PIEMONTE
FONDI STRUTTURALI E DI INVESTIMENTO EUROPEI 2014/2020
fondo sociale europeo
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Cookie e Sessioni

Corso Backend System Integrator

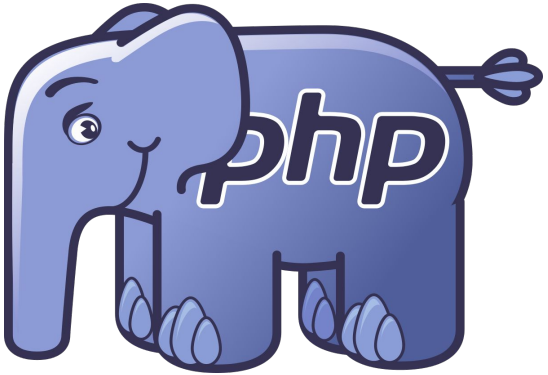
Modulo **Programmazione PHP**

Docente: Dott. Enrico Zimuel

in collaborazione con:
MINISTERO DELL'ISTRUZIONE, DELL'UNIVERSITÀ E DELLA RICERCA
UNIONE EUROPEA
REGIONE PIEMONTE
per una crescita intelligente, sostenibile ed inclusiva
www.regione.piemonte.it/europa2020
INIZIATIVA CO-FINANZIATA CON FSE

Programma

- Cookie
- Sessioni
- Leggere e scrivere in un file



Cookie

- I **cookie** sono degli header HTTP utilizzati per memorizzare informazioni di tipo chiave/valore sul client
- Vengono inviati dal server al client e sono utilizzati, prevalentemente, per tenere traccia di informazioni collegate all'utente che sta visitando il sito

Set-Cookie

```
Set-Cookie: <cookie-name>=<cookie-value>
Set-Cookie: <cookie-name>=<cookie-value>; Expires=<date>
Set-Cookie: <cookie-name>=<cookie-value>; Max-Age=<seconds>
Set-Cookie: <cookie-name>=<cookie-value>; Domain=<domain-value>
Set-Cookie: <cookie-name>=<cookie-value>; Path=<path-value>
Set-Cookie: <cookie-name>=<cookie-value>; Secure
Set-Cookie: <cookie-name>=<cookie-value>; HttpOnly
```

Cookie

- Quando un client riceve una risposta HTTP che contiene uno o più header **Set-Cookie**, deve memorizzare queste informazioni ed inviarle in tutte le future richieste, a patto che rispettino le restrizioni impostate

Sicurezza dei cookie

- Chiunque può modificare il contenuto di un cookie su un client *
- Per questo motivo nei cookie non devono essere memorizzate informazioni sensibili che possano facilitare attacchi o manomissioni di dati

* ad esempio in questo articolo puoi vedere come modificare un cookie con Chrome

Cookie Policy

- L’utilizzo dei cookie è regolamentato dal Garante della Privacy (GU n. 126, 3 giugno 2014)
- Tale provvedimento obbliga tutti i siti internet a richiedere il consenso per l’utilizzo dei cookie, pubblicando una pagina di **Privacy Policy**
- Per maggiori informazioni consultate il sito iubenda.com

Invio di cookie dal client

- Quando un client ha memorizzato un cookie, lo invia nelle richieste HTTP successive tramite l'header **Cookie**

```
Cookie: <cookie-name>=<cookie-value>
```

Cookie in PHP

- In PHP è possibile memorizzare un cookie tramite la funzione setcookie()

```
setcookie (
    string $name,
    [ string $value = "" ],
    [ int $expire = 0 ],
    [ string $path = "" ],
    [ string $domain = "" ],
    [ bool $secure = false ],
    [ bool $httponly = false ]
);
```

Esempio

- Memorizzare l'User-id 112 con scadenza 24 ore, per il dominio example.com

```
setcookie ("User-id", "112", time() + 3600*24, "/", "example.com");
```

- I cookie sono memorizzati nella variabile **$_COOKIE**

```
if (isset($_COOKIE["User-id"])) {
   // Utente già registrato
}
```

Cookie: esempio

foo = bar

Cookie: foo=bar

Set-Cookie: foo=bar

```php
<?php
setcookie('foo', 'bar');
```

```php
<?php
echo $_COOKIE['foo'];
// bar
```

Browser (client)

Server

Cookie in PHP

- Per creare un cookie:
  - setcookie(nome, valore)
- Per leggere un cookie:
  - $_COOKIE

Esercizio

- Implementare un sistema di gestione della cookie policy tramite il controllo di un cookie (es. denominato **Privacy)**
- Creare una pagina **index.php** che verifichi se esista il cookie della privacy con il valore pari a 1
- In caso negativo, la pagina deve visualizzare una Privacy Policy (es. questa) con un tasto (link) per accettarla
- In caso positivo, la pagina deve visualizzare la scritta **“Home page”**

Sessioni

- Le sessioni sono delle informazioni memorizzate sul server, associate al client che effettua la richiesta HTTP
- Il PHP genera un cookie denominato **PHPSESSID**, contenente un identificativo di sessione
- PHPSESSID = MD5(IP . timestamp . numero pseudo-casuale)

$_SESSION

- Per utilizzare la sessione è necessario richiamare la funzione session_start() all'inizio di uno script PHP
- Una volta che la sessione è avviata è possibile memorizzare un valore tramite l'array globale **$_SESSION**

Sessioni su file

- Le sessioni in PHP possono essere memorizzate in diversi modi, il sistema di default utilizza dei file
- Vengono creati dei file di testo contenenti la **serializzazione** della variabile $_SESSION
- I file vengono denominati con il valore del **PHPSESSID** e memorizzati in una cartella (es. /var/lib/php/sessions)

Serializzazione

- La **serializzazione** è un processo che consente di memorizzare lo stato di una variabile in una stringa
- In PHP si utilizzano le funzioni serialize() e unserialize() per serializzare e deserializzare il contenuto di una variabile

Esempio

```php
class Foo {
   public $a = 0;
}
$f = new Foo();
var_dump($f);

$serialize = serialize($f);
var_dump($serialize);

$u = unserialize($serialize);
var_dump($u);
```

```
// var_dump($f);
object(Foo)#1 (1) {
   ["a"]=> int(0)
}
// var_dump ($serialize);
string(26)
"O:3:"Foo":1:{s:1:"a";i:0;}"
// var_dump ($u);
object(Foo)#2 (1) {
   ["a"]=> int(0)
}
```

Sessioni: esempio

PHPSESSID = 1234

Browser (client)

Cookie:
PHPSESSID=1234

Set-Cookie:
PHPSESSID=1234

Server

```php
<?php
session_start();
$_SESSION['foo'] = 'bar';
```

scrive

/var/lib/php/sessions/1234

foo|s:3:"bar";

legge

```php
session_start();
echo $_SESSION['foo'];
// bar
```

Eliminare dati in sessione

- E’ possibile eliminare un dato in sessione con l’utilizzo di unset()

```
unset($_SESSION['User-id']);
```

- Per eliminare i dati in sessione è possibile utilizzare session_unset()
- Per distruggere la sessione è possibile utilizzare session_destroy(); questa funzione non rimuove i dati in $_SESSION

Gestore delle sessioni

- E' possibile modificare il sistema di gestione delle sessioni del PHP
  - modificando la configurazione del session.save_handler del php.ini
  - scrivendo un gestore personalizzato tramite la funzione session_set_save_handler() e/o tramite la classe SessionHandler
- Un esempio di gestore personalizzato delle sessioni con cifratura delle informazioni è disponibile su ezimuel/PHP_Secure_Session

Esercizio (da consegnare)

- Implementare un sito protetto con email e password
- Il sito deve contenere una pagina **login.php** e una pagina **home.php**
- La pagina **login.php** deve contenere il **FORM di Login** e la gestione del controllo della validità dell'email e della password
- Nel caso in cui email e password siano valide il sito deve fare un redirect verso home.php; in caso contrario deve essere riproposta la Login
- La pagina **home.php** può essere visualizzata solo se l'utente ha eseguito con successo la procedura di Login
- L'email e le password degli utenti devono essere memorizzate in un file

Grazie dell’attenzione!

Per informazioni:
enrico.zimuel@its-ictpiemonte.it

POR Piemonte
FSE 2014-2020