

Definizioni utili

1 Cookie

Un cookie è un piccolo testo, contenente informazioni uniche, che il web invia nell'header HTTP. Il browser del client tiene una lista di cookie e siti web. Quando l'utente ritorna su un sito web, il browser prenderà automaticamente i cookie (sempre che non siano scaduti).

- **Secure:** il cookie può essere inviato solo su connessioni HTTPS (default: no)
- **HttpOnly:** il cookie non è accessibile via JavaScript (default: no)
- **SameSite:**
 - **Strict:** può essere inviato solo tra lo stesso dominio
 - **Lax:** può essere inviato tra domini diversi con stessa origine
 - **None:** nessuna restrizione

1.1 Interazione Client-Server

1. Il client fa una richiesta al server
2. Il server risponde con un cookie nell'header HTTP (Set-Cookie)
3. Il client per ogni richiesta successiva invia il cookie nell'header HTTP (Cookie) controllandone la validità

1.2 In PHP

```
1 setcookie("nome","valore",tempo, "path", "domain", secure, httponly); // set
2 $_COOKIE["chiave"]; // get
```

2 Sessione

Il controllo di sessione permette di tener traccia dell'utente durante la sua interazione con un sito web.

2.1 Interazione Client-Server

1. Il client fa una richiesta al server
2. Il server risponde con un cookie contenente la session ID nell'header HTTP (Set-Cookie)
3. Il client per ogni richiesta successiva invia il cookie nell'header HTTP (Cookie) e il server lo usa per identificare la sessione e la validità

2.2 In PHP

```
1 session_start(); // sempre all'inizio del file php
2 $_SESSION["chiave"] = valore // set
3 $_SESSION["chiave"] // get
4 unset($_SESSION["chiave"]) // rilascio della variabile di sessione
5 session_destroy() // distrugge la sessione
```

3 PWA

Una PWA (Progressive Web Application) è un tipo di applicazione web che può essere installata su un dispositivo come applicazione standalone. Devono essere servite su una connessione sicura a causa del Service Worker.

3.1 Application Shell Architecture

Prevede il caricamento iniziale di una struttura base dell'applicazione web (applicazione shell) che rimane costante e viene popolata dinamicamente con i dati. Usa chiamate asincrone e modifica del DOM.

3.2 Service Worker

Script JavaScript che funziona in background, separato dall'applicazione web, che gestisce la cache, le notifiche push e le richieste di rete, permette di fruire dell'applicazione anche offline.

3.3 Manifest JSON

File che descrive come la PWA deve essere visualizzata sui dispositivi degli utenti:

- **name** e **short_name**: nome e nome breve della PWA
- **icons**: array di oggetti che rappresentano le icone che verranno usate dalla PWA
- **start_url**: URL da cui iniziare la navigazione
- **display**: specifica come deve essere visualizzata la PWA
- **theme_color** e **background_color**: per personalizzare l'aspetto della PWA