JAVA A PROFUNDIDAD

Ciclo de vida del Servlet

El ciclo de vida de un servlet puede ser definido como el proceso total desde su creación hasta la destrucción del mismo. A continuación, se describe el recorrido hecho por un servlet.

- El servlet es inicializado llamado al método init().
- El servlet llama al método service() para procesar la petición (request) del cliente.
- El servlet termina su ciclo llamando al método destroy().
- Finalmente, el servlet es recogido por el garbage collector de la JVM.

Ahora, veamos en detalle cada una de las partes del ciclo de vida de los servlets.

El método init().

El método init() es llamado una sóla vez. Este es llamado sólo cuando el servlet es creado, y no se realiza ninguna llamada del usuario después. Por lo tanto, se utiliza para inicializaciones de una sola vez.

El servlet normalmente es creado cuando un usuario invoca por primera vez una url correspondiente al servlet, pero también puede especificar que el servlet se cargue cuando se inicie por primera vez el servidor.

Cuando un usuario invoca un servlet, se crea una única instancia de cada servlet, con cada solicitud de usuario que da como resultado un nuevo hilo que se transfiere a los métodos doGet o doPost según corresponda. El método init() simplemente crea o carga algunos datos que se utilizarán a lo largo de la vida del servlet.

La declaración del método init() es como sigue:

```
1.  public void init() throws ServletException {
2.      // Código de inicialización ...
3.  }
```

El método service()

El método service() es el método principal de los servlets. Este método realiza la tarea actual. El contenedor de servlets (servidor web) llama al método service() para manejar las peticiones que llegan desde el cliente (browsers) y escribe la respuesta formateada de vuelta al cliente.

Cada vez que el servidor recibe una petición para un servlet, el servidor crea un nuevo hilo (thread) y llama al método service. El método service determina el tipo HTTP de la petición (GET, POST, PUT, DELETE, PATCH, ETC) y llama a los métodos doGet(), doPost(), doDelete(), doPut(), etc, dependiendo del tipo de petición recibida desde el cliente.

Los métodos doGet() y doPost() son los métodos más frecuentemente usados en cada petición.

El método doGet()

Una petición GET proviene de una solicitud normal de una URL o de un formulario HTML (Vease el tema de lo formularios HTML más adelante) que no tiene método especificado y debe ser manejado por el método doGet().

```
1.  public void doGet(HttpServletRequest request, HttpServletResponse response)
2.     throws ServletException, IOException {
3.     // Código del servlet.
4.  }
```

El método doPost()

Una petición por el método POST desde un formulario HTML (el cual debe especificar que utilizará el método POST), es manejada por el método doPost().

```
1.  public void doPost(HttpServletRequest request, HttpServletResponse response)
2.     throws ServletException, IOException {
3.     // Código del servlet.
4.  }
```

El método destroy()

El método destroy() es llamado solo una vez al final del ciclo de vida de un servlet. Este método permite al servlet realizar algunas acciones finales antes de destruirlo como cerrar una conexión a la base de datos.

Después que el método destroy() es llamado, el servlet es marcado para recolectarse por el garbage collector. La definición del método destroy es como sigue:

```
1.  public void destroy()
2.  {
3.     // Código de finalización.
4.  }
```