

Introducció

- HTTP és un protocol sense estat (stateless).
- Què significa sense estat i quines són les conseqüències per als dissenys d'aplicacions web?
 - Cada sol·licitud entre client i servidor és independent.
 - No emmagatzema cap informació sobre una petició o una resposta.
 - Es perden els valors de les variables i els objectes que mantenen la connexió amb la base de dades.
 - A cada petició tot ha de començar des de zero.
- Tot allò que diferent a la simple consulta a una pàgina independent, gairebé sempre serà necessari mantenir l'estat de la sessió.
- HTTP no ens ofereix cap mecanisme propi per mantenir l'estat d'una sessió

Introducció

- Les denominades **aplicacions Web** és més que una simple consulta d'una pàgina independent.
- Una **aplicació Web** és aquella en la que és necessari:
 - Autenticar l'usuari.
 - Conèixer els passos de la navegació i les preferències de l'usuari.
 - Actualitzar una base de dades.
- A les **aplicacions Web** necessitem alguna cosa que ens permeti vincular una petició amb un altre que es produeixi després.
- Aquesta vinculació pot ser immediatament després o un minut o una hora més tard.
- El manteniment de l'estat de sessió ens ho ofereix.

Què és una sessió?

- Una **sessió** és una seqüència de peticions a un servidor web realitzades per un client des d'un navegador dins de l'àmbit d'una aplicació web.
- Les característiques d'una sessió són:
 1. **Persistència temporal:** Una sessió comença quan un usuari visita una aplicació web i finalment finalitza quan l'usuari tanca el navegador o després d'un temps d'inactivitat.
 2. **Identificador únic de sessió (session ID):** Quan es crea una sessió, el servidor genera un identificador únic (generalment una cadena aleatòria) que es guarda al servidor.
 3. **Emmagatzematge d'informació:** Les dades de la sessió guardades al servidor poden incloure informació com el nom d'usuari, preferències, o altres dades específiques de l'usuari.

Què és l'estat d'una sessió?

- L'estat d'una sessió en el context de programació web és el conjunt de dades i la informació que es guarda en una sessió activa.
- Aquesta informació o conjunt de dades permet mantenir un procés relacionat de peticions dins d'una aplicació.
- Aquest estat permet al servidor recordar informació sobre un usuari específic i el seu comportament al llarg d'un conjunt de peticions HTTP, en lloc de tractar cada interacció de manera independent.
- L'estat d'una sessió és compon de totes les dades emmagatzemades al servidor i associades a l'identificador únic de sessió (Session ID) de l'usuari.

Què és l'estat d'una sessió?

- Aquestes dades poden variar segons l'aplicació, però habitualment inclouen els següents elements:
 1. Informació de l'usuari:
 - Dades personals, com ara el nom d'usuari, el correu electrònic o el número d'identificació (ID).
 - Aquesta informació s'utilitza per personalitzar l'experiència o per autenticar l'usuari en diferents parts de l'aplicació.
 2. Preferències o configuracions:
 - Configuracions seleccionades per l'usuari, com l'idioma preferit, el tema visual de l'aplicació, o altres ajustaments específics.
 - Exemple: Un usuari que prefereix veure la interfície en català.
 3. Progrés d'una interacció:
 - Informació temporal relacionada amb accions de l'usuari que no han conclòs, com el contingut d'una cistella de la compra o les respostes en un formulari.

Com resoldre la limitació de HTTP

- Per resoldre aquesta limitació del protocol HTTP existeixen solucions que s'estan aplicant des de fa temps en les aplicacions web.
- Solucions per gestionar l'estat en aplicacions web:
 1. Ocultar dades dins de formularis HTML
 - Les dades necessàries es passen d'una pàgina a una altra mitjançant camps ocults en formularis.
 2. Emmagatzemar la informació en el servidor (en taules o arxius)
 - Informació com el carret de la compra o l'estat d'inici de sessió es poden guardar en taules de bases de dades o fitxers.
 3. Emmagatzemar la informació en el client
 - Utilitzar per a desar informació al dispositiu de l'usuari.

Ocultar dades dins de formularis

- Els elements **input de tipus hidden** (<input type="hidden">) dins dels formularis es poden utilitzar per emmagatzemar dades que sempre estaran disponibles durant la navegació de l'usuari.
- Les dades necessàries per mantenir una sessió (ID usuari, productes cistella, etc.) es poden incloure en un o diversos camps ocults del formulari.
- Cada vegada que es processa una petició al servidor, aquest té accés a tota la informació pel que sigui necessari.
- Cada petició reenvia aquestes dades al servidor.

```
<form method="POST" action="carret.php">  
  <input type="hidden" name="productes" value="producteX,producteY">  
  <button type="submit">Enviar</button>  
</form>
```

Guarda informació en el servidor

- Una solució més robusta per mantenir l'estat en aplicacions web és emmagatzemar les dades necessàries al servidor.
- Això permet mantenir la informació que el client no pot manipular directament, garantint seguretat i coherència.
- Com funciona?
 1. Quan un usuari inicia una sessió o realitza una acció, la informació es guarda en una base de dades o fitxer al servidor.
 2. A cada petició posterior, pot recuperar aquesta informació utilitzant un identificador únic associat a l'usuari (com per exemple l'ID de sessió).
- Per raons de capacitat de recursos en el servidor, l'emmagatzematge de dades en el servidor és una solució vàlida quan la quantitat d'usuari no és alta.

Guarda informació en el client

- Aquesta alternativa és similar a l'anterior amb la diferència que no consumeix recursos en el servidor.
- Actualment és l'opció més utilitzada en les aplicacions Web.
- La informació emmagatzemada en l'equip client són les **cookies**.
- Com que la informació està emmagatzemada fora del servidor, mai es pot estar segur que la cookie estigui disponible quan sigui necessari.
- La informació necessària per mantenir la sessió es pot perdre perquè l'usuari podria:
 - Canviar d'equip.
 - Eliminar la cookie.

Què és una cookie?

- Un petit arxiu de fins a 4 KB que el servidor web emmagatzema a l'ordinador del client.
- Inclouen un parell clau = valor que retorna al servidor en futures sol·licituds.
- Només es poden llegir des del domini que els va emetre.
- Es poden classificar en:
 - Primàries: Configurades pel domini principal del lloc web.
 - De tercers: Configurades per altres dominis (exemple: publicitat).
- Cada navegador pot emmagatzemar les cookies en diferents ubicacions.
- La majoria de navegadors permeten desactivar les cookies.

Creació de cookies amb PHP

- Funció bàsica:

setcookie(nom_cookie, valor_cookie, [expiry_time], [cookie_path], [domain], [secure]);

- Paràmetres principals:
 - **nom_cookie**: Nom identificador de la cookie (obligatori).
 - **valor_cookie**: Contingut o informació de la cookie (obligatori).
 - **expiry_time**: Temps de caducitat (opcional). Ex.: time() + 3600(1 hora).
 - **cookie_path**: Ruta del servidor on és accessible (exemple: "/" per a tot el domini). Si no s'informa, agafa la ruta d'accés establerta en el servidor.
 - **domini**: Domini que pot accedir a la cookie (opcional). Si no s'informa, agafa la ruta d'accés establerta en el servidor.
 - **segur**: Indica si la cookie es transmet per un canal segur HTTPS (opcional, predeterminat és false).

Exemples pràctics de cookies en PHP

- **Exemple 1:** Crear una cookie que dura 60 segons

```
setcookie("user_name", "Guru99", time() + 60, '/');  
echo "La cookie s'ha configurat per 60 segons.";
```

- **Exemple 2:** Llegir una cookie

```
if (isset($_COOKIE['user_name'])) {  
    echo "Hola, " . $_COOKIE['user_name'];  
} else {  
    echo "No hi ha cap cookie configurada.";  
});
```

Exemples pràctics de cookies en PHP

- **Exemple 3:**

```
setcookie(  
    "user_name",    // Nom de la cookie  
    "JohnDoe",      // Valor  
    time() + 3600,  // Caduca en 1 hora  
    "/subcarpeta",  // Només accessible des de /subcarpeta  
    ".exemple.com"  // Accessible des de exemple.com i subdominis  
);
```

Aquesta cookie:

- Ruta (cookie_path) : Només estarà disponible dins www.exemple.com/subcarpeta.
- Domini (domini) : Qualsevol subdomini d' exemple.com podrà accedir a la cookie si es troba dins de la ruta especificada.

Variables de sessió?

- Una sessió és una variable global emmagatzemada al servidor.
- Cada sessió té un identificador únic (ID de sessió).
- Aquest identificador s'emmagatzema en una cookie al navegador de l'usuari.
- Si no hi ha suport per a cookies, l'ID de sessió pot enviar-se a través de la URL.
- Les sessions poden emmagatzemar més dades que les cookies (mida relativament gran).
- Les dades de la sessió es perden automàticament quan tanca el navegador.

Creació d'una sessió en PHP

- Per començar una sessió a PHP, s'ha de cridar la funció `session_start()`.
- Això habilita l'ús de la superglobal `$_SESSION`, un array associatiu que permet guardar i recuperar dades de la sessió.
- Exemple:

```
session_start(); // Comença la sessió
```

```
if(isset($_SESSION['page_count'])) {  
    $_SESSION['page_count'] += 1;  
} else {  
    $_SESSION['page_count'] = 1;  
}
```

Finalització de la sessió en PHP

- Es pot destruir la sessió amb **session_destroy()** i eliminar les dades de `$_SESSION`.
- Quan una sessió s'elimina, l'estat associat desapareix. Fins que això no passa, el servidor manté aquesta informació disponible per a cada petició d'aquest usuari.

Quan utilitzar les variables de sessió?

- Passar informació entre pàgines: Les sessions permeten transferir dades de manera eficient.
- Seguretat millorada: Emmagatzema dades al servidor, protegint-les contra manipulacions per part de l'usuari.
- Alternativa a les cookies: Útil quan els navegadors no suporten cookies.
- Aplicacions de gran capacitat: Ideal per a aplicacions com cistelles de compra que necessiten emmagatzemar més de 4KB de dades.