Security

XSS

Cos’è:

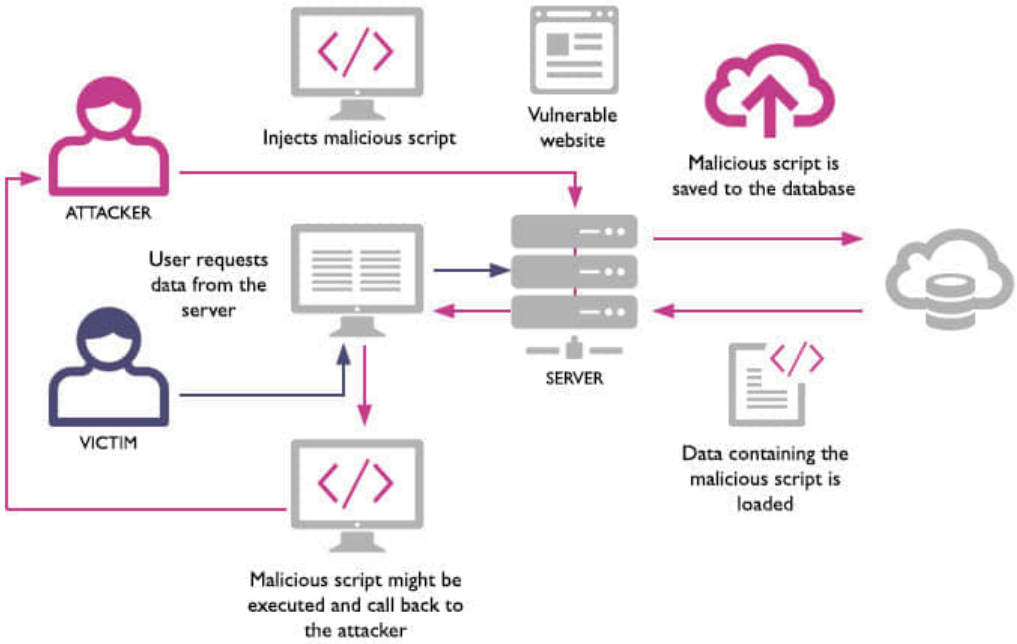
XSS: Cross-site scripting

Il **cross-site scripting** (**XSS**) è una vulnerabilità che affligge siti web dinamici che impiegano un insufficiente controllo dell'input nei form.

https://it.wikipedia.org/wiki/Cross-site_scripting

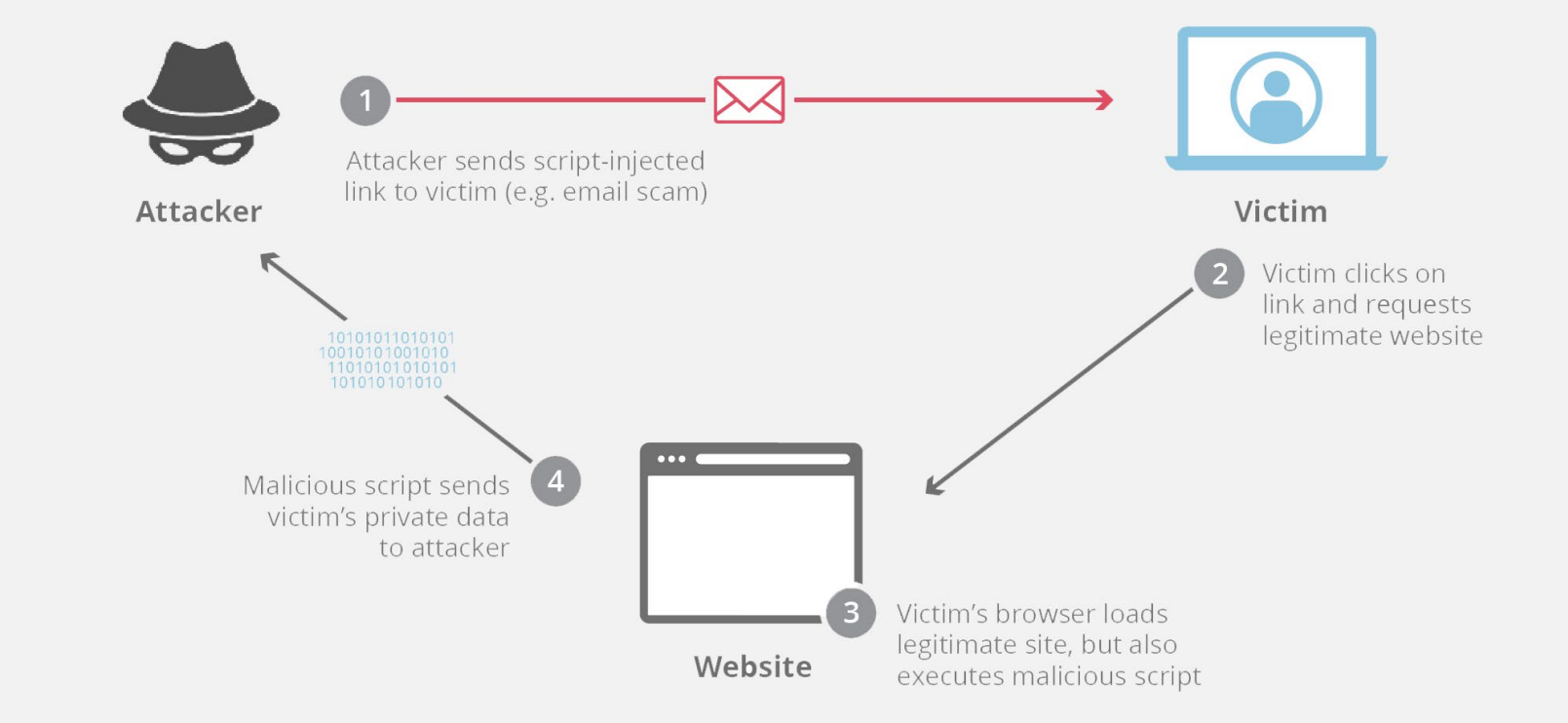
Demo:

XSS storicizzato su DB

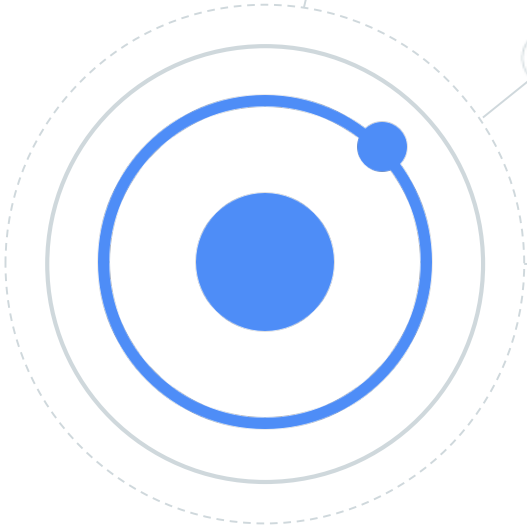


Demo:

XSS solo su client (phishing)



Ionic

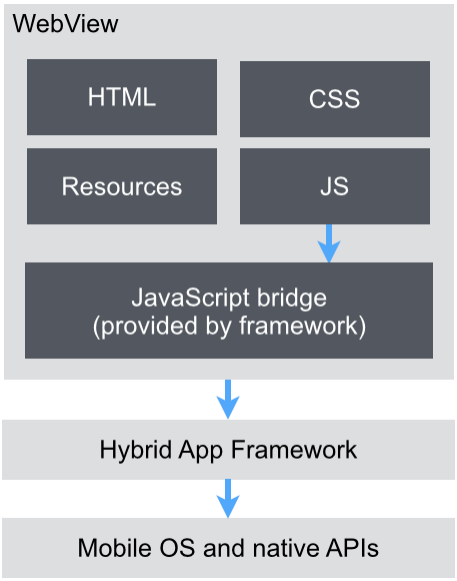


Programma

- Di cosa si tratta?
- Le tecnologie utilizzate
- Setup ambiente
- La struttura del progetto base
- I componenti base
- Alcuni servizi utili
- …

WebView

L'applicazione vive in un browser contenuto nell'app

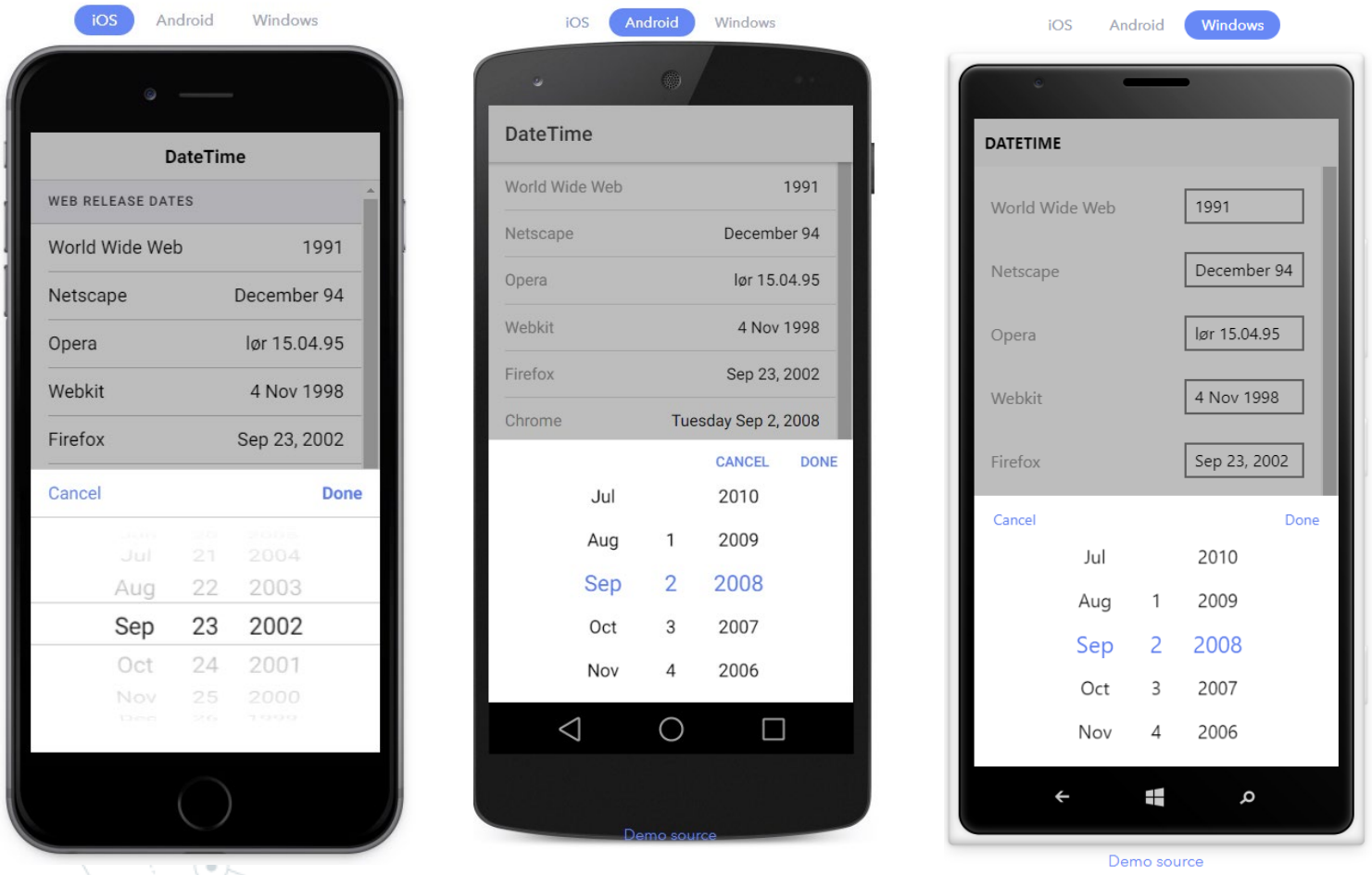


Ionic permette di sviluppare applicazioni mobile ibride sfruttando tecnologie web che tentano di riprodurre il comportamento nativo del sistema

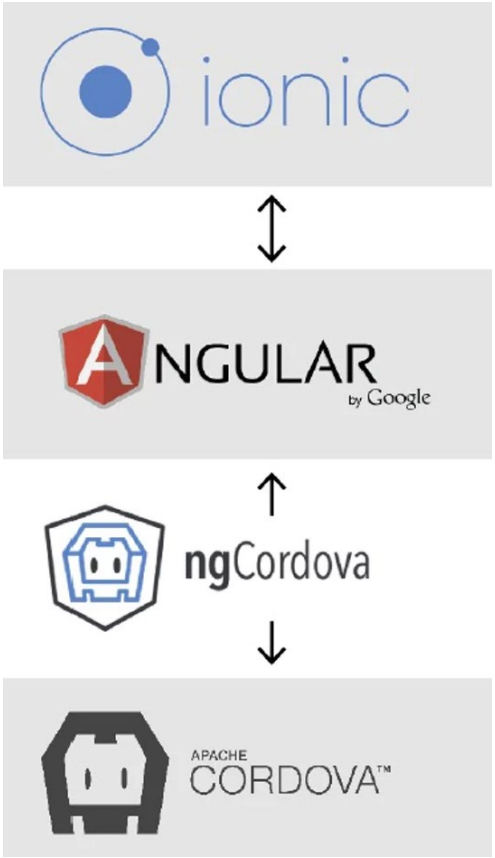
Di cosa si tratta?

Un solo tag html:

```
<ion-datetime displayFormat="MM/DD/YYYY" [(ngModel)]="myDate"></ion-datetime>
```



Le tecnologie utilizzate



UI Framework

Framework

Interfacciamento con l'hardware
(dalla versione 3 di ionic si chiama «ionic/native»)

WebView fornita dal progetto Apache Cordova
(Capacitor è il nuovo progetto Ionic per sostituire Cordova)

Le tecnologie utilizzate

TypeScript: linguaggio di programmazione (superset di JavaScript)

HTML5: linguaggio di markup per pagine web

Sass/scss: estensione del css per definire fogli di stile

Setup

- Installare nodejs LTS
- Installare un IDE come VSCODE
- Installare Ionic DevApp sul proprio smartphone
- Eseguire: npm install -g ionic
- Eseguire: ionic start «nomeprogetto»
- Scegliere «conference» come esempio di app
- Condividere la stessa rete tra notebook e smartphone oppure usare il remote debugging di Chrome
- Entrare nella cartella del progetto ed eseguire: «ionic serve –c»

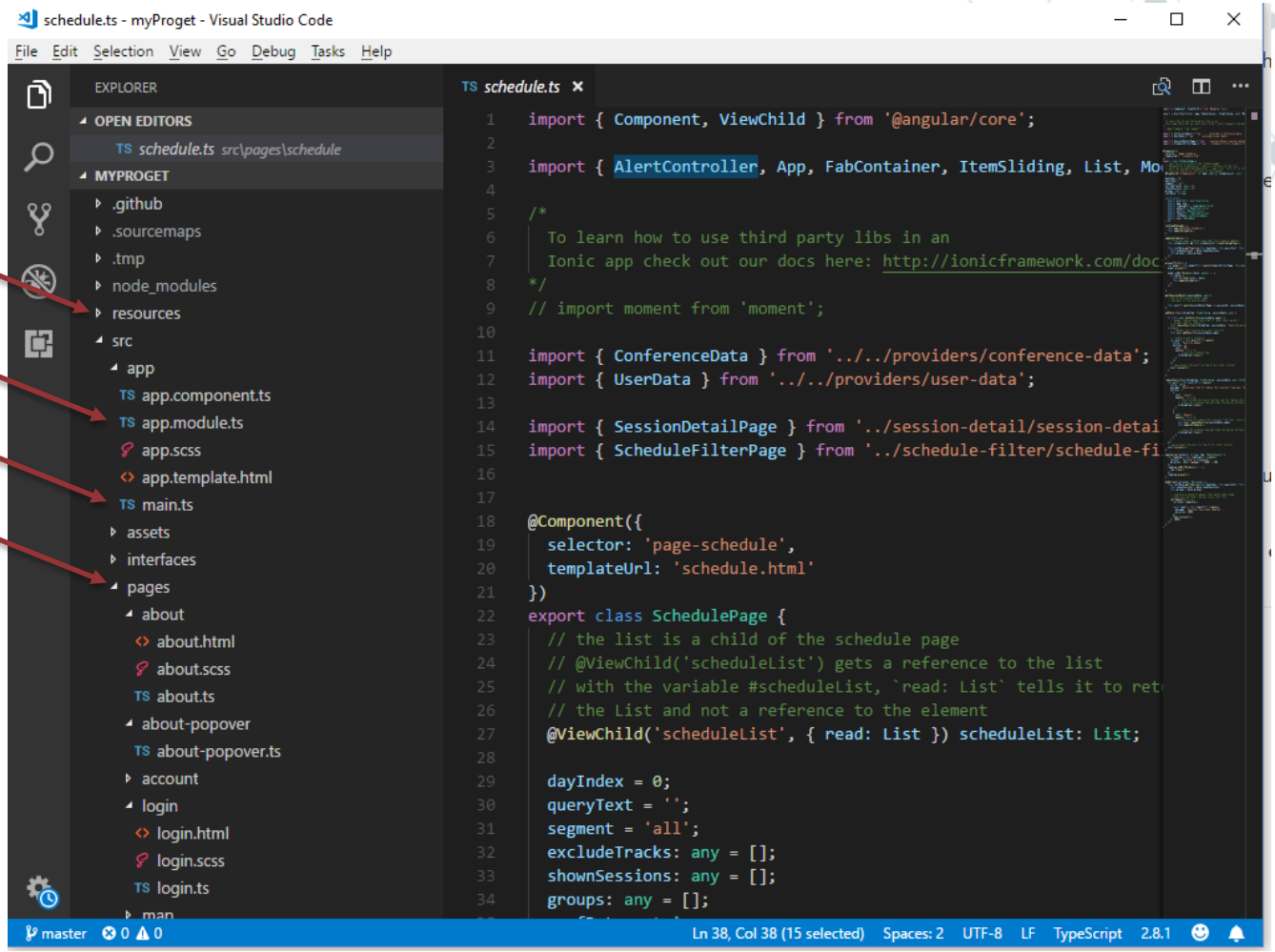
Struttura

Immagini e risorse

app

main

pages

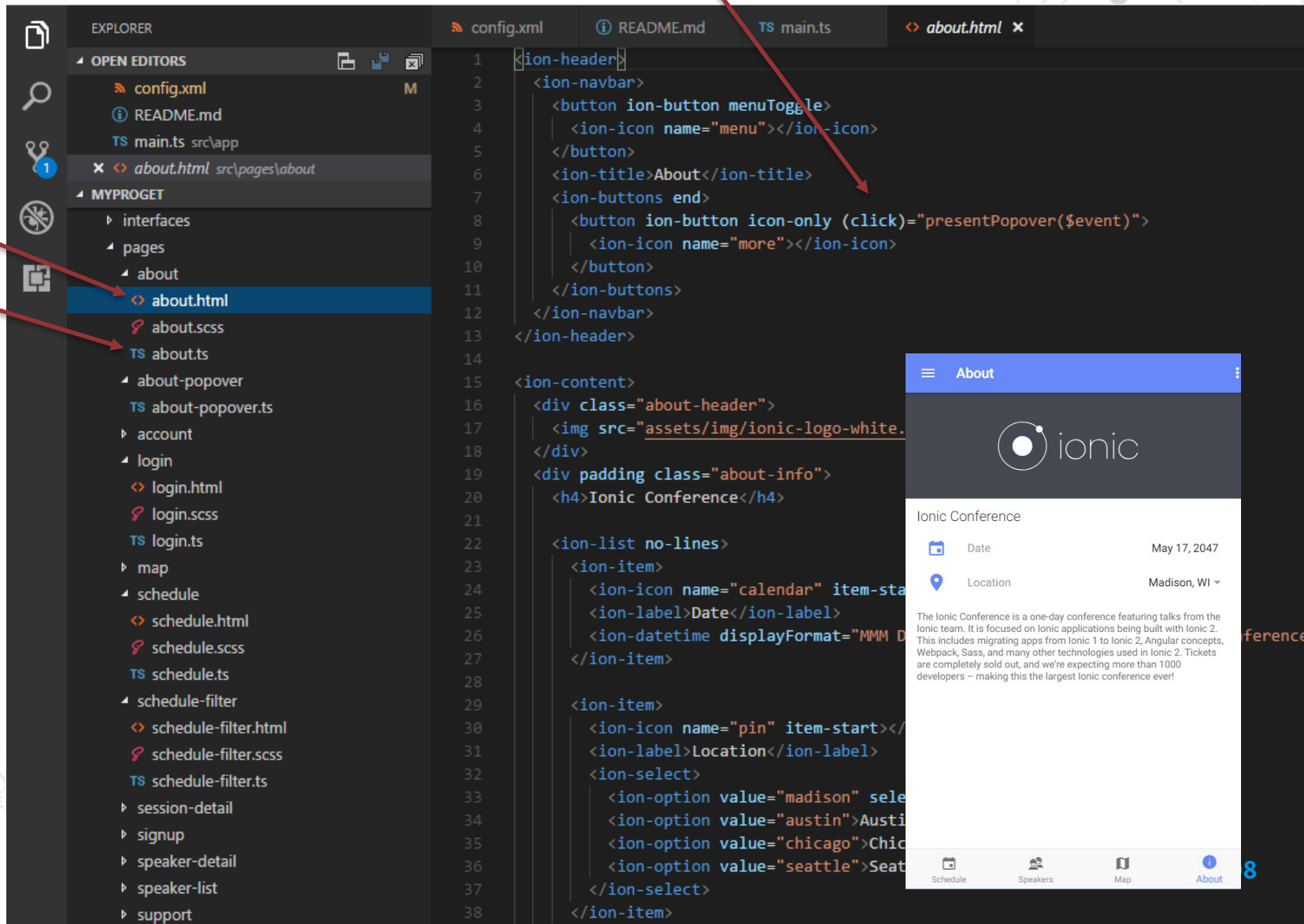


Pagina

Azione

view

controller



Componenti per UI

- ActionSheetController
- AlertController
- App
- Avatar
- Badge
- Button
- Checkbox
- Chip
- Col
- Config
- Content
- DateTime
- Events
- FabButton
- FabContainer
- FabList
- Footer
- Grid
- Haptic
- Header
- HideWhen
- Icon

https://ionicframework.com/docs/components/

Menus

Menu is a side-menu navigation that can be dragged out or toggled to show. The content of a menu will be hidden when the menu is closed.

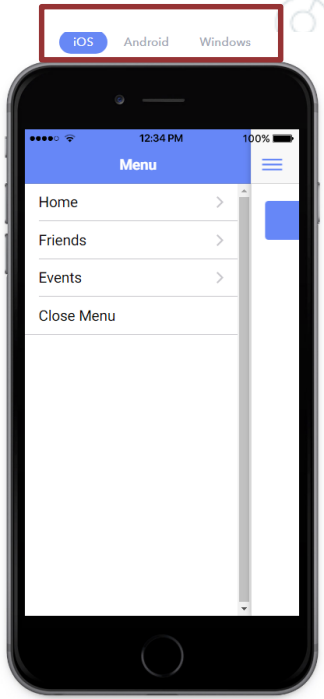
Menu adapts to the appropriate style based on the platform.

For more information, Check out the API docs.

Basic Usage

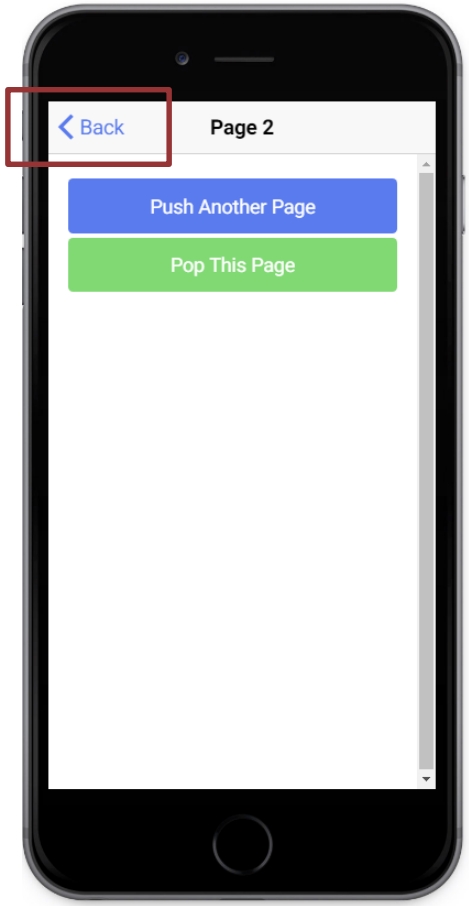
Demo Source

```
<ion-menu [content]="content">
  <ion-header>
    <ion-toolbar>
      <ion-title>Menu</ion-title>
    </ion-toolbar>
  </ion-header>
  <ion-content>
    <ion-list>
      <button ion-item (click)="openPage(homePage)">
        Home
      </button>
      <button ion-item (click)="openPage(friendsPage)">
        Friends
      </button>
      <button ion-item (click)="openPage(eventsPage)">
        Events
      </button>
      <button ion-item (click)="closeMenu()">
        Close Menu
      </button>
    </ion-list>
  </ion-content>
</ion-menu>
```



Componente NavController

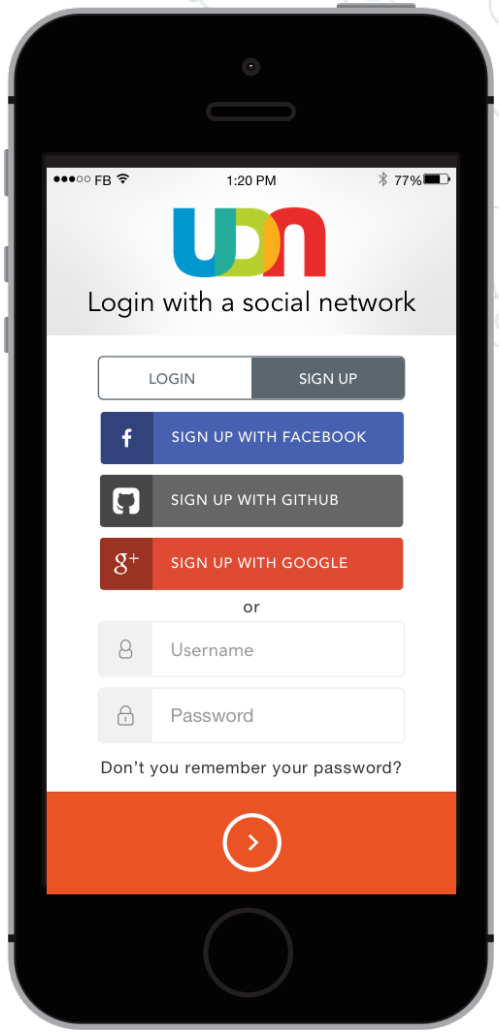
NavController is the base class for navigation controller components like `Nav` and `Tab`. You use navigation controllers to navigate to pages in your app. At a basic level, a navigation controller is an array of pages representing a particular history (of a Tab for example). This array can be manipulated to navigate throughout an app by pushing and popping pages or inserting and removing them at arbitrary locations in history.



Oauth 2

Easy access delegation

Oauth 2 rfc6749



Si basa su un principio molto semplice:

Garantire l'accesso ad applicazioni terze a delle risorse private senza condividere la propria password

Perché non condividere la propria password?

- non si possono gestire livelli di autorizzazione differenti
- non si può garantire che l'autorizzazione venga utilizzata nel contesto scelto
- per revocare il permesso sono obbligato a cambiare password

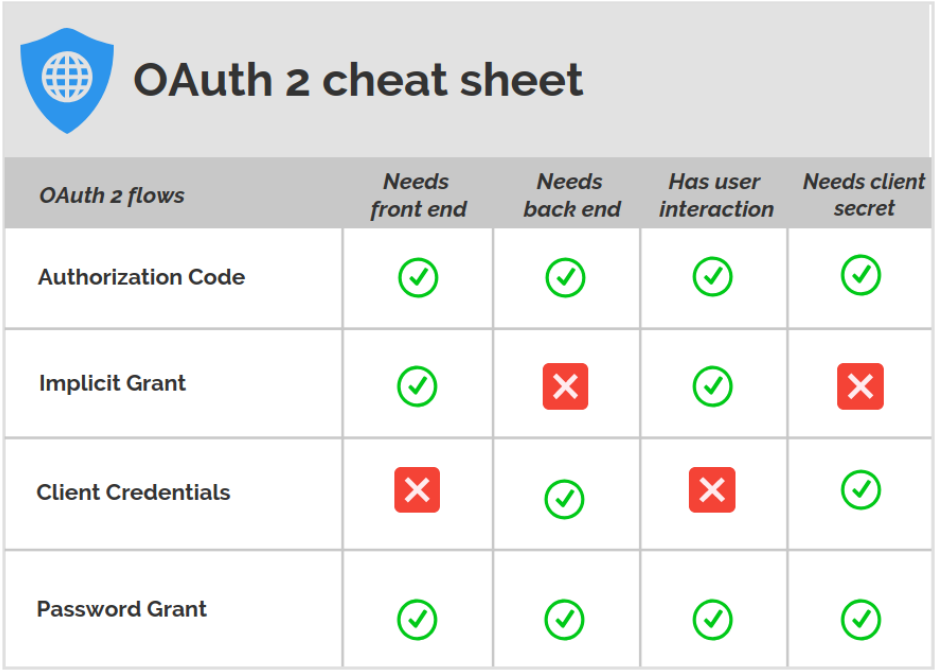
OAuth è nato quindi con il presupposto di garantire l'accesso delegato ad un client specifico per determinate risorse sul server per un tempo limitato, con possibilità di revoca.

https://it.wikipedia.org/wiki/OAuth

Oauth 2 rfc6749

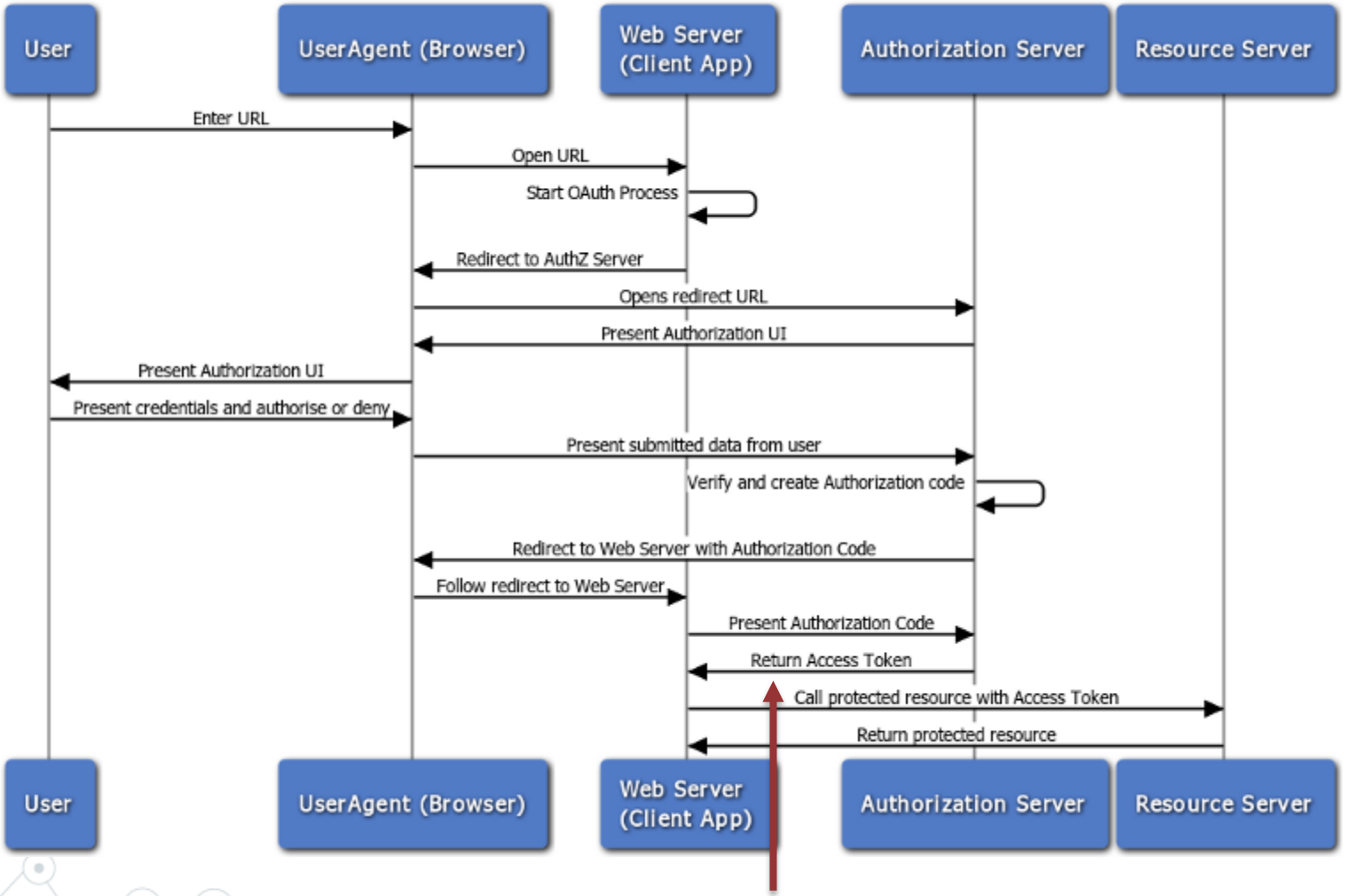
Esistono differenti «flow» di autorizzazione descritti dal protocollo:

- **Authorization Code Grant**
- **Implicit Grant**
- **Client Credential Grant**
- **Password Grant**



https://itnext.io/an-oauth-2-0-introduction-for-beginners-6e386b19f7a9

Oauth 2: Authorization Code Flow



Bearer Tokens are the predominant type of access token used with OAuth 2.0.
A Bearer Token is an opaque string, not intended to have any meaning to clients using it.

Oauth 2: Complicato?

https://auth0.com/pricing/

https://firebase.google.com/pricing

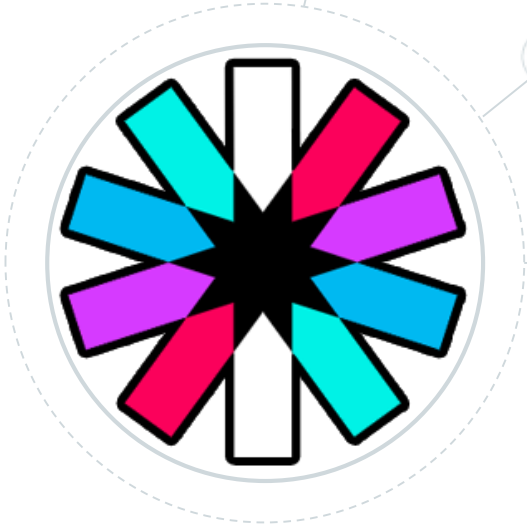
https://auth0.com/blog/ionic-framework-how-to-get-started/

https://auth0.com/docs/quickstart/spa/angular2/01-login

https://github.com/angular/angularfire

https://github.com/angular/angularfire/blob/master/docs/auth/getting-started.md

JWT

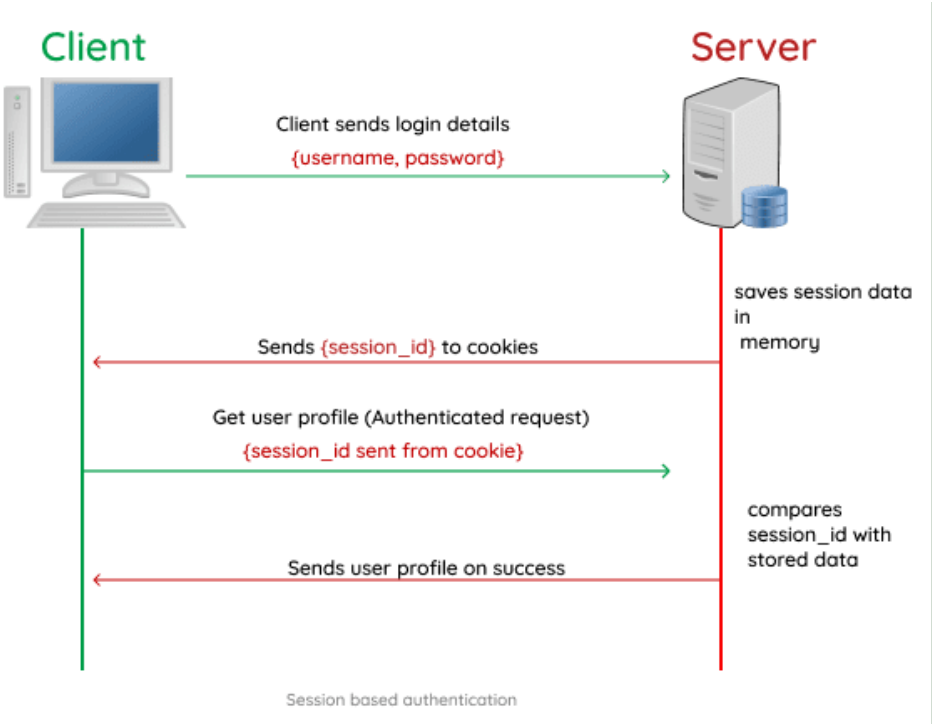


JWTs can be used as OAuth 2.0 Bearer Tokens

JSON Web Token (JWT) rfc7523

Perché è stato introdotto?

Utilizzo classico delle sessioni



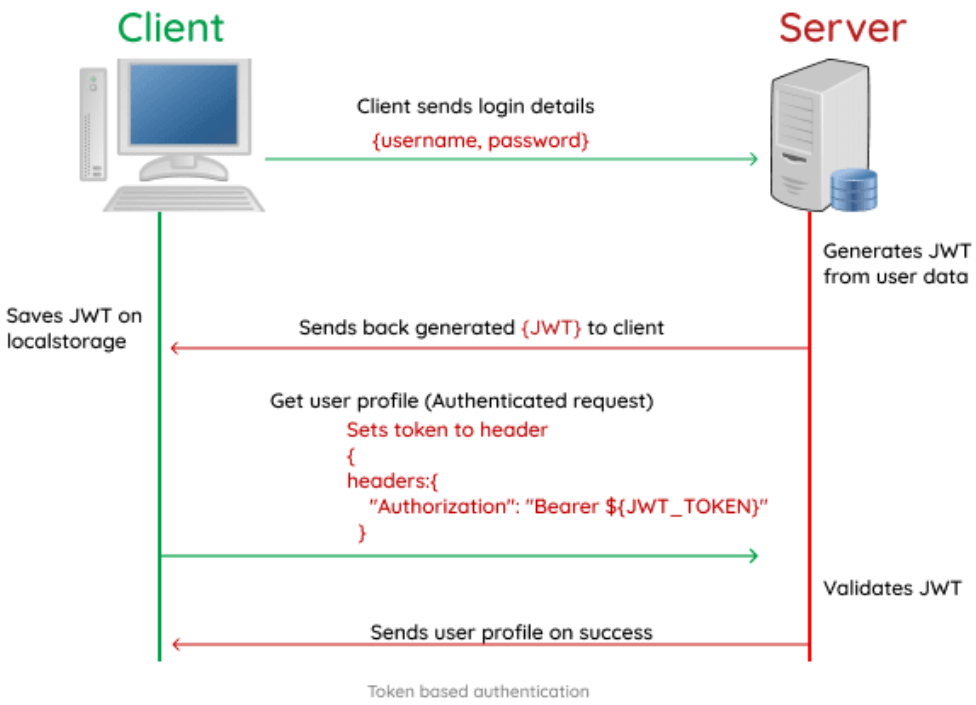
Svantaggio delle sessioni:

- Devo andare sul database a verificare la validità della sessione utente
- Nelle architetture a microservizi l'uso della sessione non è agevole
- Non scalabile con architetture a cluster
- Su una SPA ho un maggiore tempo di sviluppo rispetto ad architetture server MVC

JSON Web Token (JWT) rfc7523

Perché è stato introdotto?

Utilizzo del token



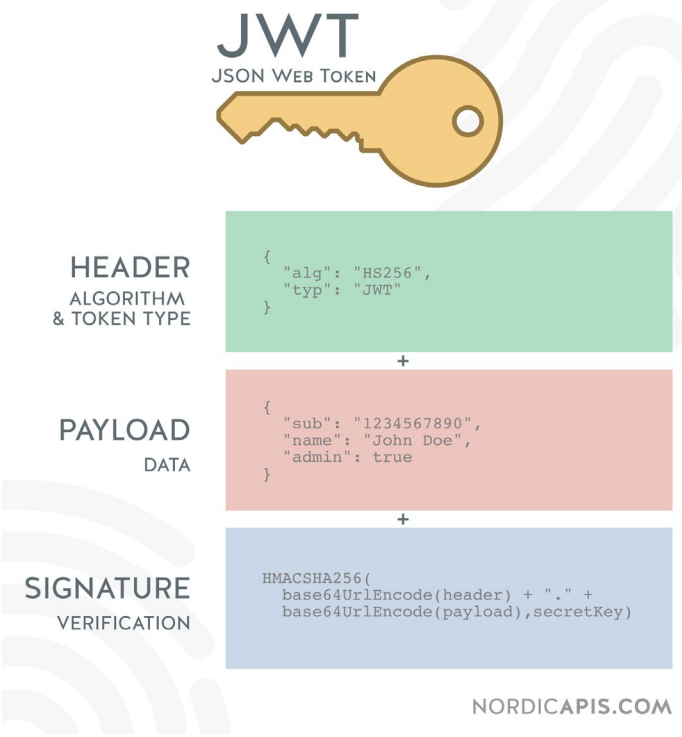
Differenze?

- Non ho storage di sessioni
- Non devo gestire sessioni «morte» lato server
- I servizi possono essere realmente statefull

Ok ma dove è la vera differenza?

JSON Web Token (JWT) rfc7523

https://jwt.io/



Cosa posso mettere nel JWT?

- Informazioni riguardante l'utente
- Informazioni sui permessi dell'utente
- Una data di scadenza
- Una signature per validare il JWT
- Qualsiasi altra informazione

NIENTE PASSWORD NEL JWT!

Lo scopo di un JWT non è di crittografare i dati, quindi evitare la lettura di dati sensibili durante il trasporto (esiste SSL), ma consente alla parte ricevente di fidarsi che i dati ricevuti sono rimasti inalterati durante il trasporto.