MOBILE WEB APIS

MOBILE WEB APIS

- URLs Beyond the Web
- Geolocation
- Device Orientation
- Capturing Pictures, Audio, Video
- Other APIs
- Not only mobile
- Summary

URLS BEYOND THE WEB

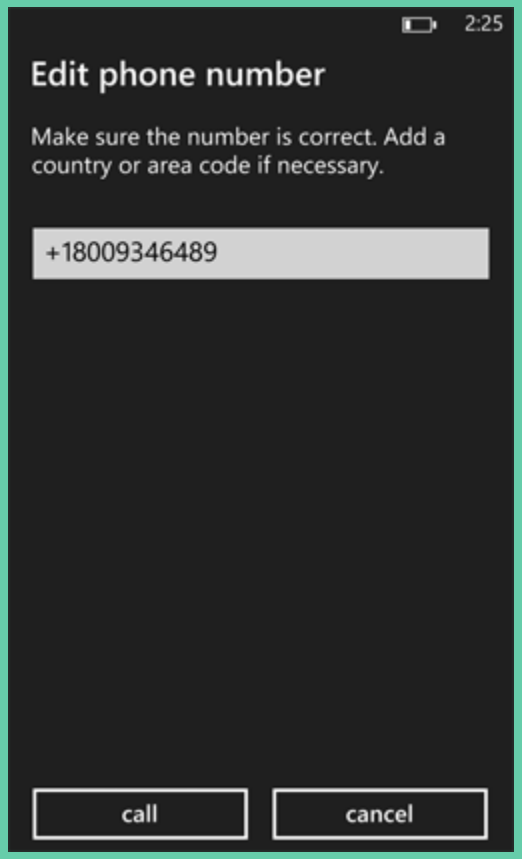
- Phone call and text message links
- Deep Linking into Apps

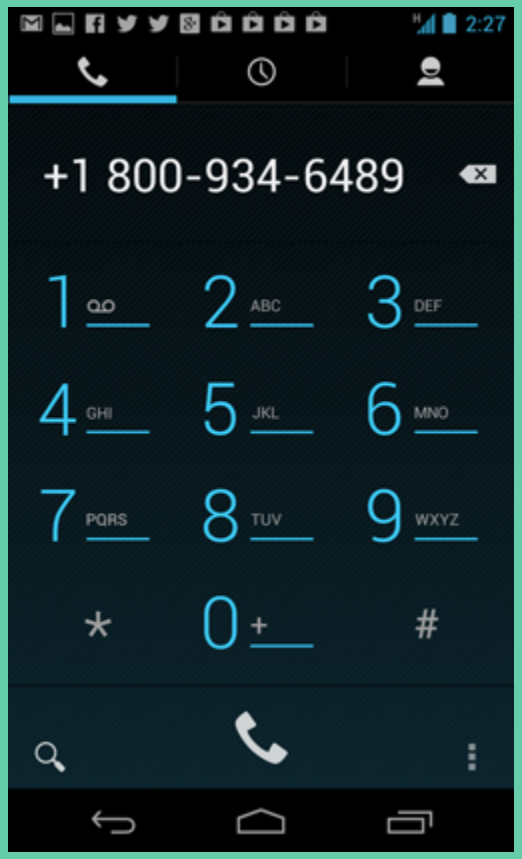
PHONE CALL AND TEXT MESSAGE LINKS

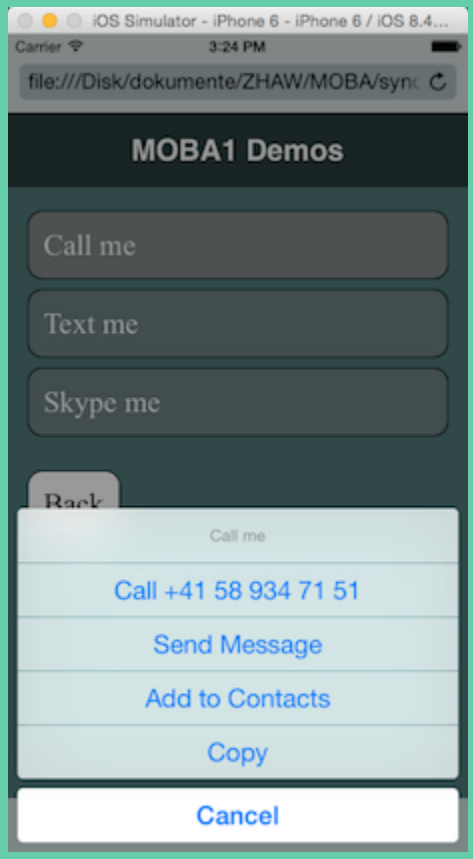
```
<a href="tel:+41589347151">
  Order Pizza Now!
</a>

<a href="sms:+41589347151?body=hello%20there">
  Text me!
</a>
```

PHONE CALL AND TEXT MESSAGE LINKS







PHONE CALL AND TEXT MESSAGE LINKS

URI scheme rather than a Web API

- The tel URI for Telephone Numbers: RFC3966
- URI Scheme for [...] Short Message Service (SMS): RFC5724

Wikipedia: Uniform Resource Identifier IANA: Uniform Resource Identifier (URI) Schemes

OTHER URI SCHEMES

- Some URI schemes are platform or app specific
- For example: Skype, FaceTime

```
skype:<username|phonenumber>[?[add|call|chat|sendfile|userinfo]]
facetime://<address>|<MSISDN>|<mobile number>
```

DEEP LINKING

- So far: usually it’s a web browser that responds to URLs
- Now: native apps can, too

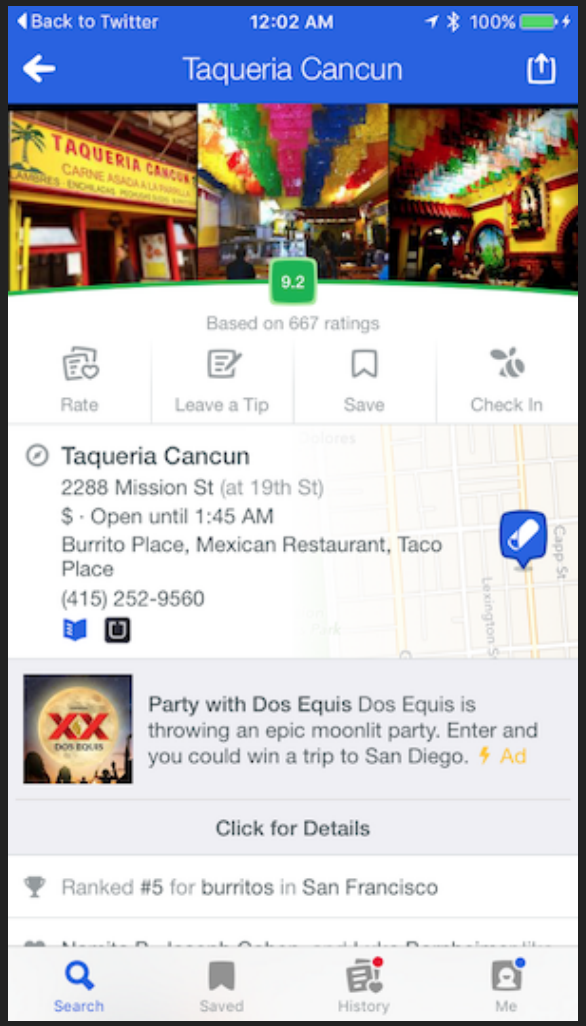
iOS: Support Universal Links
Google to offer improved app-to-app links for Android M

HOW DEEP LINKING WORKS IN IOS 9

> *Say one of your friends tweets a Foursquare link to her favorite restaurant. Before, when you tapped the link on Twitter for iOS, it simply launched the web view as it does for millions of other links—even if you had the Foursquare app installed. [...] Now when you tap that Foursquare link, iOS 9 intelligently routes you to that exact place inside of the Foursquare app bypassing Safari altogether.*

Source

HOW DEEP LINKING WORKS IN IOS 9



- Clicked on a link in the Twitter app
- Foursquare app opens with linked page
- Or: web browser if app isn’t there
- Link back to the Twitter app

HOW DEEP LINKING WORKS IN IOS 9

- Create file called *apple-app-site-association*
- Insert JSON data about the URLs that your app can handle
- Upload it to your HTTPS web server
- Prepare your app to handle universal links

↓ example ↓

JSON FILE WITH APP DATA

```
{
    "applinks": {
        "apps": [],
        "details": [
            {
                "appID": "9JA89QQLNQ.com.apple.wwdc",
                "paths": [ "/wwdc/news/", "/videos/wwdc/2015/*" ]
            },
            {
                "appID": "TeamID.BundleID2",
                "paths": [ "*" ]
            }
        ]
    }
}
```

DEEP LINKING

- Apps more and more as part of the Web?

EXAMPLES

http://www.jackivers.me/blog/2015/9/17/list-of-universal-link-ios-9-apps

RELATED TOPIC: SEARCH

- Extend search to cover app content
- Google Search now indexes iOS 9 apps...

GEOLOCATION

DETECT LOCATION

	ACCURACY	POSITIONING TIME	BATTERY LIFE
GPS	10m	2-10 minutes (only indoors)	5-6 hours on most phones
WiFi	50m (improves with density)	Almost instant (server connect and lookup)	No additional effect
Cell tower triangulation	100-1400m (based on density)	Almost instant (server connect and lookup)	Negligible
Single cell tower	500-2500m (based on density)	Almost instant (server connect and lookup)	Negligible
IP	Country: 99% City: 46% US, 53% International Zip: 0%	Almost instant (server connect and lookup)	Negligible

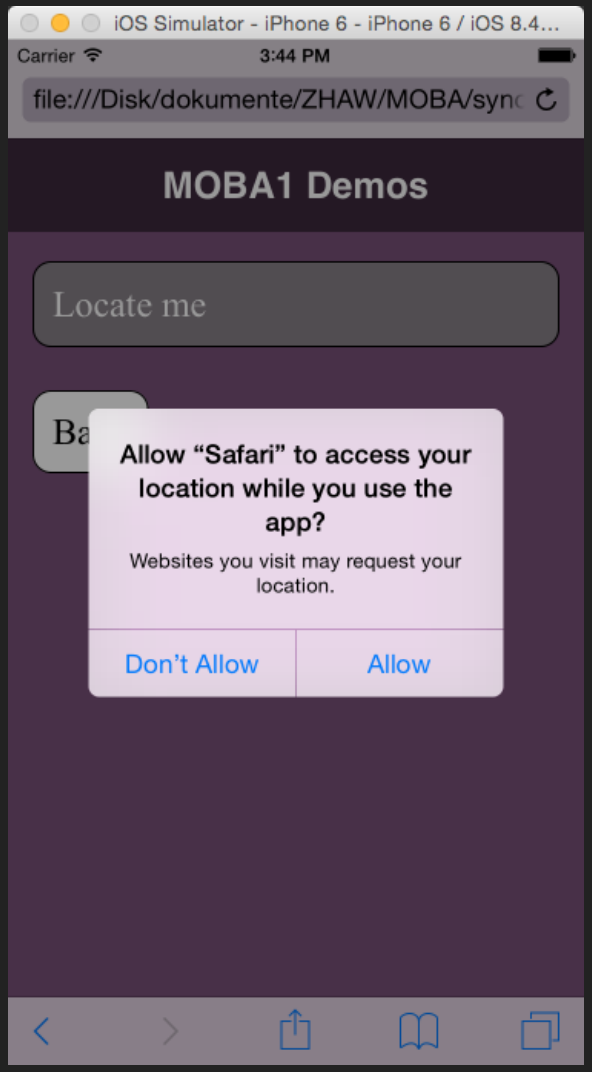
TABLE 3.1: An overview of the different ways a modern mobile device can detect your location. Smartphones make hybrid use of GPS, WiFi, and cell tower triangulation; laptops and desktops use WiFi, IP, and only rarely GPS.

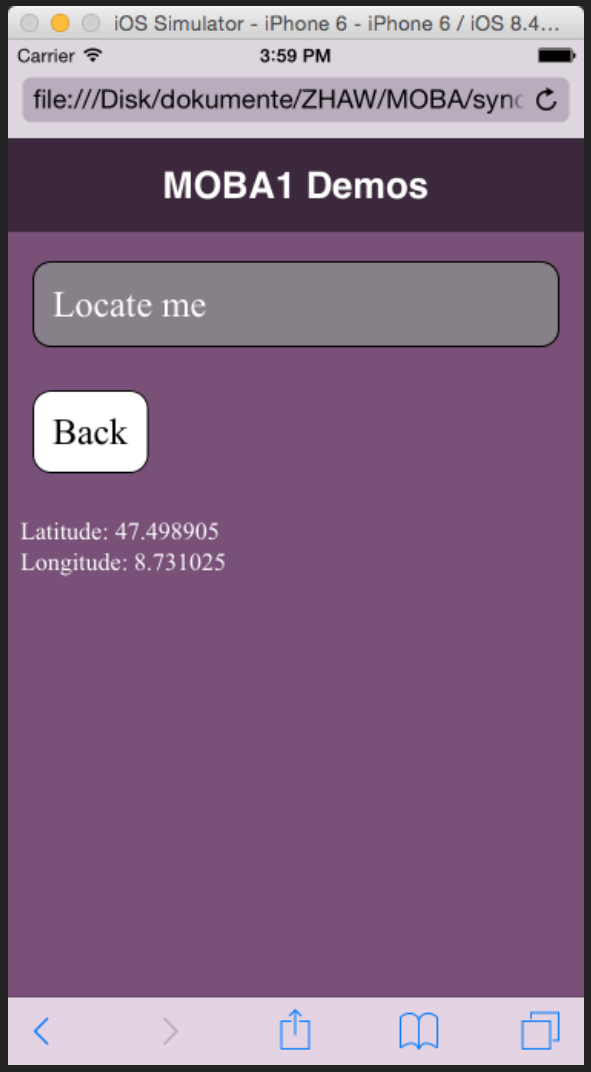
GEOLOCATION

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(success, fail);
}

function success(position) {
  alert('Latitude: '+ position.coords.latitude +
  ', Longitude: '+ position.coords.longitude);
}
```

ACCESS PERMISSION



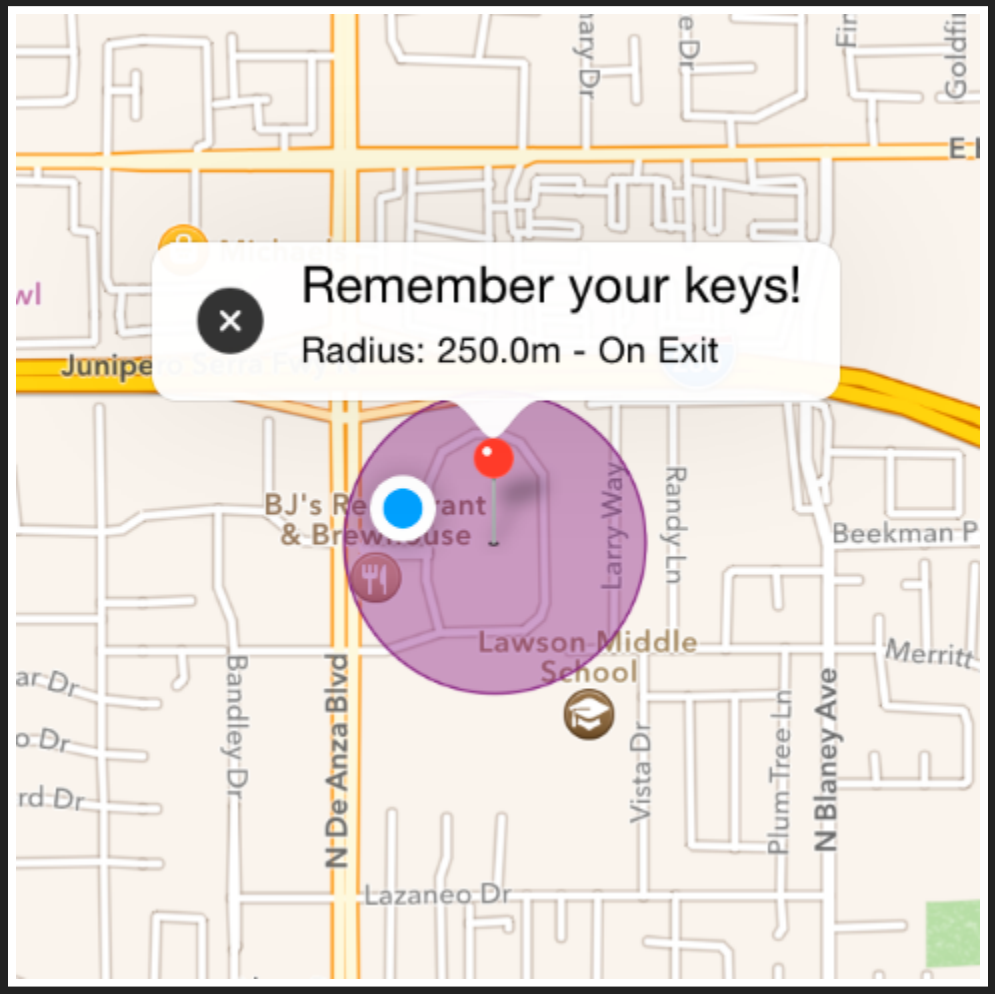


GEOFENCING

- Register for notifications when device enters defined regions
- Geofencing API
- Build on Service Workers API
- Currently Chrome only

↓ example ↓

GEOFENCING


Remember your keys!
Radius: 250.0m - On Exit
Lawson Middle School
N De Anza Blvd
N Blaney Ave
Larry Way
Randy Ln
Vista Dr
Plum Tree Ln
Bandley Dr
Lazaneo Dr
Beekman P
Merritt

DEVICE ORIENTATION

DEVICE ORIENTATION

Physical orientation and movement of the device

- Gyroscope
- Accelerometer
- Compass

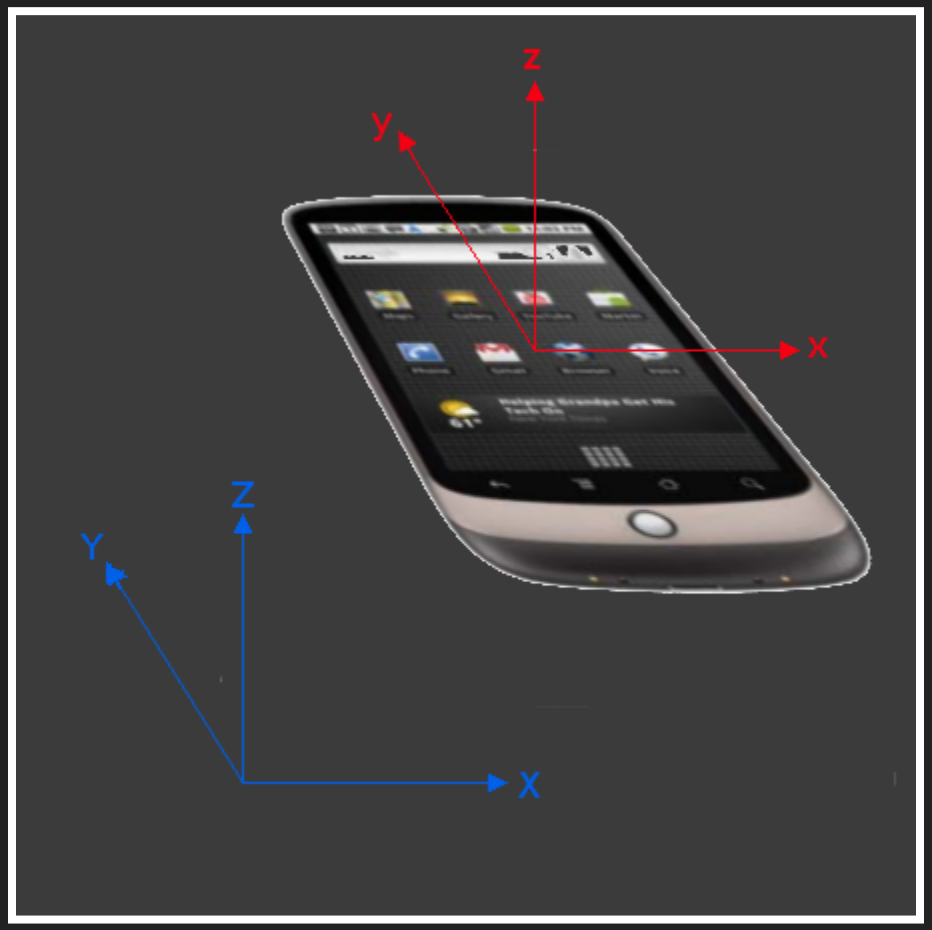
DEVICE ORIENTATION EVENTS

Events (fired on the window object)

- deviceorientation
- devicemotion
- compassneedscalibration

W3C: DeviceOrientation Event Specification

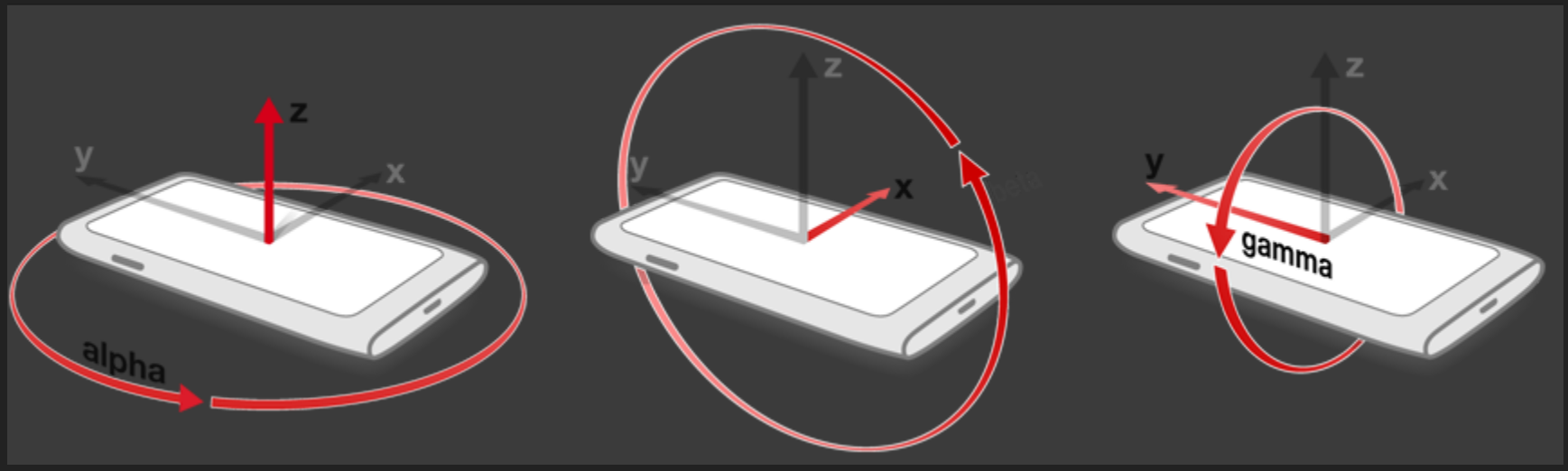
EVENT: DEVICEORIENTATION



EVENT: DEVICEORIENTATION

- Fired when the accelerometer detects an orientation change
- Event object is of type DeviceOrientationEvent
- It has four properties:
  - `alpha`: the angle around the z-axis (0..360)
  - `beta`: angle around the x-axis (-180..180)
  - `gamma`: angle around the y-axis (-90..90)
  - `absolute`: whether data is relative to the Earth's coordinate system

EVENT: DEVICEORIENTATION



EVENT: DEVICEORIENTATION

```
if (window.DeviceOrientationEvent) {
  window.addEventListener('deviceorientation', function (e) {
    a = Math.floor(e.alpha);
    b = Math.floor(e.beta);
    g = Math.floor(e.gamma);
    el.style.transform
      = 'rotateZ('+a+'deg) rotateX('+b+'deg) rotateY('+g+'deg)';
    ...
  }, true);
}
```

EVENT: DEVICEMOTION

- Fires when the device accelerates or decelerates
- Event object is of type DeviceMotionEvent
- It has four properties:
  - `acceleration`:
    relative to the Earth frame on the x, y, and z axes
  - `accelerationIncludingGravity`:
    takes Earth's gravity into account
  - `rotationRate`:
    in degrees per second (alpha, beta, and gamma properties)
  - `interval`:
    interval at which data is obtained in ms

EVENT: COMPASSNEEDSCALIBRATION

- Fired if compass needs calibration
- Calibrating will increase the accuracy of the deviceorientation data
- Should be used to inform the user
- Should also instruct the user how to callibrate the compass

DEVICE ORIENTATION EVENTS

- Partial support in mobile browsers
- Some known inconsistencies between browsers

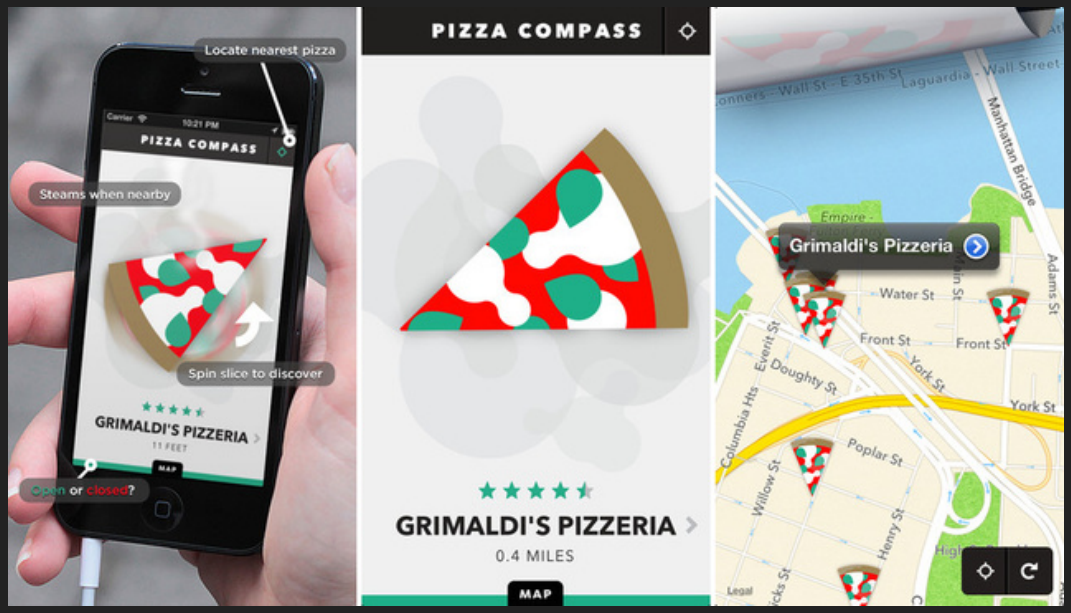
https://w3c.github.io/deviceorientation/spec-source-orientation.html

http://caniuse.com/#feat=deviceorientation

http://code.tutsplus.com/tutorials/an-introduction-to-the-device-orientation-api--cms-21067

http://www.html5rocks.com/en/tutorials/device/orientation/

EXAMPLE: PIZZA COMPASS



- Native app that could be implemented as a Web app:
  - Get location - navigator.geolocation
  - Use compass - deviceorientation event (use alpha)
  - Order a pizza - tel: URI scheme

CAPTURING AUDIO & VIDEO

CAPTURING AUDIO & VIDEO

- For many years: browser plugins necessary (Flash, Silverlight)
- Several variants of "Media Capture APIs" have evolved
- Things got messy over time...
- W3C: Device APIs Policy (DAP) Working Group

HTML MEDIA CAPTURE

- Using `<input type="file">`
- Added new values for the accept parameter

```
<!-- Current implementation (if supported) -->
<input type="file" accept="image/*" capture="camera">

<!-- Newly proposed specification -->
<input type="file" accept="image/*" capture>
```

HTML MEDIA CAPTURE

- W3C Candidate Recommendation 09 September 2014
- Moderate browser support
  (IE: not available, Safari: partial)
- No realtime effects
- Many thought HTML Media Capture was too limiting...

DEVICE ELEMENT

- Just a short episode...
- Was implemented by Opera
- Now obsolete

```
<device type="media" onchange="update(this.data)"></device>
<video autoplay></video>
```

```
function update(stream) {
  document.querySelector('video').src = stream.url;
}
```

WEBRTC

- WebRTC (Web Real Time Communications)
- Gateway into that set of APIs:
  `navigator.getUserMedia()`
- Provides access to multimedia streams (video, audio)
- Just this one method
- Parameters:
  - Object of constraints
  - Success callback
  - Failure callback

GETUSERMEDIA

Feature detection:

```
function hasGetUserMedia() {
  return !!(navigator.getUserMedia || navigator.webkitGetUserMedia ||
            navigator.mozGetUserMedia || navigator.msGetUserMedia);
}

if (hasGetUserMedia()) {
  // Good to go!
} else {
  alert('getUserMedia() is not supported in your browser');
}
```

GETUSERMEDIA

To request both audio and video, pass the following object:

```
{
  video: true,
  audio: true
}
```

Alternatively, pass a constraints object:

```
{
  video: {
    mandatory: { minWidth: 1280, minHeight: 720, minFrameRate: 30 },
    optional: [ { minFrameRate: 60 } ]
  },
  audio: true
}
```

GETUSERMEDIA: EXAMPLE

```
<video autoplay></video>
```

```
var specs = {video: true, audio: true};

// Not showing vendor prefixes
navigator.getUserMedia(specs, function(localMediaStream) {
  var video = document.querySelector('video');
  video.src = window.URL.createObjectURL(localMediaStream);
  video.onloadedmetadata = function(e) {
    // Ready to go. Do some stuff
  };
}, errorCallback);
```

GETUSERMEDIA: APIS

- The API shown in the examples is already outdated
- New API based on MediaStream objects and MediaDevices
- W3C: Media Capture and Streams

Articles on getUserMedia (outdated API):

- An Introduction to the getUserMedia API
- Capturing Audio & Video in HTML5

GETUSERMEDIA: APIS

```
// Current specification
// Returns a Promise<MediaStream> object
navigator.mediaDevices.getUserMedia(constraints);

// Former specification
navigator.getUserMedia(constraints, successCallback, errorCallback);
```

GETUSERMEDIA: APIS

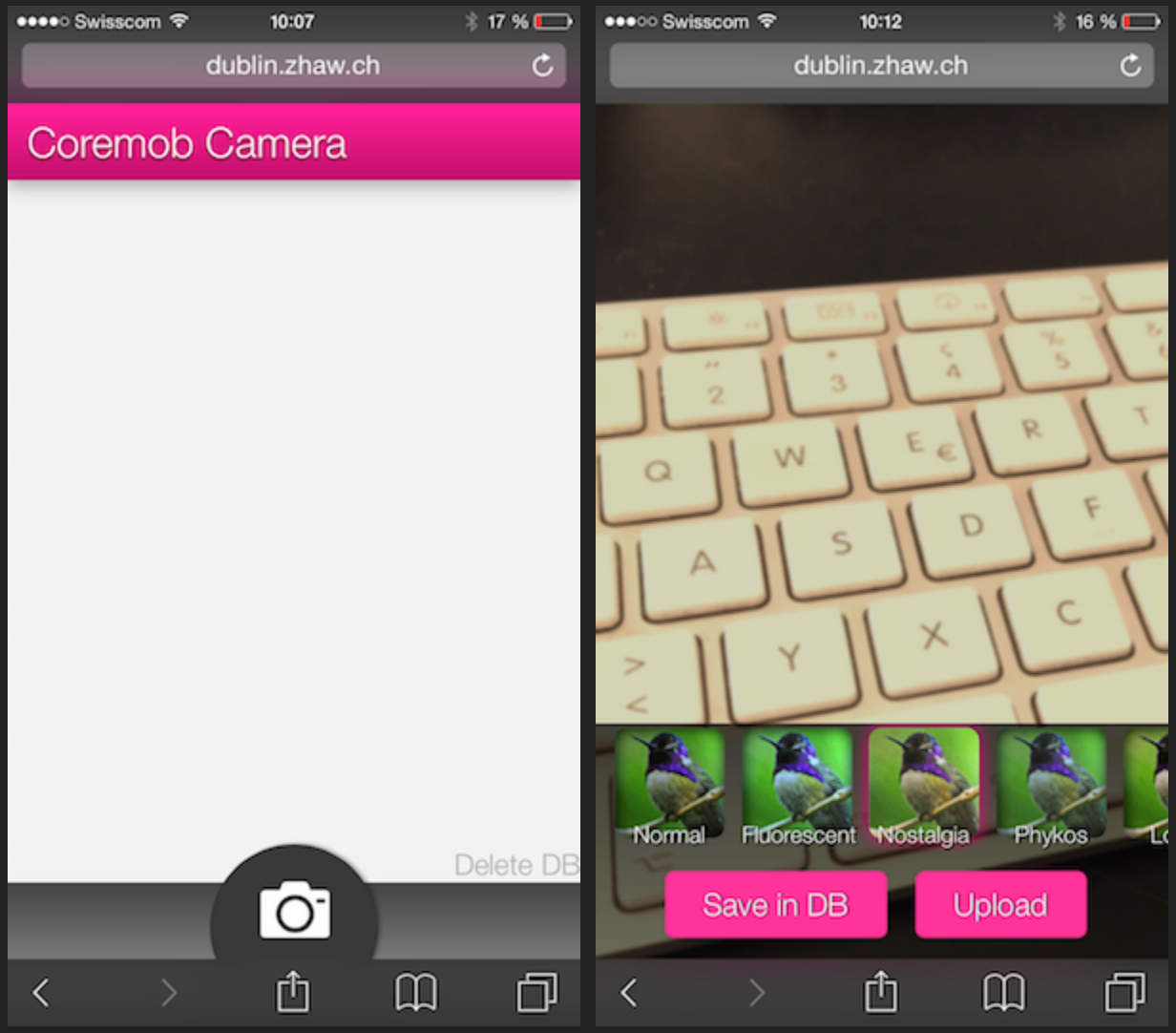
- Technology’s specification currently not stable
- MDN: "This is an experimental technology"
- MDN article: MediaDevices.getUserMedia()
- Moderate browser support: caniuse.com/#feat=stream

HTML MEDIA CAPTURE: EXAMPLE

- Back to HTML media capture
- Works in many mobile browsers
- Sample camera app: github.com/coremob/camera

↓ more ↓

HTML MEDIA CAPTURE: EXAMPLE



HTML MEDIA CAPTURE: EXAMPLE

- Take a picture with the native camera via HTML Media Capture
- The camera returns the pictures as a File object from FileReader()
- drawImage() to draw the image object in canvas
- getImageData() to obtain an ImageData object containing a copy of the pixel data for a context, then tweak the pixels (filter effect)
- canvas.toBlob() to store the blob locally with IndexedDB*
- Upload the final photo with XHR2/CORS

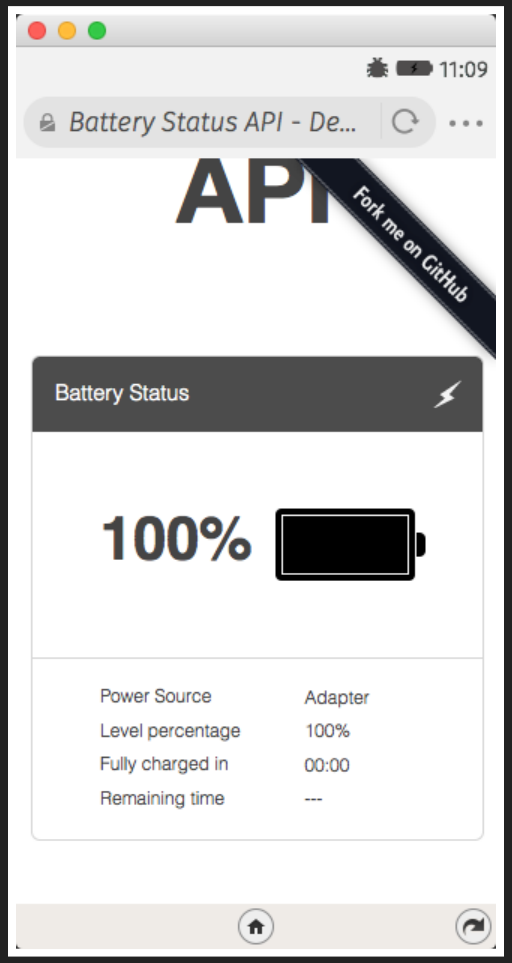
Demo Video

OTHER APIS

OTHER APIS

- Battery Status API
- Vibration API
- Ambient Light API
- Proximity API

BATTERY STATUS API



BATTERY STATUS API

```
var battery = navigator.battery || navigator.webkitBattery;

battery.addEventListener('chargingchange', updateStatus);
battery.addEventListener('levelchange', updateStatus);

function updateStatus() {
  alert('Battery status: ' + battery.level * 100 + ' %');
  if (battery.charging) {
    alert('Battery is charging...');
  }
}
```

BATTERY STATUS API

- Currently not supported by iOS/Safari and IE/Edge
- caniuse.com/#feat=battery-status
- www.w3.org/TR/battery-status/

JSFiddle

VIBRATION API

```
var vibrate = navigator.vibrate || navigator.mozVibrate;

vibrate(1000); // vibrate for 1sec

vibrate([1000, 500, 2000]);
// vibrates for 1sec, still for 0.5 seconds,
// and vibrates again for 2sec
```

VIBRATION API

- Method to access the vibration mechanism of the device
- Currently not supported by iOS/Safari and IE/Edge
- caniuse.com/#feat=vibration
- www.w3.org/TR/vibration/

AMBIENT LIGHT API

- Provides information about the ambient light level

```
window.addEventListener('devicelight', function(e) {
   alert(e.value); // value in double
});
window.addEventListener('lightlevel', function(e) {
   alert(e.value); // value in string: dim|normal|bright
});
```

AMBIENT LIGHT API

- Weak browser support, partial support only in Firefox
- caniuse.com/#feat=ambient-light
- www.w3.org/TR/ambient-light/

JSFiddle

PROXIMITY API

- Provides information about the distance between a device and an object
- Event `deviceproximity`
  - Returns three numbers: proximity, min and max sensing distance
- Event `userproximity`
  - Returns true if device is near an object

```
window.addEventListener('deviceproximity', function(e) {
   alert(e.value);
});
```

PROXIMITY API

- Weak browser support, only supported by Firefox
- caniuse.com/#feat=proximity
- www.w3.org/TR/proximity/

NOT ONLY MOBILE

NOT ONLY MOBILE

- Application Cache
- Service Workers
- Web Workers
- Web Storage
- IndexedDB
- Multimedia
- Page Visibility API
- Fullscreen API
- Other APIs

APPLICATION CACHE

- Web applications that work offline
- Browser keeps a copy of the necessary files
- Has some issues
- Good browser support

https://html.spec.whatwg.org/multipage/browsers.html#offline
http://caniuse.com/#feat=offline-apps

SERVICE WORKERS

- Persistent background processing
- Including hooks for web applications while offline
- Currently poor browser support (only Chrome)

https://slightlyoff.github.io/ServiceWorker/spec/service_worker
http://caniuse.com/#feat=serviceworkers

WEB WORKERS

- Means for web content to run scripts in background threads
- Worker thread can perform tasks without interfering with the UI
- A *shared worker* is accessible by multiple scripts
- Web Workers: good browser support (IE since 10)
- Shared workers: only Firefox and Chrome

WEB STORAGE

- Storing data locally, name/value pairs
- Good browser support

http://www.w3.org/TR/webstorage/
http://caniuse.com/#feat=namevalue-storage

INDEXEDDB

- Storing data client-side
- Allows indexed database queries
- Supported in newer browsers
  - Missing features in IE/Edge
  - Buggy behaviour in Safari

http://www.w3.org/TR/IndexedDB/
http://caniuse.com/#feat=indexeddb

MULTIMEDIA

- Video & Audio Players (elements `video` and `audio`)
- Canvas graphics (element `canvas`)
- Scalable Vector Graphics (element `svg`)
- Good browser support

PAGE VISIBILITY API

- Determining whether a document is visible on the display
- Event: `visibilitychange`
  (fired on the document object)
- Attribute: `document.hidden`
  (true or false)
- Attribute: `document.visibilityState`
  (hidden, visible, preview, prerender)

↓ more ↓

PAGE VISIBILITY API

- W3C: Page Visibility
- Good browser support
- caniuse.com/#feat=pagevisibility
- MDN: Using the Page Visibility API
- Demo: Video stops when page is not visible
- Another demo

FULLSCREEN API

- Allows content to take up the entire screen
- For example a video or canvas element
- WHATWG: Fullscreen API
- Good support on desktop browsers (with vendor prefixes)
- Weak support on mobile browsers
- http://caniuse.com/#feat=fullscreen

↓ usage ↓

FULLSCREEN API

```
// test if available
if (document.fullscreenEnabled) { ... }

// make an individual element full-screen
document.getElementById("myimage").requestFullscreen();

// returns the current element which is full-screen or null
var elem = document.fullscreenElement;
// listen to fullscreen changes
document.addEventListener("fullscreenchange", function() { ... });
// fire event when fullscreen failed
document.addEventListener("fullscreenerror", function() { ... });
```

- How to Use the HTML5 Full-Screen API (Again)
- Demo

OTHER APIS

- The Network Information API
  - Access connection information of the device
  - Discontinued
- Navigation Timing API
  - Access timing information related to navigation and elements
  - caniuse.com/#feat=nav-timing

↓ usage ↓

OTHER APIS

- Clipboard API
  - Provide copy, cut and paste events
  - Also provide access to the OS clipboard
  - caniuse.com/#feat=clipboard
- Gamepad API
  - Support input from USB gamepad controllers
  - caniuse.com/#feat=gamepad

SUMMARY

SUMMARY

- A lot of APIs available
- Several of these apply to mobile development
- Various degrees of maturity
- Browser support varies considerably
- Many APIs are still not stable

Device APIs Working Group

HYBRID APPS

- Embed web app in native application
- More consistent APIs across platforms
- Provide features not available on some platforms

Example

Plugin APIs of Apache Cordova

ADVICES

- Don’t use browser detection, use feature detection
- Consult comparison tables
  - mobilehtml5.org
  - caniuse.com
- Test on as many devices as possible

Tests on mobilehtml5.org

SOURCES

SOURCES

- HTML5 compatibility on mobile browsers with testing on real devices
  mobilehtml5.org
- Can I use, browser support tables
  caniuse.com
- Google: Device Access & Integration
  developers.google.com/web/fundamentals/device-access/
- HTML5 Mobile: Mobile Loves JS, Tomomi Imura
  girliemac.com/presentation-slides/html5-mobile-approach/deviceAPIs.html