APEX Unplugged
Building Oracle Application Express Applications That Can Survive Without the Internet

Dan McGhan
Senior Technical Consultant
My background

- Dan McGhan  
- Senior consultant with Enkitec 
  - Joined in March 2013 
  - dmcghan@enkitec.com 
- Co-Author, Expert Oracle Application Express 
- Regular presenter at various conferences 
- A.K.A “that guy that wrote that plug-in”
About Enkitec

- Oracle Platinum Partner
  - Established in 2004
  - Headquartered in Dallas, TX
  - Locations throughout the US & EMEA
- Specialties include
  - Exadata Implementations
  - Development Services
    - PL/SQL / Java / APEX
  - DBA/Data Warehouse/RAC
  - Business Intelligence
Can you hear me now?

Hello???
MISSION IMPOSSIBLE
New tools

- Cache Manifest
- On/Offline Events
- Local Storage
New tools with 2 apps

• Cache Manifest
• On/Offline Events
• Local Storage
MISSION: ABORTED
APEX Unplugged???

• A more accurate title:

  • How to create an APEX application such that when the Internet is not available it can switch to a cached offline application with reduced functionality and then synchronize with the online application when the Internet is available again
No longer just theory

- Concepts have been implemented successfully
- Now 2 good use cases
  - Survive offline
    - Most logic done in APEX
    - Minimal custom code
  - Thrive offline
    - APEX provides authentication and means to download and upload data
    - Mostly custom code
Part 1: Creating the base applications
Demo

- Creating the online APEX application
  - 3 pages: Home, Listview & Form combo on emp
  - Just use the wizard
Demo

- Creating the offline JQM application
  - Created offline directory in listener docroot
  - Added jQuery & JQM
  - 3 pages that mimic the online pages
Part 2: Making the JQM app offline(able)
Cache Manifest basics

- HTML5 storage feature
  - Allows access to web apps without a network connection
  - Tells browsers to store files indefinitely
- Really just a file that lists required files
  - File must be served with MIME type: text/cache-manifest
  - Must be added to HTML element of app
Adding the MIME type to glassfish

- Modify default-web.xml
- GLASSFISH_HOME\domains\domain1\config
A simple example

<!-- clock.html -->
<!DOCTYPE HTML>
<html>
<head>
  <title>Clock</title>
  <script src="clock.js"></script>
  <link rel="stylesheet" href="clock.css">
</head>
<body>
  <p>The time is: <output id="clock"></output></p>
</body>
</html>

/* clock.css */
output { font: 2em sans-serif; }

/* clock.js */
setInterval(function () {
  document.getElementById('clock').value = new Date();
}, 1000);
Adding a cache manifest

CACHE MANIFEST
clock.html
clock.css
clock.js

<!-- clock.html -->
<!DOCTYPE HTML>
<html manifest="clock.appcache">
  <head>
    <title>Clock</title>
    <script src="clock.js"></script>
    <link rel="stylesheet" href="clock.css">
  </head>
  <body>
    <p>The time is: <output id="clock"></output></p>
  </body>
</html>
My cache manifest

CACHE MANIFEST

CACHE:
/offline/index.html
/offline/employees.html
/offline/employee-details.html
/offline/libraries/custom/hr.js
/offline/libraries/jquery-mobile/jquery.min.js
/offline/libraries/jquery-mobile/jquery.min.map
/offline/libraries/jquery-mobile/jquery.mobile-1.3.2.min.css
/offline/libraries/jquery-mobile/jquery.mobile-1.3.2.min.js
/offline/libraries/jquery-mobile/jquery.mobile-1.3.2.min.map
/offline/libraries/jquery-mobile/images/ajax-loader.gif
/offline/libraries/jquery-mobile/images/icons-18-black.png
/offline/libraries/jquery-mobile/images/icons-18-white.png
/offline/libraries/jquery-mobile/images/icons-36-black.png
/offline/libraries/jquery-mobile/images/icons-36-white.png

NETWORK:
* 

FALLBACK:
/offline/index.html
Including the cache manifest

- In the APEX app
  - Convenient, no need to go to offline application
  - Ended up with multiple masters due to APEX URLs

- In the JQM app
  - Requires visit to JQM app to cache application
  - Cleaner, only caches what you want
Chrome and the cache manifest

- Developer tools
Chrome and the cache manifest

- AppCache Internals
Demo

- Going offline
  - Added cache manifest
  - Pointed index.html to cache manifest
Part 3: Transitioning between on/offline
4 main timings to address

• Online app not available
  • Already handled via fallback in cache manifest

• User explicitly goes offline or online

• Ajax request fails

• Need to update offline app
Problem: User goes offline

• Some browsers offer an explicit offline mode

• Others determine offline automatically
  • Airplane mode
  • Unplug the network cable
  • Disable wireless

• Can vary by browser manufacture
  • What should happen if one is connected to a network but without Internet connectivity?

• Need to provide transition to offline app
Solution

- Browsers have online/offline events
  - Triggered on the window object
  - navigator.onLine provides access to current status
- JavaScript added to **ONLINE** app

WHEN offline event triggered
  remove home icon
  add button to offline app
END WHEN

WHEN online event triggered
  remove button to offline app
  add home icon
END WHEN
Problem: User goes online

- What goes offline can come back online
- Need to provide transition to offline app
- To avoid logging in again we need the APEX session
  - But where should we get it from?
Local storage

- Local storage provides client side storage
- Better than cookies as data isn’t transferred
- 5MB maximum per domain
- Everything stored as strings
Saving APEX session in LS

• Use local storage to save session id

```javascript
localStorage.setItem('apexSessionId', 'APP_SESSION');
```
Chrome and local storage

- Developer tools
Solution

- JavaScript added to **OFFLINE** app

  WHEN online event triggered
      add button to online app
  END WHEN

  WHEN offline event triggered
      remove button to online app
  END WHEN
Demo

- Handling user explicitly going offline or online
Problem: Ajax request fails

- Mobile apps load pages and other content via Ajax
- If Ajax is used without connectivity an error will occur
Solution

- Utilize jQuery’s Ajax error handler
- JavaScript added to **ONLINE** app

```
WHEN ajax error
    redirect to offline app
END WHEN
```
Demo

- Handling failed Ajax requests
Problem: Offline app updated

- At some point you may update the offline app
- But the browser cached old versions of the files
  - Will not go out and grab newer files automatically
Solution

- Tell the browser to update the cache
  - Automatically done on load if cache manifest is modified
  - Pragmatically via `applicationCache.update()`
- Browser will download files and trigger events
  - Events triggered for downloading, updating, errors...
  - May need to manually swap in the new cache
    - Just reload the page
Using comments to force updates

- Some updates will involve adding/removing files
  - This requires the cache manifest be updated
  - The update to the manifest causes the browser to download the files again

- Some updates will only modify existing files
  - Changing the cache manifest is not technically required
  - Making changes to comments can serve as a workaround

CACHE MANIFEST
#version 1
Solution

• JavaScript added to **ONLINE** app

```
WHEN new app version available
    add link to offline app
END WHEN
```

• JavaScript added to **OFFLINE** app

```
WHEN app cache event triggered
    IF app cache update ready
        THEN
            reload page
        END IF
END IF
END WHEN
```
Demo

- Handling offline app updates
Part 4: Working with JSON & Local Storage
• JavaScript Object Notation
  • Simple structured data format (XML light)
  • Perfect for storing data in the browser for offline use
• No comprehensive support for JSON is Oracle
  • Should see something in the future
• Consider PL/JSON till then
• Or for basic JSON object support try:

```javascript
apex_util.json_from_sql(
  'SELECT ename, empno, job ' ||
  'FROM emp ' ||
  'ORDER BY ename'
);
```
Overcoming LS “strings only”

- Local storage stores simple key/value strings
  - We need to convert JSON when using it with local storage
  - `JSON.stringify`
    - Turns a JSON object into a string equivalent
    - Can be used going into local storage
  - `JSON.parse`
    - Turns the string back into an object
    - Can be used going out of local storage
Storing JSON in LS

- JavaScript added to ONLINE app

WHEN sync button clicked
   call ajax process
   store json output as string in ls
END WHEN
JavaScript added to **OFFLINE** app

WHEN emps page loaded
  parse json string to get object
  loop over emps to build emps list
END WHEN

WHEN emp selected from emps page
  parse json string to get object
  init emp details page with values
  transition to emp details page
END WHEN
### Updating JSON in LS

- **JavaScript added to** OFFLINE **app**

<table>
<thead>
<tr>
<th>WHEN existing emp modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>parse json string to get object</td>
</tr>
<tr>
<td>update existing emp with new values</td>
</tr>
<tr>
<td>stringify json object and store in ls</td>
</tr>
<tr>
<td>END WHEN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WHEN new emp created</th>
</tr>
</thead>
<tbody>
<tr>
<td>parse json string to get object</td>
</tr>
<tr>
<td>add new emp with new values</td>
</tr>
<tr>
<td>stringify json object and store in ls</td>
</tr>
<tr>
<td>END WHEN</td>
</tr>
</tbody>
</table>
Demo

- Manipulating data offline
Part 5: Syncing data when online
Caution: Lost updates ahead

- Data in LS must eventually be synced with the DB
- Be careful if users can modify the same data as others while offline
  - Lost update detection/handling is almost always custom
  - Best to avoid it all together if possible
Showing that updates are ready

- JavaScript added to **ONLINE** app

```
WHEN home page loaded
    parse json string to get object
    count modified and new emps
    add count bubble to sync button
END WHEN
```
Pushing updates to server

- JavaScript/Ajax proc added to app

WHEN sync button clicked
  parse json string to get object
  loop over emps to build delimited string
  pass string to ajax process
END WHEN

WHEN ajax process called
  parse delimited string to array of emps
  loop over emps
    parse emp string to array of attrs
    update emp with new values
  emit updated json object
END WHEN
Demo

- Syncing data online
Gotchas

- Always use full paths to access files offline
  - Fallback keeps online path
- Clear standard cache when needed
  - Use version parameter
  - Web developer
Hopes, next steps & questions

- I’d like to see built-in support for offline in APEX
  - Would start with clean(er) URLs
  - Eventually lead to declarative options/capabilities

Next steps
- Try the demo app
- Learn more

- Any questions???