obra

- Mi primer aplicación web
- Mi primer servlet
- Usando postman
- Servlet RestFul





ACADEMIK

Gracias por su atención

Escríbenos a: [cursos@nabenik.com](mailto: cursos@nabenik.com)

www.academik.io