DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

TITOLO

Indice

Sommario

Introduzione

Capitolo 1

Strumenti Web utilizzati nelle PWA

Questa sezione copre tecnologie web usate nelle Web App che non sono vincolate a uno specifico framework di sviluppo web: di esse si fornirà una descrizione delle relative funzionalità e di una loro possibile implementazione in JavaScript.

Alcuni screenshot di questa sezione sono stati catturati dai Developer Tools (DevTools) del browser: per visualizzarli su PC è sufficiente premere F12 (valido per Firefox e per qualunque browser Chromium-based come Google Chrome o Microsoft Edge). In particolare, le immagini fanno riferimento ai DevTools di Google Chrome.

1.1 Cookies

1.1.1 implementazione

I Cookie hanno un'interfaccia molto primitiva: non sono definite, infatti, delle funzioni per l'aggiunta, rimozione o la modifica di Cookie. L'unico modo per inserire, modificare, leggere ed eliminare Cookie è mediante l'attributo `document.cookie`[1].

Per inserire un nuovo Cookie basta assegnare una nuova stringa a `document.cookie`[1]: tale stringa dovrà presentarsi nel formato `"name=value; optionalField1=optionalValue1; optionalField2= optionalValue2;..."`[1]. La coppia `"name=value"` deve essere specificata, altrimenti l'inserimento fallirà silenziosamente, tutte le coppie successive sono invece opzionali. I campi opzionali sono i seguenti:

- `"expires="`: definisce la data di scadenza del Cookie come stringa UTC; per esprimere una data in tale formato è possibile usare il metodo `toUTCString()` della classe `Date` definita in JavaScript[1]. Se non sono specificati né `expires` né `max-age` allora il Cookie scadrà al termine della sessione[1].

- `"max-age="`: specifica la durata del Cookie in secondi[1].

- `"secure"`: indica che il Cookie deve essere trasmesso solo attraverso un protocollo sicuro[1].

- `"partitioned"`: indica che il Cookie deve essere memorizzato in memoria partizionata[1]. Si Supponga di accedere a un sito `A` che carica

contenuti da un sito di terze parti. Al momento del caricamento quest'ultimo imposta un Cookie sul dispositivo dell'utente. Si supponga ora di spostarsi a un sito `B`, che carica anch'esso contenuti dallo stesso sito di prima. Se il Cookie non è partizionato, allora il sito di terze parti sarà in grado di accedere al Cookie definito precedentemente, difatti, in questo caso, la chiave del Cookie è definita solo dal suo host. Se, invece, il Cookie è partizionato, allora il sito di terze parti non sarà in grado di accedere al Cookie definito durante la navigazione in `A` in quanto, in questo caso, la sua chiave è definita dalla coppia host + sito in cui è caricato il contenuto[2]. Un partitioned Cookie permette di garantire maggiore sicurezza, dato che impedisce il tracciamento *cross-site* dell'utente[2].

- `"domain="`: specifica il dominio a cui il Cookie sarà inviato; se non inserito allora assume un valore di default che coincide con l'host del documento. Si ha inoltre che il Cookie è visibile ai sottodomini solo quando questo parametro è esplicitato[1].

- `"samesite="`: definisce quando inviare il Cookie al server[1]. Esso può assumere valore `lax` se il Cookie può essere inviato solo in occasione di *same-site requests* e *top-level navigation requests*[1][1] (in questo secondo caso, però, il Cookie può essere inviato solo attraverso *safe requests*, come `GET` o `HEAD` ma non `POST`[3]), `strict` se si vuole impedire l'invio del Cookie attraverso cross-site requests [1], `none` se non si applica

[1] cioè una navigazione a un altro sito che porta alla modifica del contenuto della barra degli indirizzi[3]

alcun vincolo[1] (in quest'ultimo caso è però richiesto che sia esplicitato il parametro `secure`[4]).

- `"path="`: specifica il percorso in cui il Cookie è visibile; il Cookie sarà visibile nella cartella specificata e in tutte le subdirectory. Se non inserito allora assume il valore di default `"/"` (la root directory). Il Cookie potrà essere inviato solo dalle parti del sito contenute nel path[4].

In DevTools è possibile mostrare una visuale dettagliata di tutti i Cookie installati, con anche la possibilità di eliminare quelli indesiderati: per browser Chromium-based basta cliccare su "Application" e poi "Cookies" nel menù a sinistra, per Firefox invece "Archiviazione" e poi "Cookie".
Una volta inserito un Cookie esso non può essere modificato direttamente; è possibile solo sostituire questo con un altro: per farlo basta assegnare a `document.cookie` un nuovo Cookie con lo stesso nome di quello da sovrascrivere[5].
Per quanto riguarda l'eliminazione dei Cookie, l'unica strategia disponibile è quella di sovrascrivere il Cookie con un altro avente scadenza già passata[5].
L'attributo `document.cookie` può anche essere acceduto in lettura: in tal caso si ottiene una lista delle sole coppie nome-valore di tutti i Cookie salvati in quel momento[1]. Per visualizzare tutti gli altri parametri è necessario farlo da DevTools.

Bibliografia

[1] "Document: cookie property." https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie.

[2] "Cookies Having independent Partitioned State (CHIPS)." https://developer.mozilla.org/en-US/docs/Web/Privacy/Guides/Privacy_sandbox/Partitioned_cookies.

[3] Prima risposta al thread *"What is the difference between SameSite="Lax" and SameSite="Strict""* di Stack Overflow https://stackoverflow.com/questions/59990864/what-is-the-difference-between-samesite-lax-and-samesite-strict.

[4] "Using HTTP cookies." https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Cookies.

[5] "Javascript cookies." https://www.w3schools.com/js/js_cookies.asp.