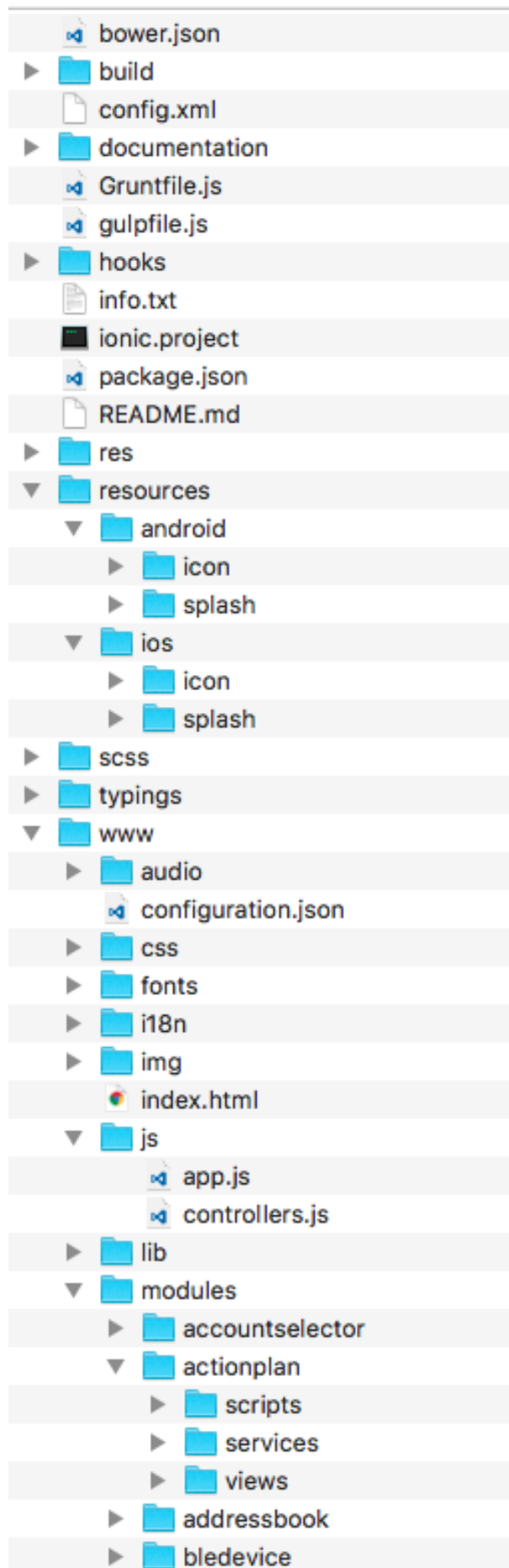


Buenas Pablo,

Ante todo disculpa mi tardanza en responder. Aunque es muy recurrente por mi parte siempre pasa que cuando quieres dedicarle tiempo a algo es cuando más te aprietan en el trabajo e incluso en lo personal. No es excusa, no he querido que pasara más tiempo sin mostraros algo.

Me gustaría comentarte en este documento un análisis sobre vuestra aplicación e igualmente un pequeño análisis sobre un prototipo que os he preparado en Ionic 3.

En cuanto a la arquitectura de vuestra aplicación me parece correcta. Sin embargo, creo que podría haber algún tipo de problema a la hora de escalar. Creo que con ayuda de la estructura tipo de un proyecto IONIC habéis separado perfectamente los principales focos (traducciones, librerías externas, constantes, estilos, vistas, ...). He querido sacar de este saco los archivos .js, que si bien están perfectamente clasificados (controladores, factorías, servicios y directivas) creo que podrían organizarse mejor pensando, como he mencionado anteriormente en la escalabilidad y el aumento de funcionalidades. Te adjunto un pantallazo que creo que va a ser la mejor forma de explicarlo (has de tener en cuenta que yo estoy influenciado por esta forma de trabajar):



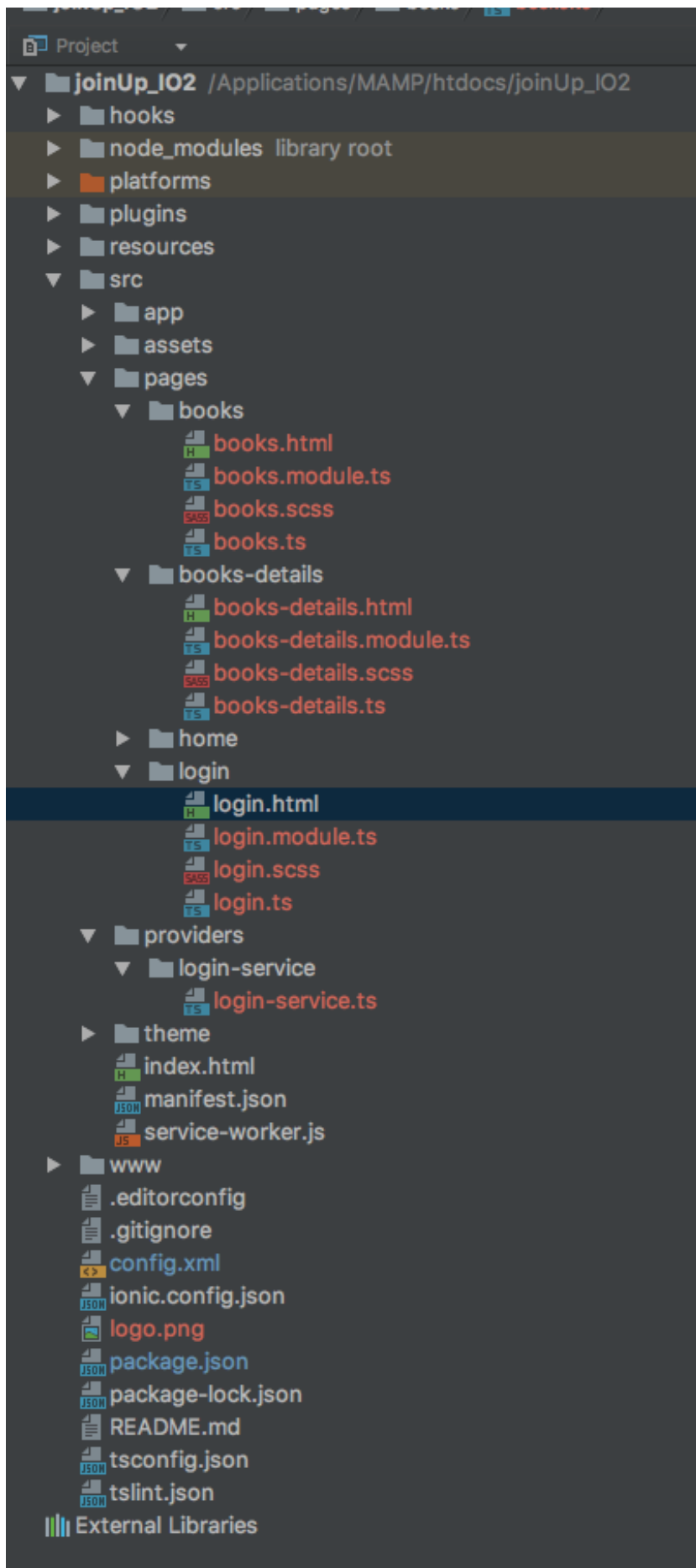
Es una estructura típica de proyecto Angular – Ionic 1. Por defecto igual a la vuestra pero difieren en lo siguiente:

- La carpeta resources, para dividir los assets por plataformas.
- La carpeta i18n, para las traducciones.
- La carpeta js, sólo tiene el app.js y el controlador principal.
- LA carpeta modules. Para mí es la que realmente da valor a esta forma de organización. En cada subcarpeta se incluyen los archivos correspondientes a cada funcionalidad de la aplicación, y se divide en subcarpetas correspondientes a vistas, archivos js y servicios. De esta forma una nueva funcionalidad sólo implica una nueva subcarpeta bajo 'modules'.

Obviamente es una opinión personal, sin llegar a discutir vuestra forma de trabajo e influenciado por esta forma de trabajar. Sin embargo, verás que es un protocolo bastante similar a la estructura por defecto que adoptan los proyectos de las nuevas versiones de Ionic 2 y 3. Las principales características:

- Desaparecen las factorías
- Las directivas se siguen usando, pero lo que más importancia tiene son los componentes que te permiten una reusabilidad de código muy buena.
- Ionic pasa a gestionar las vistas como páginas, y con un sistema de pila que lo hace muy sencillo (similar a Android).
- Cada elemento generado (excepto los providers) tiene asociado 3 archivos que se corresponden con su controlador (.ts), su vista (.html) y sus estilos (independientes entre sí). Además en Ionic 3 se añade un cuarto archivo para conceptos de carga rápida y optimización.
- Ionic 2 incorpora una gestión de los pluggins de cordova que lo hace muy sencillo de manejar.

Perdona que te esté soltando todo este parrafazo. Es para comentarte la estructura del proyecto Ionic 3 que te he subido a Bitbucket y que identifiques los archivos con los que te comento.



Esa estructura que ves ahí es un proyecto Ionic 3, en concreto versión 3.5. El proyecto se compone de 3 páginas y un provider, todos ellos creados desde el CLI. El provider no llama a vuestro back, los datos están mockeados pero me ha parecido adecuado al menos dejarlo implementado.

En cuanto a arquitectura, lo más interesante son los 4 archivos que se generan cuando generas un página o componente. Con esta arquitectura están perfectamente separados las capas de la aplicación.

#### PAGES / COMPONENTES

.ts -> Controladores

.html -> Vistas

.scss -> Estilos independientes de cada elemento.

#### PROVIDERS

.ts -> Servicios. Es bastante recomendable hacer aquí una subcarpeta 'api' que agrupe los servicios que hacen las llamadas a la API.

#### Tests

Creo que se escapa a este documento la estructura adecuada para los tests tanto unitarios como de aceptación.

#### Material subido.

Puedes encontrar el APK probado en la ubicación:

platforms/Android/build/outputs/apk/Android-debug.apk

He probado también la versión IOS, pero siento no poder generar el IPA, no tengo certificado personal.

Obviamente, muchas funcionalidades no están completas. La idea es que sólo sea un prototipo.