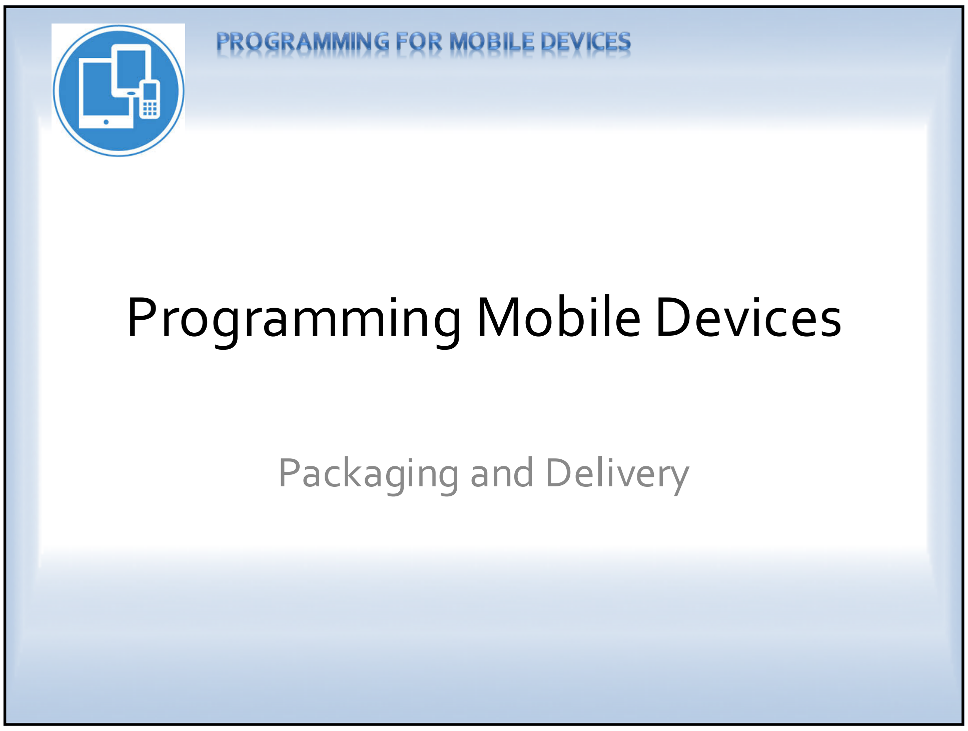
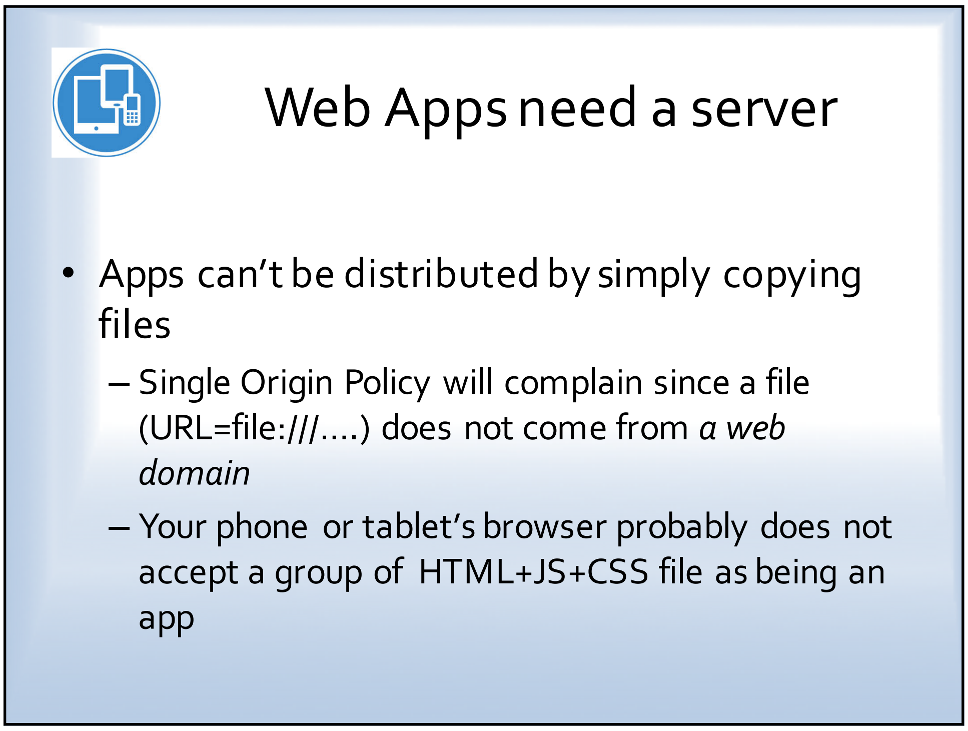


PROGRAMMING FOR MOBILE DEVICES

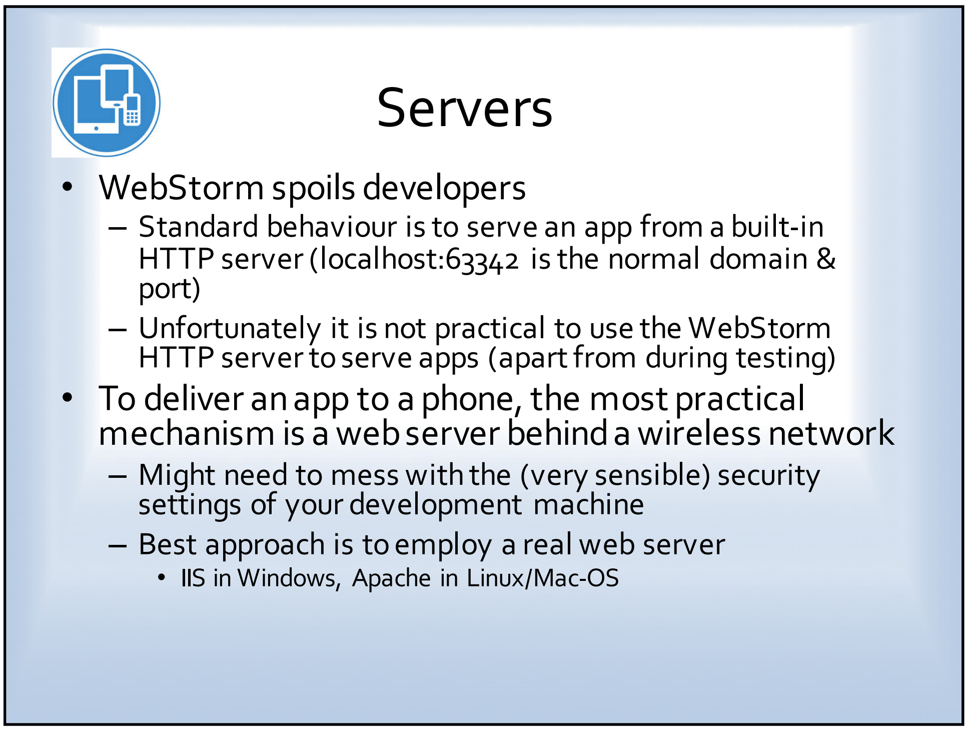
Programming Mobile Devices

Packaging and Delivery



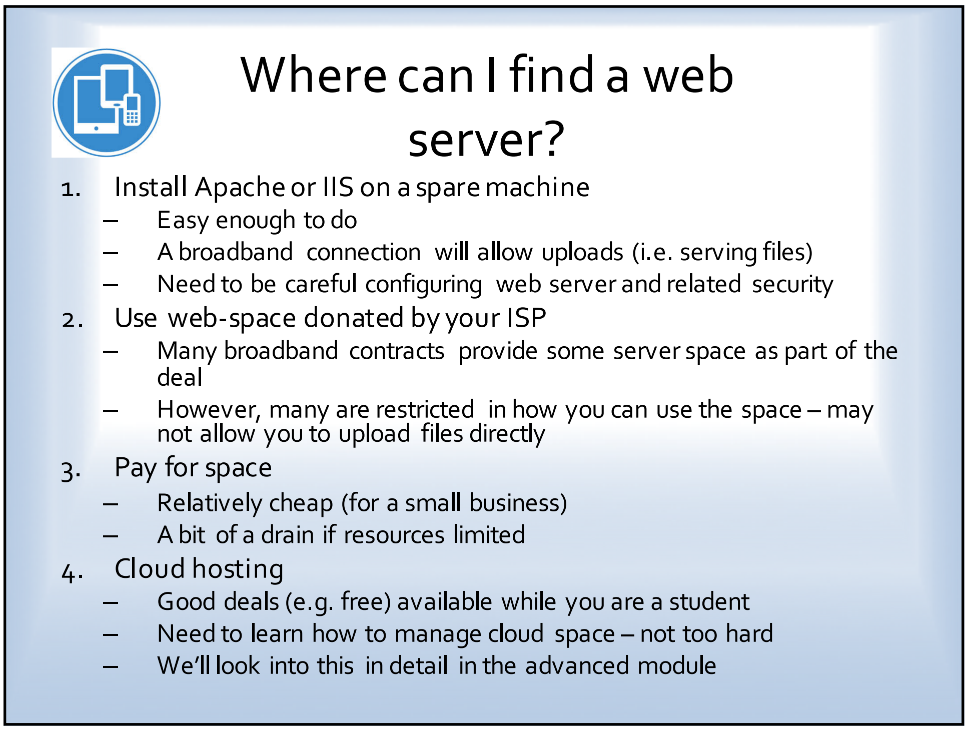
Web Apps need a server

- Apps can't be distributed by simply copying files
  - Single Origin Policy will complain since a file (URL=file:///....) does not come from *a web domain*
  - Your phone or tablet's browser probably does not accept a group of HTML+JS+CSS file as being an app



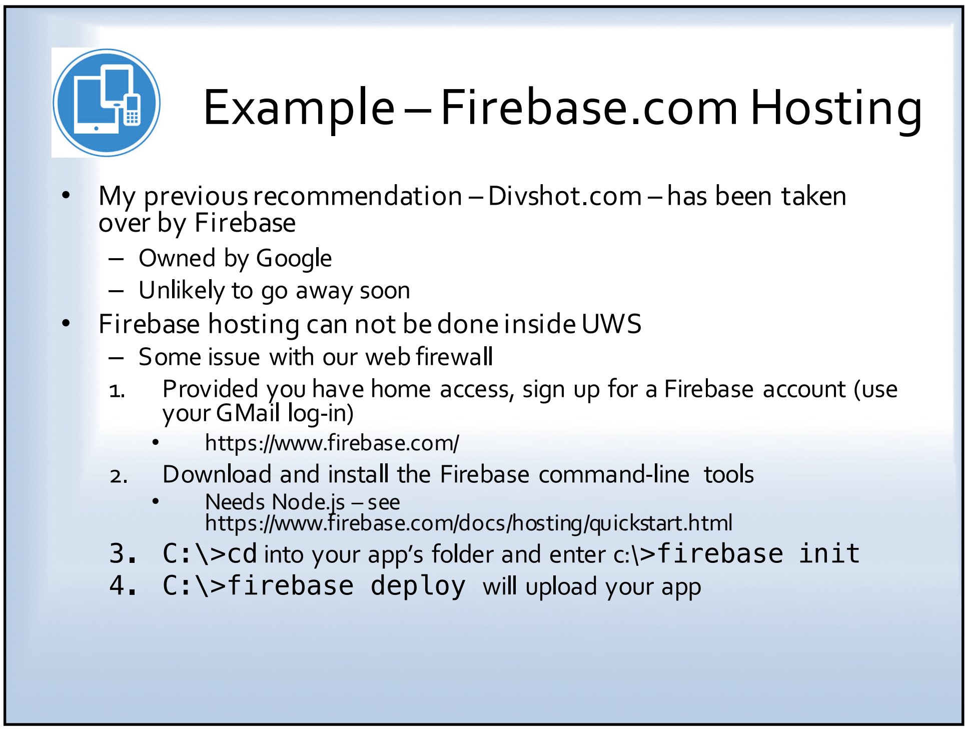
Servers

- WebStorm spoils developers
  - Standard behaviour is to serve an app from a built-in HTTP server (localhost:63342 is the normal domain & port)
  - Unfortunately it is not practical to use the WebStorm HTTP server to serve apps (apart from during testing)
- To deliver an app to a phone, the most practical mechanism is a web server behind a wireless network
  - Might need to mess with the (very sensible) security settings of your development machine
  - Best approach is to employ a real web server
    - IIS in Windows, Apache in Linux/Mac-OS



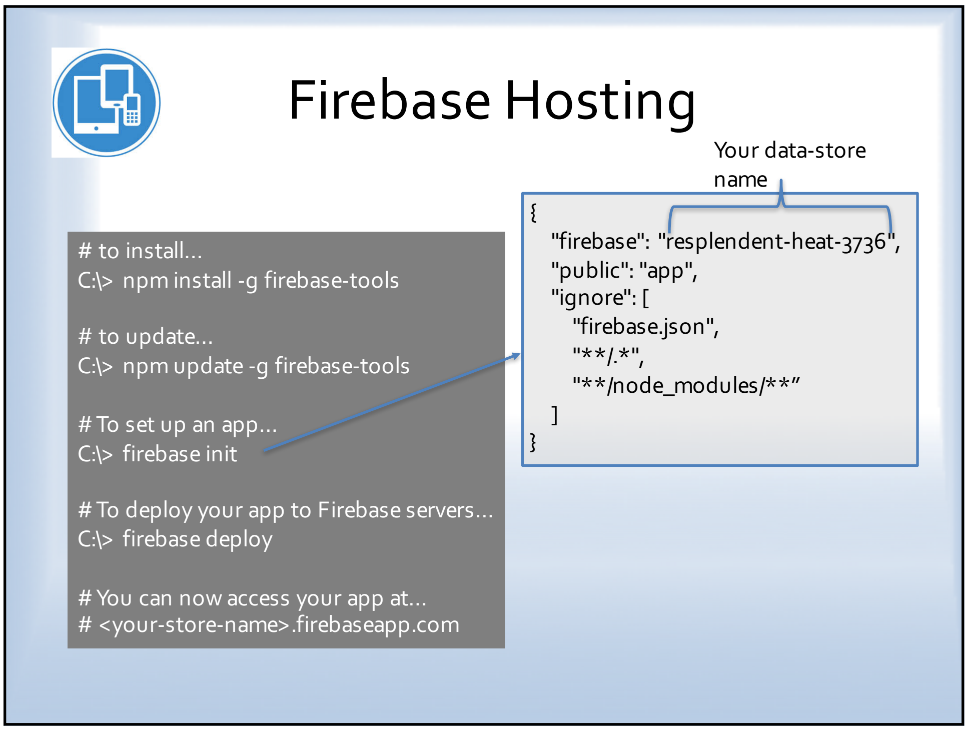
Where can I find a web server?

1. Install Apache or IIS on a spare machine
   - Easy enough to do
   - A broadband connection will allow uploads (i.e. serving files)
   - Need to be careful configuring web server and related security
2. Use web-space donated by your ISP
   - Many broadband contracts provide some server space as part of the deal
   - However, many are restricted in how you can use the space – may not allow you to upload files directly
3. Pay for space
   - Relatively cheap (for a small business)
   - A bit of a drain if resources limited
4. Cloud hosting
   - Good deals (e.g. free) available while you are a student
   - Need to learn how to manage cloud space – not too hard
   - We'll look into this in detail in the advanced module



Example – Firebase.com Hosting

- My previous recommendation – Divshot.com – has been taken over by Firebase
  - Owned by Google
  - Unlikely to go away soon
- Firebase hosting can not be done inside UWS
  - Some issue with our web firewall
  1. Provided you have home access, sign up for a Firebase account (use your GMail log-in)
     - https://www.firebase.com/
  2. Download and install the Firebase command-line tools
     - Needs Node.js – see https://www.firebase.com/docs/hosting/quickstart.html
  3. `C:\>cd` into your app's folder and enter c:\>`firebase init`
  4. `C:\>firebase deploy` will upload your app

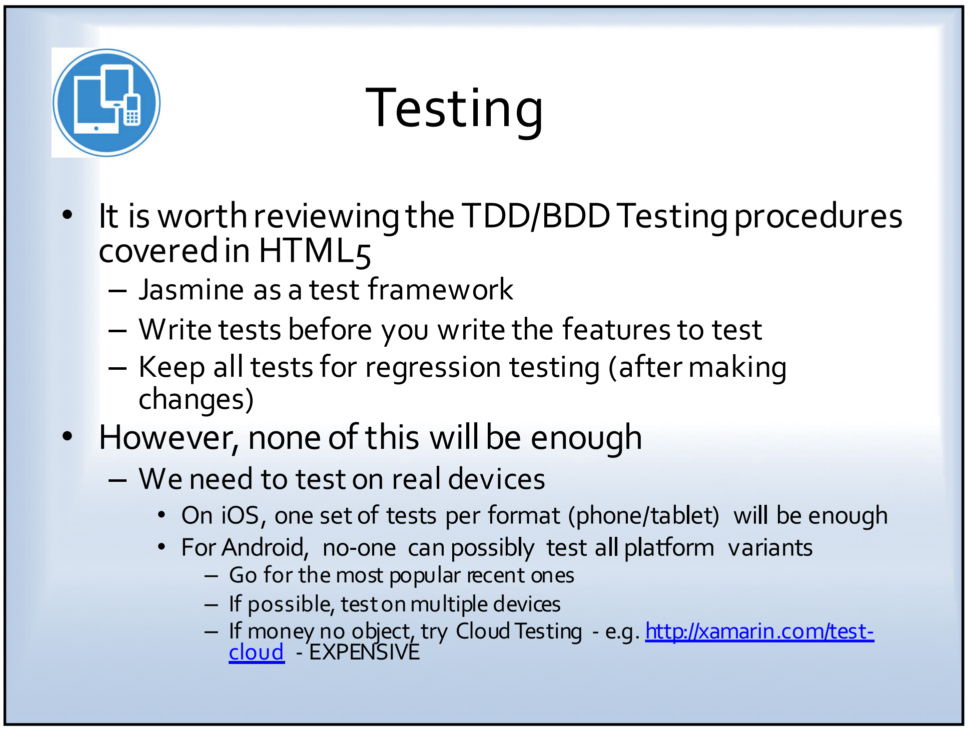


Firebase Hosting

```
# to install...
C:\> npm install -g firebase-tools

# to update...
C:\> npm update -g firebase-tools

# To set up an app...
C:\> firebase init

# To deploy your app to Firebase servers...
C:\> firebase deploy

# You can now access your app at...
# <your-store-name>.firebaseapp.com
```

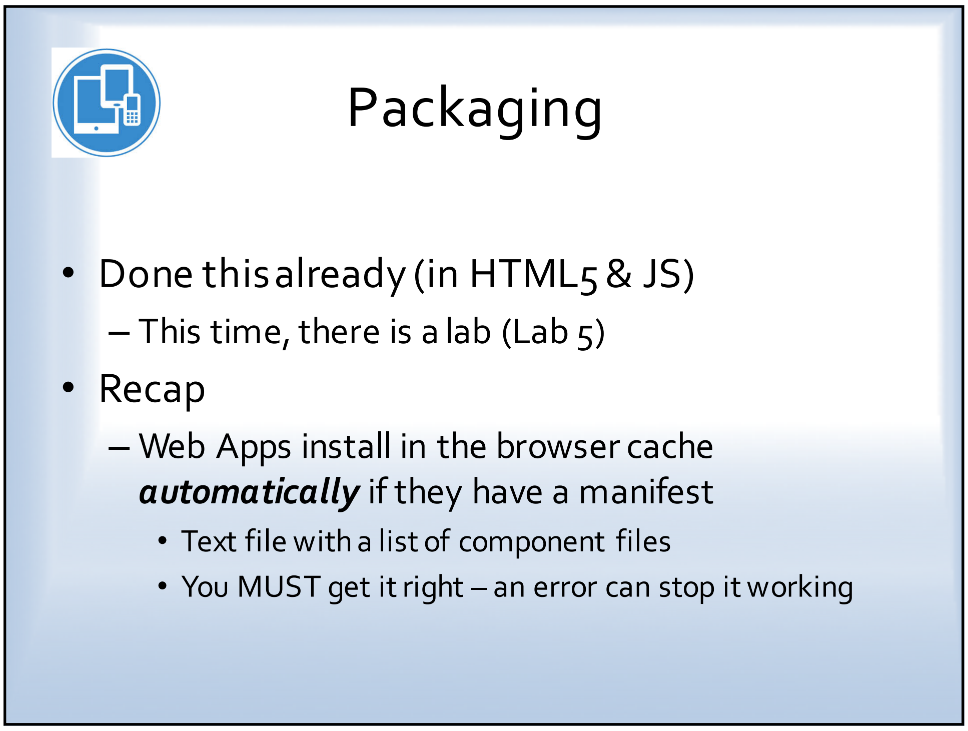
Your data-store name

```
{
  "firebase": "resplendent-heat-3736",
  "public": "app",
  "ignore": [
    "firebase.json",
    "**/.*",
    "**/node_modules/**"
  ]
}
```



Testing

- It is worth reviewing the TDD/BDD Testing procedures covered in HTML5
  - Jasmine as a test framework
  - Write tests before you write the features to test
  - Keep all tests for regression testing (after making changes)
- However, none of this will be enough
  - We need to test on real devices
    - On iOS, one set of tests per format (phone/tablet) will be enough
    - For Android, no-one can possibly test all platform variants
      - Go for the most popular recent ones
      - If possible, test on multiple devices
      - If money no object, try Cloud Testing - e.g. http://xamarin.com/test-cloud - EXPENSIVE



Packaging

- Done this already (in HTML5 & JS)
  - This time, there is a lab (Lab 5)
- Recap
  - Web Apps install in the browser cache ***automatically*** if they have a manifest
    - Text file with a list of component files
    - You MUST get it right – an error can stop it working

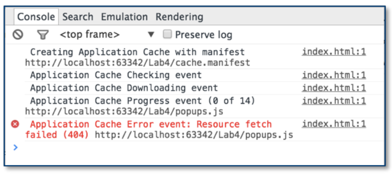
Manifest file

```
CACHE MANIFEST

# 30-04-2014 Version 0.2

NETWORK:
*

CACHE:
index.html
js/lab4.js
js/popups.js
js/maps.js
js/utils.js
http://code.jquery.com/…/jquery.mobile-1.4.0.css
http://code.jquery.com/jquery-1.10.2.min.js
http://code.jquery.com/…/jquery.mobile-1.4.0.js
images/favicon.ico
```

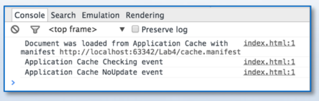
- This format is necessary
- Three key sections:
  - NETWORK:
    - URLS that must be online (* is everything not in the CACHE)
  - CACHE:
    - URLs to content that should be stored locally
  - FALLBACK:
    - URLs that are for NETWORK content, but have a version for offline use
  - # for comments
- THE RULE: if the manifest can be downloaded, it will be
  - Browser then follows directives
    - CACHE files from browser cache
    - NETWORK files from network (if possible)
  - IF A SINGLE BYTE HAS CHANGED, the app should be re-downloaded when possible

Manifest behaviour

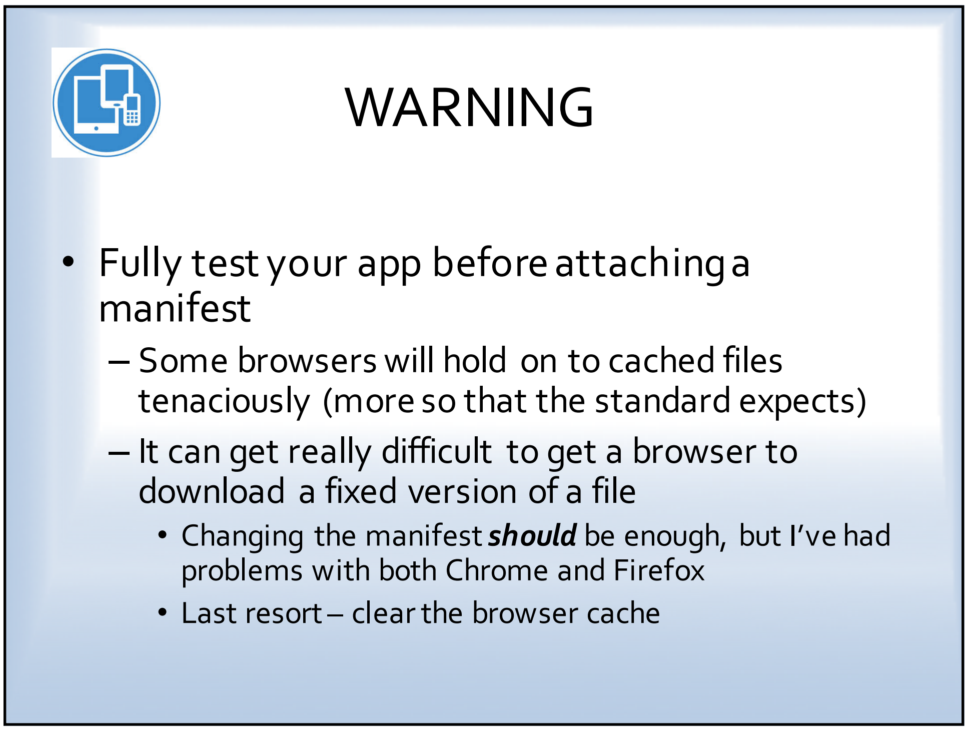
- When the manifest is out of step with the app files, there will be Resource fetch failed (404) errors visible in the console
- When the manifest truly reflects the app files, there will be a **NoUpdate event**
- First step in cache debugging is to open a console window and then refresh the app
  - Usually that is enough



```
Creating Application Cache with manifest                     index.html:1
http://localhost:63342/Lab4/cache.manifest
Application Cache Checking event                              index.html:1
Application Cache Downloading event                           index.html:1
Application Cache Progress event (0 of 14)                    index.html:1
http://localhost:63342/Lab4/popups.js
Application Cache Error event: Resource fetch                 index.html:1
failed (404) http://localhost:63342/Lab4/popups.js
>
```



```
Document was loaded from Application Cache with               index.html:1
manifest http://localhost:63342/Lab4/cache.manifest
Application Cache Checking event                              index.html:1
Application Cache NoUpdate event                              index.html:1
>
```



WARNING

- Fully test your app before attaching a manifest
  - Some browsers will hold on to cached files tenaciously (more so that the standard expects)
  - It can get really difficult to get a browser to download a fixed version of a file
    - Changing the manifest ***should*** be enough, but I've had problems with both Chrome and Firefox
    - Last resort – clear the browser cache