

Building Progressive Web Apps

Daniele Bochicchio

Chief Digital Officer @icubedsrl

Founder @aspitalia

Microsoft Regional Director, ASP.NET MVP

daniele@aspitalia.com | @dbochicchio



Microsoft
Regional Director



Agenda

- PWA: cosa sono?
- PWA: come si creano?
- PWA: parto da zero?
- PWA: adattare un sito esistente
- Performance

Progressive Web Apps

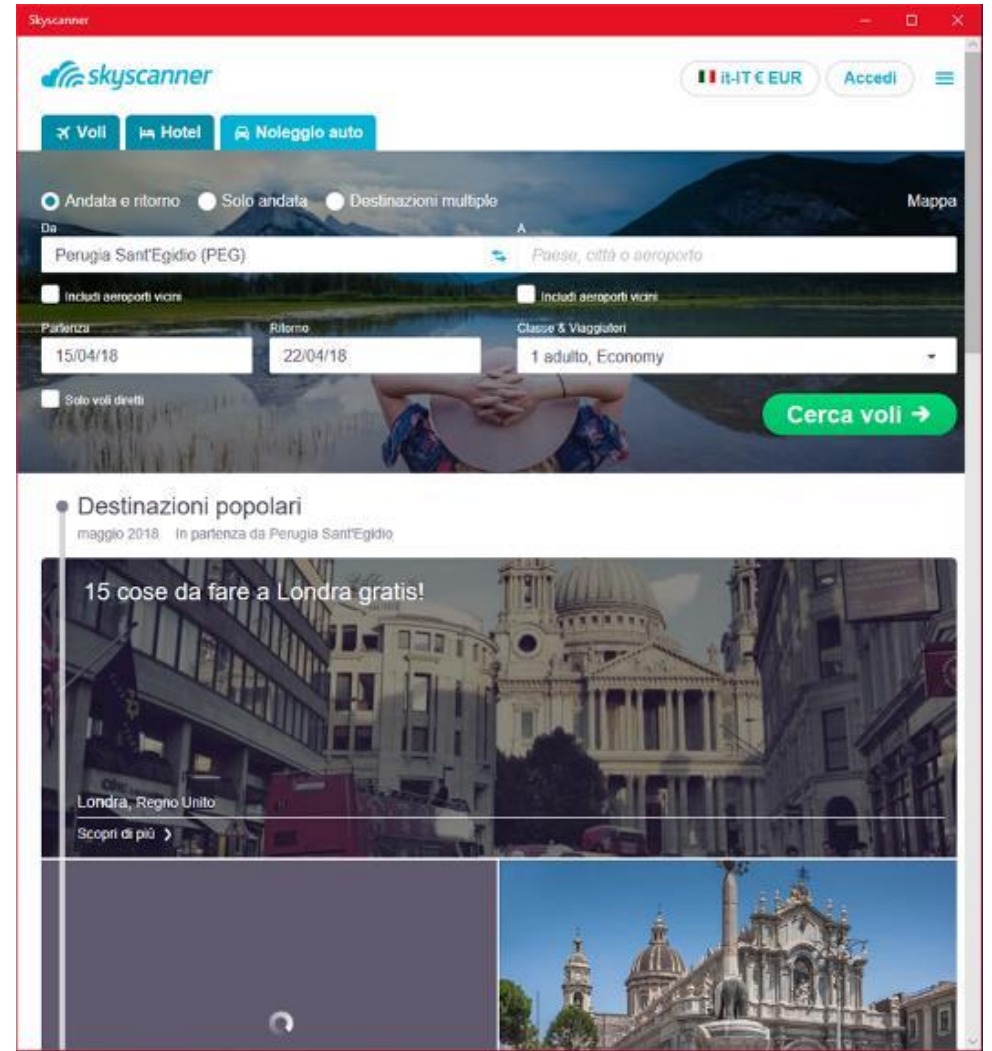
PWA = applicazioni web che diventano app

Si possono scrivere con i framework preferiti

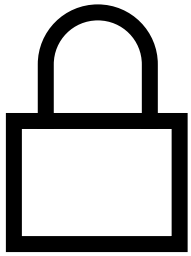
- Angular, React, vue.js – perfino jQuery

Le PWA devono

- essere responsive e fluide
- funzionare offline



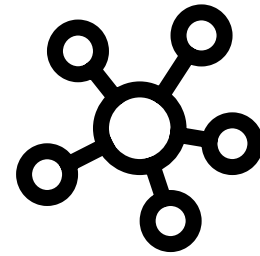
Pre-requisiti di una PWA



HTTPS



Web
Application
Manifest



Service
Worker

Progressive Web Apps

Tecnicamente, una PWA sfrutta questi concetti

Manifest: descrive l'app e i suoi contenuti

Service Worker:

- consente di eseguire processi in background, ricevere notifiche push

- gestisce la cache

- gestisce le chiamate di rete

PWA: da web ad app

Una PWA può essere "generata" a partire da un sito web esistente

Non serve ripartire da zero

Adattare un sito è abbastanza semplice

Vedi <https://www.pwabuilder.com>

PWA: codice minimo da aggiungere

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"><title>Insider Dev Tour</title>
<link rel="manifest" href="manifest.json">
<script>
// Check compatibility for the browser we're running this in
if ("serviceWorker" in navigator) {
if (navigator.serviceWorker.controller) {
console.log("[PWA Builder] active service worker found, no need to register");
} else {
// Register the service worker
navigator.serviceWorker.register("pwabuilder-sw.js", {
scope: "./"
})
.then(function (reg) {
console.log("[PWA Builder] Service worker has been registered for scope: " +
reg.scope);
});
}
}
</script>
```

Demo

PWA: from zero to hero

Progressive Web Apps

In futuro, grazie a **WebAssembly**, diventerà possibile avere PWA che sono state scritte in un qualsiasi linguaggio e compilate in WASM, un formato portabile (binario) per VM, con codice eseguito direttamente nel browser

Per .NET, tenete d'occhio Blazor (sessione più tardi in questa track)

Sono **applicazioni native** a tutti gli effetti

Cos'è WebAssembly

WebAssembly è un nuovo runtime (+ formato di codice) che gira nativamente sui browser

È un linguaggio di basso livello simile all'assembly con un formato binario compatto che gira con performance paragonabili al codice native.

Aggiunge un compilation target (.wasm) per linguaggi come C, C++, Rust (o .NET) che li fa girare nel web.

È pensato per girare insieme a Javascript, con cui può girare insieme e con cui può condividere l'accesso a DOM/routine.



Caricare un file wams in memoria

```
var importObject = {
  imports: {
    imported_func: function(arg) {
      console.log(arg);
    }
  }
};

fetch('simple.wasm').then(
  response => response.arrayBuffer()
).then(bytes => WebAssembly.instantiate(bytes, importObject)
).then(result => result.instance.exports.exported_func()
);
```

Com'è fatto un file .wasm

```
;; simple.wasm
(module
  (func $i (import "imports" "i") (param i32))
  (func (export "e")
    i32.const 42
    call $i))
```

Esempi di Web Assembly

<https://s3.amazonaws.com/mozilla-games/ZenGarden/EpicZenGarden.html>

WebGL 1.0, BananaBread Engine

<https://www.funkykarts.rocks/demo.html>

C++ to WASM using Emscripten

Demo

<https://david.blob.core.windows.net/idt2019/wasm/index-video.html>

Web Components

Un insieme di standard web per costruire componenti che si comportano e sembrano come component native della pagina.

Incapsulano markup, CSS e codice in modo tale che funzionino con qualsiasi pagina o framework.

Supporto cross-browser con polyfill per supporto per browser più vecchi.

Web Component di esempio

```
<script>
  customElements.define('x-new-markup', class extends HTMLElement {
    connectedCallback() {
      this.innerHTML = "<b>Hello, welcome to the  </b>";
    }
  });
</script>
```

```
<x-new-markup>Club</x-new-markup>
```

Hello, welcome to the Club

Cosa manca per avere vere app?

- Notifiche push
 - Ci pensa il Service Worker
- Background task
 - Ci pensa il Service Worker :)
- Accesso all'hardware
 - Specifiche ad hoc: Web Bluetooth (BT LE), Web USB, Web Midi, WebVR/XR e altre in arrivo

Esempio di notifiche push con Service Worker

```
self.addEventListener('push', function (event) {
  event.waitUntil(handlePushEvent(event));
});

//Respond to the user clicking the toast notification
self.addEventListener('notificationclick', function (event) {
  console.log('On notification click: ', event.notification.tag);
  event.notification.close();

  const urlToOpen = new URL('/chat', self.location.origin).href;
  console.log(urlToOpen);

  // This looks to see if the current is already open and focuses it
  event.waitUntil(clients.matchAll({
    type: 'window'
  }).then(function (clientList) {
    for (var i = 0; i < clientList.length; i++) {
      var client = clientList[i];
      console.log('client.url:' + client.url);

      if (client.url === urlToOpen && 'focus' in client)
        return client.focus();
    }
    if (clients.openWindow)
      return clients.openWindow('/chat');
  }));
});
```

Demo

From Angular to PWA with love
<https://cday.azurewebsites.net/>

about://flags

Experimental Web Platform Features
Service Worker Push Messaging

Chi ha già fatto una PWA?

- Facebook
- Twitter
- Pinterest
- Google
- Trivago
- E tanti altri!

Demo

ASPIItalia.com PWA

`about://flags`

Experimental Web Platform Features

Service Worker Push Messaging

Nuovi libri su .NET Core, C# 8, VB 2019

Nuovi libri su

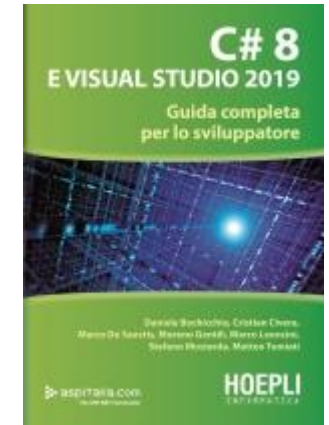
ASP.NET Core 2.1

C# 8 e .NET Core 3

VB 2019 e .NET Core 3

<http://books.aspitalia.com/>

CommunityDays.it



Grazie!

- Il materiale sarà online nei prossimi giorni su <http://www.communitydays.it>