**SECURE CONTEXT**

Un contesto sicuro è quando si ha una ragionevole certezza che il contenuto sia stato consegnato in modo sicuro (tramite HTTPS per esempio)

e per il quale la comunicazione con contesti non sicuri è limitata.

Molte API e funzionalità Web sono accessibili solo in un contesto sicuro. L'obiettivo principale dei contesti sicuri è quello di

impedire ai man-in-the-middle attackers di accedere ad API che potrebbero ulteriormente compromettere la vittima di un attacco.

**PERCHÉ ALCUNE FUNZIONALITÀ DOVREBBERO ESSERE LIMITATE?**

Alcune API sul Web sono molto potenti e consentono a un utente malintenzionato di eseguire quanto segue e altro:

- Invadere la privacy di un utente;

- Ottenere l'accesso a basso livello del computer di un utente;

- Ottienere l'accesso a dati, come le credenziali di un utente.

**QUANDO UN CONTESTO È CONSIDERATO SICURO?**

Un contesto sarà considerato sicuro quando viene consegnato in modo sicuro e quando non può essere utilizzato per fornire accesso a API sicure in un contesto non sicuro. In pratica, ciò significa che data una pagina in un contesto sicuro tutte le pagine lungo la sua catena devono essere anch'essere state consegnate in modo sicuro.

Ad esempio, una pagina consegnata in modo sicuro tramite TLS non è considerata un contesto protetto se ha un documento padre o antenato che non è stato consegnato in modo sicuro; in caso contrario, la pagina sarebbe quindi in grado di esporre le API sensibili all'antenato distribuito in modo non sicuro. Allo stesso modo, se un documento rilasciato da TLS viene aperto in una nuova finestra da un contesto non sicuro senza che venga specificato nessun altro , la finestra aperta non viene considerata come un contesto protetto.

File consegnati localmente come http: // localhost e file: // path sono considerati come consegnati in modo sicuro.

I contesti non locali devono essere pubblicati su https: // e dove i protocolli utilizzati non devono essere considerati deprecati.

**RILEVAMENTO DI FUNZIONALITÀ**

Le pagine possono utilizzare il rilevamento delle funzioni per verificare se si trovano in un contesto protetto o meno utilizzando il valore isSecureContext booleano, esposto nell'ambito globale.

if (window.isSecureContext) {

// Page is a secure context so service workers are now available

navigator.serviceWorker.register("/offline-worker.js").then(function () {

...

});

}

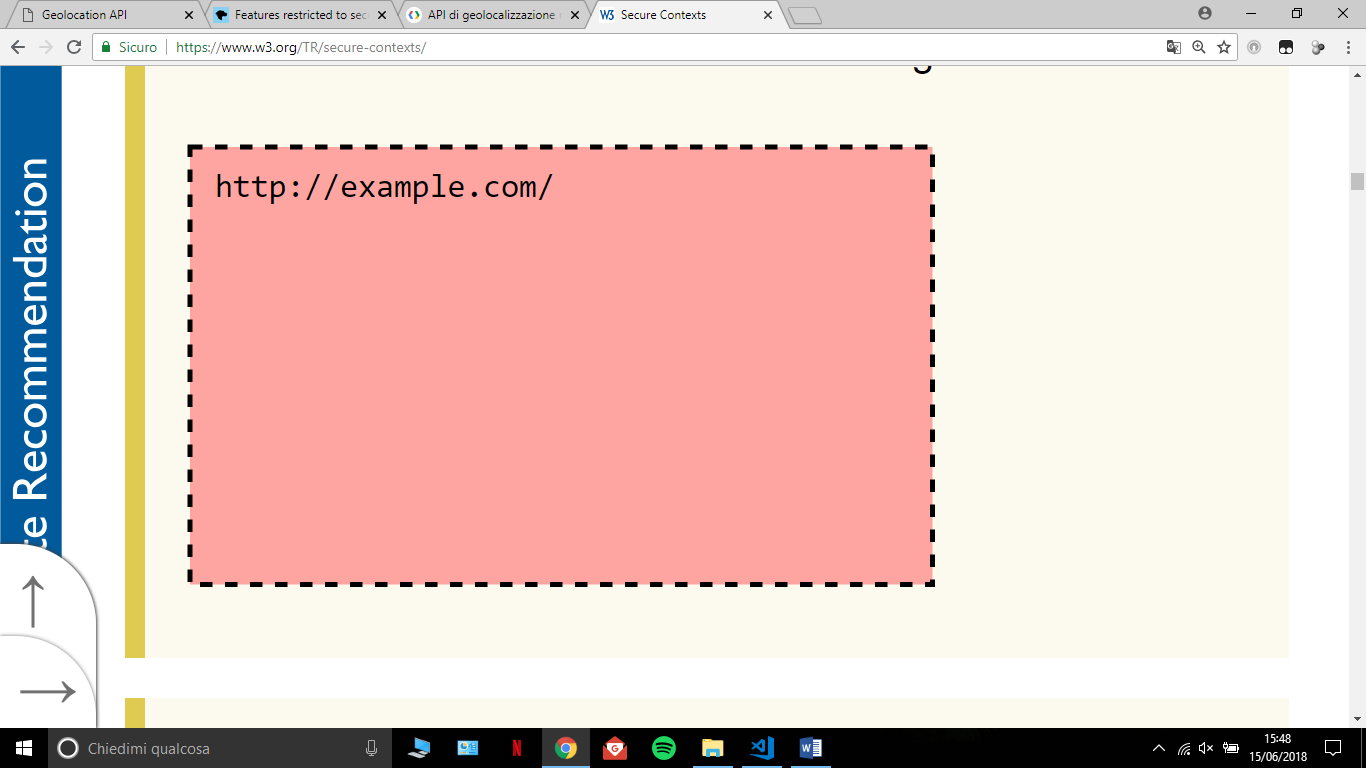
**CONTESTI DI SICUREZZA (DA W3C)**

<https://www.w3.org/TR/secure-contexts/>

**TOP-LEVEL DOCUMENTS**

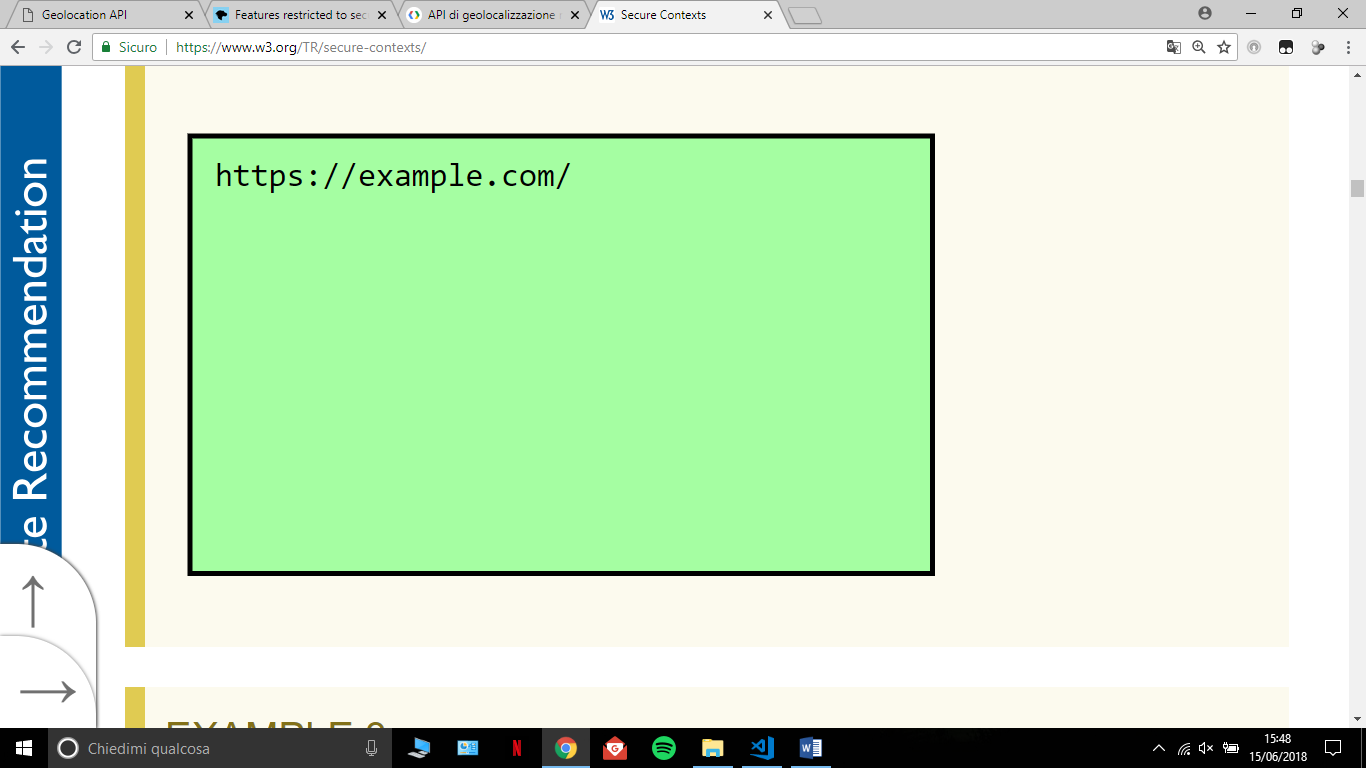
1 – HTTP

http non è un contesto sicuro, perché non viene distribuito su un canale crittografato.



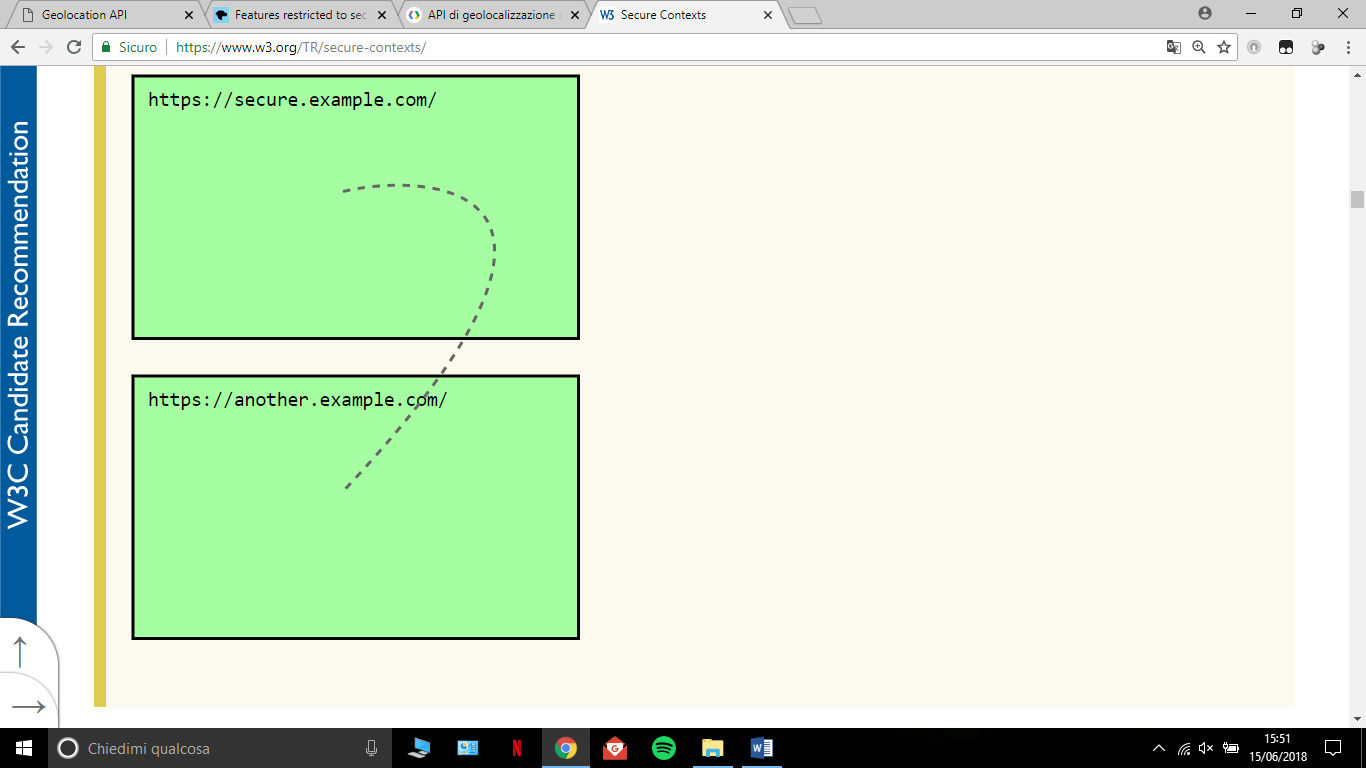
2 – HTTPS

https è sicuro perché distribuito su un canale crittografato.



3 – HTTPS -> HTTPS

se viene aperta una pagina https da una pagina https, anche la pagina aperta sarà sicura.



4 – HTTP -> HTTPS

Se una pagina http apre una finestra http in una nuova finestra, lo stato della nuova finestra dipende da come è stato aperto. Se la pagina http può ottenere un riferimento alla pagina https, la nuova finestra non è un contesto sicuro.

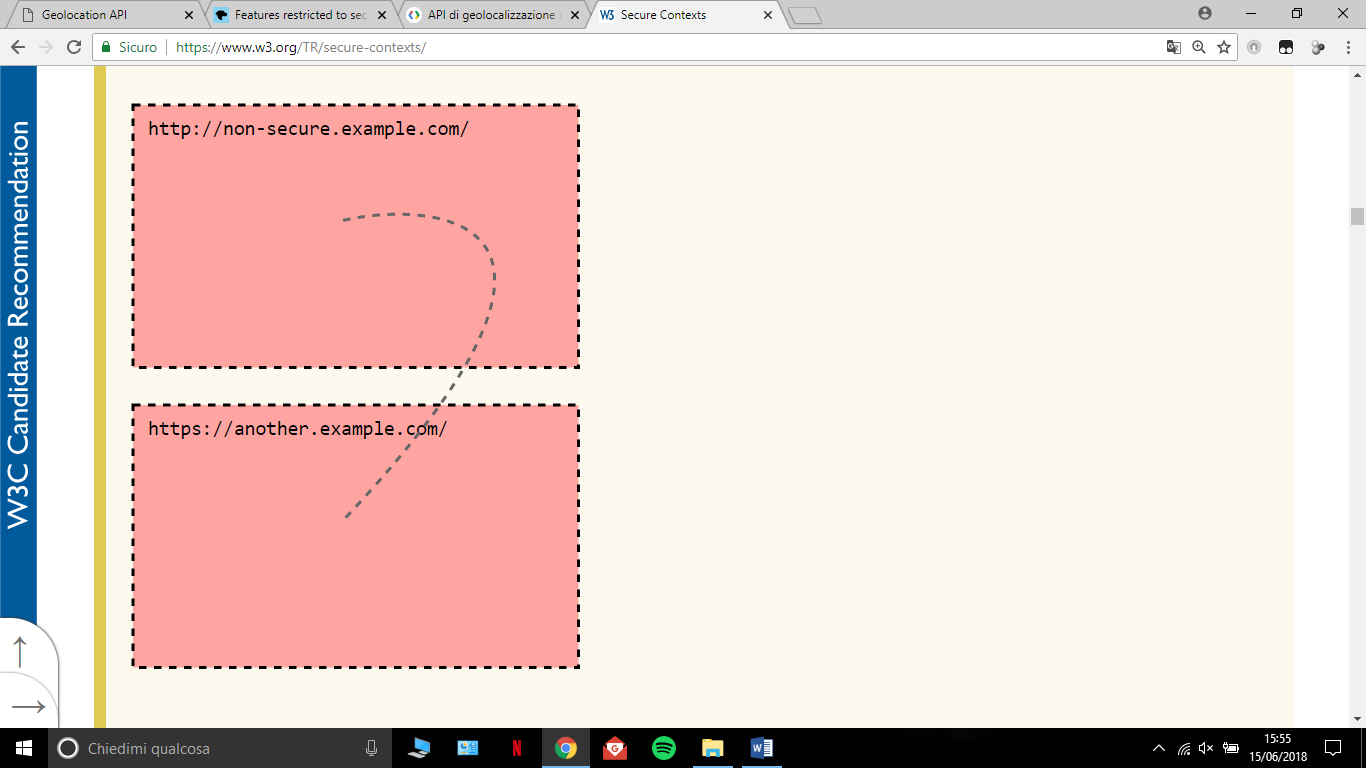
Ciò significa che quanto segue produrrà entrambi contesti non sicuri:

<a href="https://example.com/" target="\_blank"> Link! </a>

<Script>

var w = window.open ("https://example.com/");

</ Script>



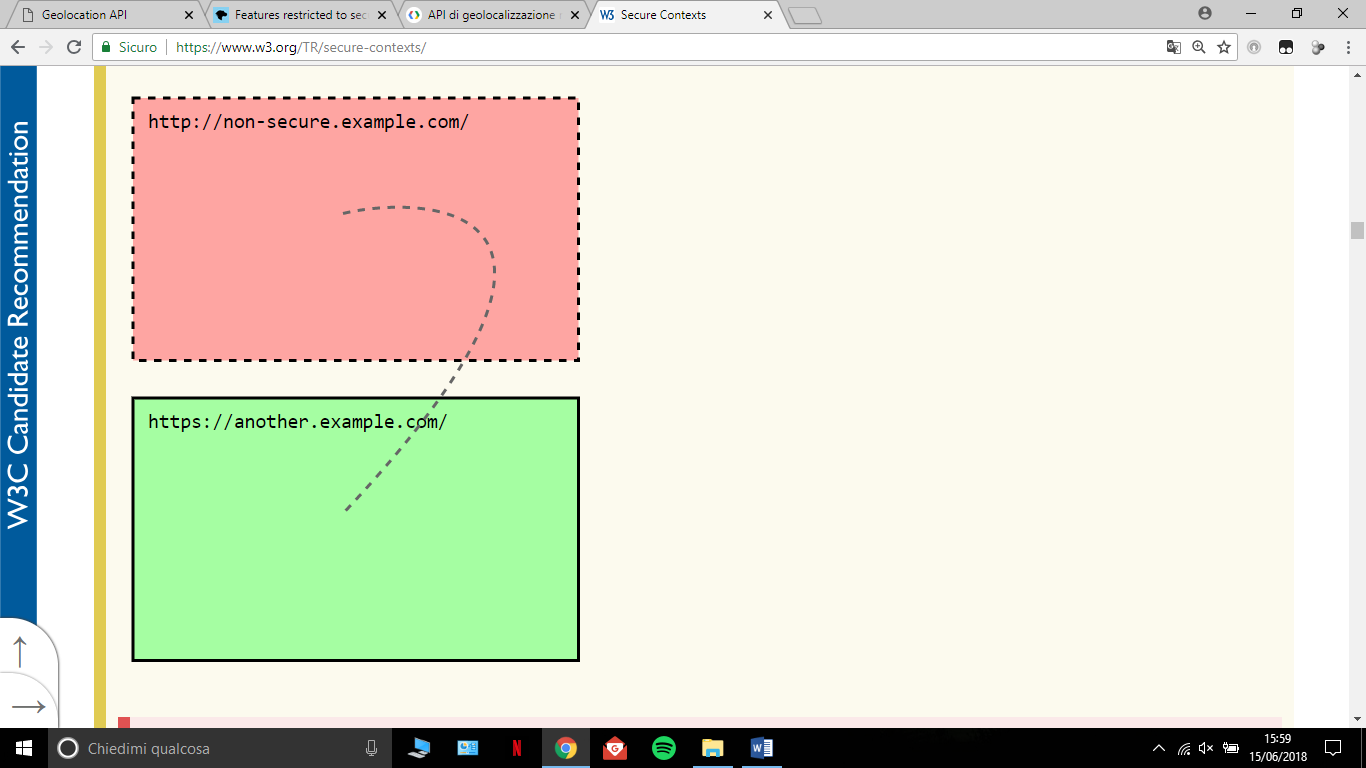
Il collegamento può essere interrotto tramite la relazione di collegamento [noopener](https://html.spec.whatwg.org/multipage/semantics.html#link-type-noopener) (Questa parola chiave non crea un [collegamento ipertestuale](https://html.spec.whatwg.org/multipage/links.html#hyperlink) , ma [annota](https://html.spec.whatwg.org/multipage/links.html#hyperlink-annotation) tutti gli altri collegamenti ipertestuali creati dall'elemento ) il che significa che quanto segue genererà contesti sicuri:

<a href="https://example.com/" rel=" [noopener](https://html.spec.whatwg.org/multipage/semantics.html#link-type-noopener) "target="\_blank"> Link! </a>

<Script>

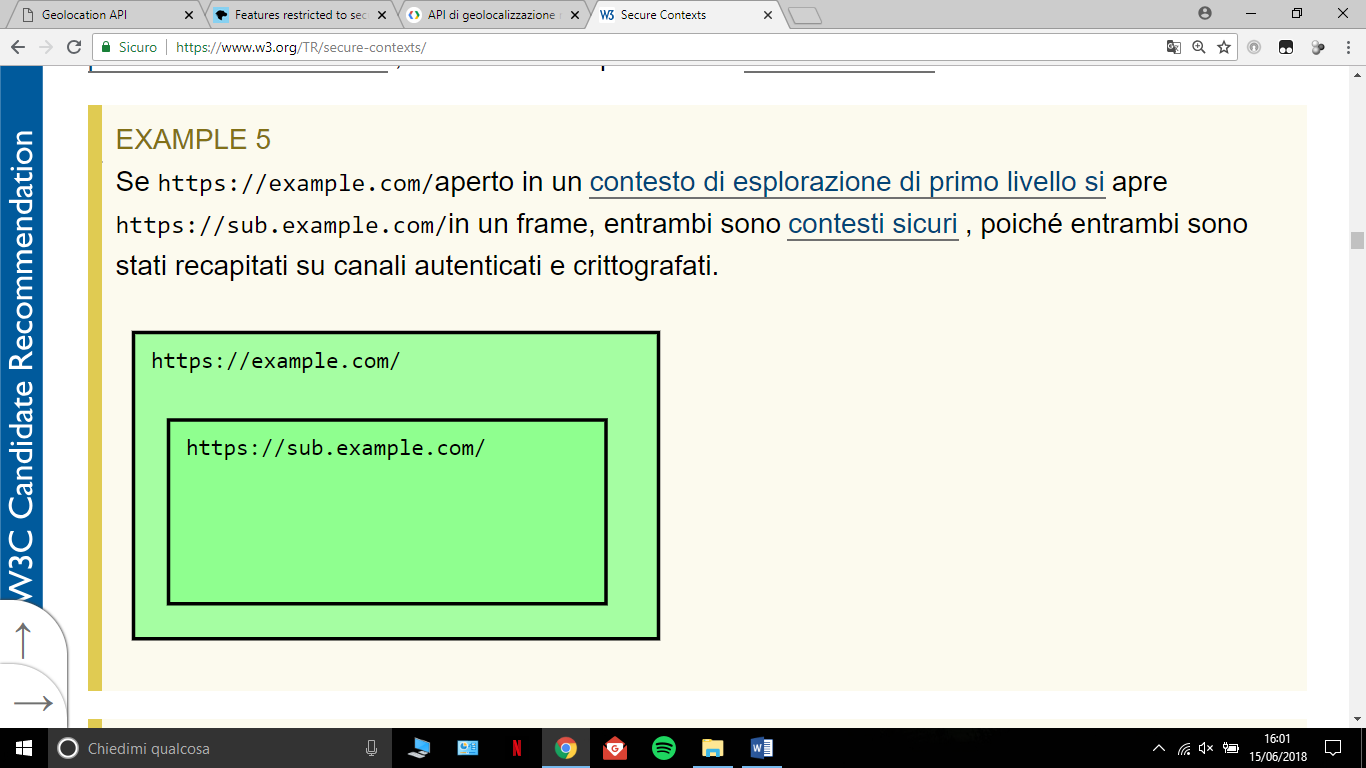
var w = window.open ("https://example.com/", "", " [noopener](https://html.spec.whatwg.org/multipage/semantics.html#link-type-noopener) ");

</ Script>

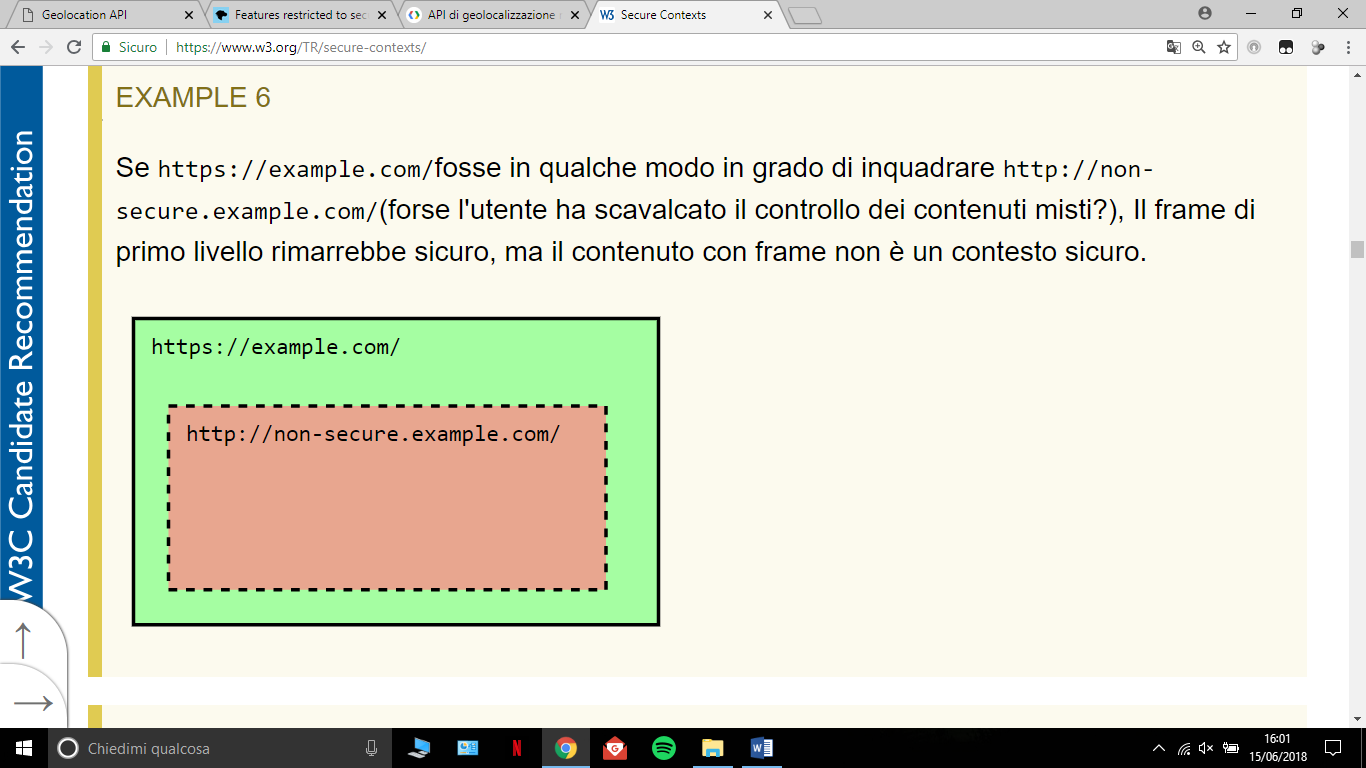


**FRAME DOCUMENTS**

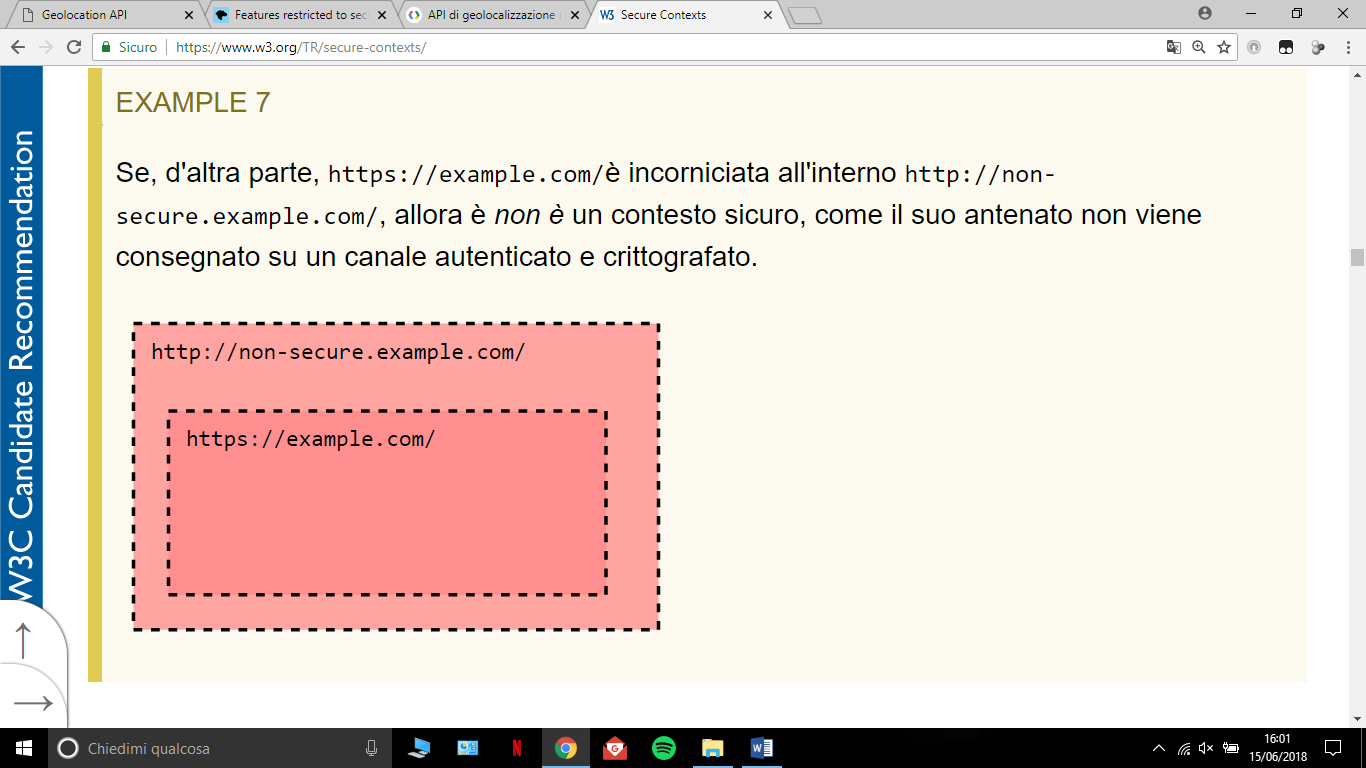
5 – HTPPS DENTRO HTTPS



6 – HTTP DENTRO HTTPS



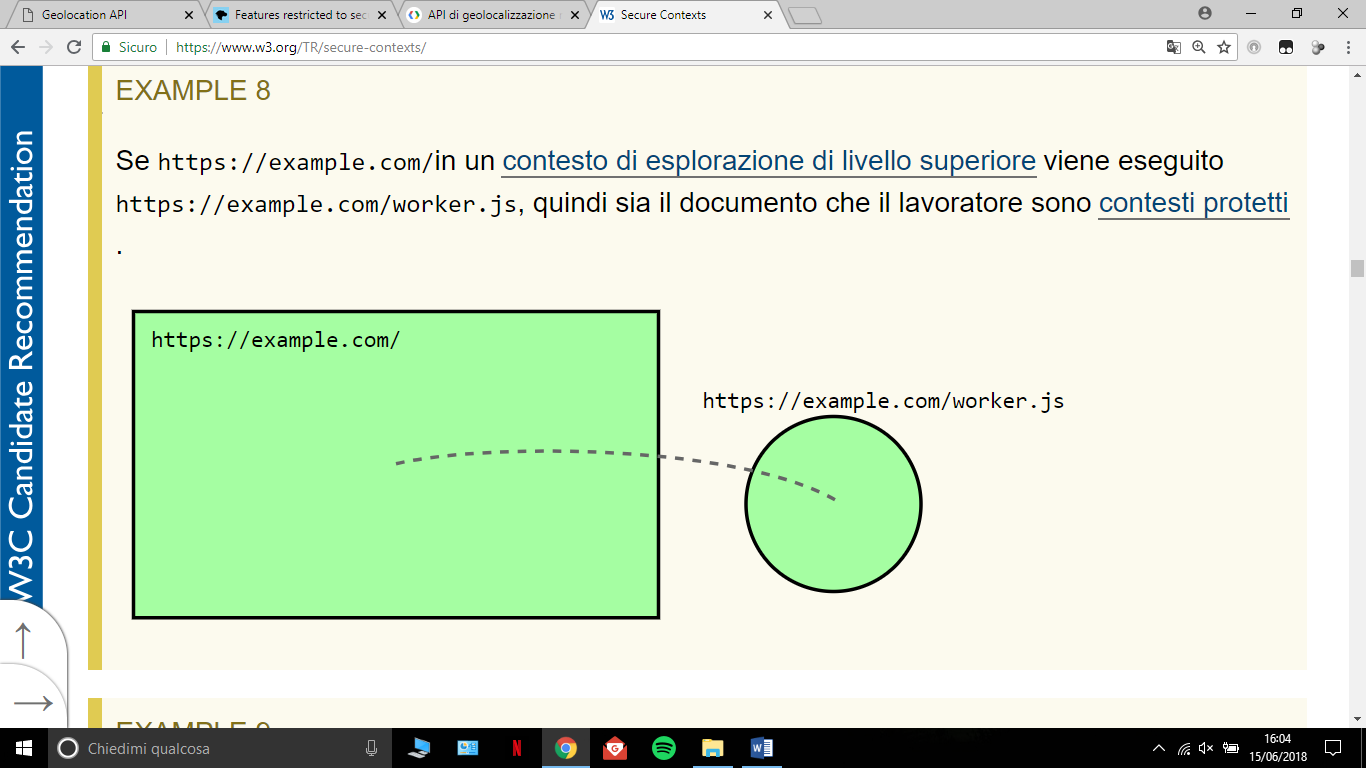
7 – HTTPS DENTRO HTTP



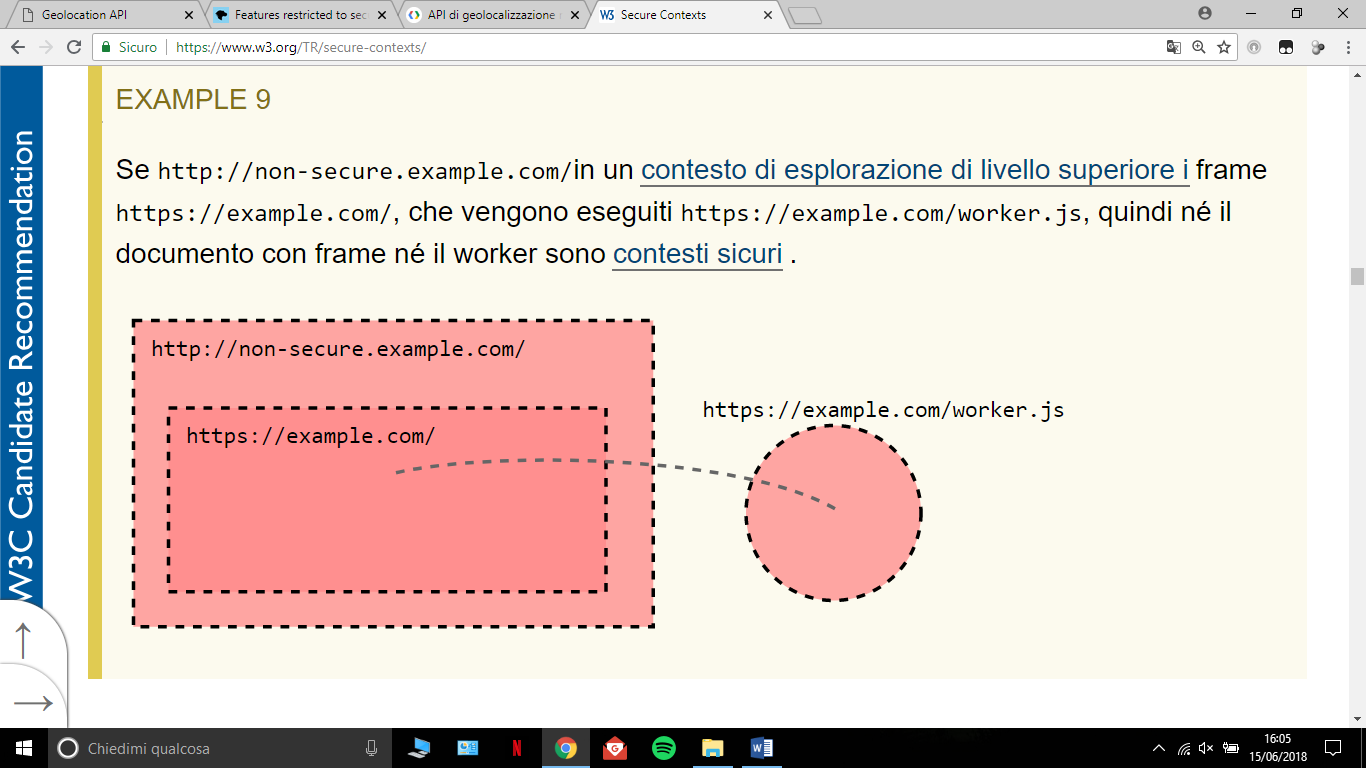
**WEB WORKERS**

I web workers sono simili ai documenti frame documents. Sono [contesti sicuri](https://www.w3.org/TR/secure-contexts/#secure-context) quando vengono consegnati da [origini potenzialmente affidabili](https://www.w3.org/TR/secure-contexts/#potentially-trustworthy-origin) , solo se il loro proprietario è di per sé un [contesto sicuro](https://www.w3.org/TR/secure-contexts/#secure-context) :

8 – HTTPS -> HTTPS



9 – HTTPS(DA HTPP) -> HTTPS

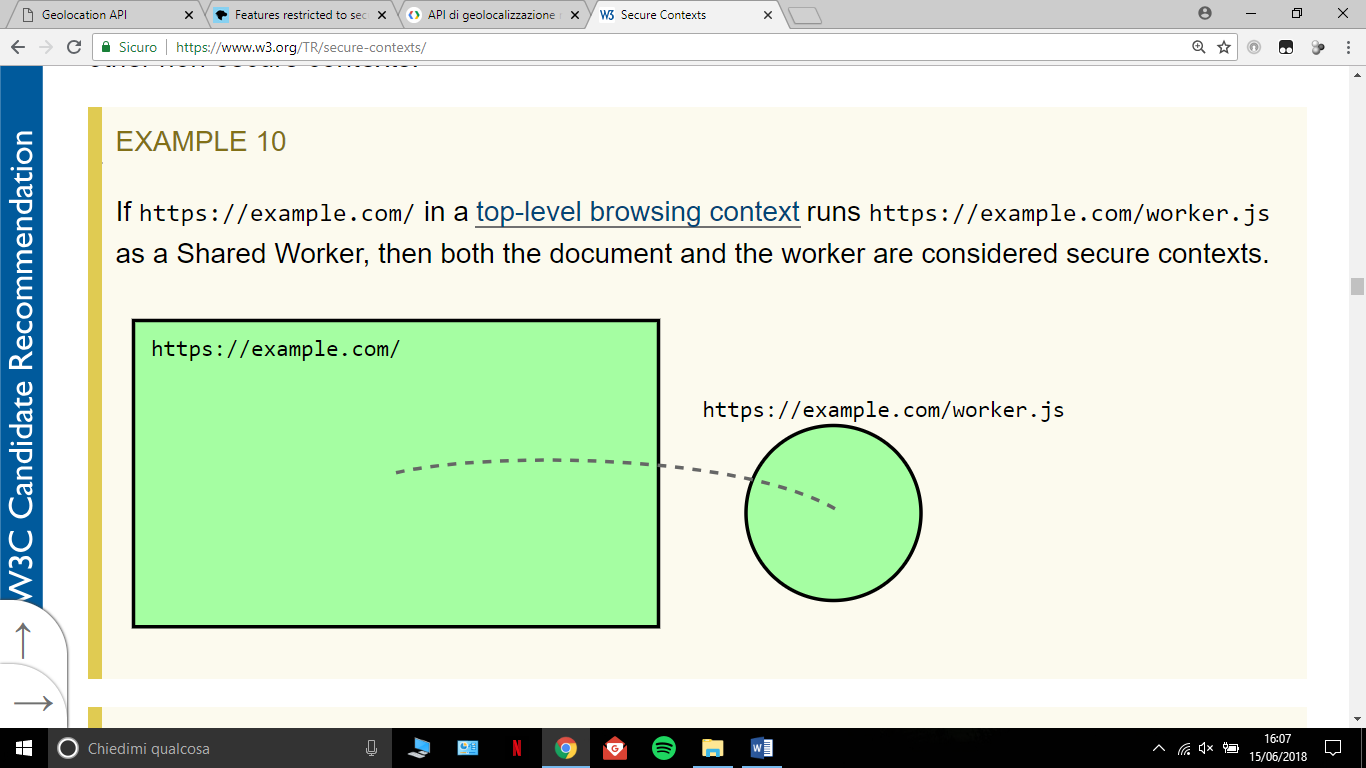


**SHARED WORKERS**

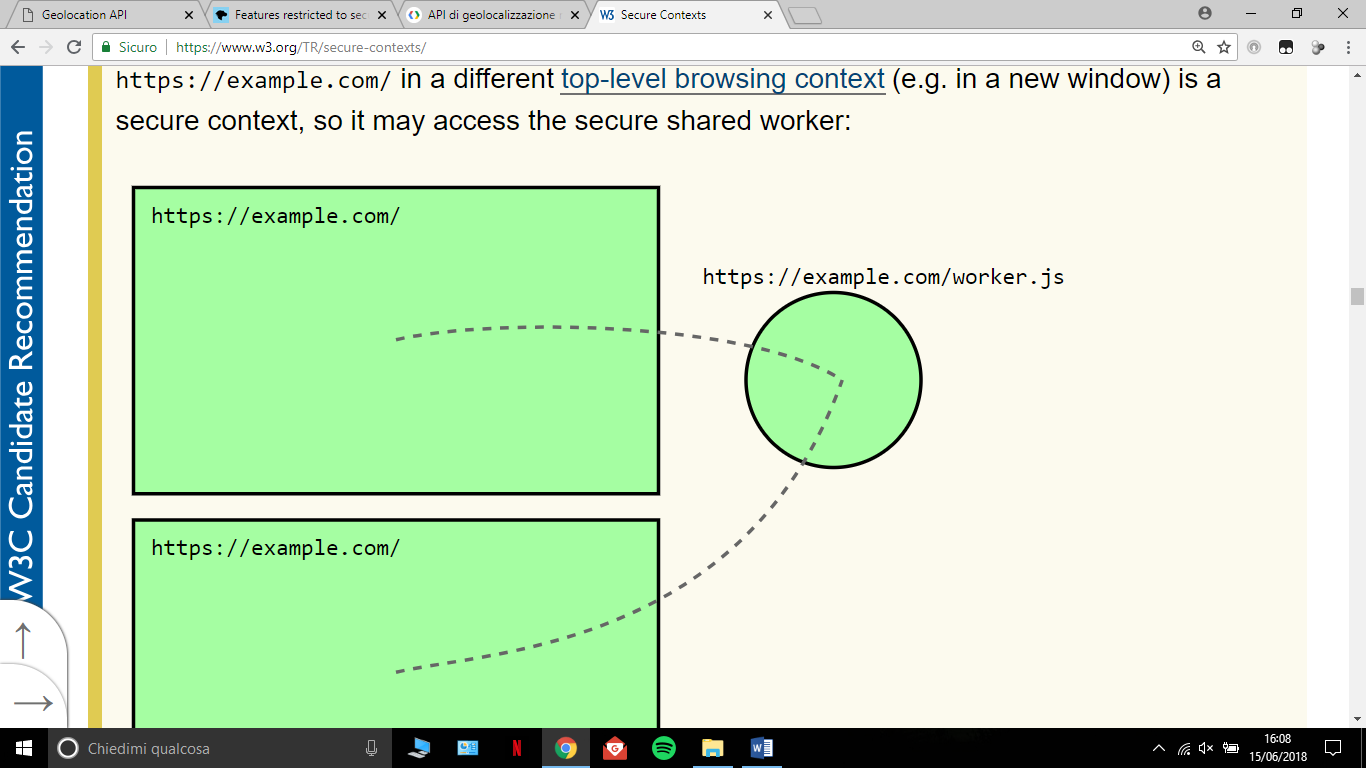
Più contesti possono essere associati a un shared worker. Se un [contesto sicuro](https://www.w3.org/TR/secure-contexts/#secure-context) crea un shared worker, allora è un [contesto sicuro](https://www.w3.org/TR/secure-contexts/#secure-context) e può essere collegato solo da altri [contesti sicuri](https://www.w3.org/TR/secure-contexts/#secure-context). Se un contesto non sicuro crea un shared worker, allora è non è un [contesto sicuro](https://www.w3.org/TR/secure-contexts/#secure-context), e può essere attaccato solo da altri contesti non protetti.

10 –

Se <https://example.com/> in un  [top-level browsing context](https://www.w3.org/TR/html51/browsers.html#top-level-browsing-context) esegue <https://example.com/worker.js> come uo shared worker, allora sia il documento che lo shared worker sono considerati contesti sicuri.

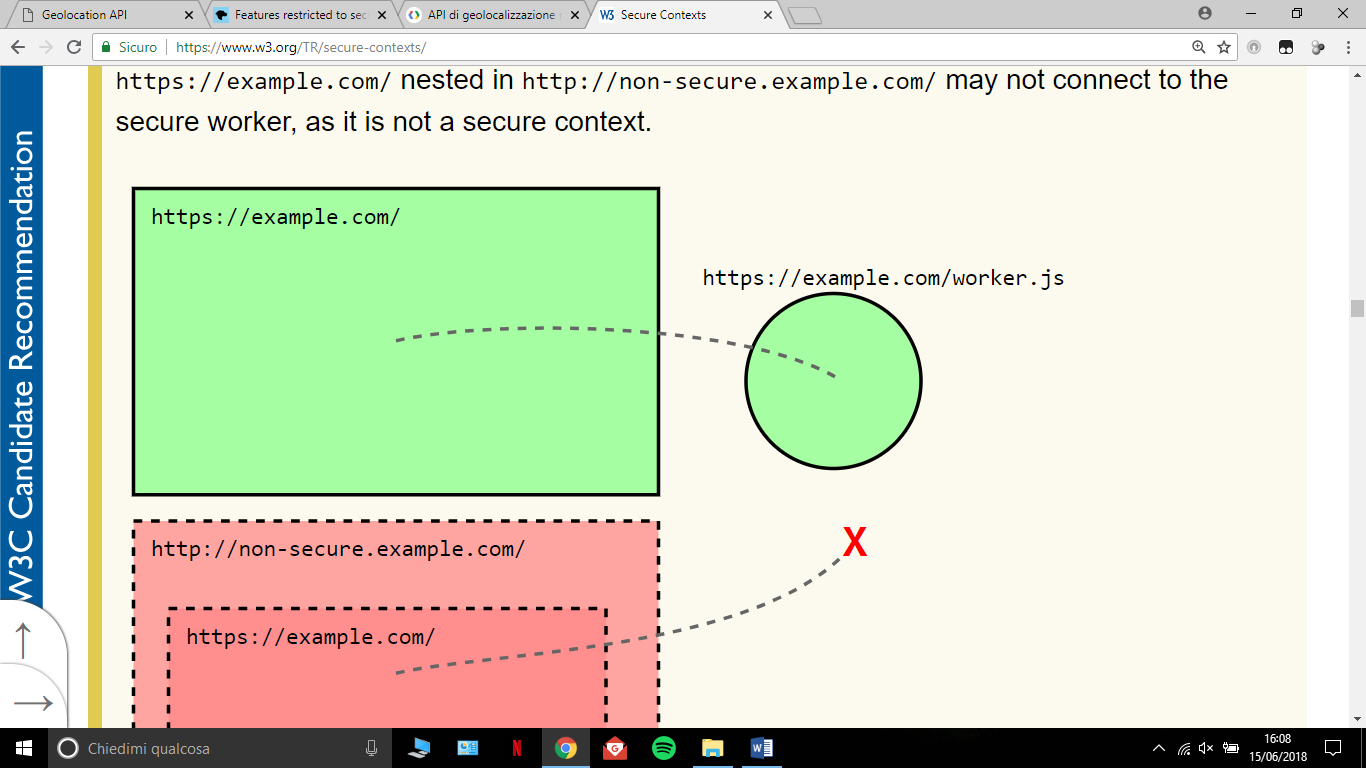


11 –

https://example.com/in un [contesto di navigazione di primo livello](https://www.w3.org/TR/html51/browsers.html#top-level-browsing-context) diverso (ad esempio in una nuova finestra) è un contesto sicuro, quindi può accedere allo shared worker sicuro:

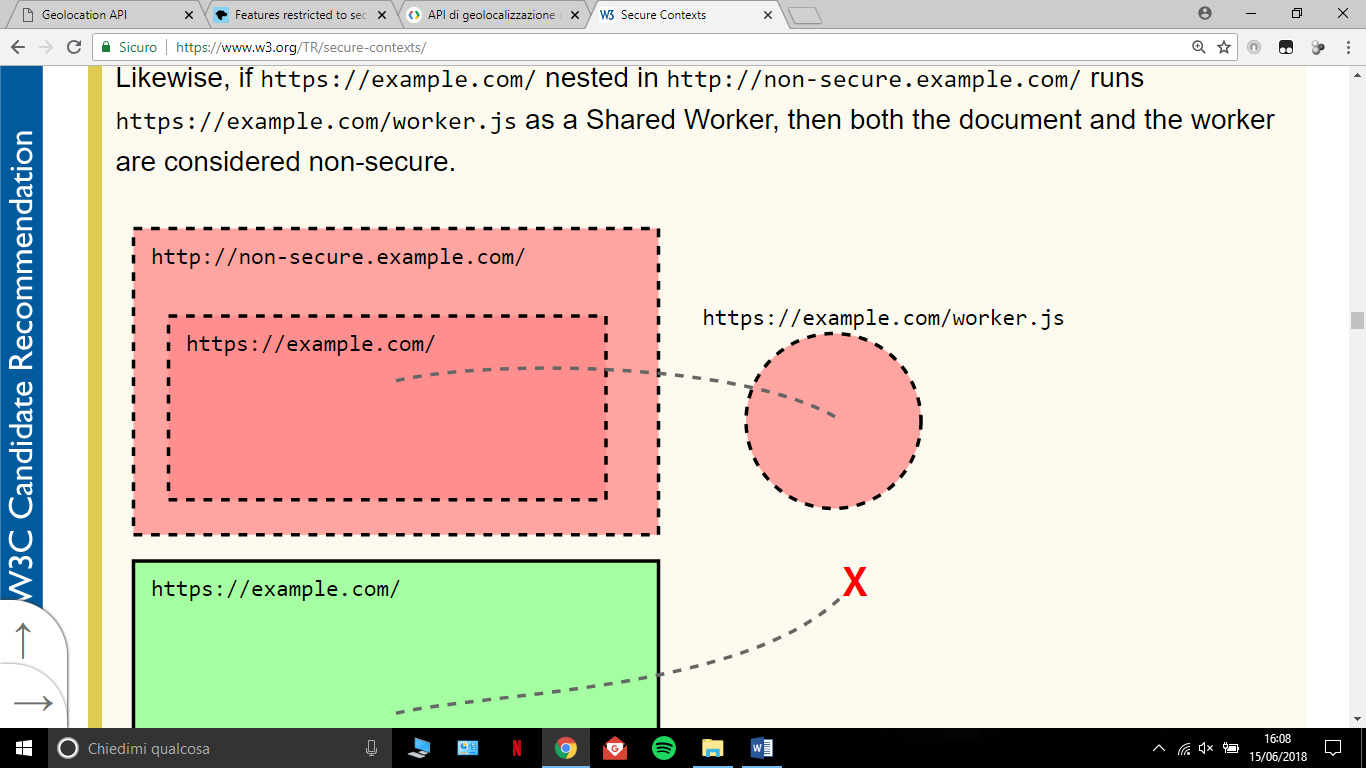
12 –

<https://example.com/> nidificato in <http://non-secure.example.com/> potrebbe non connettersi allo shared worker, in quanto non è un contesto sicuro.



13 –

Allo stesso modo, se <https://example.com/> nidificato in <http://non-secure.example.com/> esegue <https://example.com/worker.js> come uno shared worker, allora sia il documento che lo shared worker sono considerati non sicuri.



**SERVICE WORKERS**

I service workers sono sempre [contesti sicuri](https://www.w3.org/TR/secure-contexts/#secure-context). Solo i [contesti protetti](https://www.w3.org/TR/secure-contexts/#secure-context) possono registrarli e possono avere solo clienti che sono [contesti sicuri](https://www.w3.org/TR/secure-contexts/#secure-context) .

14 –

Se <https://example.com/> in un top-level browsing context registra <https://example.com/service.js> , allora sia il documento che il Service Worker sono considerati contesti sicuri.

