

# Security

XSS



Cos'è:

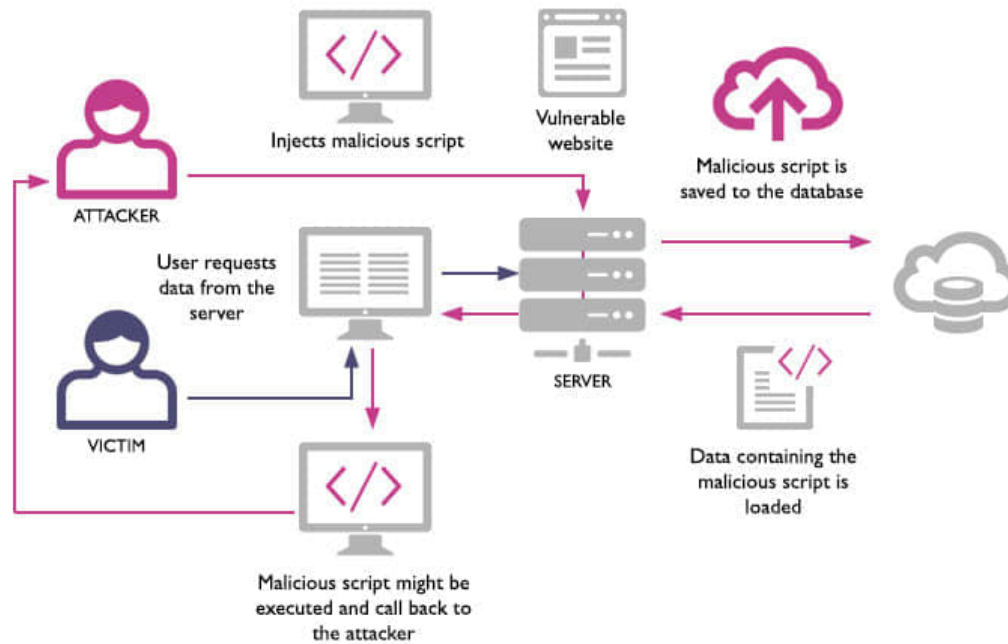
**XSS:** Cross-site scripting

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](https://it.wikipedia.org/wiki/Cross-site_scripting)

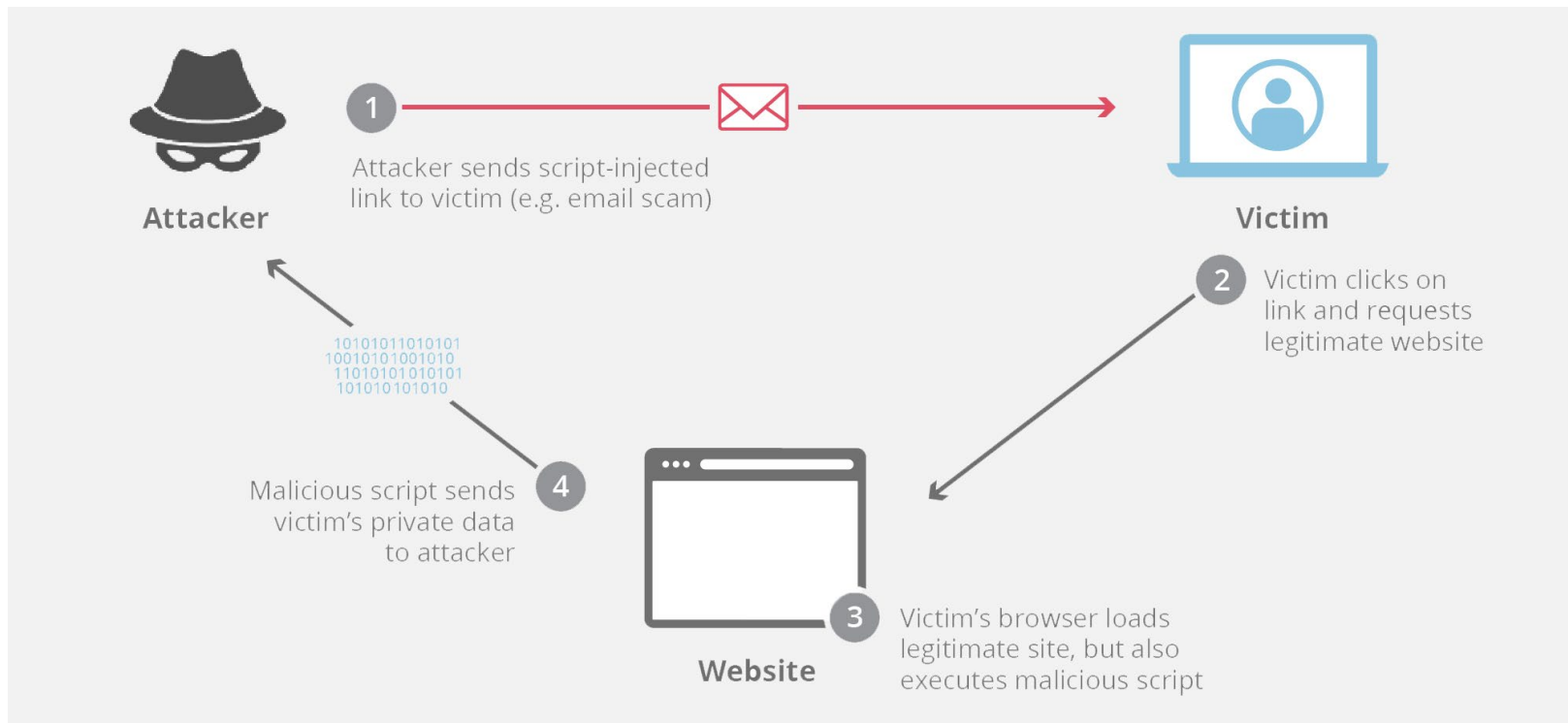
## Demo:

XSS storicizzato su DB

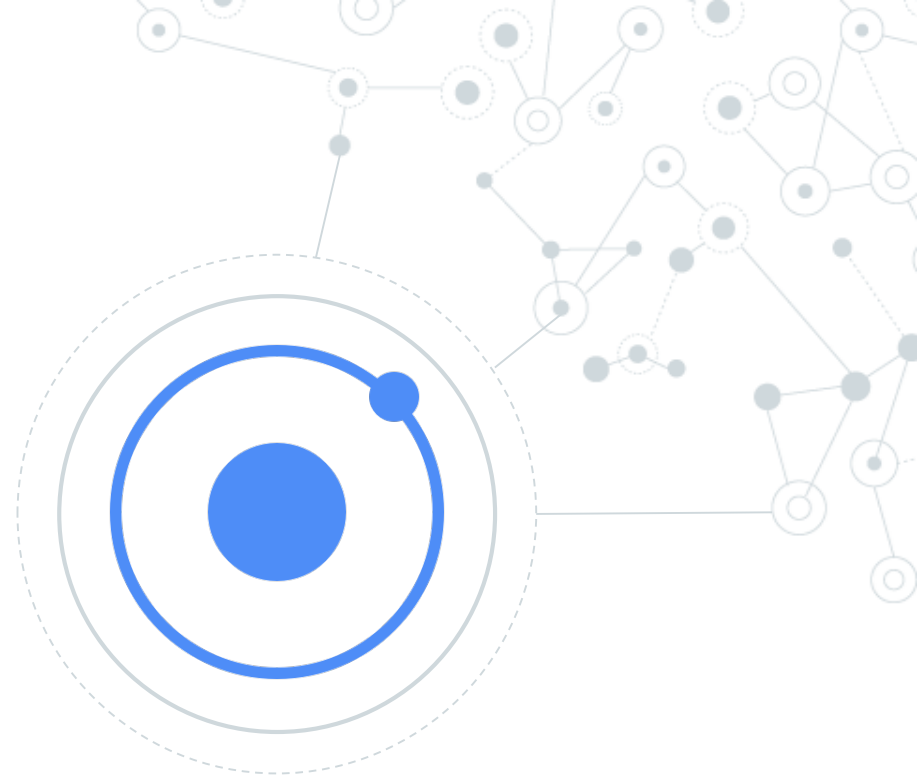


## Demo:

### XSS solo su client (phishing)



Ionic





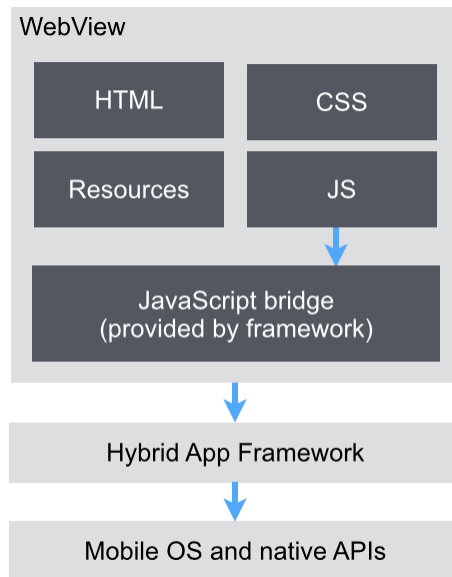
# Programma

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

Di cosa si tratta?

## WebView

L'applicazione vive in un browser contenuto nell'app

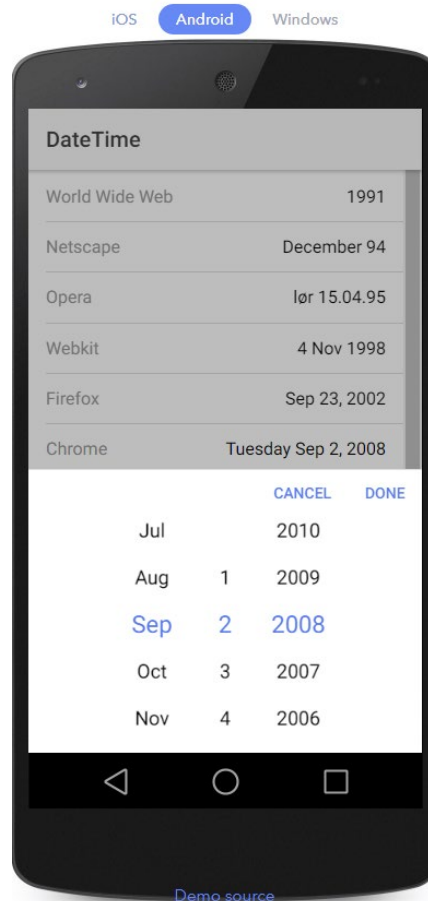
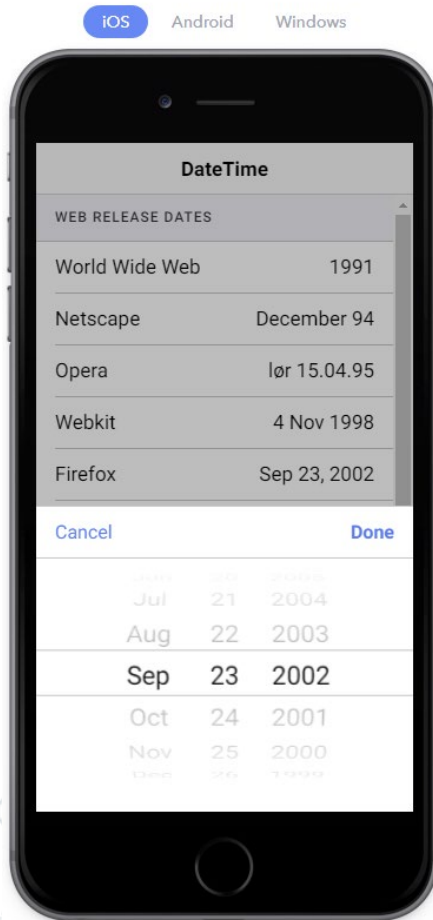


**Ionic permette di sviluppare applicazioni mobile ibride sfruttando tecnologie web che tentano di riprodurre il comportamento nativo del sistema**

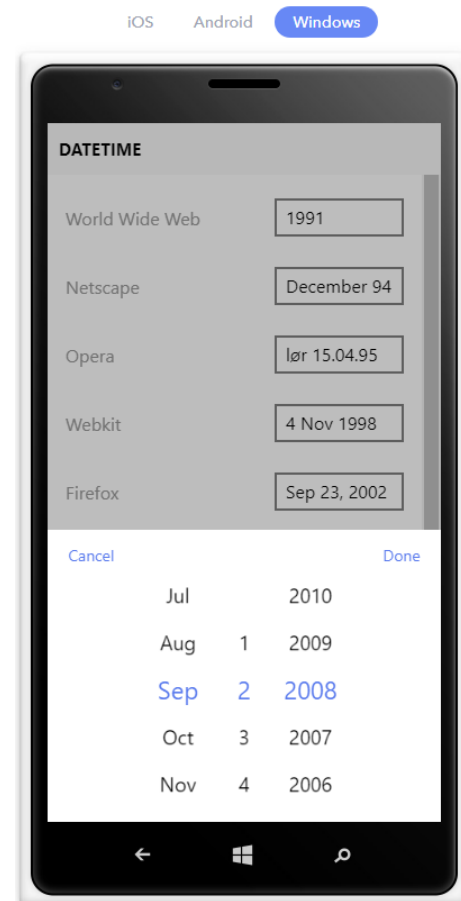
# Di cosa si tratta?

Un solo tag html:

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



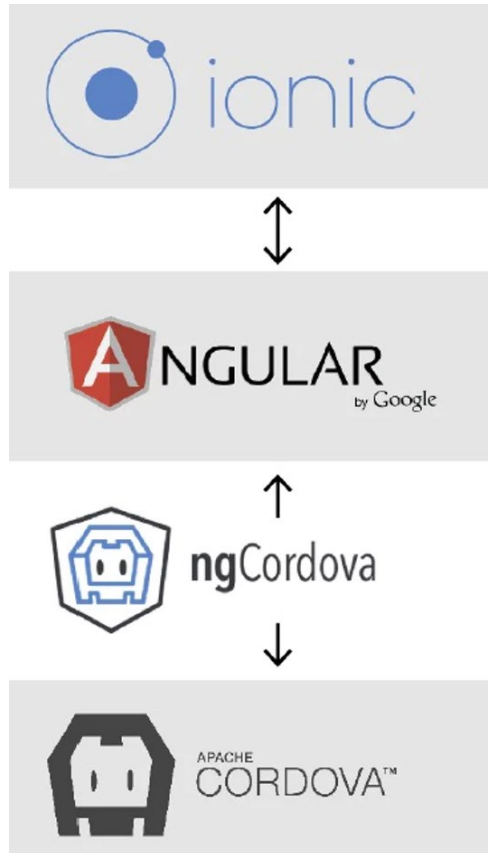
[Demo source](#)



[Demo source](#)



## 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

```
schedule.ts - myProget - Visual Studio Code
File Edit Selection View Go Debug Tasks Help

EXPLORER
├─ OPEN EDITORS
│   └─ TS schedule.ts src\pages\schedule
├─ MYPROGET
│   ├── .github
│   ├── .sourcemaps
│   ├── .tmp
│   ├── node_modules
│   ├── resources
│   └─ src
│       ├── app
│       │   ├── TS app.component.ts
│       │   ├── TS app.module.ts
│       │   ├── app.scss
│       │   ├── app.template.html
│       │   ├── TS main.ts
│       │   ├── assets
│       │   ├── interfaces
│       │   └─ pages
│       │       ├── about
│       │       │   ├── about.html
│       │       │   ├── about.scss
│       │       │   └─ about.ts
│       │       ├── about-popover
│       │       │   ├── TS about-popover.ts
│       │       ├── account
│       │       ├── login
│       │       │   ├── login.html
│       │       │   ├── login.scss
│       │       │   └─ login.ts
│       │       └─ man
│       └─ main

TS schedule.ts x
1  import { Component, ViewChild } from '@angular/core';
2
3  import { AlertController, App, FabContainer, ItemSliding, List, Mo
4
5  /*
6   To learn how to use third party libs in an
7   Ionic app check out our docs here: http://ionicframework.com/doc
8  */
9  // import moment from 'moment';
10
11 import { ConferenceData } from '../providers/conference-data';
12 import { UserData } from '../providers/user-data';
13
14 import { SessionDetailPage } from '../session-detail/session-detail
15 import { ScheduleFilterPage } from '../schedule-filter/schedule-fi
16
17
18 @Component({
19   selector: 'page-schedule',
20   templateUrl: 'schedule.html'
21 })
22 export class SchedulePage {
23   // the list is a child of the schedule page
24   // @ViewChild('scheduleList') gets a reference to the list
25   // with the variable #scheduleList, 'read: List' tells it to ret
26   // the List and not a reference to the element
27   @ViewChild('scheduleList', { read: List }) scheduleList: List;
28
29   dayIndex = 0;
30   queryText = '';
31   segment = 'all';
32   excludeTracks: any = [];
33   shownSessions: any = [];
34   groups: any = [];
```

# Pagina

# Azione

view

controller

The image shows a development environment (VS Code) with the Explorer on the left, the Editor in the center, and a mobile app preview on the right. The Explorer shows a project structure with 'about.html' selected. The Editor shows the HTML code for 'about.html'. The mobile preview shows the rendered 'About' page with Ionic logo and conference details.

```
1 <ion-header>
2   <ion-navbar>
3     <button ion-button menuToggle>
4       <ion-icon name="menu"></ion-icon>
5     </button>
6     <ion-title>About</ion-title>
7   <ion-buttons end>
8     <button ion-button icon-only (click)="presentPopover($event)">
9       <ion-icon name="more"></ion-icon>
10    </button>
11  </ion-buttons>
12 </ion-navbar>
13 </ion-header>
14
15 <ion-content>
16   <div class="about-header">
17     
19   <div padding class="about-info">
20     <h4>Ionic Conference</h4>
21
22   <ion-list no-lines>
23     <ion-item>
24       <ion-icon name="calendar" item-start>
25       <ion-label>Date</ion-label>
26       <ion-datetime displayFormat="MMM D
27     </ion-item>
28
29     <ion-item>
30       <ion-icon name="pin" item-start></
31       <ion-label>Location</ion-label>
32       <ion-select>
33         <ion-option value="madison" sele
34         <ion-option value="austin">Austi
35         <ion-option value="chicago">Chic
36         <ion-option value="seattle">Seatt
37       </ion-select>
38     </ion-item>
```

Mobile App Preview (About page):

- Header: About
- Logo: ionic
- Section: Ionic Conference
- Details:
  - Date: May 17, 2047
  - Location: Madison, WI
- Text: The Ionic Conference is a one-day conference featuring talks from the Ionic team. It is focused on Ionic applications being built with Ionic 2. This includes migrating apps from Ionic 1 to Ionic 2, Angular concepts, Webpack, Sass, and many other technologies used in Ionic 2. Tickets are completely sold out, and we're expecting more than 1000 developers - making this the largest Ionic conference ever!
- Footer: Schedule, Speakers, Map, About

# 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.

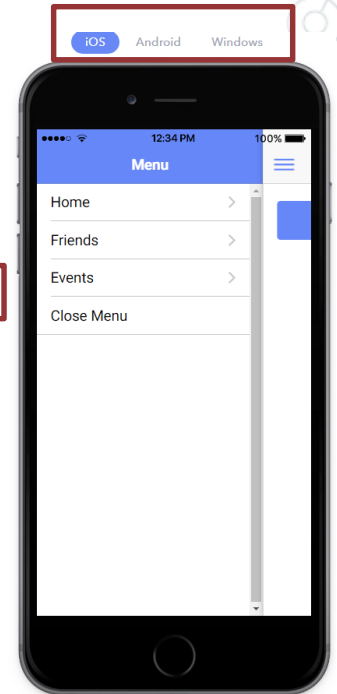
Menu adapts to the appropriate style based on the platform.

For more information, Check out the [API docs](#).

### Basic Usage

```
<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>
```

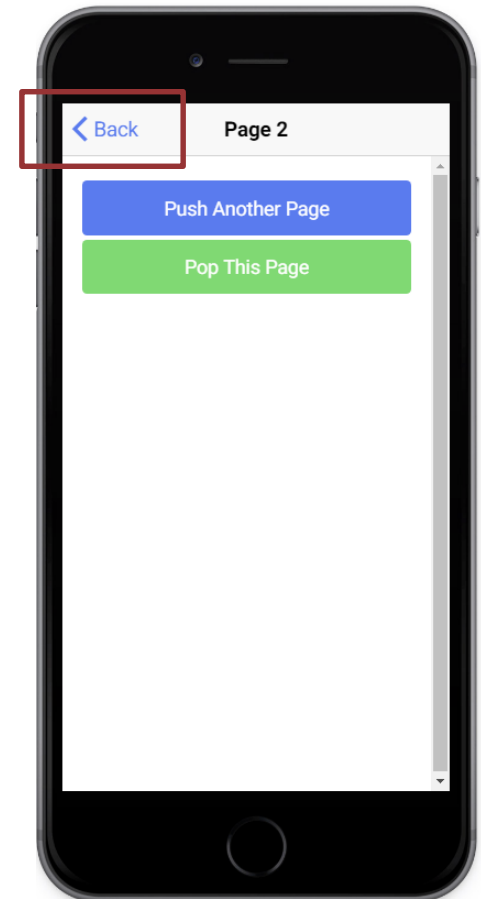
[Demo Source](#)



Attiva Windows  
Passa a Impostazioni per attivare Windows.

# 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:

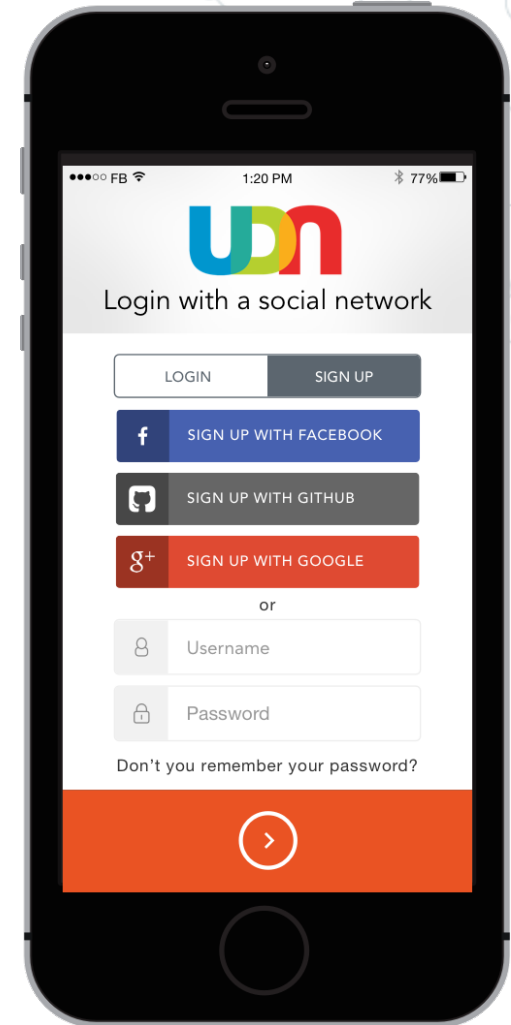
Garantire l'accesso ad applicazioni terze a delle risorse private senza condividere la propria password

## 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.

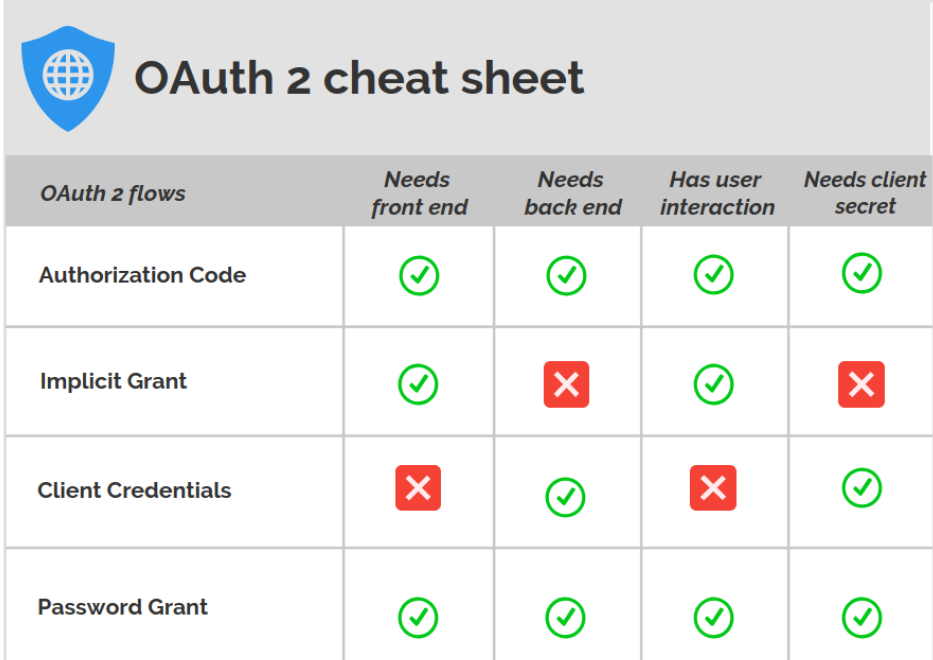
<https://it.wikipedia.org/wiki/OAuth>



# Oauth 2 rfc6749

Esistono differenti «flow» di autorizzazione descritti dal protocollo:

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

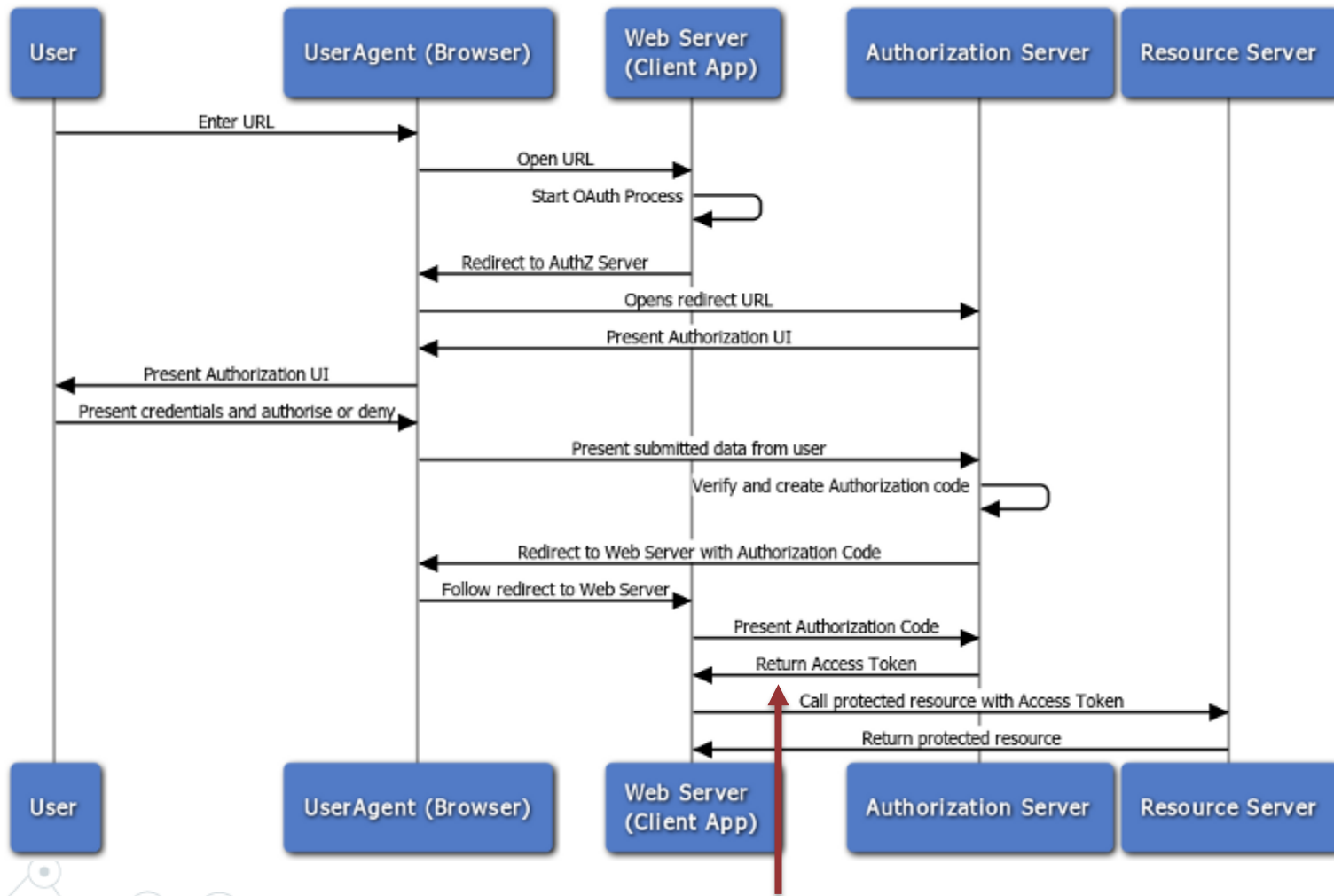


The image shows a cheat sheet for OAuth 2 flows. It features a blue shield icon with a globe inside, followed by the title "OAuth 2 cheat sheet". Below the title is a table with five columns: "OAuth 2 flows", "Needs front end", "Needs back end", "Has user interaction", and "Needs client secret". The rows represent the four grant types: Authorization Code, Implicit Grant, Client Credentials, and Password Grant. Each cell in the table contains a green checkmark (✓) or a red X (✗) to indicate the requirements for each flow.

<i>OAuth 2 flows</i>	<i>Needs front end</i>	<i>Needs back end</i>	<i>Has user interaction</i>	<i>Needs client secret</i>
Authorization Code	✓	✓	✓	✓
Implicit Grant	✓	✗	✓	✗
Client Credentials	✗	✓	✗	✓
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/>



Firebase Authentication

<https://firebase.google.com/pricing>

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

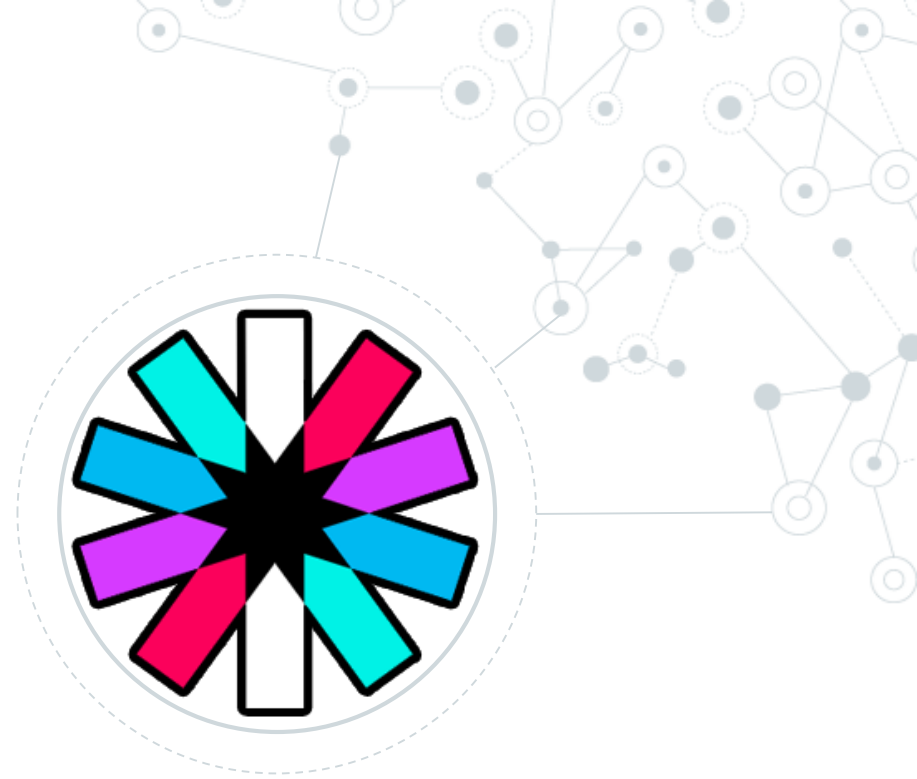
<https://auth0.com/docs/quickstart/spa/angular2/01-login>

<https://github.com/angular/angularfire>

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

# JWT

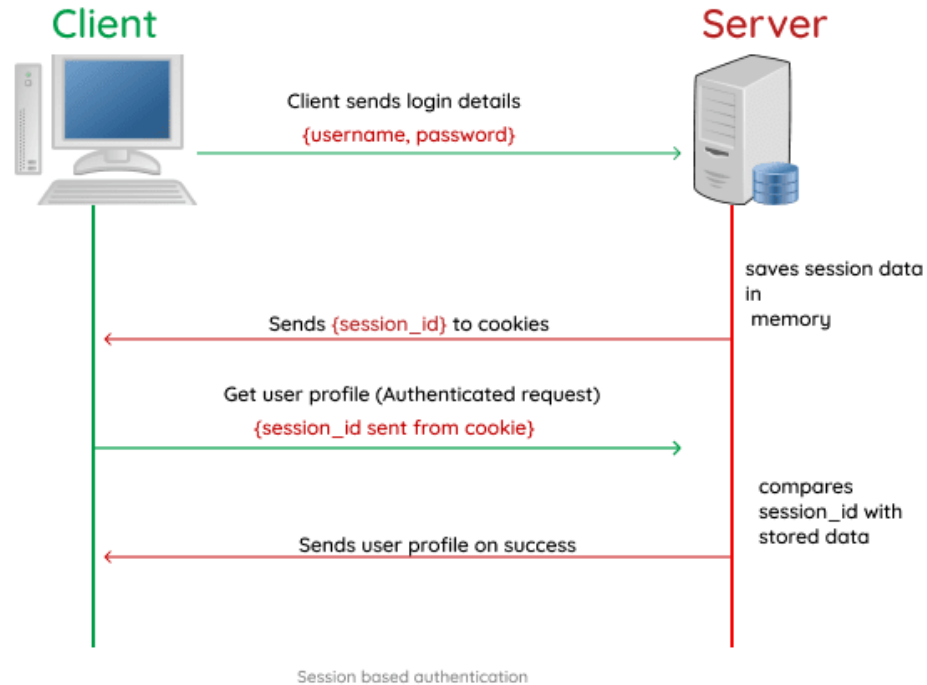
JWTs can be used as OAuth 2.0 [Bearer Tokens](#)



# JSON Web Token (JWT) rfc7523

## Perché è stato introdotto?

### Utilizzo classico delle sessioni



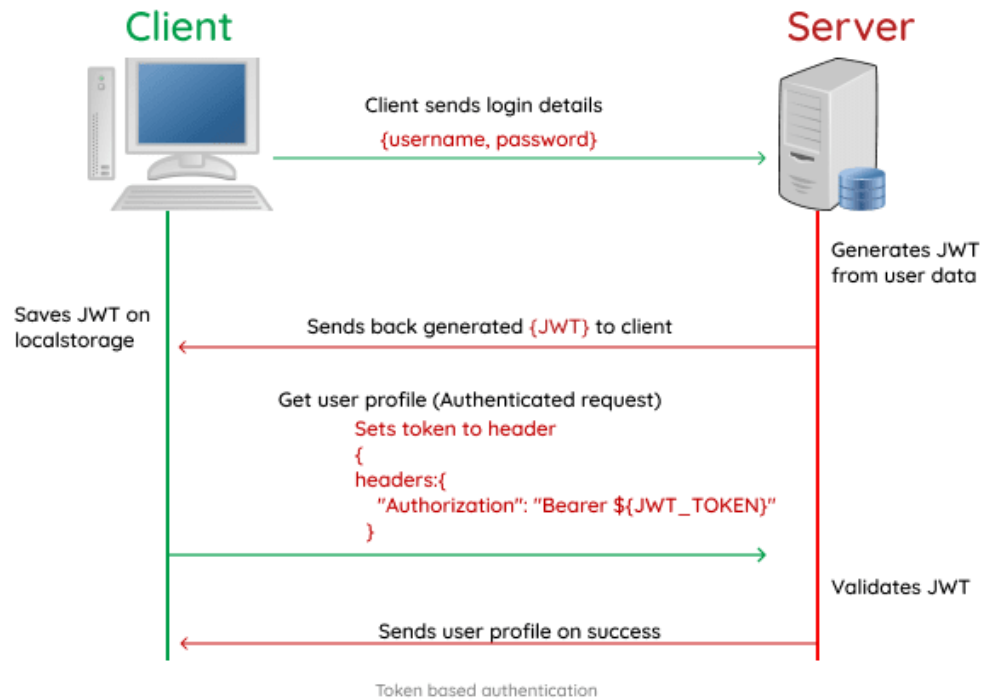
### Svantaggio delle sessioni:

- 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!**

JWT  
JSON WEB TOKEN



HEADER  
ALGORITHM  
& TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

+

PAYLOAD  
DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

+

SIGNATURE  
VERIFICATION

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),secretKey)
```

NORDICAPIS.COM

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.