فهرست

[برنامه‌نویسی موبایل 3](#_Toc463020533)

[توسعه نرم‌افزارهای چند سکویی 3](#_Toc463020534)

[چهارچوب کوردووا بنیاد آپاچی 5](#_Toc463020535)

[تاریخچه 5](#_Toc463020536)

[نحوه عملکرد 6](#_Toc463020537)

[نقشه گوگل 6](#_Toc463020538)

[کوردووا و نقشه گوگل 7](#_Toc463020539)

[نحوه عملکرد افزونه 8](#_Toc463020540)

[پیاده‌سازی برنامه بکمک کوردووا 8](#_Toc463020541)

[نصب خط فرمان کوردووا (فون‌گپ) 8](#_Toc463020542)

[ایجاد پروژه 9](#_Toc463020543)

[افزودن سکوها 10](#_Toc463020544)

[بررسی و نصب پیش‌نیازها برای ایجاد و اجرا 10](#_Toc463020545)

[پیش‌نیازهای ایجاد برنامه اندروید 10](#_Toc463020546)

[نصب پیش نیازها 11](#_Toc463020547)

[تعیین متغییرهای محیطی 12](#_Toc463020548)

[بررسی پیش‌نیازها 12](#_Toc463020549)

[ساخت برنامه 13](#_Toc463020550)

[آزمون و بررسی خروجی برنامه 13](#_Toc463020551)

[مدیریت افزونه‌ها 13](#_Toc463020552)

[بروز رسانی کوردووا و سکوها 14](#_Toc463020553)

[کتابخانه‌ها و چهارچوبها 15](#_Toc463020554)

[انگولار جی‌اس 15](#_Toc463020555)

[اونسن یو‌آی 15](#_Toc463020556)

[در مورد Bower 17](#_Toc463020557)

[نصب و بروز رسانی bower و بسته‌های آن 17](#_Toc463020558)

[مقدار دهی اولیه bower 17](#_Toc463020559)

[کدها و توضیحات پروژه 20](#_Toc463020560)

[ساختار برنامه 20](#_Toc463020561)

[فایل شاخص 21](#_Toc463020562)

[کدهای جاوااسکریپت 21](#_Toc463020563)

[ماژولها 21](#_Toc463020564)

[کنترلرها 22](#_Toc463020565)

[سرویسها 23](#_Toc463020566)

[منابع 24](#_Toc463020567)

نرم‌افزار موبایل موقعیت یاب MyMap

برنامه‌نویسی موبایل

گوشی تلفن هوشمند[[1]](#footnote-1) به گوشی قابل حملی اطلاق می‌شود که مجهز به سیستم عامل پیشرفته‌ای است که ویژگی‌های رایانه‌های شخصی را با سایر امکاناتی کاربردی که در حین جابجایی و حرکت می‌تواند مفید باشد را تلفیق کرده است. گوشی‌های هوشمند که عموما در قطع جیبی ساخته می‌شوند، معمولا امکانات گوشی‌های همراه از قبیل دریافت و برقراری تماس و پیام متنی را با ویژگی‌های سایر وسایل قابل حمل دیجیتال از قبیل دستیار شخصی دیجیتال[[2]](#footnote-2)، ضبط و پخش صوت، موسیقی و فیلم، لوازم ناوبری با جی‌پی‌اس[[3]](#footnote-3) و ... تلفیق می‌کنند. علاوه بر آن امکان اتصال به اینترنت و یا اجرای نرم‌افزارهای مختلف در اغلب آنها فراهم شده است.

اولین گوشی هوشمند در سال ۱۹۹۹ توسط یک شرکت ژاپنی به نام ان‌تی‌تی دوکومو[[4]](#footnote-4) عرضه شد و پس از آن در قرن ۲۱ به سرعت استفاده از این گوشی‌ها گسترش یافت. از سال ۲۰۱۲ بیشتر گوشی‌ها مجهز به شبکه پرسرعت نسل چهارم[[5]](#footnote-5)، حسگر حرکت[[6]](#footnote-6) و امکانات پرداخت بوسیله موبایل شدند و در سه ماهه سوم این سال شمار گوشی‌های هوشمند به یک میلیارد رسید و همینطور در اوايل سال ۲۰۱۳ میلادی فروش گوشی‌های هوشمند از گوشی‌های معمول پیشی گرفت.

توسعه نرم‌افزارهای چند سکویی

توسعه نرم‌افزارهای چند سکویی موضوعی نیست که با ظهور گوشی‌های هوشمند معرفی شده باشد بلکه پیش از این نیز موضوع در موارد دیگر مطرح بوده که مهمترین مورد قابل ذکر تعریف نرم‌افزار برای سیستم‌های عامل رایانه شخصی است که برخی از توسعه دهندگان اقدام به توسعه نرم‌افزار مستقل برای سیستمهای عامل متفاوت می‌نمودند و برخی دیگر به کمک ماشینهای مجازی[[7]](#footnote-7) جاوا و مانند آن یک نرم‌افزار برای چندین سیستم عامل تهیه و ارائه می‌نمودند. لذا مطرح شدن دوباره این بحث در توسعه نرم‌افزار گوشی‌های هوشمند به دلیل فراگیر شدن استفاده از آنها در مدت زمان کم و همچنین معرفی سیستم عاملهای هوشمند متنوع باعث شد که موضوع با جدیت بیشتری مطرح شده و راه‌ حلهای جدیدی مطرح شود.

به عنوان مثال در سالهای قبل تفاوت در سیستمهای عامل عمدتاً محدود به سه سیستم عامل می‌شد: توزیعهای مبتنی بر لینوکس لینوس تروالدز، ویندوز مایکروسافت و سیستم عامل شرکت اپل؛ ولی در مورد سیستم‌عاملهای جدید، شرکت اپل سیستم عامل iOS و مایکروسافت سری جدیدی از سیستم عامل Windows Phone را تعریف کرد که با ادغام آن در سیستم عامل خود از یک سیستم عامل برای گوشی‌ها، تبلتها، لپ‌تاپها و رایانه‌های شخصی استفاده می‌کند. در این میان، بر خلاف دو سیستم قبلی که بدلیل محدود بودن امکان تنوع توسط صاحبان امتیاز آنها محدود شده بود، در مورد لینوکس همانطور که توزیعهای متنوعی با هسته لینوکس برای سیستمهای رومیزی[[8]](#footnote-8) معرفی شده بود در مورد لوازم هوشمند نیز سیستم عاملهای متفاوتی مطرح شد که از آن جمله می‌توان به اندروید[[9]](#footnote-9) ، سیستم‌عامل فایرفاکس[[10]](#footnote-10) ، اوبونتو لمسی[[11]](#footnote-11) و تایزن[[12]](#footnote-12) اشاره کرد. علاوه بر موارد بالا شرکتهایی نظیر بلک‌بری[[13]](#footnote-13) نیز وجود دارند که ساز و کار و سیستم عامل مربوط به خود را دارند.

همانطور که گفته شد، گوشی‌های هوشمند،‌ این امکان را فراهم می‌آورند تا نرم‌افزارهای مختلف بر روی آنها قابل اجرا باشد و شرکتهای توسعه دهنده سیستم عامل تلفنهای هوشمند به کمک کیت‌های توسعه نرم‌افزار، این امکان را برای سایر توسعه دهندگان فراهم آوردند تا نرم‌افزارهای خود را برای سیستم عامل آنها ارائه دهند. در ابتدای کار تنوع لوازم هوشمند محدود بود ولی با توسعه این صنعت و با افزوده شدن به شمار سیستم‌های و همچنین با تعریف مفهوم اینترنت اشیاء[[14]](#footnote-14) دستگاههای هوشمند دیگر محدود به گوشی و رایانه شخصی نمی‌شد بلکه در اکثر دستگاه‌های دیجیتال قابل استفاده بودند. همین امر باعث شد تا در توسعه نرم‌افزار با طیف وسیعی پردازنده‌ها با توان عملیاتی متفاوت و همچنین صفحات نمایش با اندازه‌های و اشکال مختلف (از یک صفحه ۱ اینچی در ساعت هوشمند تا یک تلویزیون هوشمند با صفحه نمایشی بزرگتر از ۱۹ اینچ به شکل ساده، خمیده، مربعی، گرد و ...) روبرو شویم.

پوشش مربوط به تفاوتهای سخت‌افزاری در هر سیستم عامل به کمک محیط توسعه یک‌پارچه‌ای[[15]](#footnote-15) که توسط توسعه دهنده سیستم عامل یا گروه دیگری ارائه می‌شود تا حدود زیادی پشتیبانی می‌شد ولی ایراد کار اینجا بود که توسعه دهنده نرم‌افزار باید فرآیند توسعه را به ازای هر سیستم‌عامل جدا‌گانه انجام می‌داد و برای هرکدام مسیر توسعه[[16]](#footnote-16)، ردگیری موضوع[[17]](#footnote-17)، رفع ایراد[[18]](#footnote-18) و ... مستقل (در تمامی و یا در برخی موارد) تعریف می‌کرد.

بنابراین در مسیر توسعه نرم‌افزار نیاز به محیطهای توسعه‌ای که خروجی مناسب برای سیستمهای مختلف ایجاد نماید به شدت محسوس بوده و به این منظور رویکردهای مختلفی برای توسعه پیشرو[[19]](#footnote-19) تعریف و مورد استفاده قرار گرفته است. ابزارهای توسعه پیشرو بر واسط کاربری[[20]](#footnote-20) و تجربه کاربری[[21]](#footnote-21) تمرکز کرده و توانایی‌های زیر را فراهم نموده‌اند:

* ابزارهای توسعه واسط کاربری.
* کیتهای توسعه نرم‌افزار به منظور دسترسی به قابلیتهای دستگاه.
* سازگاری و پشتیبانی از چند سکویی.

چهارچوب کوردووا بنیاد آپاچی

تاریخچه

کوردووا[[22]](#footnote-22) (فون‌گپ[[23]](#footnote-23)) یکی از چهارچوب[[24]](#footnote-24)های محبوب برای توسعه نرم‌افزار موبایلی است که توسط شرکت نیتوبی[[25]](#footnote-25) خلق گردید. در سال ۲۰۱۱ شرکت ادوبی[[26]](#footnote-26) با خرید نیتوبی این محصول را با نام تجاری فون‌گپ به بازار عرضه کرد و نسخه‌ای متن‌باز[[27]](#footnote-27) از آن توسط بنیاد آپاچی تحت عنوان کوردووا ارائه گردید. این چهارچوب به منظور ایجاد نرم‌افزار برای چند سکویی، به جای وابستگی به واسطهای برنامه نویسی تعریف شده برای آن، تنها با استفاده از قابلیتهای تعریف شده در تکنولوژی‌های وب (سی‌اس‌اس ۳ [[28]](#footnote-28)، اچ‌تی‌ام‌ال ۵ [[29]](#footnote-29) و جاوااسکریپت[[30]](#footnote-30)) بهره می‌برد و به این روش وابستگی به سکو را در نرم‌افزار به حد‌اقل می‌رساند. برنامه‌ها در پوششی در سکوی مورد نظر اجرا شده و برای دستیابی به توانایی‌های دستگاه –مانند سنسورها، داده‌، شبکه و...- بر اتصالات[[31]](#footnote-31) مطابق استاندارد واسط برنامه نویسی اتکا دارد.



تصویر 1 - ساختار برنامه در Cordova

کوردووا تا سال ۲۰۱۲ برای ایجاد برنامه‌های هر سکو نیاز به سیستمی با سکو سازگار آن داشت، برای مثال برای ایجاد برنامه برای iOS نیاز به یک رایانه اپل[[32]](#footnote-32) و برای ایجاد برنامه برای ویندوز و یا ویندوزفون، نیاز به رایانه‌ای با سیستم عامل ویندوز بود. ولی بعد از سپتامبر ۲۰۱۲ شرکت ادوبی با ارائه سرویس سازنده فون‌گپ[[33]](#footnote-33) این امکان را فراهم آورد تا بدون نیاز به چندین سیستم (و یا سیستم عامل) و با بارگذاری[[34]](#footnote-34) فایلهای خود در این سرویس، خروجی را بصورت نرم‌افزار در سکو‌های دلخواه دریافت نمائید.

نحوه عملکرد

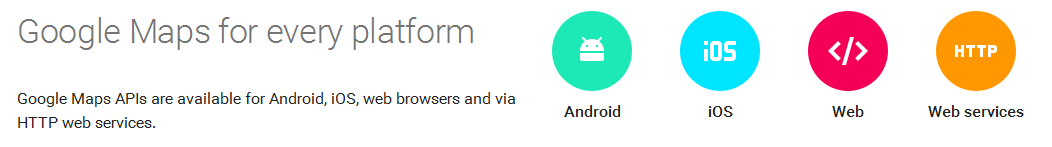
نحوه عملکرد بصورت اجمالی به این صورت است که با اجرای برنامه در سیستم مورد نظر ابتدا نرم‌افزار کوردووا یک نمای وب[[35]](#footnote-35) ایجاد نموده و صفحه اچ‌تی‌ام‌ال مورد نظر را بر روی آن اجرا می‌کند، در این حالت کنترل برنامه در اختیار کدهای جاوااسکریپت ایجاد شده توسط توسعه‌دهنده قرار خواهد گرفت. در صورتی که نرم‌افزار نیاز به برقراری ارتباط به سخت‌افزار دستگاه داشته باشد، نیاز به افزونه‌[[36]](#footnote-36)های کوردووا خواهد بود. این افزونه‌ها در واقع کتابخانه‌هایی شامل دو بخش هستند. یک بخش کدهای جاوااسکریپتی که برنامه توسعه دهنده از آن استفاده می‌کند؛ بخش دیگر آن به زبان برنامه‌نویسی سیستم‌عامل مورد نظر و کوردووا در این میان وظیفه برقراری ارتباط بین دو قسمت افزونه از قبیل ارسال درخواستها، دریافت و بازگرداندن پاسخها و ...) را برعهده دارد. برخی از این افزونه‌ها توسط خود کوردووا و تعداد زیادی نیز توسط توسعه‌دهنده‌های دیگر ارائه شده‌اند. بعنوان نمونه می‌توان به افزونه‌های باتری، شبکه، جی‌پی‌اس و ... اشاره کرد.

نقشه گوگل

همه چیز مهیای یک سفر لذت‌بخش است. نقشه بزرگی تهیه کرده‌اید و برای برخی استان‌های مختلف که در مسیرتان قرار دارند نیز نقشه‌های جداگانه با جزئیات بیشتر خریده‌اید. همه چیز خوب است ولی این روش دیگر کمی قدیمی به نظر می‌رسد. شاید بهتر باشد با نرم‌افزارهای نقشه کمی بیشتر آشنا شویم.

آیا روزهای پیش از وجود نقشه‌های گوگل را به خاطر دارید؟ نقشه‌های بزرگ کاغذی که بعد از استفاده زحمت زیادی برای تا کردن آنها متقبل می‌شدیم در نهایت هم هیچ‌وقت نقشه‌ها به درستی تا نمی‌شدند. در مقابل امروزه ما به نرم‌افزارهایی دسترسی داریم که با کمک جی‌پی‌اس در سراسر دنیا کار می‌کنند. دیگر نیازی به نگرانی در مورد گم شدن نداریم.

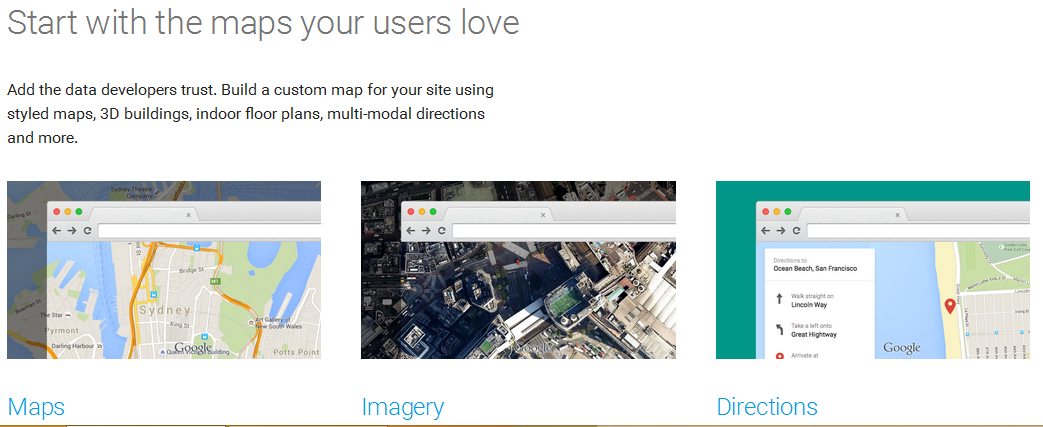
اگر چه نرم‌افزار رسمی گوگل برای استفاده از سرویس نقشه در گوشی‌ها و سایر سکوهای نرم‌افزاری[[37]](#footnote-37) ارائه شده، ولی گاهی به روشی غیر از این نرم‌افزارها برای استفاده از نقشه نیازمندیم. برای مثال در یک نرم‌افزار به نمایش نقشه با قابلیت‌ها تعریف شده در نقشه گوگل (نمایش موقعیت فعلی، نمایش نقاطی در اطراف نقطه مورد نظر و همچنین اطلاعات اضافی مربوط به موقعیتهای نمایش داده شده، مسیر یابی و ... ) نیاز داریم. برای این منظور گوگل واسط برنامه‌نویسی نرم‌افزار[[38]](#footnote-38) خود را در اختیار توسعه دهندگان قرار داده تا به کمک آن بتوانند بدون نیاز به پیاده‌سازی مجدد نقشه بصورت اختصاصی (به نوعی اختراع مجدد چرخ)، از قابلیتهای نقشه بهره‌مند شوند.



تصویر 2- سکوهای پشتیبانی شده در نقشه گوگل

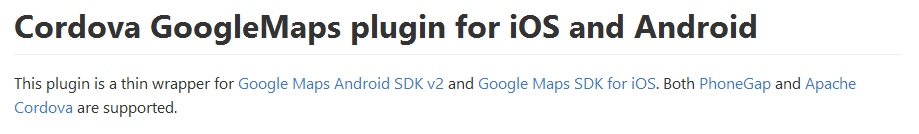
این شرکت با توجه به سکو‌های مختلف کاربردهای واسطهای متفاوتی تعریف کرده که از جمله آنها می‌توان به واسط وب، اندروید[[39]](#footnote-39) و آی‌او‌اس[[40]](#footnote-40) اشاره کرد.

کوردووا و نقشه گوگل



تصویر 3 - نقشه گوگل

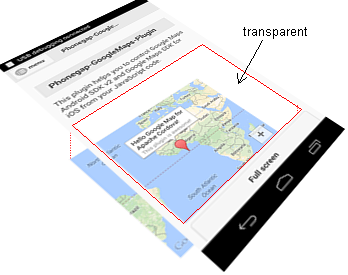
همانطور که گفته شد گوگل برای سکو‌های مختلف واسطهای مورد استفاده آنها را تدارک دیده است. از این جمله می‌توان به واسط وب نقشه اشاره کرد که با فراخوانی یک فایل جاوااسکریپت به راحتی می‌توان نقشه را در وب‌سایت خود نمایش دهید. حال کافیست قسمتی از (یا کل) نمای وب برنامه را به عنوان محل نمایش نقشه به واسط معرفی نمود تا نقشه در محدوده مورد نظر نمایش داده شود. با کمک این واسط می‌توان نقاط دلخواه را بر روی نقشه نمایش داد، فاصله بین دو نقطه را بدست آورد و یا حتی بین آنها مسیریابی نمود. از آنجایی که کوردووا از نمای وب استفاده می‌کند، به همین دلیل نسبت به نرم‌افزارهای بومی کمی کندتر است. همین ایراد بر واسط نقشه نیز وارد است زیرا بر روی نمای وب قرار می‌گیرد. برای رفع این ایراد می‌توان از روش دیگر استفاده از نقشه گوگل در برنامه‌های مبتنی بر کوردووا که افزونه‌‌ها است استفاده کرد. افزونه نقشه از کیت توسعه نرم‌افزار بومی گوگل استفاده می‌کند که نسبت به واسط وب از نظر سرعت عملکرد بهتری از خود نشان می‌دهد.



تصویر 4- افزونه نقشه گوگل برای کوردووا

نحوه عملکرد افزونه

واسط برنامه نویسی نقشه گوگل در نسخه ۱.۲.۰ امکان افزودن اچ‌تی‌ام‌ال بر روی نقشه را از فراهم آورد. در واقع نقشه یک نمای بومی[[41]](#footnote-41) است که هیچ ارتباطی با وب ندارد و این به معنی نمایش دو نمای متفاوت بر روی یکدیگر است. برای نمایش محتوی بر روی آن نقشه زیر نمای وب قرار می‌گیرد بنابر‌این برای قابل رویت بودن نقشه، المانهای وبی که بر روی آن قرار می‌گیرند باید شفاف باشند که خود افزونه به این امر رسیدگی می‌کند.

این افزونه توسط ماساهی کاتسوماتا[[42]](#footnote-42) بصورت مستقل توسعه داده شده است به این معنی که توسط خود گوگل پشتیبانی نشده و به همین دلیل نمی‌توان انتظار داشت با تغییرات نقشه گوگل بروز رسانی شود. همچنین هنوز تمامی قابلیتهای نقشه در آن پشتیبانی نشده و برنامه‌ای برای پوشش آنها در کوتاه مدت وجود ندارد.

پیاده‌سازی برنامه بکمک کوردووا

نصب خط فرمان کوردووا (فون‌گپ)

برای آنکه بتوان پروژه‌ای را با کمک کوردووا تعریف نماییم، ابتدا لازم است تا خود را نصب نماییم.[[43]](#footnote-43) برای نصب کوردووا نیاز است محیط نودجی‌اس[[44]](#footnote-44) قبلا بر روی سیستم نصب شده باشد. از آنجایی که نودجی‌اس در حال حاضر در دو نسخه پشتیبانی طولانی مدت[[45]](#footnote-45) (ورژن ۴.۵) و نسخه پایدار (۶.۶) ارائه می‌گردد بنابراین برای نصب نسخه دلخواه می‌توان از یکی از دو دستور زیر باری دانلود آن استفاده نمود:

$ curl -sL https://deb.nodesource.com/setup\_4.x | sudo -E bash -

$ curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash –

سپس با دستور زیر اقدام به نصب نودجی‌اس خواهیم کرد:

$ sudo apt-get install -y nodejs

حال با دستور زیر کوردووا را بر روی سیستم نصب خواهیم نمود:

$ sudo npm install -g cordova

پس از نصب شدن کوردووا بر روی سیستم دستور cordova به دستورات خط فرمان افزوده شده و امکان استفاده از آن فراهم خواهد شد.

ایجاد پروژه

برای ایجاد پروژه به کمک خط فرمان از دستور زیر استفاده می‌نماییم:

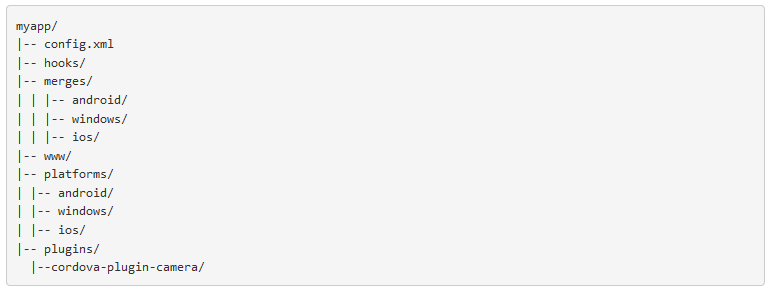
$ cordova create path [id [name [config]]]

**path**:‌ مسیر پوشه‌ای که کوردووا برای ایجاد پروژه باید ایجاد نماید.

**id**: مقداری که در خصوصیت id در فایل config.xml قرار خواهد گرفت و ساختار نوشتاری آن بصورت شاخص سبک دامنه معکوس است. در صورت عدم تخصیص مقدار پیش‌فرض (io.cordova.hellocordova) استفاده می‌شود.

**name**: عنوان برنامه که در خصوصیت name فایل config.xml قرار می‌گیرد. مقدار پیش‌فرض در صورت عدم تخصیص HelloCordova خواهد بود.

**config**: مقداری بصورت متن جی‌سان[[46]](#footnote-46) که بصورت کلید و مقدار در مسیر <path>/.cordova/config.json ذخیره خواهد شد.



تصویر 5- ساختار ایجاد شده توسط دستور cordova create myapp

تمامی دستوراتی که برای مدیریت یک برنامه توسط کوردووا اجرا می‌شود باید در داخل پوشه پروژه (در اینجا myapp) فراخوانی شوند. بنابراین اولین کاری که بعد از ایجاد پروژه باید صورت گیرد، ورود به پوشه مورد نظر است:

$ cd myapp

افزودن سکوها

با دستور زیر می‌توانید سکوهای مورد نیاز خود را به پروژه اضافه کنید:

$ cordova platform add android --save

$ cordova platform add ios --save

در مثال فوق دو سکوی اندروید و آی‌اواس به پروژه اضافه شدند. حال برای بررسی سکوهای اضافه شده می‌توانیم از دستور زیر استفاده نماییم:

$ cordova platform ls

لیست سکوهای پشتیبانی شده توسط کوردووا به قرار زیر است:

* Amazon Fire OS.
* Android.
* BlackBerry 10.
* Firefox OS.
* iOS.
* Ubuntu.
* Windows Phone 8.
* Windows.



تصویر 6 - سیستم عاملهای پشتیبانی شده توسط کوردووا

بررسی و نصب پیش‌نیازها برای ایجاد و اجرا

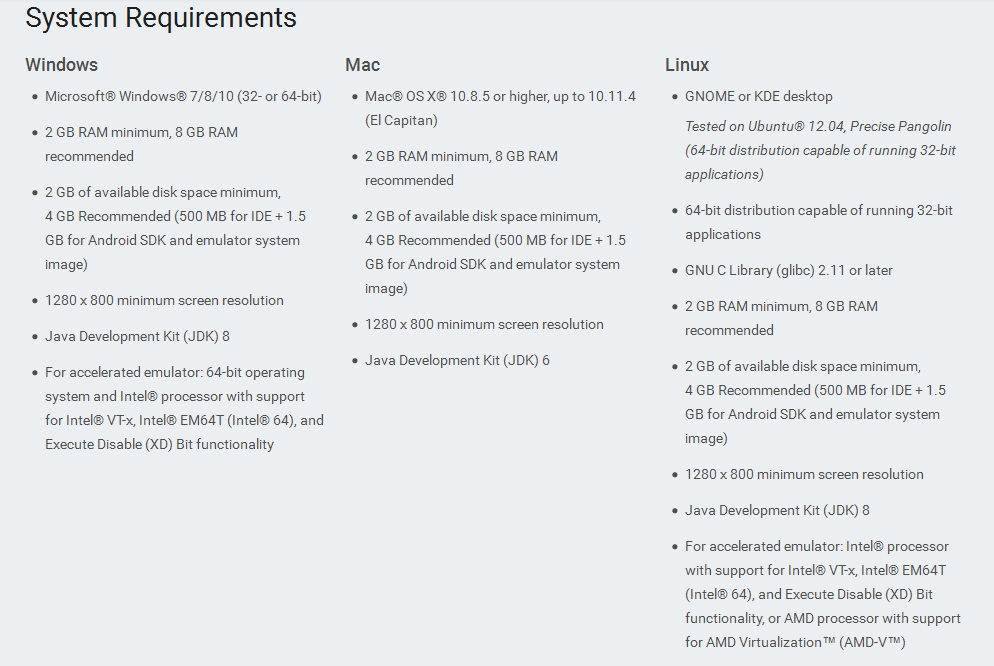
برای ایجاد و اجرای برنامه‌ها، نیازمند نصب کیت توسعه نرم‌افزاری[[47]](#footnote-47) مربوط به سکوی مورد نظر هستیم. بعنوان جایگزین، اگر شما از مرورگر برای توسعه استفاده می‌کنید می‌توانید سکوی مرورگر را اضافه کنید که پیش‌نیازی ندارند.

پیش‌نیازهای ایجاد برنامه اندروید

همانطور که گفته شد، کوردورا برای ایجاد و اجرای برنامه برای هر سکو نیازمند کیت توسعه سکو مورد نظر می‌باشد. به همین منظور پیش از شروع ایجاد پروژه برای هر سکو باید این کیت را فراهم نمود. از آنجایی که برنامه نوشته شده قرار است بر روی سکوی اندروید اجرا شود، لذا توضیحات مرتبط با آن را درج می‌نماییم.

کوردوا برای اندروید نیازمند کیت اندروید می‌باشد که بر روی لینوکس، اواس‌ایکس و ویندوز قابل نصب است.

­­­­­­­­­­­­­­



تصویر 7 – پیش نیازهای نصب Android SDK

کوردووا در آخرین نسخه بسته اندروید[[48]](#footnote-48) خود از واسط برنامه‌نویسی اندروید سطح ۲۳ پشتیبانی می‌کند. در زیر جدول سطوح پشتیبانی نسخه‌های اخیر کوردووا را مشاهده می‌کنید:

|  |  |
| --- | --- |
| نسخه بسته اندروید کوردووا | سطوح پشتیبانی شده اندروید |
| 5.x.x | 14-23 |
| 4.1.x | 14-22 |
| 4.0.1 | 10-22 |
| 3.7.x | 10-21 |

جدول 1- نسخه‌های پشتیبانی شده اندروید در نسخه‌های اخیر کوردووا.

نصب پیش نیازها

موارد زیر برای ایجاد و اجرای برنامه اندروید باید بر روس سیستم نصب باشند:

* **کیت توسعه جاوا (JDK[[49]](#footnote-49)):** نسخه مورد نیاز، نسخه ۷ و یا بالاتر می‌باشد.
* **کیت توسعه اندروید:** برای این منظور «ابزار مستقل کیت توسعه اندروید[[50]](#footnote-50)» یا «محیط توسعه اندروید[[51]](#footnote-51)» را نصب نمایید. در صورتی که تنها قصد توسعه نرم‌افزار با ابزارهای فعلی کوردووا را دارید نصب کیت توسعه مستقل کافی است.
* افزودن بسته‌های کیت توسعه: بعد از نصب کیت توسعه اندروید، متناسب با سطح ای‌پی‌آی هدف برخی بسته‌ها باید نصب شوند. به این منظور ابزار مدیریت کیت توسعه را اجرا کرده و از نصب بودن این موارد اطمینان حاصل نمایید:
  + کیت توسعه سکو اندروید مرتبط با نسخه هدف.
  + ابزار ساخت[[52]](#footnote-52) کیت توسعه نسخه ۱۹.۱.۰ یا بالاتر.
  + مخزن پشتیبانی اندروید[[53]](#footnote-53).

تعیین متغییرهای محیطی

واسط خط فرمان[[54]](#footnote-54) کوردووا به منظور کارکرد صحیح نیازمند برخی متغییرهای محیطی[[55]](#footnote-55) است. اگر چه واسط بصورت خودکار سعی در مقدار دهی این متغییرها می‌کند، ولی در برخی موارد نیاز است تا توسعه دهند، بصورت دستی این مقادیر را تعریف و تعیین نماید:

* JAVA\_HOME: آدرس محل نصب کیت توسعه جاوا.
* ANDROID\_HOME: آدرس محل نصب کیت توسعه نرم‌افزار اندروید.
* بهتر است آدرس پوشه‌های tools و platform-tools از کیت توسعه اندروید نیز در متغیر PATH اضافه شود.

علاوه بر تنظیماتی که در بالا بیان شد تنظیمات دیگری (مانند تعریف شبیه ساز اندروید) نیز وجود دارد که با توجه به عدم نیاز در برنامه پیش رو از توضیح آنها صرفنظر گردید.

بررسی پیش‌نیازها

برای بررسی فراهم بودن پیش‌نیازها، می‌توان از دستور زیر استفاده نمود:

$ cordova requirements

Requirements check results for android:

Java JDK: installed .

Android SDK: installed

Android target: installed android-19,android-21,android-22,android-23,Google Inc.:Google APIs:19,Google Inc.:Google APIs (x86 System Image):19,Google Inc.:Google APIs:23

Gradle: installed

Requirements check results for ios:

Apple OS X: not installed

Cordova tooling for iOS requires Apple OS X

Error: Some of requirements check failed

در مثال فوق پیش نیازهای اندروید رعایت شده ولی از آنجایی که برای ایجاد نرم‌افزار iOS نیاز به سیستم عامل اپل است، لذا در بررسی پیش‌نیازهای اپل با خطا مواجه می‌شویم. به همین دلیل سکوی آی‌اواس را حذف می‌کنیم:

$ cordova platform rm ios

ساخت برنامه

بصورت پیش‌فرض با اجرای دستور create ساختار ساده‌ی برنامه مبتنی بر وب ایجاد می‌شود که صفحه آغازین آن در پوشه پروژه با آدرس www/index.html قابل دسترس است. هرگونه مقدار دهی اولیه باید به عنوان بخشی از رسیدگی‌کننده به رخداد آمادگی دستگاه[[56]](#footnote-56) که در مسیر www/js/index.js قرار دارد، صورت پذیرد.

برای ایجاد خروجی پروژه برای همه سکوها کافیست دستور زیر را اجرا کنید:

$ cordova build

با اجرای دستور زیر خروجی ایجاد شده محدود به سیستم عامل اندروید خواهد شد:

$ cordova build android

آزمون و بررسی خروجی برنامه

کیتهای توسعه برنامه‌های موبایل معمولا با یک شبیه‌ساز که تصویری[[57]](#footnote-57) از دستگاه را اجرا می‌کند همراه شده‌اند. بنابراین این امکان وجود دارد تا برنامه خود را در آن اجرا کرده و وضعیت برنامه خود در سکو مربوطه را بررسی نمایید. برای اجرای شبیه ساز در کوردووا کافیست از دستور زیر استفاده نمایید:

$ cordova emulate [<platform>]

این دستور برنامه را دوباره ساخته و اگر دستور شبیه سازی سکوی هدف خاصی را تعیین نکرده باشد، برنامه را بر روی تمام شیبه‌سازهای موجود نصب و اجرا می‌نماید. علاوه بر آن این امکان وجود دارد تا با دستور زیر خروجی برنامه را بر روی دستگاه حقیقی متصل مشاهده نمود:

$ cordova run [<platform>]

مدیریت افزونه‌ها

با ویرایش برنامه پیش‌فرض می‌توان از برتری‌های فن‌آوری‌های وب بهره برد. ولی برای برنامه‌ای که قرار است از امکانات سطح دستگاه[[58]](#footnote-58) استفاده کند، باید از افزونه‌ها استفاده کنید.

افزونه‌ها، یک واسط برنامه‌نویسی جاوااسکریپت برای عملکردهای بومی دستگاه در اختیار توسعه دهنده قرار می‌دهند. افزونه‌ها غلبا بر روی مدیر بسته‌های نودجی‌اس[[59]](#footnote-59) میزبانی می‌شوند و می‌توان با استفاده از صفحه جستجوی افزونه‌ها یا دستور زیر در آنها جستجو کرد:

$ cordova plugin search <search\_phrase>

برخی از واسطهای کلیدی توسط پروژه متن باز آپاچی تهیه شده‌اند با عنوان افزونه‌ واسطهای برنامه‌نویسی هسته[[60]](#footnote-60) خوانده می‌شوند. برای مثال برای اضافه کردن افزونه موقعیت جغرافیایی از دستور زیر استفاده می‌کنیم:

$ cordova plugin add cordova-plugin-geolocation

بروز رسانی کوردووا و سکوها

برای اطلاع از نسخه‌ای که بر روی سیستم نصب است می‌توان از دستور زیر استفاده نمود:

$ npm info cordova version

و برای بروز رسانی[[61]](#footnote-61):

$ sudo npm update -g cordova

به منظوز بروز رسانی سکو‌های نصب شده بود:

$ cordova platform update android –save

کتابخانه‌ها و چهارچوبها

همانطور که پیشتر بیان شد، نرم‌افزارهای تهیه شده توسط کوردووا مبتنی بر وب بوده و برای مدیریت عملکردهای متفاوت در قبیل مدیریت وقایع، راستی‌آزمایی داده‌ها و ... نیاز به استفاده از جاوااسکریپت است. در این بین می‌توان به تنهایی از خود جاوااسکریپت بهره برد یا اینکه از کتابخانه‌ها و چهارچوبهای تعریف شده برای آن بهره برد. در اینجا به توضیح مختصر و اجمالی در مورد کتابخانه‌ها و چهارچوبهای مورد استفاده در این برنامه می‌پردازیم.

انگولار جی‌اس

تا زمانی انگولارجی‌اس[[62]](#footnote-62) برای زبان برنامه نویسی دارت[[63]](#footnote-63) خود معرفی نکرده بود شاید بهترین تعریف برای آن چهارچوب محبوب جاوااسکریپت بود. ولی از آنجایی که گوگل نقشه‌های بلندپروازانه‌ای برای دارت و جایگزینی آن با جاوااسکریپت دارد، پس این پروژه قدرتمند خود را برای دارت نیز تعریف کرده و در هر دو زبان قابل استفاده است. در واقع انگولار با کمک گرفتن از قابلیتهای جدیدی که در نسخه ۵ اچ‌تی‌ام‌ال تعریف شده دست به کار شد و با افزودن به دایره لغات (تگ[[64]](#footnote-64)های) آن باعث راحت شدن ایجاد برنامه‌های وب شد. البته این بخش کوچکی از این چهارچوب است و در عین حال با تعریف کتابخانه‌های مختلف و همچنین ساده سازی سفارشی‌سازی و توسعه امکانات انگولار سبب شد تا مورد استقبال توسعه‌دهندگان قرار گیرد.

اونسن یو‌آی

علاوه بر کتابخانه‌های ایجاد شده برای ایجاد برنامه‌های موبایلی با استفاده از برنامه‌های مبتنی بر وب، نیاز به ایجاد ظاهری مشابه نرم‌افزارهای موبایل برای این گونه سایتها و برنامه‌ها کاملا قابل تشخیص بود که در این بین، رویکردهای متفاوتی وجود داشت. برای مثال می‌توان به پروژه‌های بوت‌استرپ[[65]](#footnote-65) شرکت توییتر، جی‌کوئری موبایل[[66]](#footnote-66)، آیونیک[[67]](#footnote-67)، اونسن‌یو‌آی[[68]](#footnote-68)، انگولاریوآی برای موبایل[[69]](#footnote-69) و... اشاره کرد.

بوت‌استرپ به ایجاد انقلاب در طراحی واکنش‌گرا معروف است که با توسعه دهنده این امکان را می‌دهد تا با اطمینان از اینکه برنامه او در گستره‌ی وسیعی از صفحات نمایش بدون ایراد (یا با کمترین مشکل) به نمایش در خواهد آمد با کمترین دغدغه‌ای به کار خود بپردازد.

مدیران جی‌کوئری نیز که سعی داشتند مانند گذشته توسعه‌دهندگان وب دوباره را به سوی خود جذب نمایند و از رقبای خود عقب نمانند، دو کتابخانه جدید برای واسط کاربری[[70]](#footnote-70) و برنامه‌های ترکیبی[[71]](#footnote-71) موبایل ارائه نمودند که بر خلاف اقبال خوبی که مدیون اعتبارات کسب شده توسط جی‌کوئری بود با گذشت زمان بدلیل محدودیتهایی که نسبت به سایر چهارچوبها و کتابخانه‌ها داشت[[72]](#footnote-72) تقریبا از صحنه رقابت محو شد.

آیونیک و اونسن‌یوآی رویه تقریبا مشابه‌ای را پیش گرفتند و با تعریف تمهای مختلف و همچنین ظاهری مشابه برنامه‌های بومی هر سیستم عامل،‌ باعث شدند تا توسعه‌دهندگان زیادی به سوی این دو جذب شوند. ولی به هر حال هر کدام ویژگی‌های منحصر به فرد خود را دارا بودند از جمله اینکه آیونیک از سیستم‌عاملهای آی‌او‌اس ۶+ و اندروید ۱/۴+ پشتیبانی می‌کند و پشتیبانی از ویندوز‌فون و فایر‌او‌اس را نیز در نقشه راه خود قرار داده است. در حالی که اونسن‌یو‌آی از سیستم عاملهای آی‌اواس ۶+‌، اندروید ۳/۲+ و فایر‌فاکس، از کروم و سافاری نیز پشتیبانی می‌کند.

در مورد Bower

ساده‌ترین راه برای اضافه کردن کتابخانه‌ها و چهارچوبهای وب استفاده از شبکه تحویل محتوی[[73]](#footnote-73) است، زیرا هم تراکنشهای بین سیستم‌ کاربر و سرور سایت کمتر می‌شود و همچنین اگر سایتهای مختلف از این روش برای ارجاع استفاده کنند در نهایت با دریافت هر فایل از این شبکه، در مورد سایر سایتها از فایل ذخیره شده استفاده می‌شود و از دانلود مجدد آن خودداری می‌کند. ولی در یک برنامه ترکیبی که قرار است صفحه (صفحات) وبی را در گوشی کاربر در غالب یک برنامه به اجرا در آورد دیگر نیازی به این کار نیست. حتی ممکن است کاربر را در استفاده اول دچار مشکل نماید.

علاوه بر آن گاهی یک کتابخانه یا چهارچوب به کتابخانه یا چهارچوبی دیگر وابسته است که مدیریت این وابستگی‌ها و اینکه به کدام نسخه از آن وابسته است در زمان نصب و در زمان بروز رسانی کتابخانه یا چهارچوب بسیار مهم است. برای این منظور از نرم‌افزار مدیریت بسته پیش‌رو[[74]](#footnote-74) Bower استفاده می‌کنیم. Bower در هنگام نصب با بررسی تمامی بسته‌های مورد نیاز مراحل نصب را مدیریت می‌کند و این امکان را فراهم می‌کند که در صورت نیاز در مسیر توسعه بین نسخه‌های متفاوت یک کتابخانه (یا چهارچوب) جابجا شد یا آنرا بروزرسانی کرد.

برای مثال این برنامه به چهارچوب انگولارجی‌اس، اونسن‌یو‌آی (واسط گرافیکی موبایل) و کتابخانه ngStorage (کتابخانه مدیریت LocalStorage و SessionStorage انگولار‌جی‌اس) نیاز دارد که می‌توان در حین یا بعد از مقداردهی اولیه[[75]](#footnote-75) bower برای پروژه آنها را تعریف نمود.

نصب و بروز رسانی bower و بسته‌های آن

bower یکی از بسته‌ها نودجی‌اس است که به راحتی با کمک مدیر بسته‌ّهای نود قابل قابل نصب است به شرط آنکه گیت[[76]](#footnote-76) بر روی سیستم نصب باشد:

$ sudo npm install -g bower

و پس از انجام عملیات نصب، دستور bower در خط فرمان در دسترس قرار خواهد گرفت که می‌توان به کمک آن بسته‌های مورد نیاز را نصب کرد.

$ bower install angular

$ bower install onsenui

$ bower install ngstorage

مقدار دهی اولیه bower

در هر پروژه می‌توان بدون نیاز به مقدار دهی اولیه bower اقدام به نصب نسخه‌ّهای مختلف کتابخانه‌های front-end کرد. ولی مواردی وجود دارد که بر مبنای آن بهتر است در هر پروژه bower را مقدار دهی نماییم. از جمله این موارد می‌توان به بروز رسانی بسته‌ها اشاره کرد. علاوه بر آن در هنگام توسعه نرم‌افزار و در حین کنترل نسخه تنها چیزی که از کتابخانه‌های خارجی باید در کنترل نسخه وارد شود، مرجعی است که طبق آن هریک از اعضای تیم توسعه که آنرا در سیستم خود کلون[[77]](#footnote-77) می‌کند یا هر فرد دیگری که انشعابی[[78]](#footnote-78) از آن گرفت، بتواند آن بسته را به برنامه خود اضافه نماید. با این رویکرد دیگر فضای اضافی برای ذخیره وابستگی‌ها هدر نخواهد شد.

برای این منظور مدیر بسته‌ّهای bower -در صورت مقدار دهی اولیه- این امکان را فراهم می‌نماید تا در بدون نیاز به حفظ فایلهای بسته‌ّهای مورد نیاز، بتوان در هر زمان وابستگی‌های مربوطه را به سیستم اضافه نمود. برای مقدار دهی اولیه bower ابتدا به پوشه برنامه رفته و دستور زیر را اجرا می‌کنیم:

$ bower init

پس از اجرای این دستور یک سری سوالات پرسیده می‌شود که در ایجاد فایل بیانیه[[79]](#footnote-79) با نام "bower.json" مورد استفاده قرار می‌گیرد. همانطور که از نام این فایل بر می‌آید، محتویات آن شامل یک شیء جی‌سون[[80]](#footnote-80) است که حاوی تنظیمات و اطلاعات مربوط به نرم‌افزار و وابستگی‌ّهای آن. لازم به ذکر است این دستور فقط یک دستور کمکی جهت سهولت در ایجاد و تعریف ویژگی‌[[81]](#footnote-81)های این فایل بیانیه و در صورت تمایل می‌توان فایل را بصورت دستی[[82]](#footnote-82) تعریف و مقداردهی نمود. در زیر فایل بیانیه مربوط به پروژه مشاهده می‌کنید:



تصویر 8 - ساختار فایل bower.json

همانطور که در بالا مشاهده می‌کنید وابستگی‌های برنامه به قرار زیر است:

* انگولارجی‌اس - جدیدترن نسخه با شماره اصلی ۱.
* اونسن‌یو‌آی - نسخه ۲ / نامزد انتشار[[83]](#footnote-83) ۱۵ (آخرین نسخه موجود نرم‌افزار در هنگام توسعه).
* ان‌جی‌استوریج – جدیدترین نسخه زیرمجموعه نسخه ۳/۰.

در صورتی که بعد از ایجاد فایل بیانیه bower قصد افزودن بسته جدیدی به آنرا داشته باشید کافیست در انتهای دستور نصب بسته از گزینه '--save' استفاده نمایید:

$ bower install angular --save

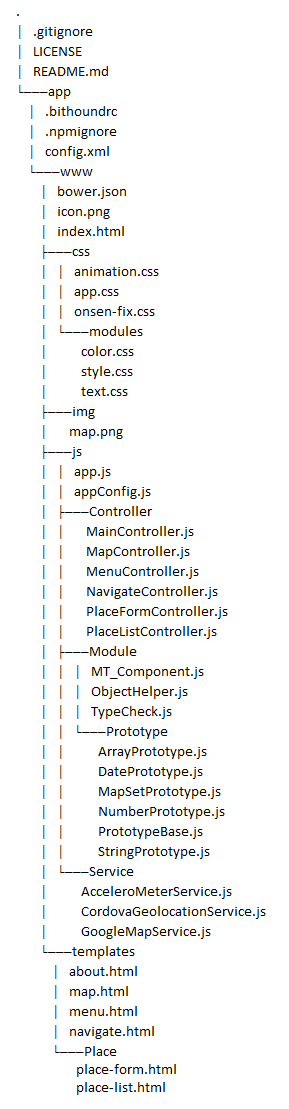
با اجرای این دستور bower علاوه بر افزودن بسته مورد نظر (در اینجا انگولارجی‌اس) آنرا به لیست وابستگی‌ها در فایل بیانیه نیز اضافه می‌کند. در صورتی که فایل بیانیه را در اختیار داشته و قصد افزودن وابستگی‌ها به برنامه خود را دارید کافی است دستور زیر را اجرا نمایید تا تمامی وابستگی‌ّهای لیست شده در بیانیه، به پروژه اضافه شوند:

$ bower install

به منظور بروز رسانی بسته‌ّها از دستور بروز رسانی استفاده می‌کنیم:

$ bower update

کدها و توضیحات پروژه



تصویر 9 - ساختار درختی فایلهای برنامه.

در این قسمت کدهای برنامه را توضیح خواهیم داد. مسلما برای جلوگیری از اطاله مطلب، از توضیح ریز کدها خودداری خواهیم کرد.

ساختار برنامه

در پوشه اصلی برنامه علاوه بر پوشه app فایل مربوط به مجوز، فایل توضیحات پیشفرض گیت و فایل .gitignore –که شامل مواردی است که توسط گیت ردگیری نمی‌شوند - وجود دارد. در پوشه app علاوه دو فایل پیکره‌بندی مربوط به برنامه‌های دیگر، یک پوشه با عنوان www وجود دارد که حاوی فایلهای برنامه وب هستند. علاوه بر آن یک فایل config.xml نیز وجود دارد که اطلاعات پیکره‌بندی سیستم‌عاملهایی که قرار است برنامه خروجی برای آنها داشته باشیم در این فایل قرار می‌کیرد.

در ساختار اصلی این پوشه (app) دو پوشه دیگر نیز وجود دارد که بدلیل نادیده گرفتن توسط گیت در ساختار روبرو مشاهده نمی‌شوند. این دو پوشه دارای عنوان platform (برای سکوهای هدف) و plugin (حاوی افزونه‌های مورد استفاده در برنامه) است.

در پوشه علاوه وب فایل بیانیه[[84]](#footnote-84) bower (bower.json) و آیکون برنامه، یک فایل اچ‌تی‌ام‌ال – محل آغاز برنامه است – وجود دارد و سایر فایلهای مورد نیاز در پوشه‌های زیرین این پوشه دسته بندی شده و در فایل ایندکس (index.html) به آنها ارجاع داده شده. ساختار پوشه‌ّها بدین صورت است:

* **CSS:** حاوی تمامی سبکها[[85]](#footnote-85)ی مورد استفاده در برنامه که در مسیر توسعه به برنامه افزوده شده‌اند.
* **Templates:** قالبهای اچ‌تی‌ام‌ال مورد استفاده به عنوان صفحات مختلف برنامه در این پوشه قرار دارند.
* **JS:** شامل تمامی فایلهای جاوااسکریپت برنامه با تفکیک زیر می‌باشد:
  + **فایلهای app, appConfig:** شامل بخشهای اصلی و عموی برنامه و مقادیر اولیه.
  + **کنترلرهای انگولار:** که هرکدام وظیفه مدیریت یک صفحه و یا بخشی از آنرا برعهده دارند.
  + **سرویسهای انگولار:** که برخی از وظایف برنامه از قبیل مدیریت ارتباطات با نقشه و افزونه منطقه جغرافیایی[[86]](#footnote-86) را بر عده دارند.
  + **پوشه ماژولها:** کهشامل برخی کتابخانه‌های کمکی و تعریف ویژگی برای پیش‌الگوهای[[87]](#footnote-87) کلاسهای جاوااسکریپت (از قبیل رشته‌ها و اعداد) می‌باشد.

فایل شاخص

فایل شاخص (index.html) که به عنوان فایل اصلی معرفی شده، حاوی مرجعی به تمامی فایلهای سبکها و کدهای جاوااسکریپت است. از آنجایی که اونسن‌یوآی میتواند تمام صفحاتی که قرار است نمایش داده شود را بصورت جداگانه و توسط فایلهای قالبی که با آن معرفی می‌شود فراخوانی و نمایش دهد، لذا تنها قطعه کد اچ‌تی‌ام‌ال مورد استفاده مربوط به تعریف منو است که در آن فایلهای حاوی قالب منو و صفحه اصلی معین شده‌اند:



تصویر 10 - تعریف منو در صفحه index.html

کدهای جاوااسکریپت

همانطور که قبلا توضیح داده شد، عمده کارهای برنامه توسط کدهای جاوااسکریپت انجام می‌پذیرد که در ادامه به شرح آنها می‌پردازیم[[88]](#footnote-88).

ماژولها

شامل تمامی کتابخانه‌های سفارشی ایجاد شده توسط توسعه‌دهنده است که در اینجا قرار گرفته.

* MT\_Components: مولفه‌های[[89]](#footnote-89) مورد نیاز:
  + افزودن فیلتر صورت عنوان[[90]](#footnote-90) برای رشته‌ها.
  + افزودن رسیدگی‌کننده[[91]](#footnote-91) رخداد لمس (یا کلیک) طولانی.
* ObjectHelper.js: شامل متدهای کمکی برای کار با اشیاء جاوااسکریپت.
* TypeCheck.js: حاوی شیء TypeCheck برای بررسی نوع داده‌ای.
* پوشه Prototype: شامل یک سری توسعه برای نمونه‌های انواع داده‌ای جاوااسکریپت که در فایلهای مجزا تعریف شده‌اند.

کنترلرها

در انگولار، کنترلرها بخش اصلی کار را برعهده دارند که از آن جمله می‌توان به مدیریت جابجایی بین صفحات، تغییر وضعیت نمایش، اجازه ویرایش یک موجودیت و مدیریت ذخیره و حذف آن و ... اشاره کرد:

* **MainController:**‌ کنترلر اصلی برنامه که محدوده آن شامل تمام برنامه می‌شود به همین دلیل عمومی‌ترین وظایف - که در اینجا فقط شامل تعریف نام برنامه، اقلام فهرست[[92]](#footnote-92) و چند مقدار اولیه است- به آن سپرده شده.
* **MapController:** وظیفه مدیریت صفحه نقشه را بر عهده دارد که شامل موارد زیر می‌شود:
  + **مقدار دهی اولیه نقشه:** هر بار که برنامه اجرا می‌شود لازم است تا از طریق افزونه نقشه بومی فراخوانی شود.
  + **مدیریت وقایع مرتبط با نقشه:‌** وقایع شامل:
    - تغییر نوع نقشه.
    - اعمال تغییرات موقعیت در نمایش نقشه در صورت پی‌گیری محل کنونی کاربر.
    - افزودن/حذف نشانگر در صورت لمس نقشه توسط کاربر.
    - نمایش نقاط ذخیره شده توسط کاربر بهمراه تصویر انتخاب شده برای نقاط.
    - مدیریت جابجایی اشاره‌گر جدید.
  + مدیریت دکمه‌های تعریف شده و جابجایی بین صفحات در صورت نیاز.
    - **تغییر نوع نقشه:**
      * ماهواره‌ای.
      * ترکیبی (ماهواره‌ای و نقشه).
      * نقشه راهها.
      * همراه با پستی‌بلندی‌های جغرافیایی.
    - افزودن محل جدیدی که توسط نشانگر علامتگذاری شده.
    - تغییر وضعیت پیگیری محل کنونی با لمس دکمه موقعیت من[[93]](#footnote-93).
    - ارجاع به صفحه هدایت[[94]](#footnote-94).
    - نمایش ترافیک بر روی نقشه.
  + نمایش وضعیتهای برنامه.
* MenuContorller: مدیریت نمایش و جابجایی بین صفحات برنامه و همچنین تغییر تم.
* NavigateController: مدیریت مکانهای ناوبری تعیین شروع و پایان مسیر و نمایش ناوبری نقشه گوگل.
* PlaceListController: مدیریت فهرست مکانها
* PlaceFormController: مدیریت فرم نمایش مکانها
  + ثبت اطلاعات موقعیت جدید.
  + مشاهده و ویرایش موقعیتهای ذخیره شده.
  + دریافت آدرس موقعیت از نقشه گوگل.
  + راستی‌آزمایی و ذخیره اطلاعات.

سرویسها

در مسیر توسعه با مواردی مواجه می‌شویم که نیازمند یک نمونه از عملکردها در چندین بخش مجزا از برنامه هستیم. اگر در هر مکان این نمونه بصورت جداگانه ایجاد شود، تنها باعث مصرف حافظه بیشتر می‌شود. برای جلوگیری از اینگونه تکرارها، از سرویسها استفاده می‌کنیم؛ سرویسها در هر اجرای برنامه فقط یکبار ایجاد شده و مورد استفاده قرار می‌گیرند. در این پروژه دو سرویس تعریف شده است. سرویس موقعیت جغرافیایی که مدیریت افزونه مربوطه در کوردووا را برعهده دارد و سرویس نقشه گوگل که مدیریت ارتباطات با افزونه نقشه گوگل را برعهده دارد.

منابع

* <https://developers.google.com/maps/>
* <https://en.wikipedia.org/wiki/Smartphone>
* <http://www.cio.com/article/2952890/mobile/6-linux-based-mobile-operating-systems.html>
* <https://en.wikipedia.org/wiki/Mobile_application_development#Front-end_development_tools>
* <https://en.wikipedia.org/wiki/Apache_Cordova>
* <https://github.com/mapsplugin/cordova-plugin-googlemaps>
* <https://cordova.apache.org/docs/en/latest/guide/cli/>
* <https://bower.io/docs/api>
* ­

1. Smartphone [↑](#footnote-ref-1)
2. PDA (Personal Digital Assistant). [↑](#footnote-ref-2)
3. GPS (Global Positionning System). [↑](#footnote-ref-3)
4. NTT DoCoMo [↑](#footnote-ref-4)
5. 4G LTE [↑](#footnote-ref-5)
6. Motion Sensor. [↑](#footnote-ref-6)
7. Virtual Machine. [↑](#footnote-ref-7)
8. Desktop. [↑](#footnote-ref-8)
9. Android. [↑](#footnote-ref-9)
10. Firefox OS. [↑](#footnote-ref-10)
11. Ubuntu Touch. [↑](#footnote-ref-11)
12. Tizen. [↑](#footnote-ref-12)
13. Black Berry. [↑](#footnote-ref-13)
14. Internet of things. [↑](#footnote-ref-14)
15. IDE (Integrated Development Environment). [↑](#footnote-ref-15)
16. Road Map. [↑](#footnote-ref-16)
17. Issue Tracking. [↑](#footnote-ref-17)
18. Bug fix. [↑](#footnote-ref-18)
19. Fron-end. [↑](#footnote-ref-19)
20. UI (User Interface). [↑](#footnote-ref-20)
21. UX (User Experience). [↑](#footnote-ref-21)
22. Cordova. [↑](#footnote-ref-22)
23. PhoneGap. [↑](#footnote-ref-23)
24. Framework. [↑](#footnote-ref-24)
25. Nitobi. [↑](#footnote-ref-25)
26. Adobe. [↑](#footnote-ref-26)
27. Open source. [↑](#footnote-ref-27)
28. CSS3. [↑](#footnote-ref-28)
29. HTML5. [↑](#footnote-ref-29)
30. JavaScript. [↑](#footnote-ref-30)
31. Bindings. [↑](#footnote-ref-31)
32. Apple Computer. [↑](#footnote-ref-32)
33. PhoneGap Build. [↑](#footnote-ref-33)
34. Upload. [↑](#footnote-ref-34)
35. WebView. [↑](#footnote-ref-35)
36. Plugin. [↑](#footnote-ref-36)
37. Software Platforms. [↑](#footnote-ref-37)
38. API (Application Programminng Interface). [↑](#footnote-ref-38)
39. Android. [↑](#footnote-ref-39)
40. iOS. [↑](#footnote-ref-40)
41. Native View. [↑](#footnote-ref-41)
42. Masashi Katsumata. [↑](#footnote-ref-42)
43. از آنجایی که فون‌گپ مبتنی بر کوردووا بوده و در حال حاضر تفاوت عمده بین این دو به واسط کاربری گرافیکی(GUI) و امکانات بصری و تسهیل کننده‌ای که فون‌گپ در اختیار توسعه دهنده قرار می‌دهد، باز می‌گردد و همچنین به دلیل اینکه برای توسعه این نرم‌افزار از واسط گرافیکی استفاده نشده (توسعه نرم‌افزار در محیط اوبونتو 16.04LTS صورت گرفته و فون‌گپ برای سیستمهای مبتنی بر هسته لینوکس واسط گرافیکی ارائه نداده است.) کلیه توضیحات در این نوشته به واسط خط فرمان کوردووا اختصاص خواهد یافت که بدون تغییر خاصی در مورد فون‌گپ هم قابل استفاده است. برای مثال دستور ایجاد پروژه در کوردووا:

    $ cordova create my-map ir.lrnr.mymap

    که در فون گپ تنها نیاز به جابجای کلمه کوردووا و فون گپ خواهد بود:

    $ phonegap create my-map ir.lrnr.mymap

    ضمن اینکه برای استفاده مستقیم از دستورات کوردووا در فون‌گپ کافیست بعد از دستو فون‌گپ از دستور کوردووا استفاده نمایید:

    $ phonegap cordova create my-map ir.lrnr.mymap [↑](#footnote-ref-43)
44. NodeJS. [↑](#footnote-ref-44)
45. LTS (Long Term Support). [↑](#footnote-ref-45)
46. JSON. [↑](#footnote-ref-46)
47. SDK. [↑](#footnote-ref-47)
48. Cordova-Android package. [↑](#footnote-ref-48)
49. Java Developer Kit. [↑](#footnote-ref-49)
50. Android stand-alone SDK tools. [↑](#footnote-ref-50)
51. Android Studio. [↑](#footnote-ref-51)
52. Build-tools. [↑](#footnote-ref-52)
53. Android Support Repository. [↑](#footnote-ref-53)
54. CLI (Command Line Interface). [↑](#footnote-ref-54)
55. Environment Variables. [↑](#footnote-ref-55)
56. "devideready" event handler. [↑](#footnote-ref-56)
57. Image. [↑](#footnote-ref-57)
58. Device level features. [↑](#footnote-ref-58)
59. npm (Node Package Manager). [↑](#footnote-ref-59)
60. Core Plugin APIs. [↑](#footnote-ref-60)
61. Update. [↑](#footnote-ref-61)
62. AngularJS. [↑](#footnote-ref-62)
63. Dart. [↑](#footnote-ref-63)
64. Tag. [↑](#footnote-ref-64)
65. Twitter Bootstarp. [↑](#footnote-ref-65)
66. JQuery Mobile. [↑](#footnote-ref-66)
67. Ionic. [↑](#footnote-ref-67)
68. OnsenUI. [↑](#footnote-ref-68)
69. Mobile AngularUI. [↑](#footnote-ref-69)
70. JQuery UI. [↑](#footnote-ref-70)
71. Hybrid Application. [↑](#footnote-ref-71)
72. <http://www.gajotres.net/ionic-vs-onsenui/> [↑](#footnote-ref-72)
73. CDN (Content Delivery Network) [↑](#footnote-ref-73)
74. Front-end Package Management Software. [↑](#footnote-ref-74)
75. Initialization. [↑](#footnote-ref-75)
76. Git. [↑](#footnote-ref-76)
77. Clone. [↑](#footnote-ref-77)
78. Fork. [↑](#footnote-ref-78)
79. Manifest. [↑](#footnote-ref-79)
80. JSON Object. [↑](#footnote-ref-80)
81. Property. [↑](#footnote-ref-81)
82. Mannually. [↑](#footnote-ref-82)
83. Release Candidate. [↑](#footnote-ref-83)
84. Manifest. [↑](#footnote-ref-84)
85. Styles. [↑](#footnote-ref-85)
86. Geolocation plugin. [↑](#footnote-ref-86)
87. Prototype property definitions. [↑](#footnote-ref-87)
88. به دلیل کوتاه کردن مطالب کدهای css و همچنین قالبهای html توضیح داده نمی‌شوند. [↑](#footnote-ref-88)
89. Components. [↑](#footnote-ref-89)
90. Title case. [↑](#footnote-ref-90)
91. Handler. [↑](#footnote-ref-91)
92. Menu. [↑](#footnote-ref-92)
93. My location. [↑](#footnote-ref-93)
94. Navigate. [↑](#footnote-ref-94)