Entorns de Desenvolupament

17. Cookies

Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur

IES DR. LLUÍS SIMARRO

17. Cookies

Què és una cookie?

HTTP és un **protocol sense estat**, és a dir l'usuari sol·licita una pàgina web i HTTP torna la pàgina web. **No existeix** el **seguiment** i les **dades identificables**.

La **privadesa absoluta** pot semblar molt bé en principi, però si no podem fer un seguiment de quin usuari és quin, **no** hi ha manera d'**implementar carrets de la compra** i no podem processar **transaccions de manera segura**.

Per això es van introduir les **galetes** o **cookies**, que no són més que un xicotet tros de **dades guardades al navegador**. S'utilitzen principalment per fer un **seguiment dels usuaris**, guardar algunes **preferències** i emmagatzemar **dades temporals**.

Així doncs, una **galeta** en **PHP** és un **fitxer xicotet** amb una **mida màxima de 4KB** que el **servidor web emmagatzema a l'ordinador client**. Normalment s'utilitzen per fer un **seguiment d'informació** com ara un **nom d'usuari** que el lloc pot recuperar per **personalitzar la pàgina** quan l'usuari visite el lloc web la propera vegada. **Cada vegada** que el mateix ordinador **sol·licite** una **pàgina** amb un navegador, també **enviarà la galeta**.

Una galeta **només es pot llegir** des del **domini des del qual s'ha emès**. Les galetes solen establir-se en una **capçalera HTTP**, però JavaScript també pot establir una galeta directament en un navegador.

PHP admet **cookies** o **galetes HTTP** de manera **transparent**. Són un mecanisme per **emmagatzemar dades en el navegador remot** i, per tant, fer el seguiment o **identificar** els **usuaris** recurrents. Amb **PHP** es poden **crear** galetes mitjançant la funció `setcookie()` o `setrawcookie()`.

Les **galetes formen part** de la **capçalera HTTP**, de manera que cal **cridar a `setcookie()` abans que s'envie cap eixida al navegador**. Es tracta de la **mateixa limitació** que té la funció `header()`. Es poden utilitzar les funcions de **memòria intermèdia d'eixida** (o la configuració en **php.ini**, `output_buffering`) per **retardar l'eixida** de l'script fins que s'haja decidit si es vol configurar o no cap galeta o enviar cap capçalera.

Qualsevol galeta enviada al servidor des del client **s'inclourà automàticament** en un array global `$_COOKIE`.

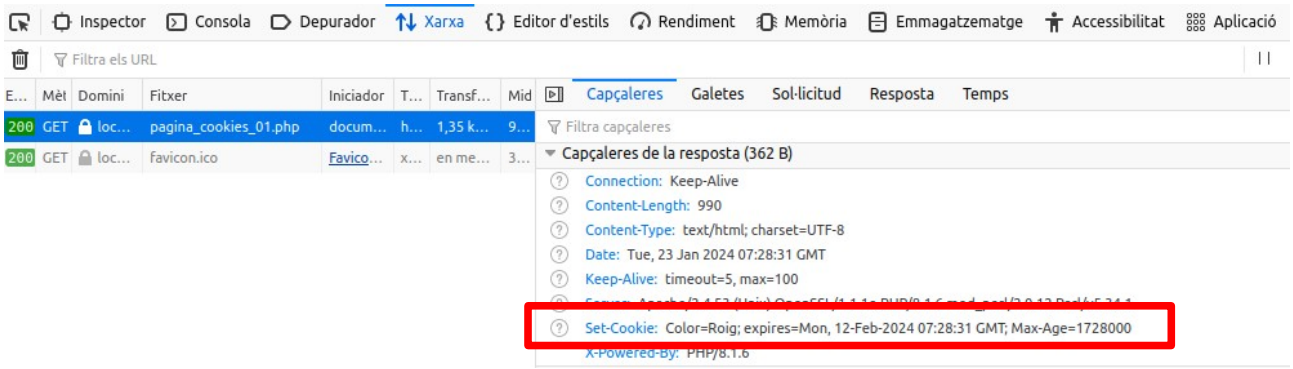
Exemple: establir una galeta:

```
<?php
    // Crear una galeta vàlida per 20 dies = 20*24*60*60 segons
    setcookie("Color", "Roig", time()+20*24*60*60);
?>
```

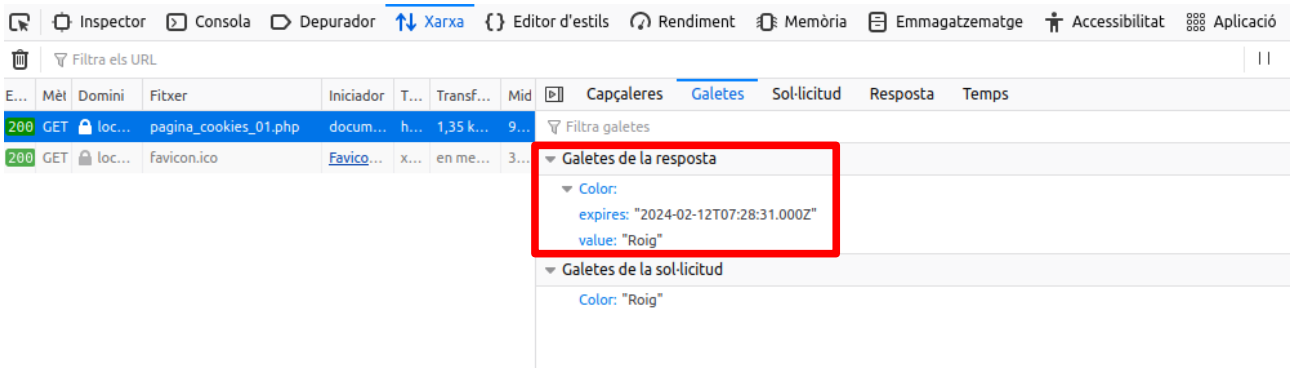
Com funcionen realment les cookies

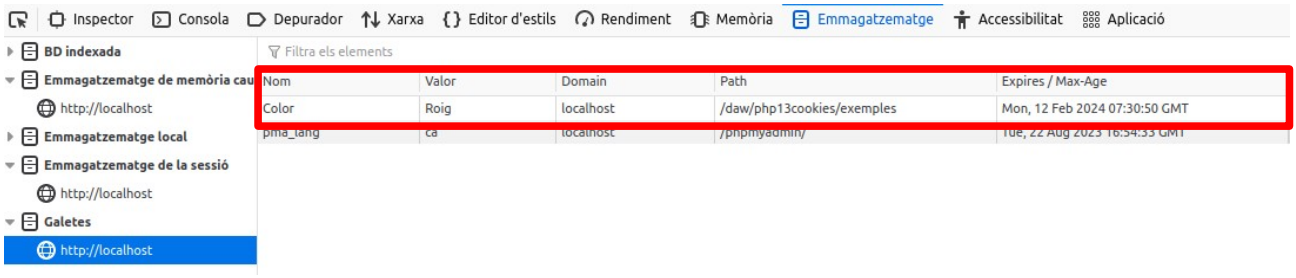
Les **galetes no es guarden al servidor**. El que realment passa amb `setcookie("Color", "Roig")` és que PHP generarà la capçalera HTTP:

Set-Cookie: Color=Roig

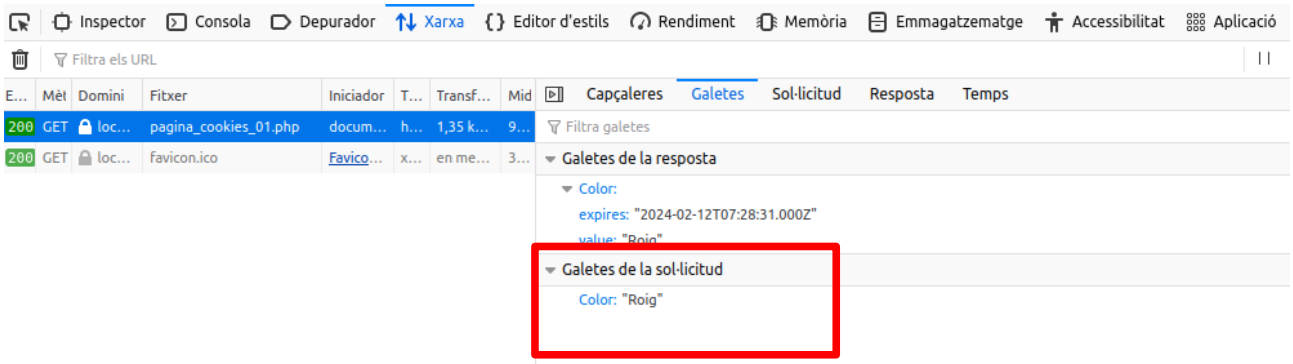


Quan el navegador reba **Set-Cookie: Color=Roig**, crearà i guardarà la galeta.





En visites posteriors, el navegador torna a enviar la galeta `Color=Red` al servidor; PHP guarda això a `$_COOKIE`.



Crear cookies

Es crea una galeta amb la funció `setcookie()`:

```
setcookie(name, value, expires, path, domain, security, httponly);
```

La **sintaxi** detallada és la següent:

```
setcookie(
    string $name,
    string $value = "",
    int $expires_or_options = 0,
    string $path = "",
    string $domain = "",
    bool $secure = false,
    bool $httponly = false
): bool
```

Veiem els **paràmetres** de la funció:

- **name**: El nom de la galeta.
- **value**: El valor de la galeta. Aquest valor **s'emmagatzema a l'ordinador del client**; és convenient no emmagatzemar-hi informació sensible. Suposant que el nom és 'Color', aquest valor es **recupera** mitjançant `$_COOKIE['Color']`.
- **expires_or_options**: El moment en què caduca la galeta. És una marca de temps d'Unix, per tant és el nombre de segons des de l'**època**. Una manera de configurar-ho és afegint el nombre de segons abans que caduque la galeta al resultat de la crida a la funció `time()`. Per exemple, `time()+60*60*24*30` establirà que la galeta caduque en 30 dies. Una altra opció és utilitzar la funció `mktime()`. Si **s'estableix a 0 o s'omet**, la galeta **caducarà al final de la sessió** (quan es tanque el navegador).
- **path**: La ruta del servidor on la galeta estarà disponible. Si s'estableix a '/', la galeta estarà disponible a tot el domini. Si s'estableix a '/temp/', la galeta només estarà disponible al directori /temp/ i tots els subdirectoris del domini com ara /temp/dir/. El valor per defecte és el directori actual on s'està configurant la galeta.
- **domain**: El (sub)domini al qual està disponible la galeta. Si es configura en un subdomini (com ara "www.exemple.com"), la galeta estarà disponible per a aquest subdomini i tots els altres subdominis d'aquest (per exemple, w2.www.exemple.com). Per fer que la galeta estiga disponible per a tot el domini (inclosos tots els subdominis d'aquest), simplement cal establir el valor al nom del

domini ('exemple.com', en aquest cas).

- **secure**: Indica que la galeta **només** s'ha de **transmetre mitjançant una connexió HTTPS segura** des del client. Quan s'estableix a `true`, la galeta només s'establirà si existeix una connexió segura.
- **httponly**: Quan és `true`, la galeta **només serà accessible mitjançant el protocol HTTP**. Això vol dir que la galeta no serà accessible mitjançant llenguatges de script, com ara JavaScript.
- **options**: Un **array associatiu** que pot tenir qualsevol de les claus `expires`, `path`, `domain`, `secure`, `httponly` i `samesite`. Si hi ha alguna altra clau, es genera un error de nivell **E_WARNING**. Els valors tenen el mateix significat que el descrit per als paràmetres amb el mateix nom. El valor de l'element `samesite` hauria de ser `None`, `Lax` o `Strict`. Si no es dóna cap de les opcions permeses, els seus valors per defecte són els mateixos que els valors per defecte dels paràmetres explícits. Si s'omet l'element `samesite`, no s'estableix cap atribut de galeta `SameSite`.

`setcookie()` defineix una galeta que s'enviarà juntament amb la resta de capçaleres HTTP. Com altres capçaleres, **les galetes s'han d'enviar abans de qualsevol eixida de l'script** (és una restricció de protocol). Això requereix que les crides a aquesta funció es facen abans de qualsevol eixida a pantalla, incloses les etiquetes `<html>` i `<head>` així com qualsevol espai en blanc.

Si l'eixida existeix abans de cridar aquesta funció, `setcookie()` fallarà i retornarà `false`. Si `setcookie()` s'executa correctament, tornarà `true`. Això no indica si l'usuari ha acceptat la galeta.

Només el paràmetre de **nom** és obligatori. Tots els altres paràmetres són opcionals.

Crear i recuperar cookies

L'exemple següent **crea** una **galeta** anomenada **"usuari"** amb el valor **"Xavi"**. La galeta caducarà al cap de 30 dies (86400 * 30).

A continuació, es recupera el valor de la galeta **"usuari"** (utilitzant la variable global

`$_COOKIE`). També fem servir la funció `isset()` per saber si la galeta està configurada:

Exemple:

```
<?php
$cookie_nom= "usuari";
$cookie_valor = "Xavi";
setcookie($cookie_nom, $cookie_valor, time() + (86400 * 30), "");
// 86400 = 1 dia
?>
<!DOCTYPE html>
<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_nom])) {
  echo "Cookie amb nom '" . $cookie_nom . "' no està creada!";
} else {
  echo "Cookie '" . $cookie_nom . "' s'ha creat!<br>";
  echo "El seu valor és: " . $_COOKIE[$cookie_nom];
}
?>
</body>
</html>
```

La funció `setcookie()` ha d'aparèixer **ABANS** de l'etiqueta `<html>`. El **valor de la galeta** es **codifica automàticament per URL quan s'envia** la galeta i es **descodifica automàticament quan es rep** (per **evitar la codificació d'URL**, cal utilitzar `setrawcookie()`).

Modificar el valor de les cookies

Per modificar una galeta, només cal **tornar a configurar** la galeta mitjançant la funció `setcookie()`:

Exemple:

```
 <?php
$cookie_nom = "usuari";
$cookie_valor_nou = "Rebeca";
setcookie($cookie_nom, $cookie_valor_nou, time() + (86400 * 30), "");
?>
<!DOCTYPE html>
<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_nom])) {
  echo "Cookie amb nom '" . $cookie_nom . "' no s'ha creat!";
} else {
  echo "Cookie '" . $cookie_nom . "' s'ha creat!<br>";
  echo "El seu valor és: " . $_COOKIE[$cookie_nom];
}
?>
</body>
</html>
```

Eliminar cookies

Per eliminar una galeta, s'utilitza també la funció `setcookie()`, però amb una data de caducitat en el passat.

Cal tindre en compte, que `$_COOKIE` no reflectirà els canvis immediatament. Caldrà desactivar manualment amb `unset($_COOKIE["cookie_nom"])` per eliminar la clau/valor de la sessió actual.

```
<?php
// establim la data d'expiració a una hora abans o a un valor molt remot
setcookie("usuari", "", time() - 3600); // setcookie("usuari", "", 1);
unset($_COOKIE["usuari"]);
?>
```

```
<!DOCUMENT html>
<html>
<body>
<?php
      echo "Cookie 'usuari' s'ha eliminat.";
?>
</body>
</html>
```

Comprovar si estan habilitades les cookies

L'exemple següent verifica si les galetes estan habilitades. Primer, provem de crear una galeta de prova amb la funció `setcookie()` i, a continuació, comptem els elements de la variable de matriu `$_COOKIE`:

Exemple:

```
<?php
setcookie("cookieProva", "prova", time() + 3600, "");
?>
<!DOCTYPE html>
<html>
<body>
<?php
if(count($_COOKIE) > 0) {
  //si hi ha alguna galeta, estan habilitades
  echo "Les cookies estan habilitades.";
} else {
  echo "Les Cookies estan deshabilitades.";
}
?>
</body>
</html>
```

Arrays en les cookies

Per a inserir arrays en les cookies, primer hem d'utilitzar `serialize()` o `json_encode()` per convertir l'**array en una cadena**.

```
<?php
// Serialitzem l'array
setcookie("array1", serialize(["Hola", "Adéu"]));
// O utilitzem json_encode
setcookie("array2", json_encode(["PHP", "Cookies"]));
?>
```

Per **recuperar** un **array des d'una cookie** hem d'utilitzar `unserialize()` o `json_decode()`, convertint la **cadena en un array de nou**.

```
<?php
// Deserialitzem l'array
$array1 = unserialize($_COOKIE["array1"]);
var_dump($array1);
// O utilitzem json_decode
$array2 = json_decode($_COOKIE["array2"]);
print_r($array2);
?>
```

Avantatges de l'ús de cookies

Entre els avantatges d'utilitzar galetes en PHP podem destacar:

- **Experiència d'usuari millorada**: les galetes permeten als llocs web **emmagatzemar informació específica de l'usuari**, com ara les **preferències i les credencials d'inici** de sessió, que es poden utilitzar per oferir una experiència d'usuari més personalitzada.

- **Dades persistents**: les galetes permeten als llocs web **emmagatzemar dades al dispositiu de l'usuari**, que poden persistir fins i tot **després que l'usuari tanque el navegador o apague el seu dispositiu**. Això fa possible que els llocs web recorden les preferències d'un usuari i les credencials d'inici de sessió en diverses visites.
- **Fàcil implementació**: les galetes PHP són fàcils d'implementar i es poden utilitzar per emmagatzemar una gran varietat de dades, cosa que les converteix en una eina versàtil per als desenvolupadors de llocs web.

Bones pràctiques amb les cookies

Per garantir la **millor experiència d'usuari** i **seguretat** possibles, és important seguir certes bones pràctiques quan s'utilitzen galetes PHP. Algunes d'aquestes bones pràctiques inclouen:

- **Utilitzar connexions segures** (HTTPS) quan es creen i s'accedeixen galetes. Això ajudarà a **protegir les dades emmagatzemades** a les galetes perquè **no siguen interceptades per tercers**.
- **No emmagatzemar dades sensibles**, com ara **credencials d'inici de sessió**, a les galetes. En comptes d'això, és **millor** utilitzar solucions d'emmagatzematge del costat del **servidor**, com ara **bases de dades**, per emmagatzemar aquest tipus de dades.
- **Utilitzar noms de galetes únics i descriptius** per a les galetes per **evitar conflictes amb altres galetes** utilitzades pel vostre lloc web o altres llocs web.
- **Limitar la quantitat de dades emmagatzemades** a les galetes només al que siga necessari. Grans quantitats de dades poden **alentir el rendiment del lloc web** i **augmentar el risc de violacions de dades**.

L'ús de les **cookies** i la seua intromissió amb la **privadesa** de l'usuari és una qüestió que està en constant debat. Una vegada acceptades, les cookies permeten **accedir a les dades del dispositiu de l'usuari** per tal de registrar la seua navegació i enviar-li contingut personalitzat. Aquest és un factor de gran interès per a les empreses,

però que fins fa poc poques vegades és notificat al ciutadà de forma clara i precisa.

Per tal de regular-ho s'ha establert alguna **legislació**:

- Llei 34/2002, de l'11 de juliol, de Serveis de la Societat de la Informació i de Comerç Electrònic. Enllaç. Resum.
- Llei Orgànica 3/2018, de 5 de desembre, de Protecció de Dades Personals i Garantia dels Drets Digitals. Enllaç. Resum.

En el Títol III de la llei 3/2018 s'adapta la normativa al principi de transparència en el tractament del **reglament europeu**, que regula el **dret** dels afectats a **ser informats sobre el tractament i recull la denominada «informació per capes»** ja generalment acceptada en àmbits com el de la videovigilància o la instal·lació de dispositius d'emmagatzematge massiu de dades (com ara les **galetes**), **facilitant a l'afectat la informació bàsica**, si bé, indicant-li una adreça electrònica o un altre mitjà que permeta accedir de forma senzilla i immediata a la restant informació.

S'exigia, entre d'altres coses, l'**obligació d'informar, a l'usuari, de manera explícita els fins de les galetes que ha d'acceptar, la seua funció i destinació**, així com l'encarregat de tractar les dades de caràcter personal que es traslladaren a cada portal web.

El Tribunal de Justícia de la Unió Europea (TJUE) es va pronunciar l'octubre del 2019 insistint en l'**obligatorietat que els competeix a les companyies d'informar degudament els usuaris sobre l'ús de cookies als seus portals**, i la necessitat de comptar amb el **consentiment exprés d'aquests darrers**.

Podem distingir "**cookies tècnicament necessàries**", que obeeixen a l'emmagatzematge de **dades vinculades al funcionament de la plataforma digital**, i "**cookies tècnicament no necessàries**", referents a **dades que no constitueixen requisits indispensables per al bon funcionament de la web**.

En essència, el que es pretén és **garantir la protecció de les dades de caràcter personal** de tots els **usuaris** que consulten qualsevol tipus de **plataforma digital**.