



Consuetudini dello sviluppo Web

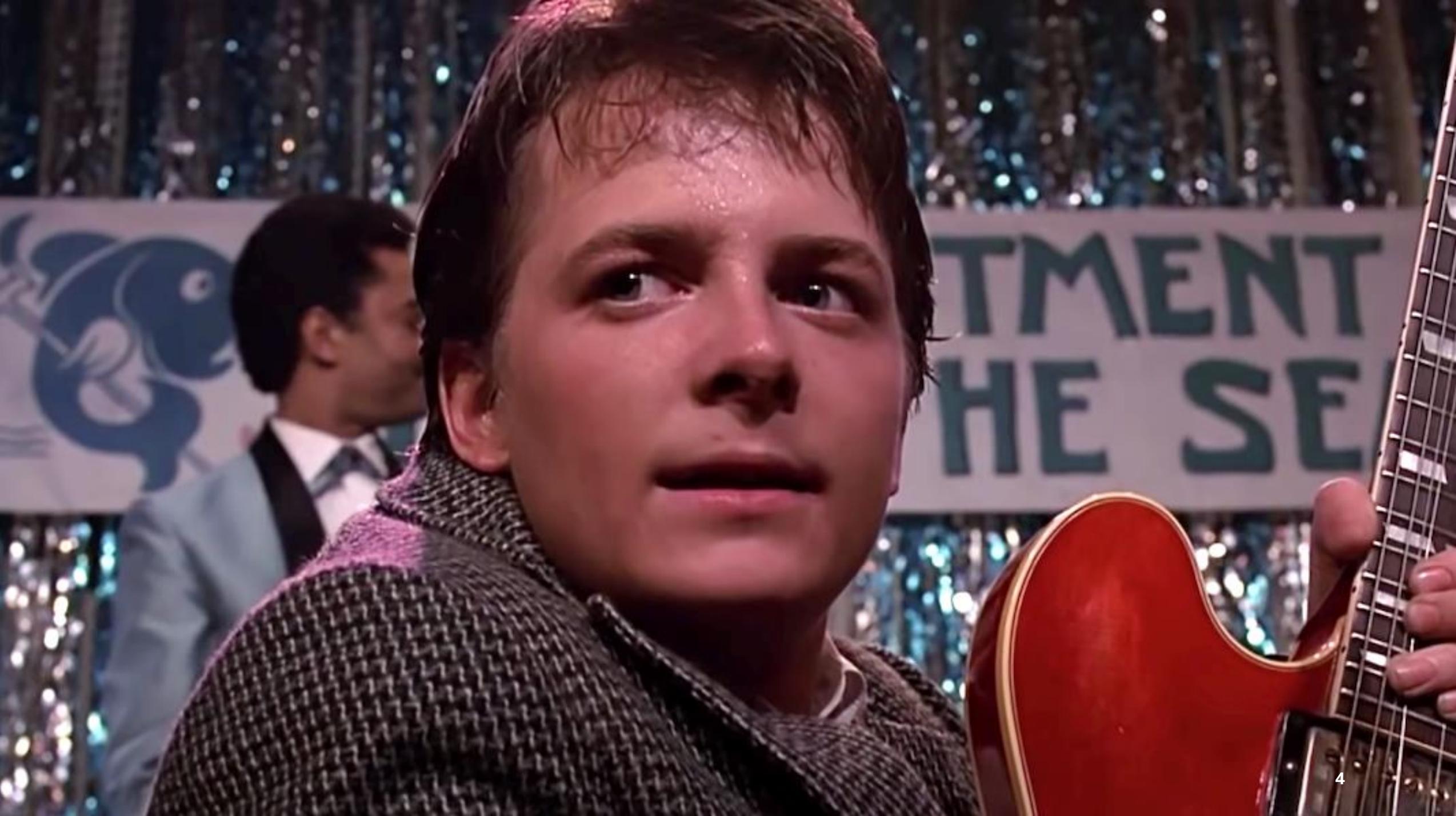
Laboratorio di interfaccia uomo-macchina 2023-2024

Panoramica

- Nella progettazione di un caso d'uso digitale, occorre tenere presente una serie di informazioni che **condizioneranno** inevitabilmente l'interazione del vostro **utente** con il vostro sistema.
- Il vostro utente avrà un suo specifico background culturale che sarà inevitabilmente condizionato da tutti gli altri prodotti che avrà utilizzato prima di arrivare a voi.
- Diventa fondamentale quindi contestualizzare la vostra progettazione in un determinato periodo storico, considerando tutte le consuetudini più diffuse e quelle emergenti.

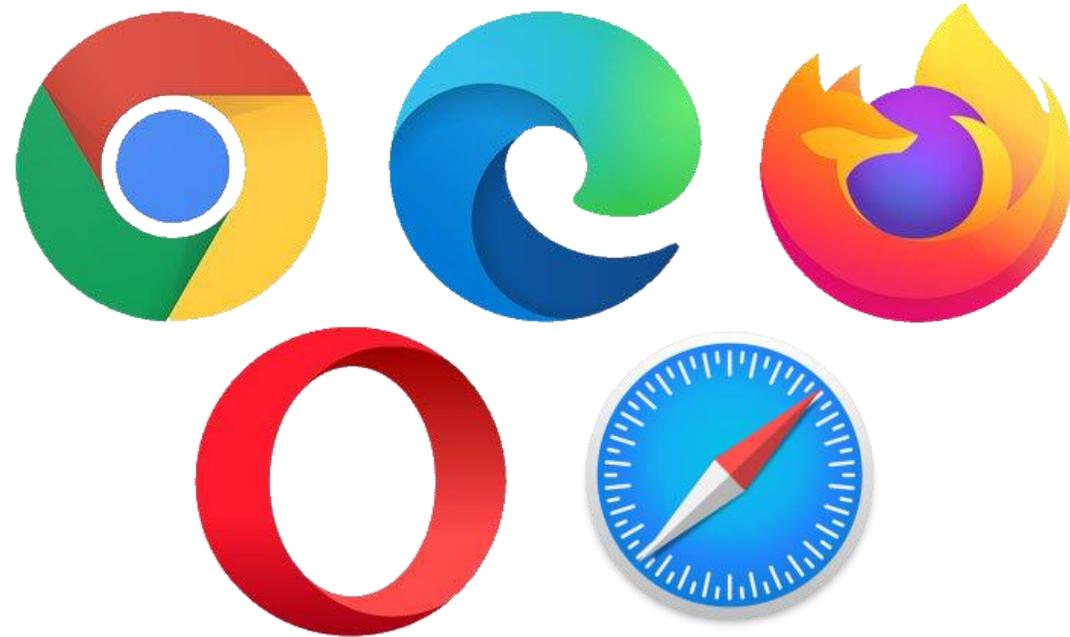
La sfida

- In uno scenario di consuetudini di utilizzo in continuo fermento ed evoluzione, il vostro compito sarà quello di realizzare un applicativo che **rimanga sempre attuale**, e che sappia evolvere nel tempo.
- Rimane una nostra sfida progettuale adottare tutte le accortezze per garantirci questa possibilità
- In questa corsa frenetica occorre fare attenzione ad alcuni rischi...



Il Browser

- E' il **principale veicolo** dell'esperienza utente.
- E' nel browser che l'utente ha il primo **metro di paragone** con qualcos'altro che ha utilizzato prima.



Standardizzazione diretta da W3C®



Consorzio internazionale con lo scopo di :

- Ottenere una standardizzazione del web
- Indirizzare i produttori verso nuove funzionalità
- Indirizzare il mercato verso l'abbandono dei vecchi browser che non supportano gli standard
- E' possibile verificare il livello di compatibilità sui singoli elementi dal sito <https://caniuse.com/>

La strada della standardizzazione

La progressiva standardizzazione dei browser più moderni ha portato a mancare una delle principali motivazioni del successo di jQuery, che si presentava come lingua franca in un contesto di browser non standardizzati.

L'attenzione alle performance, sia di caricamento che di esecuzione del client, legato alla disaffezione verso la sintassi jQuery, sta portando alla sua progressiva dismissione.

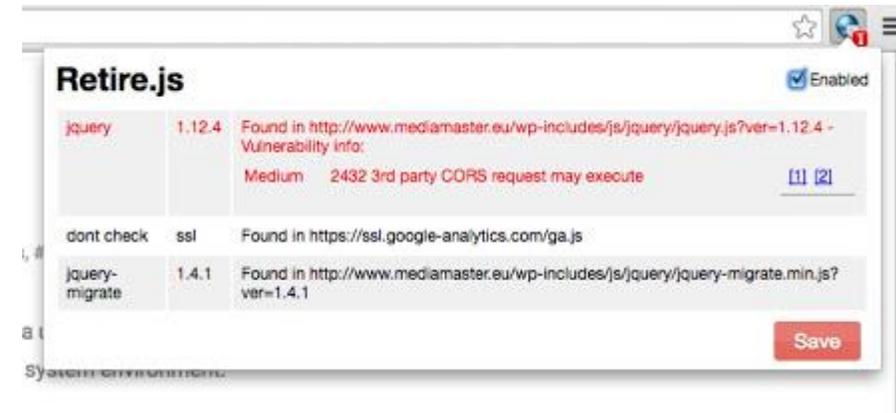
Ad esempio Bootstrap 5 non usa più jQuery come dipendenza



Attenzione alle vulnerabilità

La tendenza a limitare le librerie utilizzate e a mantenerle aggiornate non è solo legato all'ottimizzazione di velocità e compatibilità, è sempre più questione di sicurezza.

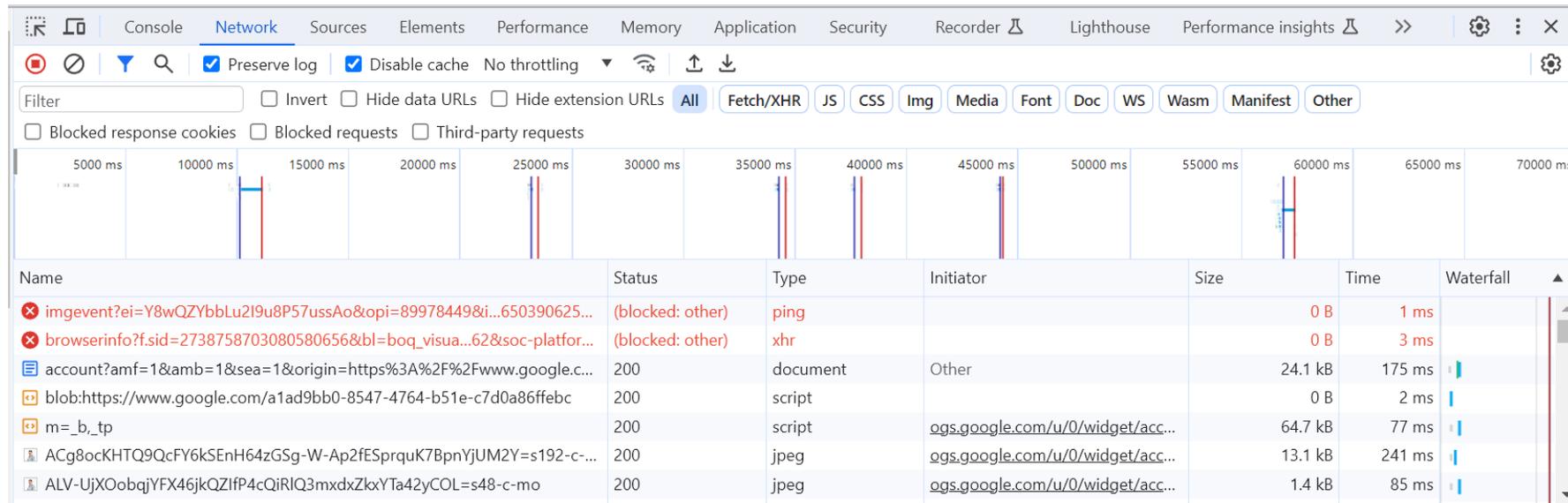
Esistono suggerimenti nell'IDE o plugin come Retire.js che segnalano se una libreria utilizzata ha falle di sicurezza note.



Il Browser

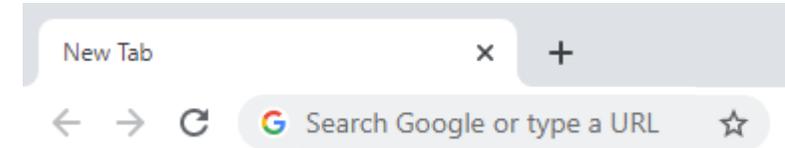
Un buon alleato “DevTools (F12)”

- Esaminare gli elementi – Elements
- Controllare i sorgenti – Sources/Console
- Controllare interazioni - Network



Il Browser

Uso delle tab

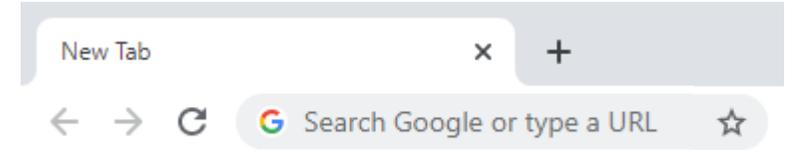


- Le tab garantiscono di poter lasciare un contenuto pronto per la navigazione mentre si continua nel flusso di navigazione corrente.
- All'interno di applicativi abilita anche il confronto tra due pagine o la copia dei valori
- Permettere di aprire un applicativo web in più tab è dato per scontato.
- Per farlo però occorre però stare attenti a due requisiti in progettazione:
 - L'url deve contenere tutte le informazioni per procedere alla nuova apertura della pagina (stateless)
 - Eventuali dati utente necessari per l'utilizzo dell'applicativo devono essere mantenuti in sessioni separate e non condivise tra le tab. Attenzione, anche i cookie sono condivisi.

Il Browser

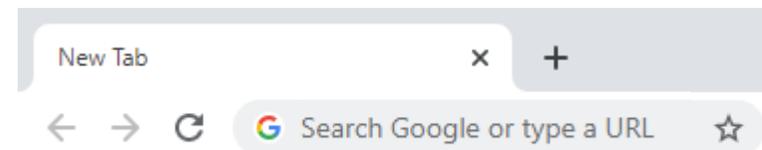
Titolo della pagina

- Nel momento in cui sono presenti più tab è molto utile attribuire un titolo corretto alla pagina, in modo da poterla selezionare correttamente senza doverla aprire.
- Occorre mostrare il nome del dato visualizzato o del caso d'uso come prima informazione.



Il Browser

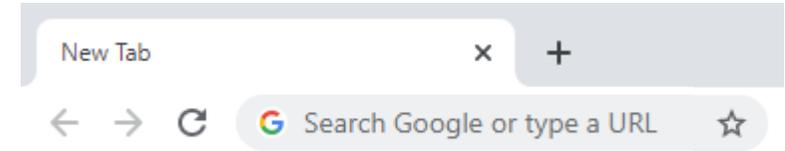
Pulsante back/forward



- Il pulsante back e forward consentono di cambiare le pagine viste nel corrente flusso di navigazione.
- L'utente si aspetta di poterle utilizzare.
- Prestate dunque attenzione:
 - Nelle navigazioni Ajax in cui non viene effettivamente ricaricata la pagina è possibile aggiungere l'url nella history con `window.history.pushState()`
 - Nelle navigazioni in cui non si vuole che la navigazione finisca nelle'url anche se di fatto abbiamo caricato una nuova pagina si può indicare che la nuova pagina sostituisca quella corrente aprendola con `window.location.replace(url);`

Il Browser

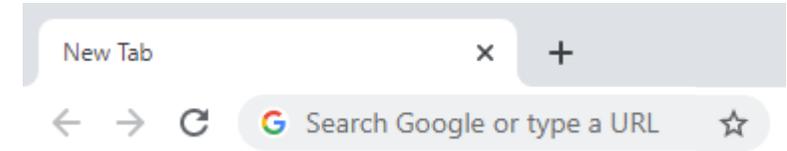
Salvataggio nei preferiti



- Se l'utente utilizzerà un caso d'uso con frequenza, è molto probabile che questo finisca nei preferiti.
- Per abilitare il salvataggio nei preferiti occorre che:
 - L'url deve contenere tutte le informazioni per procedere alla nuova apertura della pagina (stateless)
 - Eventuali dati utente necessari per l'utilizzo dell'applicativo non devono interferire con quelli di sessioni separate.
 - Il caso d'uso deve disporre di un url nel punto di ingresso desiderato

Il Browser

Pulsante aggiorna



- Il pulsante aggiorna permette di ricaricare la pagina corrente nel suo stato originario
- L'utente si aspetta che eventuali modifiche della pagina corrente vadano perse e che la pagina torni nello stato iniziale.
- L'utente non si aspetta di dover ricostruire lo stato impostato in pagine precedenti. (Esempio aggiornamento la pagina di modifica di un utente e ritorno alla maschera in cui cerco l'utente da editare)

Aggiorna

(problematica)



L'utente Pippo lavora nel backoffice e si trova a dover pigiare il pulsante salva su una maschera.

Poi, visto che adora il nostro applicativo, salva la pagina tra i preferiti.

Qualche giorno dopo Pippo clicca sul preferito che si era salvato per tornare subito alla maschera d'interesse e... che succede?

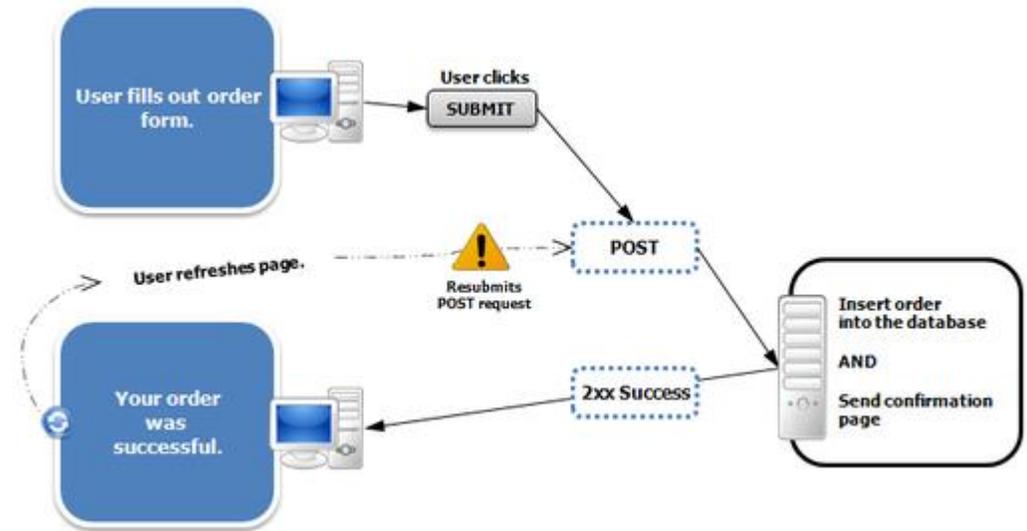
E se invece, dopo aver salvato, Pippo decide di aggiornare la pagina, che succede?

Post Redirect Get - PRG

(problematica)

Nel primo caso (preferito) Pippo chiama in GET una pagina che risponde al verbo http POST (ottiene un 405 se il nome è diverso tra la prima get e la seguente post)

Nel secondo caso (aggiorna) lo stesso form viene mandato più volte al server e, a seconda di che tipo di form si tratta, potrebbe causare problemi



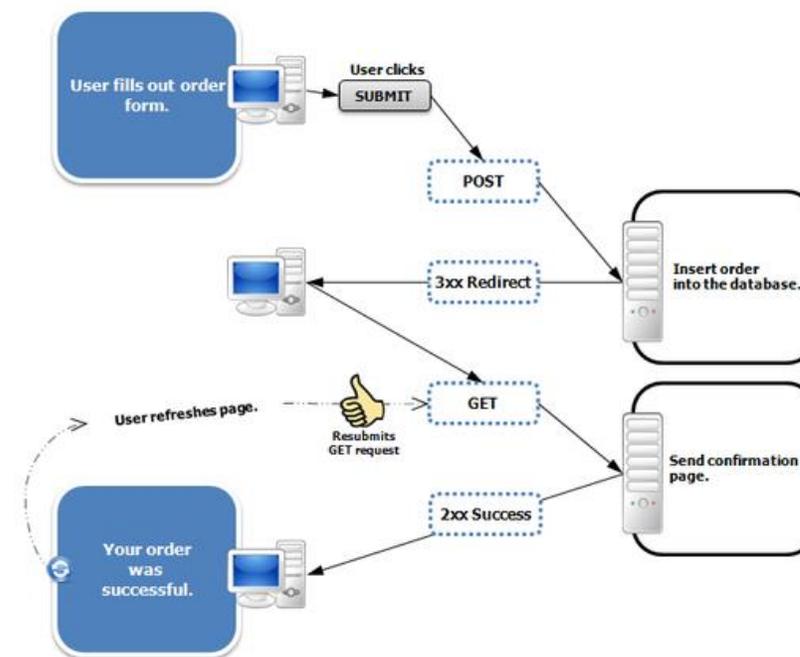
Post Redirect Get - PRG

(soluzione)

La soluzione a questo tipo di problemi è il pattern PRG (Post-Redirect-Get)

Quando l'utente esegue un'azione che genera una chiamata post il server risponde con un redirect ad una get.

Ora se l'utente decide di aggiornare la pagina o di salvarla nei preferiti, salverà l'ultima get e non la post.



La velocità

Velocity 09: Marissa Mayer, "In Search of... A better, fast



Fino 09:09

<https://youtu.be/WFsQvcdmLxc?t=14>

Marissa Mayer

È stata la prima ingegnere donna assunta da Google e una dei primi 20 impiegati della compagnia di cui fa parte dal giugno 1999



Il device

Il device è un altro fattore che influenza notevolmente l'interazione e ci costringe a non considerare più costanti elementi essenziali come:

- **Dimensione** del monitor, della quale è sempre più importante considerare dimensioni fisiche e risoluzione
- **Modalità di interazione**, mouse o touch

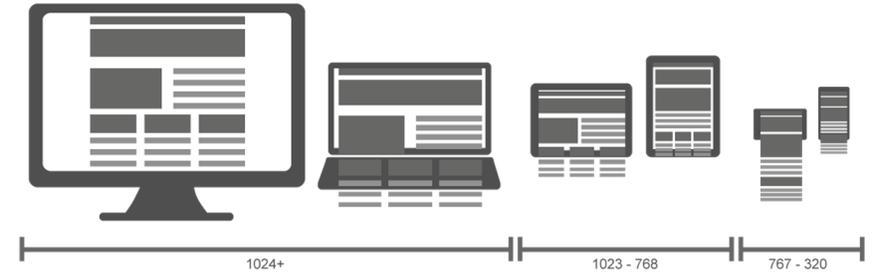


Il device

User scale

- Esiste oggi una varietà enorme di monitor, sia per quanto riguarda le dimensioni che di risoluzioni.
- Monitor anche ad alta risoluzione, utilizzati in piccoli display dei portatili, portano gli utenti ad utilizzare fattori di scala e di rendere totalmente imprevedibile la dimensioni in pixel.
- Disabilitare l'opportunità all'utente di scegliere il fattore di scala è fortemente sconsigliato, e ad esempio Chrome desktop ignora `<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />`

Il device responsive



- L'utente non si aspetta più che il contenuto sia mostrato identico indipendentemente dal device, ma si aspetta che su ogni device sia fruibile al meglio.
- Il responsive è la tecnica che ci permette di creare siti in grado di adattarsi in automatico al dispositivo.
- Il principio è ridurre la necessità dell'utente di ridimensionare e scorrere i contenuti.



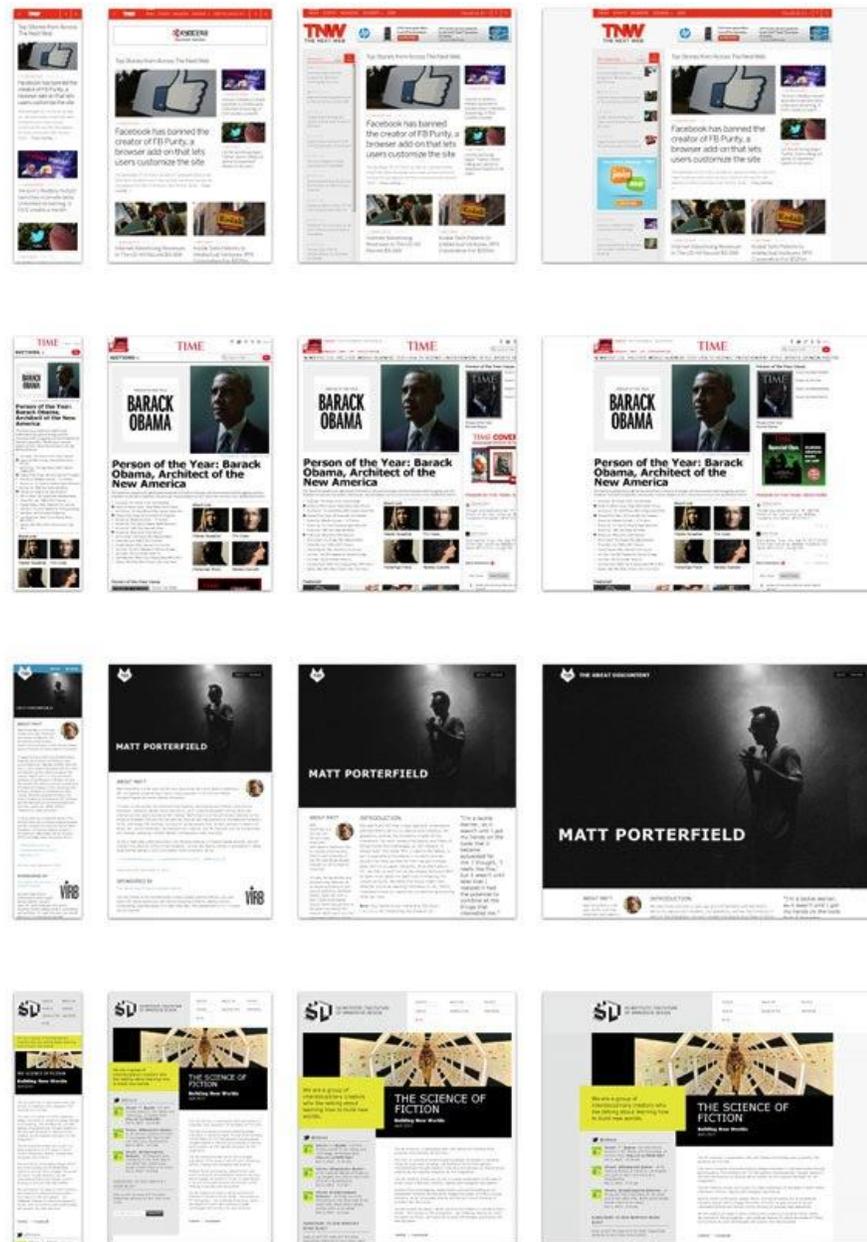
Il device

Meccanismi responsive

- Utilizzo di media queries
- Utilizzo di griglie flessibili
- Utilizzo di immagini flessibili

Con il termine «Mobile First» si intende l'ordine della progettazione che porta a realizzare prima il layout per i dispositivi con monitor più piccoli e poi quelli con dimensioni maggiori.

Progettare mobile first ci aiuta a ragionare sui contenuti e sulle informazioni di maggior risalto



Il device

Modalità di interazione

Consideriamo anche il modo in cui è l'utente a interagire con noi.

Non tutti i dispositivi hanno le stesse caratteristiche ed occorre supportarli tutti.



Il device

Modalità di interazione

Spesso considerare questi aspetti porta grosse sfide progettuali

Ad esempio, se occorre prevedere l'uso del touch:

- Gli elementi sono abbastanza grandi?
- Alcune informazioni sono veicolate con l'hover? Il touch non ha hover!

Il contesto

Il caso d'uso che andremo a progettare, pur trattando gli stessi dati, può essere sostanzialmente diverso in funzione del contesto in cui viene fruito.

Le differenze possono dipendere da molti fattori:

- Contesto culturale
- Distribuzione geografica (uso da cellulare o da pc Es: prenotazione biglietto)
- Necessità garanzie tempistiche (Es: allarme impianto produttivo)

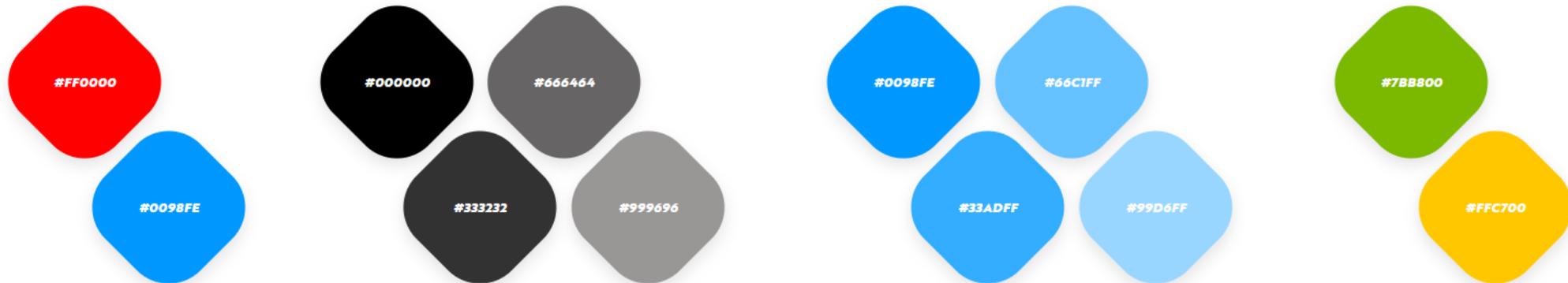


Utilizzo dei colori

Organizzazione

E' opportuno crearsi un palette di colori per:

- Creare uniformità e consistenza
- Veicolare il significato tramite i colori
- Ridurne l'uso per darne maggior risalto
- Adottare criteri specifici per la scelta della palette a partire dal colore scelto



Utilizzo dei colori

Veicolo emotivo

I colori sono anche un veicolo emotivo e in base al tipo di cultura e contesto possono veicolare significati differenti

COLOR EMOTION GUIDE



Attenzione !

In base al contesto il significato del colore può veicolare concetti differenti.

Voi cosa fareste?

Salva tutto, tranquillo

Annulla

Elimina



Scenari avanzati

- Alcune tecniche sono requisiti abilitanti di nuovi scenari che entrano nei prodotti di consumo che utilizziamo tutti i giorni.



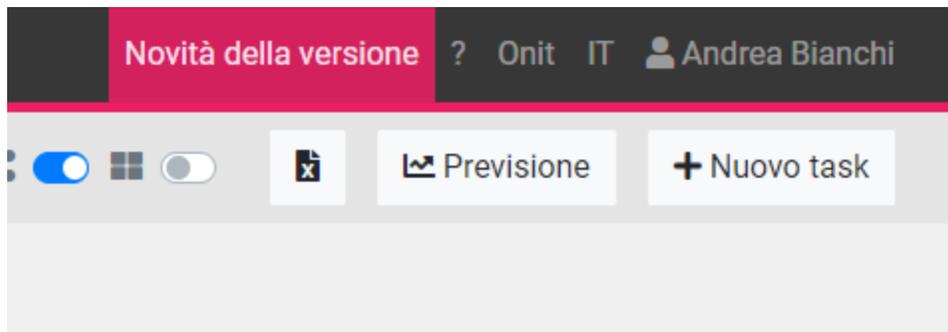
Animazioni CSS

- Nelle ultime versioni di CSS sono presenti istruzioni sempre più avanzate che permettono di realizzare elementi precedentemente impensabili.
- Tramite le animazioni è possibile applicare degli effetti molto fluidi e senza utilizzo di javascript.
- Questi effetti, possono enfatizzare l'interazione con l'utente e restituire un look moderno ed interattivo.



Animazione CSS

Lesson learned

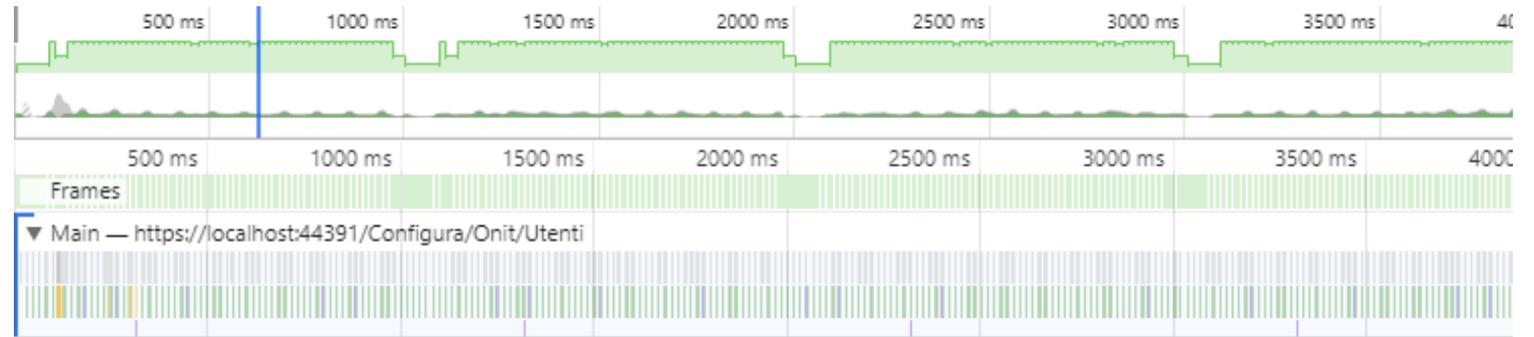
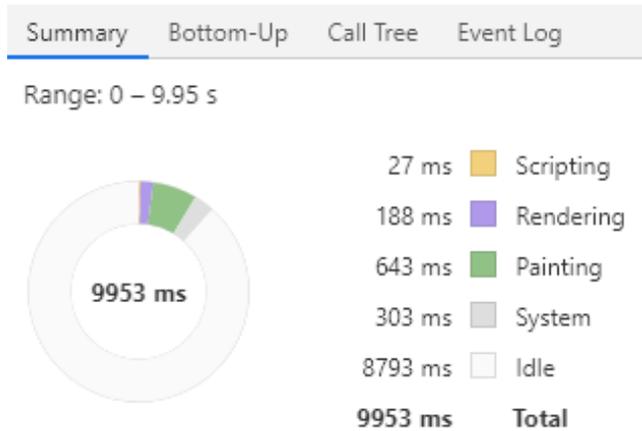


```
@keyframes onit-pulse-keyframes {  
  0% {  
    background-color: rgba($app-color, .25);  
  }  
  
  50% {  
    background-color: rgba($app-color, 0.9);  
    color: rgba($app-background-contrast-color, 0.9);  
  }  
  
  100% {  
    background-color: rgba($app-color, .25);  
  }  
}
```

Animazione CSS

Lesson learned

Risultato, mi telefona un collega che gli sto utilizzando un core al 100%



4 things a browser can animate cheaply

Position

```
transform: translate(npx, npx);
```

Scale

```
transform: scale(n);
```

Rotation

```
transform: rotate(ndeg);
```

Opacity

```
opacity: 0...1;
```

Move all your visual effects to these things.
Transition everything else at your own risk.



```
@keyframes onit-pulse-keyframes {
```

```
0% {
```

```
  background-color: rgba($app-color, .25);
```

```
}
```

```
50% {
```

```
  background-color: rgba($app-color, 0.9);
```

```
  color: rgba($app-background-contrast-color, 0.9);
```

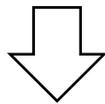
```
}
```

```
100% {
```

```
  background-color: rgba($app-color, .25);
```

```
}
```

```
}
```



Cambio implementazione

```
@keyframes onit-pulse-keyframes {
```

```
0% {
```

```
  opacity: 1;
```

```
}
```

```
50% {
```

```
  opacity: 0.5;
```

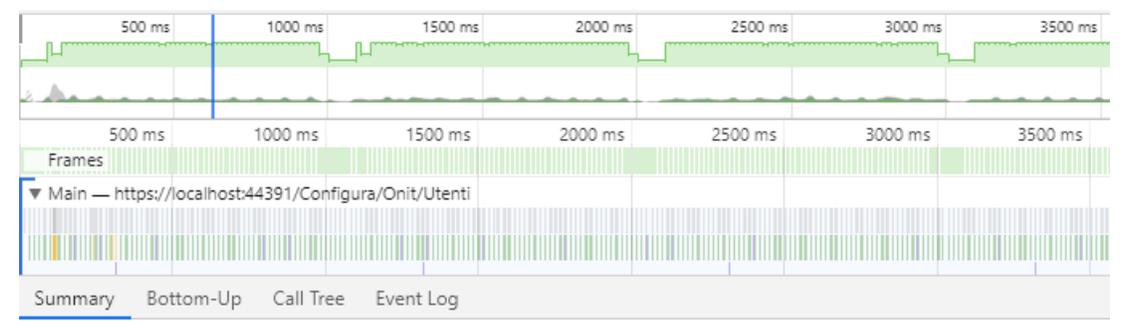
```
}
```

```
100% {
```

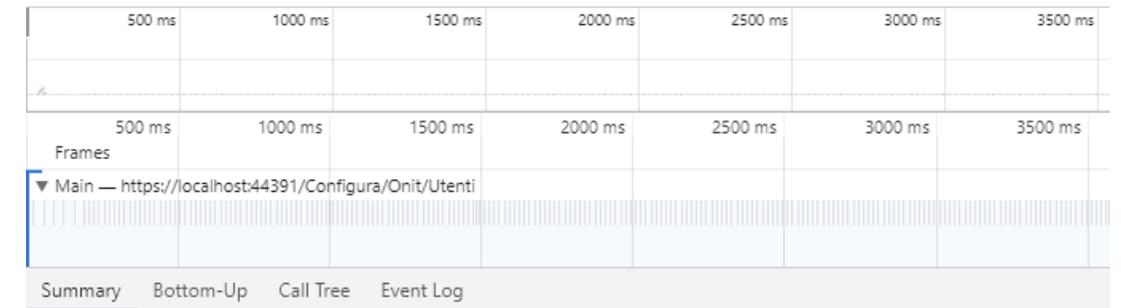
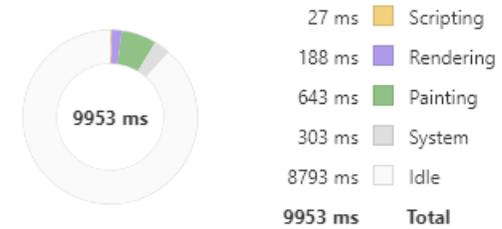
```
  opacity: 1;
```

```
}
```

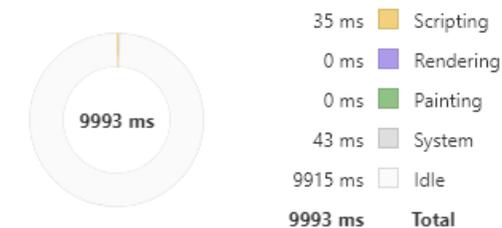
```
}
```



Range: 0 – 9.95 s



Range: 0 – 9.99 s



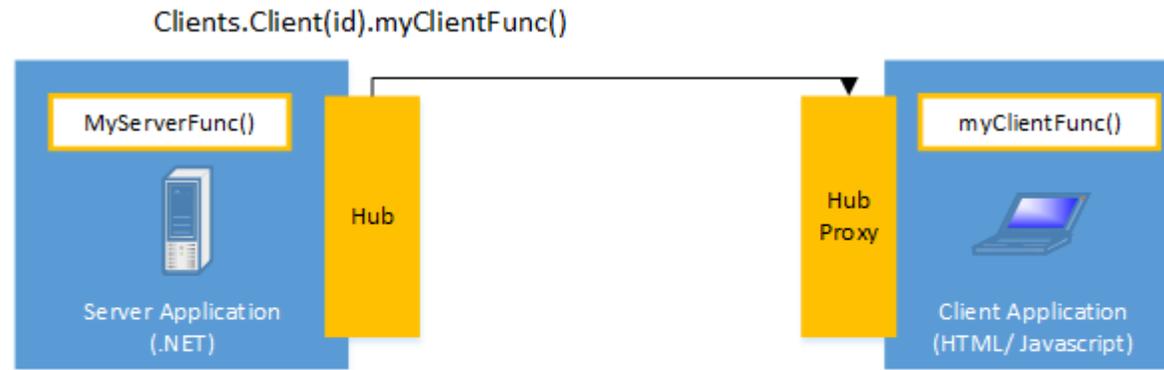
Scenari collaborativi

- Alcuni scenari, ad esempio quelli collaborativi, necessitano che uno o più client siano il più possibile allineati con i dati del server, e che alla modifica di un contenuto tutti i client vengano prontamente allineati.
- In questa situazione una possibile soluzione, non ottima, è che il client continui a chiedere dati al server per controllare eventuali aggiornamenti. Questa soluzione creerebbe presto inefficienze difficilmente sostenibili in un contesto normale di carico.
- Altra soluzione è che sia il server stesso a «contattare» i client interessati per avvisarli della presenza di un aggiornamento.

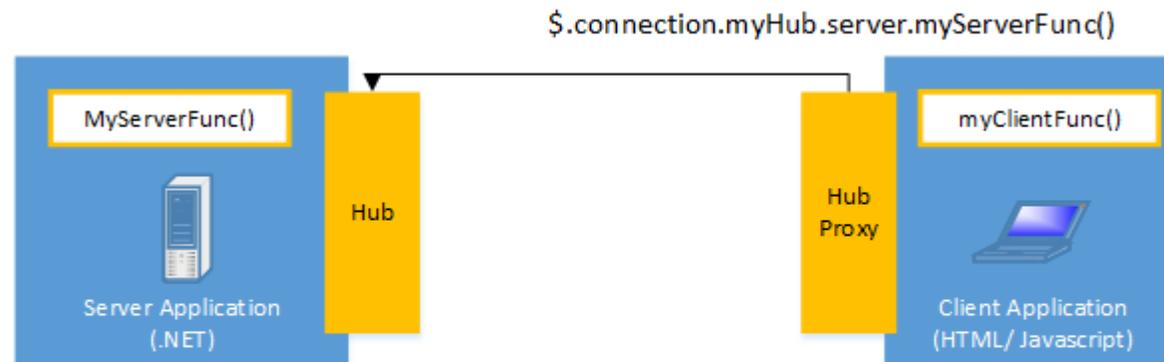
SignalR

- SignalR gestisce automaticamente la gestione delle connessioni e **consente di trasmettere messaggi a tutti i client** connessi contemporaneamente, come in una chat room. È anche possibile inviare messaggi a client specifici. **La connessione tra client e server è persistente**, a differenza di una connessione HTTP classica, che viene ristabilita per ogni comunicazione.
- SignalR supporta la funzionalità "**server push**", in cui il codice del server può chiamare il codice client nel browser usando chiamate rpc (Remote Procedure Call), anziché il modello di richiesta-risposta comune sul Web.
- SignalR usa il nuovo trasporto **WebSocket**, dove disponibile ed esegue il fallback ai trasporti meno recenti, se necessario.

SignalR



Server invocation of client method
myClientFunc()



Client invocation of server method
MyServerFunc()



Push notifications

- Ci sono situazioni in cui si ha la necessità di richiamare l'attenzione dell'utente ad un particolare messaggio/aggiornamento evento accaduto, anche mentre l'utente non sta necessariamente guardando l'applicazione. In questi contesti si parla di notifica push.
- Per gestire le push notifications lo strumento più usato (e gratuito se usate le funzioni base) è OneSignal. Presenta un setup molto rapido.
- Ogni utente può registrarsi sia da smartphone che da pc e sullo stesso dispositivo può registrarsi da più browsers, così da ricevere le notifiche dove preferisce.

Push notifications

- NOTE

- Se l'utente si registra tramite pc è necessario che l'applicazione da cui si è registrato sia aperta per ricevere le push (può non avere il focus)
La notifica push viene inserita nell'elenco delle notifiche del Sistema Operativo che si sta utilizzando.
- Se l'utente si registra tramite smartphone NON è necessario avere l'applicazione aperta per riceverle, basta avere il device acceso.
La notifica push viene visualizzata nell'elenco delle notifiche delle app.

Dark mode

La modalità Dark Mode nasce da una capacità legata ai display Oled dei cellulari, che permette di **ridurre il consumo** di energia.

Il display è infatti l'elemento che ha il maggiore consumo di batteria. Per la loro tecnologia i display Oled emettono luce da ogni singolo pixel. Con il bianco dunque si ha il massimo di consumo, mentre si ottiene un consumo nullo nel caso del colore nero.

Con la diffusione della Dark mode l'utilizzo tra gli utenti si è esteso anche per altri fattori ed oggi è utilizzato anche su display led che non ottengono nessun vantaggio in termini di consumo energetico.

Dark mode

- Supportare la Dark Mode inoltre può essere un aspetto per migliorare l'accessibilità, dato che può essere di aiuto a persone con problemi visivi.
- Infine può essere un tratto distintivo per il look dell'applicativo, esempio Netflix

L'indicazione se l'utente preferisce la dark mode può essere intercettata via css tramite l'apposito media

```
@media (prefers-color-scheme: dark) {  
}
```

CQRS

- Command and Query Responsibility Segregation è un pattern architetturale che permette di organizzare le comunicazioni da e verso un data store.
- CQRS separa le operazioni su un data store in queries (letture di dati) e comandi (aggiornamenti sui dati).
- Permette di isolare i modelli in Read Model, usato da una query, e Write Model, usato da un comando.



CQRS

- Query: Legge i dati, non deve modificarli in nessun modo. Ritorna un DTO.
- Command: Modifica uno o più dati, dovrebbe eseguire un'azione (Richiedere una giornata di ferie, Approvare un permesso) e non un'operazione atomica legata ai dati (Impostare campo A con valore true)



CQRS

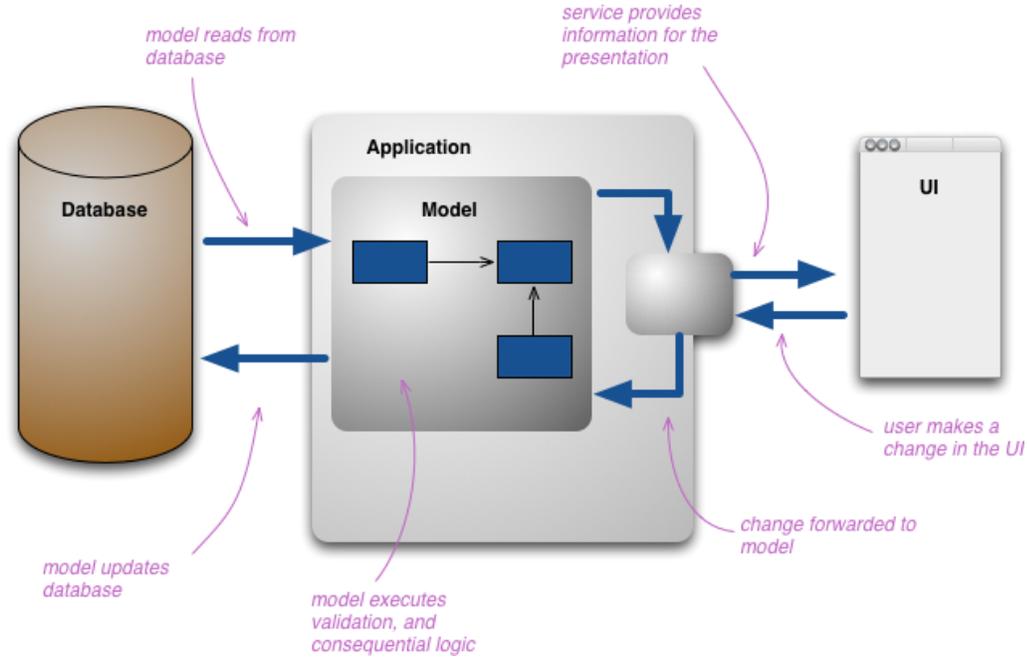
- La struttura precedentemente descritta abilita alla possibilità di separare fisicamente i dati tra dati in lettura (Read Model) e dati in scrittura (Write Model).
 - PRO: Si possono adottare soluzioni diverse per lo storage dei dati, ad hoc per la specifica funzione. Ogni lettura si semplifica ad una semplice lettura di dati, dove saranno già state applicate le regole di business.
 - CONTRO: I dati in lettura e quelli in scrittura devono avere meccanismi di sincronizzazione, anche se con tempi più rilassati, dettati dal caso d'uso.



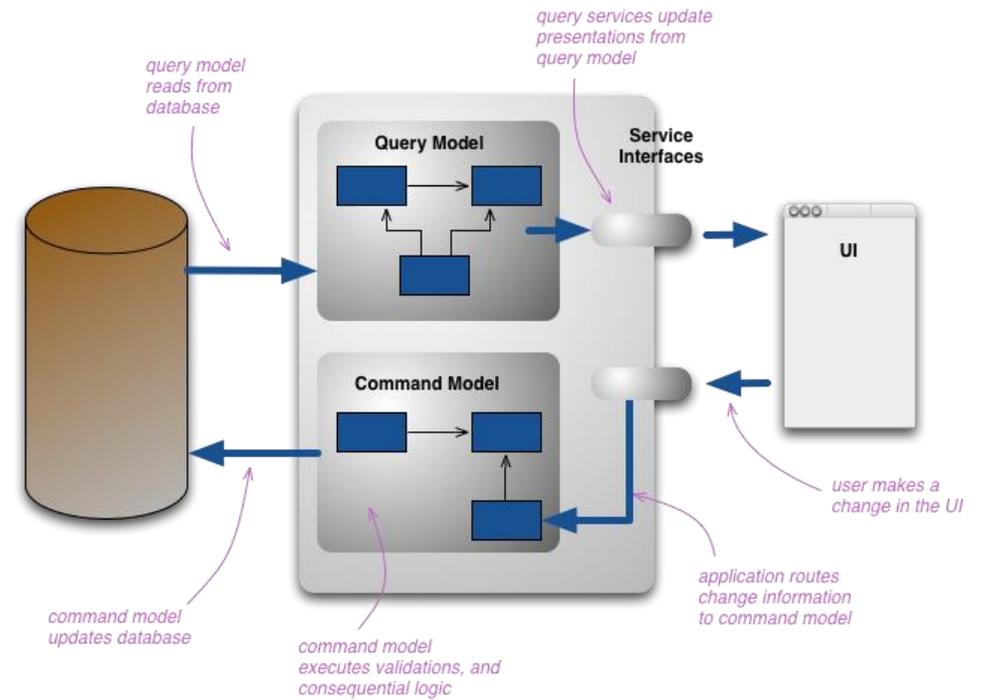
CQRS



Approccio tradizionale



Approccio CQRS



Domande ?



Approfondimenti

- https://en.wikipedia.org/wiki/Marissa_Mayer
- <https://www.w3.org/TR/css-animations-1/>
- <https://martinfowler.com/bliki/CQRS.html>

