Explain the concept of: Hybrid Mobile App Developmenthttp://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/

Like the websites on the internet, hybrid mobile apps are built with a combination of web technologies like HTML, CSS, and JavaScript. The key difference is that hybrid apps are hosted inside a native application that utilizes a mobile platform’s [WebView](http://developer.telerik.com/featured/what-is-a-webview/). (You can think of the WebView as a chromeless browser window that’s typically configured to run fullscreen.) This enables them to access device capabilities such as the accelerometer, camera, contacts, and more. These are capabilities that are often restricted to access from inside mobile browsers. Furthermore, hybrid mobile apps can include native UI elements in situations where necessary, as evidenced by [Basecamp’s approach towards hybrid mobile app development](https://signalvnoise.com/posts/3743-hybrid-sweet-spot-native-navigation-web-content).

## Explain the Pros & Cons of using Hybrid Mobile App Development compared to Native App Development.

**Hybrid**

Pros:

* If publishing to different platforms, you won’t need to rewrite your application.
* Hybrid mobile apps don’t have that “mobile web” browser look because they can include native hardware features.
* The content of a hybrid app is portable and just requires a native harness to run it.
* Since software like [Ionic](http://ionicframework.com/) or [React](https://facebook.github.io/react/) provides frameworks to make a webpage act like a native application, they can be distributed on the App Store.
* Developers have the option to package the app locally or through a server, which provides access both online and offline.

Cons:

* Since hybrid applications are relatively new in the mobile development space, automatic generation may not work on all devices, which can get especially complicated when trying to accommodate to different Android phones.
* Since hybrid app development is still new there is not as much support for any troubleshooting for unprecedented problems that may occur.
* Several vendors have started offering build-platforms for hybrid frameworks, simplifying the build knowledge that was previously required for multi-platform. Just be prepared to pay for it.
* If the App Store is able to recognize that your application is not truly native, it may be denied from the App Store.
* If your app can’t be published on the App Store, then that would reduce your monetization and distribution potential since purchase price or in-app purchases are native features.
* Since most hybrid apps are written in HTML5, they rely on the system’s browser to support the wrapper for running the application, which presents a supplantable resource that external parties could exploit beyond the normal security afforded to a native application. This heavily hitches behavior to a system component that could be replaced on customized/rooted devices, creating very difficult situations to isolate and support for errors or exploits.
* When a new iOS version is released, hybrid developers would have to rely on a third party before they are able to design hybrid applications on the new OS.
* Lack of the pure UI assets of iOS or Android may result in a slower performance of the app in general. It may not look like a mobile website, but it may feel like it at certain points.
* Phonegap, Cordova and others generate native by-products, meaning you still have to support and manage the individual packages in the app-stores. Keeping versions in sync across platforms while addressing individual bugs can be more difficult than a pure native approach.

**Native**

* Due to all of the app’s elements being included in a single native package, native applications tend to have fast graphics with fluid animations built in.
* Native applications can access exclusive native APIs in the phone’s operating system such as push notifications, camera, and in-app purchases, which would otherwise be prohibited, or provided in a cumbersome manner on a mobile web application.
* If you’re developing a native application for iPhone, they have many resources, development tools, and reading material to help you out.
* Apple definitely wants to push their brand whenever possible, so they have provided UI components from their UI libraries to make development a little less painful.

**Cons:**

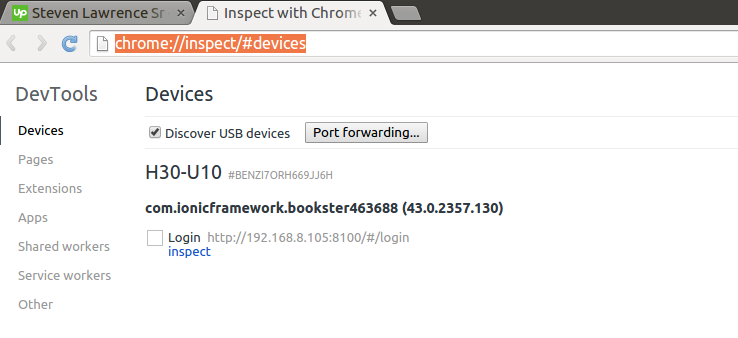
* If you intend to also publish your app to a different app store, your application will need to be rewritten in order to be a native app on another mobile OS. This usually delays features for the next platform in development. For example Snapchat implemented features such as reverse video a lot later on Android than on the Snapchat app for iOS. This is not to do with any one platform being better than another. However, Snapchat was released on iOS first, so it usually receives new features first.
* It is a time consuming process to create a native app for both iOS and Android, so it will cost you.
* Native platforms define their own rules and frameworks and inherit little from other disciplines, requiring more investment. For many businesses with existing development (ie. web, desktop app, etc) personnel, they would be unable to utilize these existing resources towards a native mobile application.
* Native applications typically require you to define phones and tablets separately, or define individual layouts. While this step is available and repeated on the web with CSS media queries, it’s usually a single layout and multiple stylesheets. The effort for native is non-transferable between iOS, Android and others since each operating system is locked into proprietary tooling.

## Explain about the "building blocks" involved in an ionic Hybrid Application

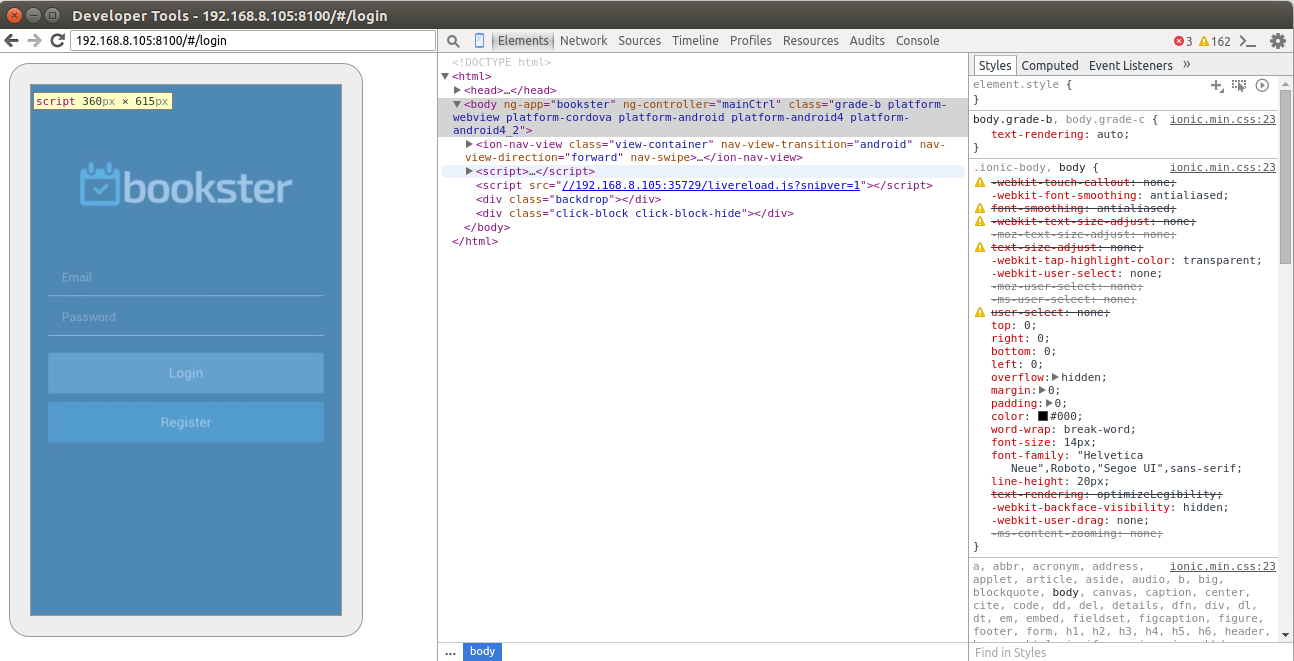
* Cordova / PhoneGap provides access to a device’s capabilities.
* Then Ionic provides a framework, which makes a webpage act like a native application.

## Explain and demonstrate ways to debug Hybrid Mobile Applications Running on a real device

The second method will not only allow you to see console logs but also inspect elements and network monitor, the same way you do with websites. Use chrome dev tools, write chrome://inspect/#devices in chrome address bar and hit enter. It will open the following page in your browser, showing all devices and hybrid applications running on it.



Now clicking on inspect will open the same inspection window like web page. See the following image for visual understanding.



## Explain how and why, it is possible for a Hybrid Application to access native phone devices like location, calendar etc.

Because a hybrid app utilizes Cordova/PhoneGap which provides access to a device’s capabilities.

## Explain when and why CORS is a problem for Hybrid Mobile Applications

### Running on a device

What happens when you run ionic run?

* Your files for the app are copied to the device (or simulator).
* The app runs, thus firing a browser on the phone/simulator to run the files that were copied over, something like: file://some/path/www/index.html.

Your origin will not exist, since you are running off a file:// URI; therefore, any request outwards will not require a CORS request.

* The easiest way to handle the CORS problem is to ultimately ask your API provider to allow all hosts. However, this isn’t always an option.

## *Explain*, using an example *you have implemented*, the "fundamentals" of an ionic application.

* Javascript
* CSS
* HTML5
* Package.json

Example:

|  |
| --- |
| angular.module('starter', ['ionic', 'ngCordova']) |
|  |

|  |  |
| --- | --- |
|  |  |
|  |  |

.run(function($ionicPlatform) {

## Explain *using an example* how your Hybrid Application communicates with a backend and how CORS problems were solved (if any)

Through an ajax REST API.

Example:

app.use(cors()); //This makes sure that our API allows all origins.

Alternatively you could use the code underneath which would mean a hybrid app wouldn’t work:

var corsOptions = {

origin: 'http://example.com',

optionsSuccessStatus: 200 // some legacy browsers (IE11, various SmartTVs) choke on 204

};

## Explain, with focus on location, technologies related to locations used on:

### Your app (client side)

* + $cordovaGeolocation.getLocation();

## Your backend

* + Ajax REST API
  + What our backend response looks like (geoJson):
    - {
    - "type": "Feature",
    - "geometry": {
    - "type": "Point",
    - "coordinates": [125.6, 10.1]
    - },
    - "properties": {
    - "name": "Dinagat Islands"
    - }
    - }