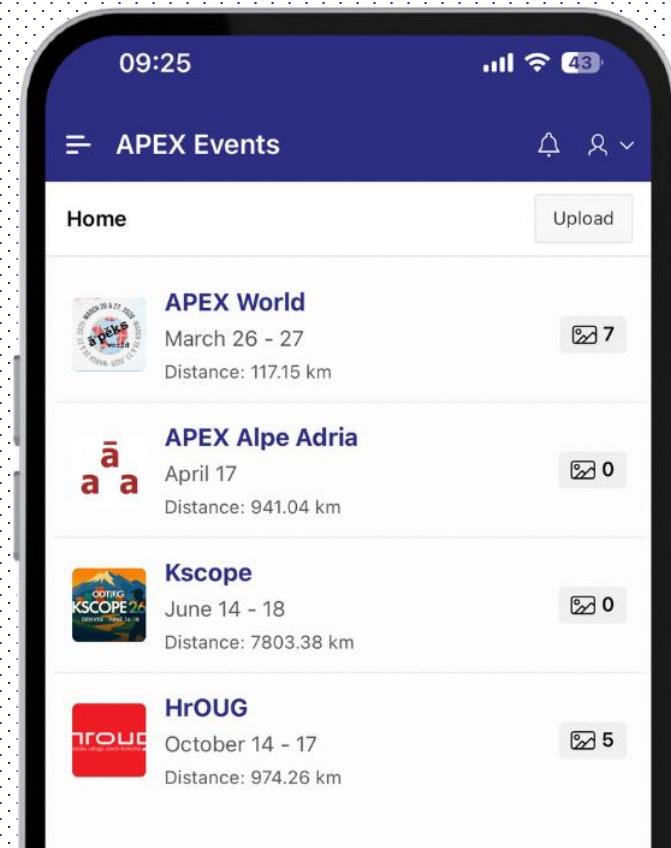


PWA



Mobile first: Building a PWA in APEX

Kevin Thyssen

March 27 – Ede, The Netherlands

Extend APEX like a PRO



APEX Office Print
Printing & Reporting
built for APEX



APEX Media Extension
Image processing for the
Oracle Database.



APEX Office Edit
Document management
for APEX



Plug-ins Pro
Highly requested features
and functionality to
extend APEX

APEX Message Service
Real-time communication
for Oracle APEX



APEX Project Eye
Productivity, Quality Assurance,
and Security tool for APEX





 Oracle ACE
Associate

Senior APEX consultant @ United Codes

Located in Belgium

15+ years experience in Oracle APEX

Cycling enthusiast



@kthyssen.bsky.social



@kevinthyssen



kevin.thyssen@united-codes.com



kevinthyssen

Progressive Web App

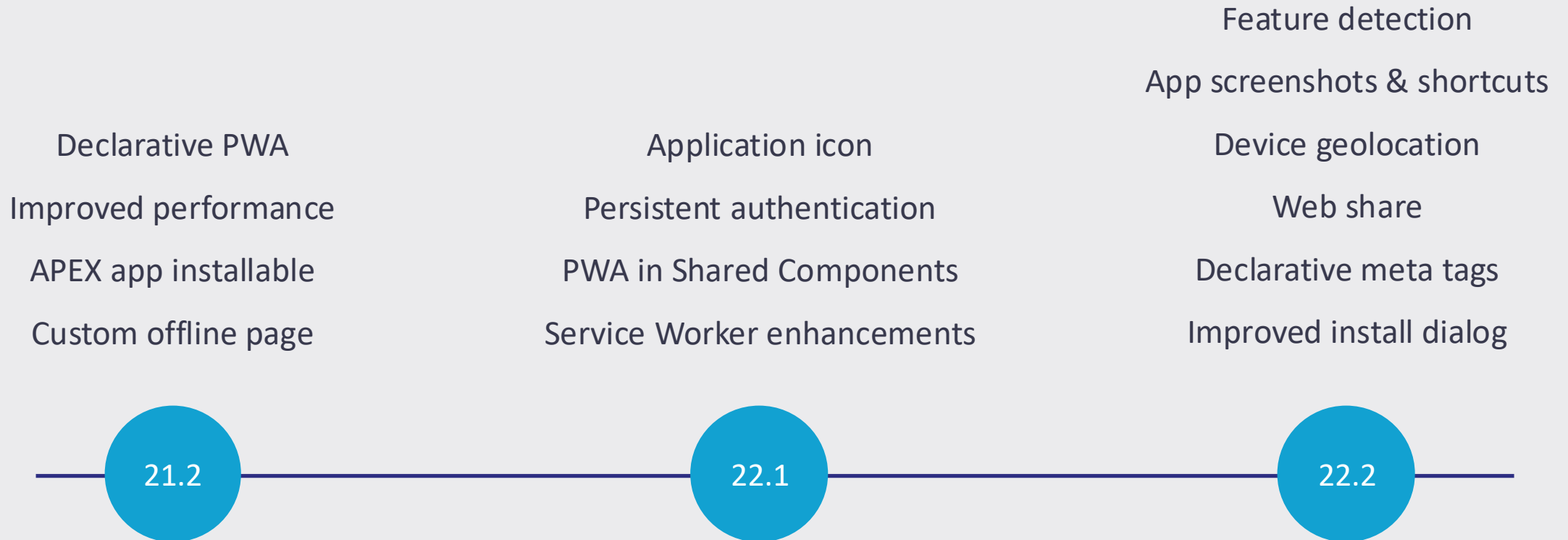
PWA

Can operate both as a website and mobile app

Built using common web technologies
JavaScript, CSS & HTML

Can take advantage of native mobile device features
without requiring an app store

PWA - APEX



PWA - APEX

Push Notifications

Improved application icon fidelity

23.1

QR Code Generator

Image Upload

23.2

PWA – APEX – Create Application

The screenshot shows the Oracle APEX App Builder interface for creating a new application. The browser address bar shows the URL: `ofxrel3koy69ptc-db23ai.adb.eu-frankfurt-1.oraclecloudapps.com`. The user is logged in as Kevin Thyssen (diversus).

Create an Application

Name: APEX Events

Appearance: Vita, Side Menu

Pages: Add Page

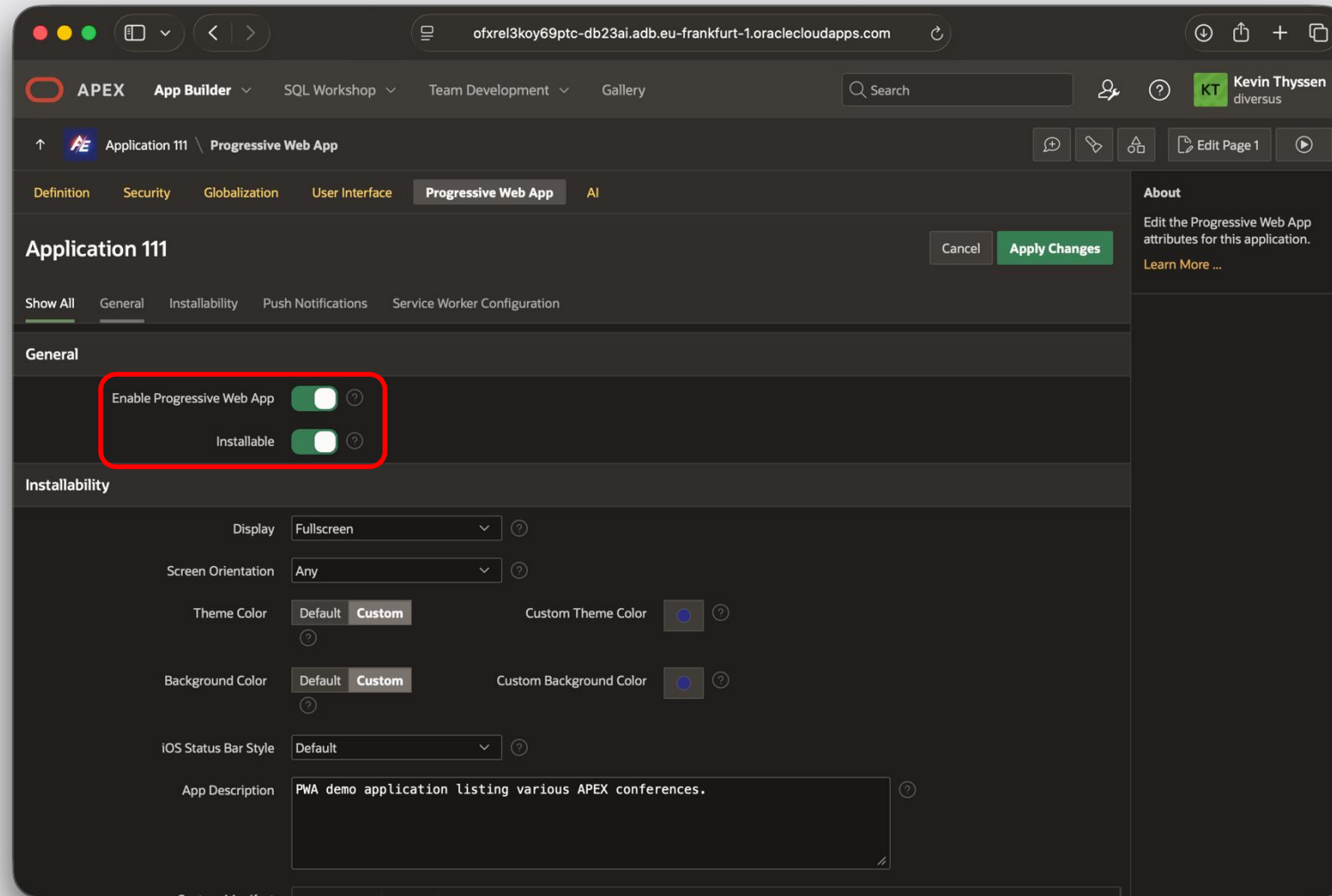
Features: Check All

- Install Progressive Web App**
Give your app the ability to be installed
- Push Notifications**
Allow users to receive push notifications
- Access Control**
Enable role-based user authorization
- Activity Reporting**
Include user activity and error reports
- Feedback**
Allow users to provide feedback
- Theme Style Selection**
Update default application look and feel
- About Page**
Add about this application page
- Configuration Options**
Enable or disable application features

Buttons: Cancel, Create Application

PWA – APEX – Existing Application

Application → Shared Components → Progressive Web App



PWA – (APEX) Requirements

Served using HTTPS

Enable Friendly URLs

The web app must have a web manifest file

A service worker must be registered

PWA – Web Manifest

JSON file that tells the browser about your PWA

How it should behave when installed on
the user's desktop or mobile device

Includes app name, icons, start url,
shortcuts, screenshots, ...



```
// Default manifest.json from APEX
```

```
{
  "id": "111",
  "name": "APEX Events",
  "start_url": "/ords/r/diversus/apex-events/home",
  "display": "fullscreen",
  "orientation": "any",
  "description": "PWA demo application listing various APEX conferences.",
  "theme_color": "#2d2e82",
  "background_color": "#2d2e82",
  "dir": "ltr",
  "lang": "en",
  "icons": [{
    "src": "/ords/r/diversus/111/files/static/v61/icons/app-icon-512.png",
    "sizes": "512x512",
    "type": "image/png",
    "purpose": "any"
  }],
  "screenshots": [{
    "src": "/ords/r/diversus/111/files/static/v61/pwa/ae-screen-1-bg.png",
    "type": "image/png",
    "label": "Startup Screen"
  }],
  "shortcuts": [{
    "name": "QR Code",
    "description": "Get the QR Code for this App",
    "url": "/ords/r/diversus/apex-events/home",
    "icons": [{
      "src": "/ords/r/diversus/111/files/static/v61/pwa/shortcut-icon-10.png",
      "sizes": "192x192"
    }]
  }]
}
```

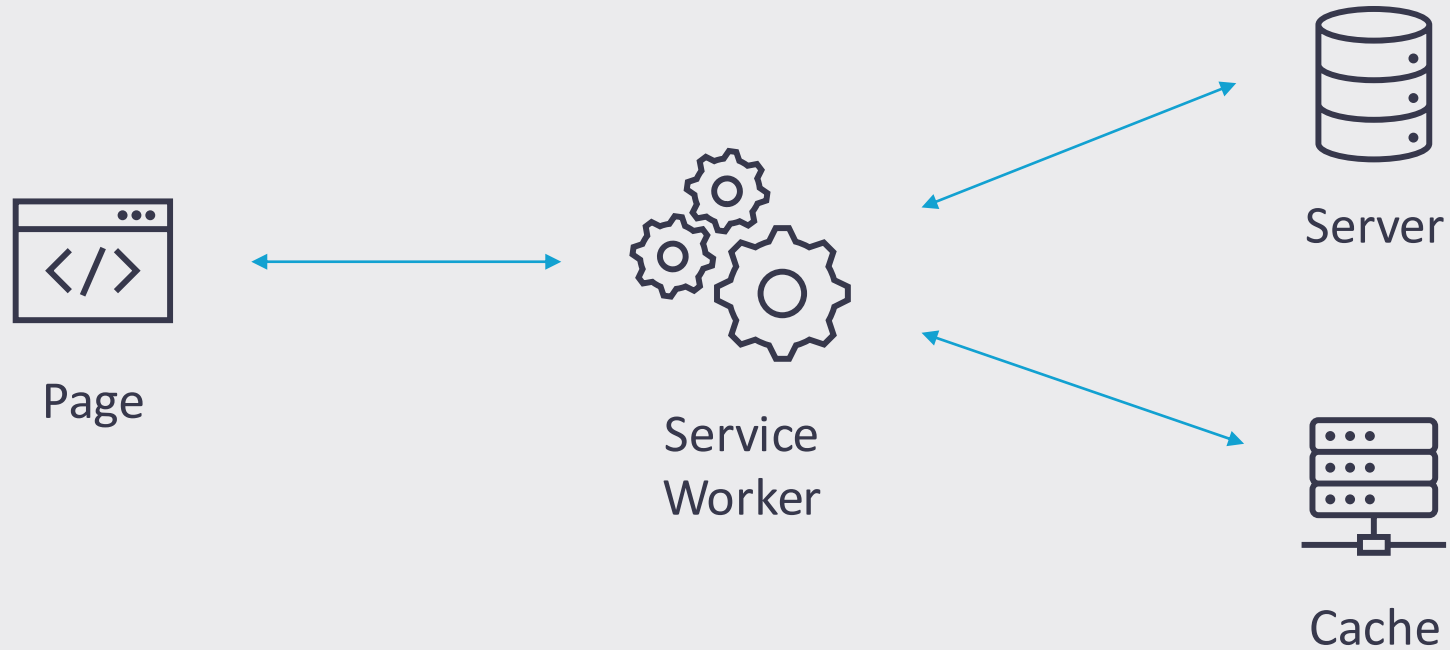
PWA – Service Worker

Foundation of your PWA, written in JavaScript

Allows intercepting and control of network requests
and asset caching from the web browser

Can execute JavaScript code even when the app is not in use

PWA – Service Worker



PWA – Service Worker - APEX

Install and activate the service worker

Serve resources from cache if cache exists

Otherwise serve from network, then put resource in cache

Serve an offline page if network fails

PWA – Service Worker - Hooks

Allows developers to modify or extend the Service Worker

Add code to existing events:
install, activate, fetch, push, ...

PWA – Service Worker - Hooks

Event: push ?

Default **Configure Hooks** Custom

Before ?

1

After ?

1 `console.log(notification);`

Preview

```
self.addEventListener("push", (event) => {  
  // "Before" hook starts  
  
  // "Before" hook ends  
  
  const notification = event.data.json();  
  self.registration.showNotification( notification.title, notification );  
  
  // "After" hook starts  
  console.log(notification);  
  // "After" hook ends  
});
```

PWA – Console

The screenshot shows the Chrome DevTools Application panel. The 'Application' tab is selected and highlighted with a red box. In the left sidebar, the 'Service workers' folder is also highlighted with a red box. The main panel displays the configuration for a service worker at the URL `https://ofxrel3koy69ptc-db23ai.adb.eu-frankfurt-1.oraclecloudapps.com/ords/r/diversus/apex-events/`. The source is `sw.js`, received on 3/17/2026 at 8:46:36 PM. The status is '#8718 activated and is running' with a 'Stop' button. The clients list shows the current page. There are input fields for testing push messages, sync tags, and periodic sync, each with a corresponding action button. The update cycle table shows two instances: one installed and one waiting.

Version	Update Activity	Timeline
▶ #8718	Install	
▶ #8718	Wait	

PWA – Persistent Authentication

Internal Workspace → Manage Instance → Security

The screenshot displays the Oracle APEX 'Security' settings page. The breadcrumb trail is 'Manage Instance > Security'. The page title is 'Security' with a green 'Apply Changes' button. Below the title, there are tabs for 'Security Settings' and 'Authorized URLs'. A navigation bar includes 'Show All', 'Security', 'HTTP Protocol', 'Real Application Security', 'Session Timeout', 'Workspace Isolation', 'Region and Web Service Excluded Domains', 'Authentication Control', and 'Password Policy'. The main content area is titled 'Security' and contains a list of settings:

- Set Workspace Cookie: Yes
- Allow Persistent Auth: Yes** (highlighted with a red box)
- Persistent Authentication Lifetime Days: 30
- Disable Administrator Login: No
- Disable Workspace Login: No
- Disable Apps Login: No
- Allow Public File Upload: No
- AI Enabled: Yes
- Allow Database Credentials Usage: No
- Restrict Access by IP Address: [Empty field]
- App Builder Proxy for AI Services: [Empty field]
- Instance Proxy: [Empty field]
- Checksum Hash Function: Most Secure
- Rejoin Sessions: Enabled for Public Sessions

On the right side, there is an 'About' section with the text: 'Manage settings that apply to the entire Oracle APEX instance.'

Push Notifications

PWA – Push Notifications (23.1)

Can be enabled declaratively in Shared Components

Push Notifications Feature pages with settings and user preferences

Send notifications using a native Page Process or APEX_PWA API

APEX views for user subscriptions and queue

PWA – Pros

No App Store required

Cross-platform compatibility

Offline functionality *

Automatic updates

Secure

PWA – Cons

Limited iOS support (improving lately, but still behind)

Reduced access to device features

Performance limitations (e.g.: heavy animations)

Browser dependency

https://oracleapex.com/jsapi

The screenshot shows a web browser window displaying the Oracle APEX JS API documentation. The browser's address bar shows 'docs.oracle.com'. The page title is 'Namespace: pwa'. On the left, there is a navigation menu with 'Index' at the top, followed by 'Namespaces' and a list of namespace paths including 'apex.pwa'. Below that are 'Interfaces' and a list of interface names. The main content area is titled 'Namespace: pwa' and features a 'QuickNav' section with a 'Functions' list: 'getDisplayMode', 'getInstallText', 'getPushSubscription', 'hasPushSubscription', 'isInstallable', 'openInstallDialog', 'subscribePushNotifications', and 'unsubscribePushNotifications'. Below this is the 'apex.pwa' section, which states that the namespace contains Oracle APEX functions for Progressive Web App features, available since version 21.2. A 'Functions' section follows, detailing the '(static) getDisplayMode()' function, its return type '{string}', and its usage for determining the current display mode (fullscreen, standalone, minimal-ui, or browser).

Index

Namespaces

- apex
- apex.actions
- apex.da
- apex.date
- apex.debug
- apex.event
- apex.item
- apex.lang
- apex.locale
- apex.message
- apex.model
- apex.navigation
- apex.navigation.dialog
- apex.navigation.popup
- apex.page
- apex.pwa**
- apex.region
- apex.server
- apex.storage
- apex.theme
- apex.util
- apex.util.delayLinger
- apex.widget

Interfaces

- actions
- cardsRegion
- facetsRegion
- htmlBuilder
- interactiveGridView
- item
- mapRegion
- model
- numberFieldItem

Namespace: pwa

QuickNav

Functions

- [getDisplayMode](#)
- [getInstallText](#)
- [getPushSubscription](#)
- [hasPushSubscription](#)
- [isInstallable](#)
- [openInstallDialog](#)
- [subscribePushNotifications](#)
- [unsubscribePushNotifications](#)

apex.pwa

The apex.pwa namespace contains Oracle APEX functions related to Progressive Web App features.

These functions are useful only when an APEX application has enabled Progressive Web App.

Since: 21.2

Functions

(static) `getDisplayMode()` → {string} ■

Get the current display mode for the PWA.

Possible values are: fullscreen, standalone, minimal-ui, browser.

The display mode is set by the developer in the application definition.

This function is used to determine if the application is currently accessed through the PWA application (eg. in fullscreen) or through the browser normally.

[Cookie Preferences](#) | [Ad Choices](#)

Demo

Overview of PWA features in APEX

Geolocation

Uploading Images using MLE

Sending Push Notifications







https://oracleapex.com/pwa

APEX PWA Reference Install App







PWA Progressive Web Apps in APEX

About

Oracle APEX provides developers with the ability to create Progressive Web Apps (PWAs) that can be easily installed on any desktop or mobile device, offering users a more native app experience. This application serves as a useful reference for developers looking to incorporate key PWA features into their own applications, helping them to create more engaging and user-friendly apps.

 Faster Performance Take advantage of advanced caching for even faster page rendering performance.	 Install Anywhere Your app can be installed on any desktop or mobile devices without downloading from an app store.
 Feels Native Your app runs fullscreen, can be added to the home screen, provide shortcuts actions, and is searchable.	 Lightweight APEX PWAs have a tiny footprint on devices and take virtually no space compared to native apps.
 Device Integration Send notifications, fetch geolocation, share APEX data to other apps, access the camera, and more.	 Fully Branded Customize your app's appearance, including home screen icons, color scheme, logo and more.

Explore

 Getting Started	 Installation	 Appearance
 App Icon	 Push Notifications	 Geolocation

https://whatpwacando.today

Danny Moerkerke – Member of the APEX DEV Team

PWA

What PWA Can Do Today

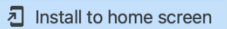
A showcase of what is possible with Progressive Web Apps today.

A Progressive Web App (PWA) is a website that can be installed on your device and provide an app-like experience [read more](#)

How to use this app

This app is itself a PWA so you can install it on your device and then check the [demos](#) below to see what is supported on your device.

When the button below becomes enabled, you can install this app.

 Install to home screen

Stay up to date

Join my email list for a weekly update on PWAs and new features of the modern web, tested and explained in plain English.


[Subscribe here](#)

Need help?


If you run into issues when implementing a PWA I can help by running an [audit](#) of it.

Demos


Check the demos to see what is supported on your device.

**Installation**






Using the beforeinstallprompt event, a native dialog can be displayed to install a web app

**Offline support**

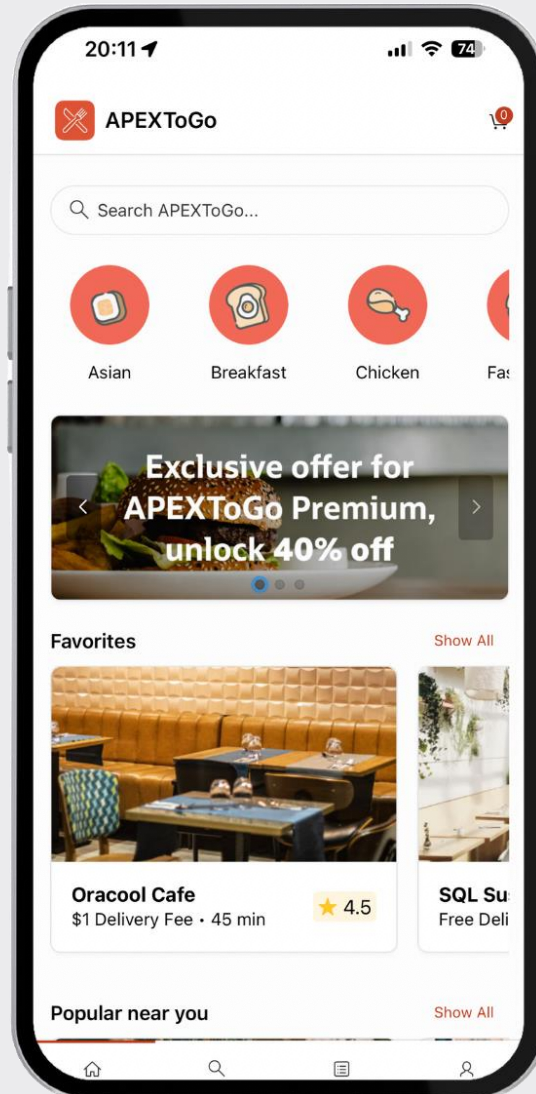
The Service Worker enables web apps to work offline

**Notifications**

The Notifications API enables web apps to display notifications, even when the app is not active.

 Home  Audit  Email list  Bugs  Reload

https://oracleapex.com/ords/r/apex_pm/apextogo



MLE Blog Posts

How to Use Server-Side JavaScript in Oracle APEX with Oracle 23ai #JoelKallmanDay

<https://lmoreaux.hashnode.dev/how-to-use-server-side-javascript-in-oracle-apex-with-oracle-23ai-joelkallmanday>

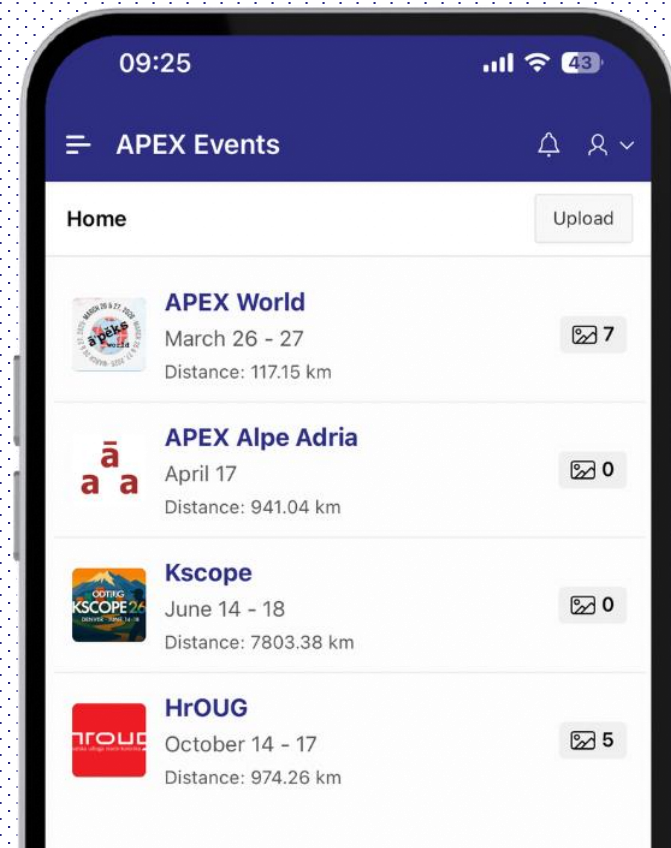
Louis Moreaux

Using the MLE PL/SQL Foreign Function Interface in APEX

<https://martincarstenbach.com/2025/11/16/using-the-mle-pl-sql-foreign-function-interface-in-apex/>

Martin Bach

PWA



Thank you!

kevin.thyssen@united-codes.com