7

Speak React Native

React Native course by U+_

Overview

- Redux persist
- Expo eject
- React native init
- Splash screen & icons

Redux persist

Redux persist

- Allows redux to store its state between sessions (i.e. resume app state even after it’s killed)
- Both saving and loading state work automatically when redux persist is configured

Installing

```
$ yarn add redux-persist
```

Initialization - imports & config

```
import { persistStore, persistReducer } from "redux-persist"
import storage from "redux-persist/lib/storage"

const persistConfig = {
 key: "root",
 storage,
}
const reducers = {...}
```

Initialization - persistor

```
const persistedReducer = persistReducer(
  persistConfig,
  combineReducers(reducers),
)
const store = createStore(
  persistedReducer,
  ...
)
const persistor = persistStore(store)
```

Initialization - PersistGate

```
<Provider store={store}>
  <PersistGate loading={null} persistor={persistor}>
    <App />
  </PersistGate>
</Provider>
```

Blacklisting parts of state

Use either blacklist or whitelist

```
const persistConfig = {
   key: "fintech",
   storage,
   blacklist: ["registration"], // Reducer keys that you
do NOT want stored to persistence here
   whitelist: ["app"], // Optionally, just specify the
keys you DO want stored to
};
```

Expo eject

Ejecting expo project

- When do we need it?
  - Custom native functionality
  - Native libs unsupported by Expo
  - Custom build process

```
$ expo eject
```

ExpoKit vs. regular eject

- ExpoKit lets us continue using Expo SDK imports even after eject while allowing us to make changes to the app's native code
- Regular eject means a completely Expo-free project, more control but also more potential issues

Creating react native app

Init new app

```
$ react-native init projectName
```

Running standalone app

```
$ react-native run-android

$ react-native run-ios
```

Automatic libraries linking

- First install the library using yarn
- Then, if the lib supports it, use the following:

```
$ react-native link libraryName
```

Manual libraries linking

- http://facebook.github.io/react-native/docs/linking-libraries-ios.html#manual-linking

Splash screen & icons

Splash screens

- 2 possible approaches
  - Dummy images (bad one)
  - Storyboards (iOS) / Screens (Android) (good one)

Splash screen with dummy images

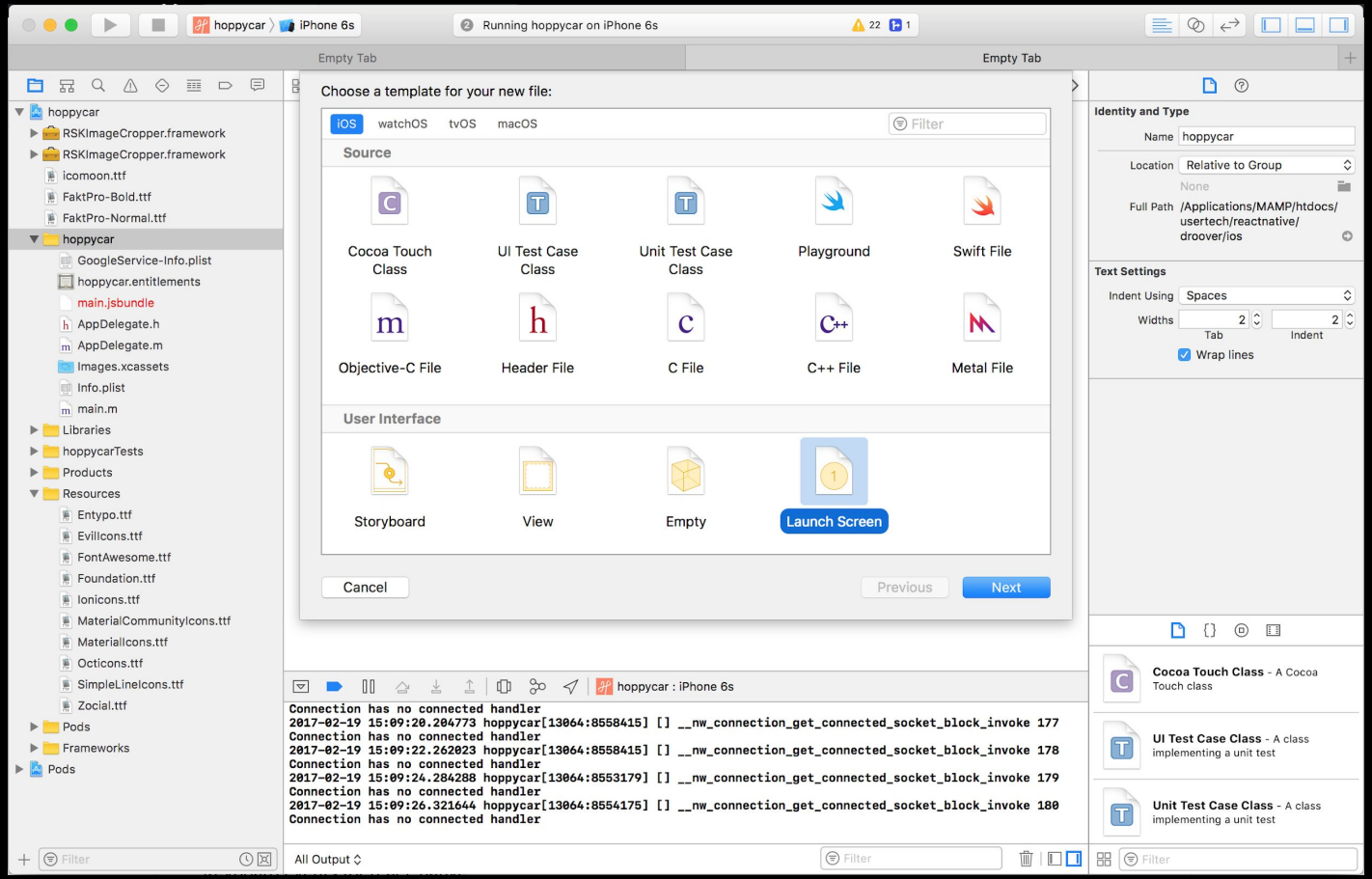
- https://medium.com/the-react-native-log/change-default-launch-screen-in-react-native-ios-app-544f94f1e947

Splash screen with storyboards

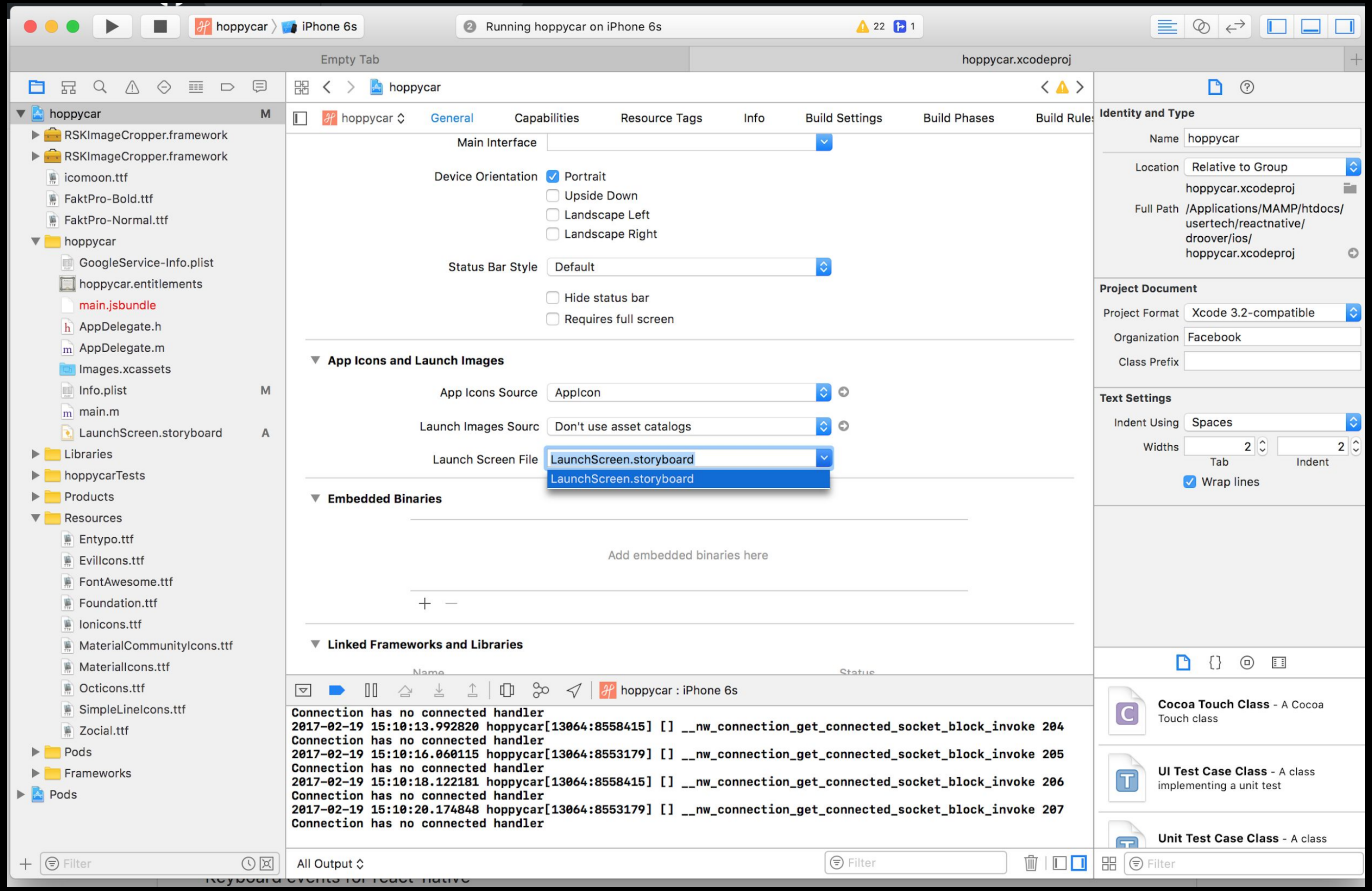
`support-v4:+`

- Scalable for different resolutions
- Prevent image duplicity
- https://medium.com/handlebar-labs/how-to-add-a-splash-screen-to-a-react-native-app-ios-and-android-30a3cec835ae

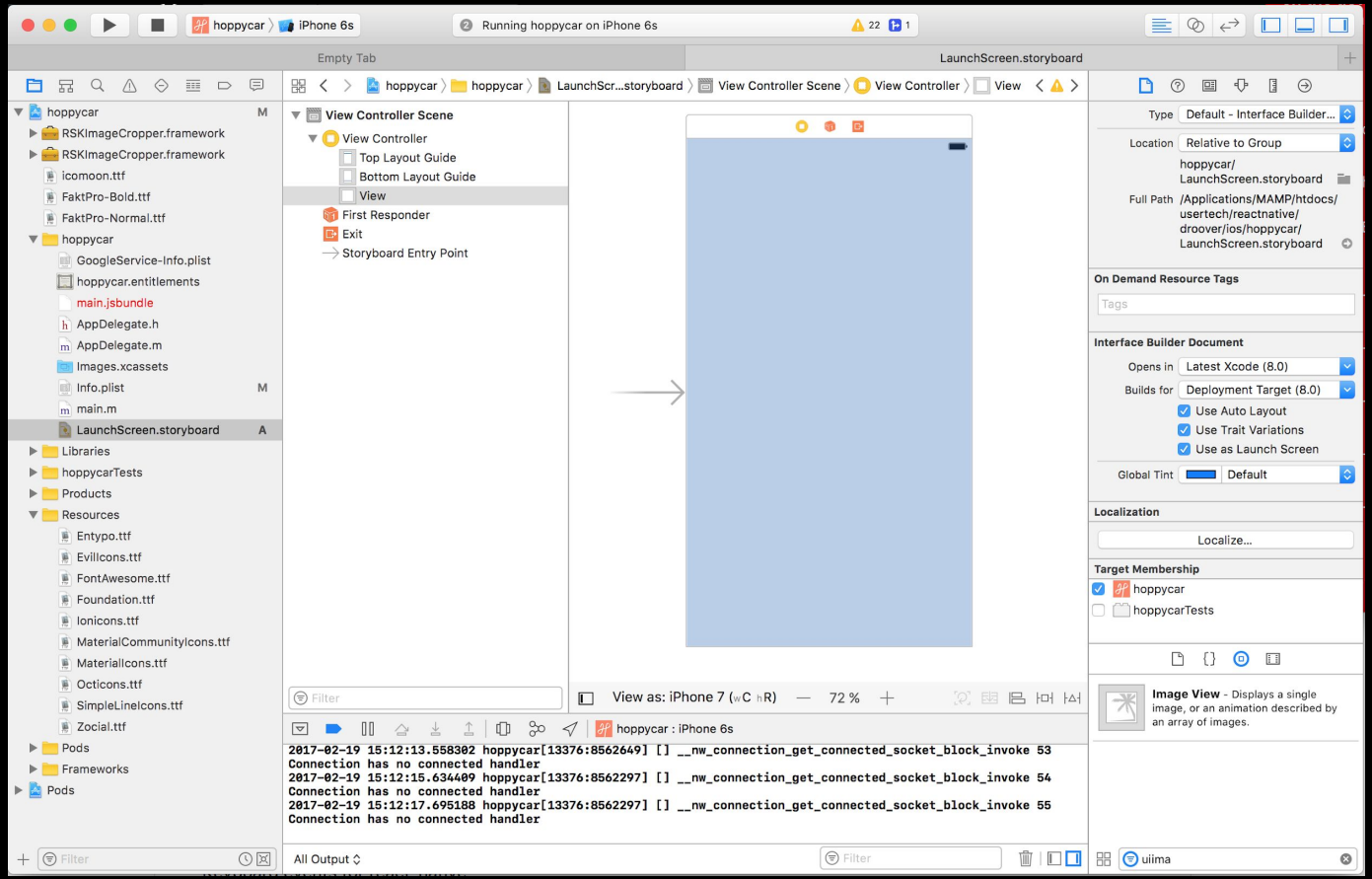
Create new Launch Screen file



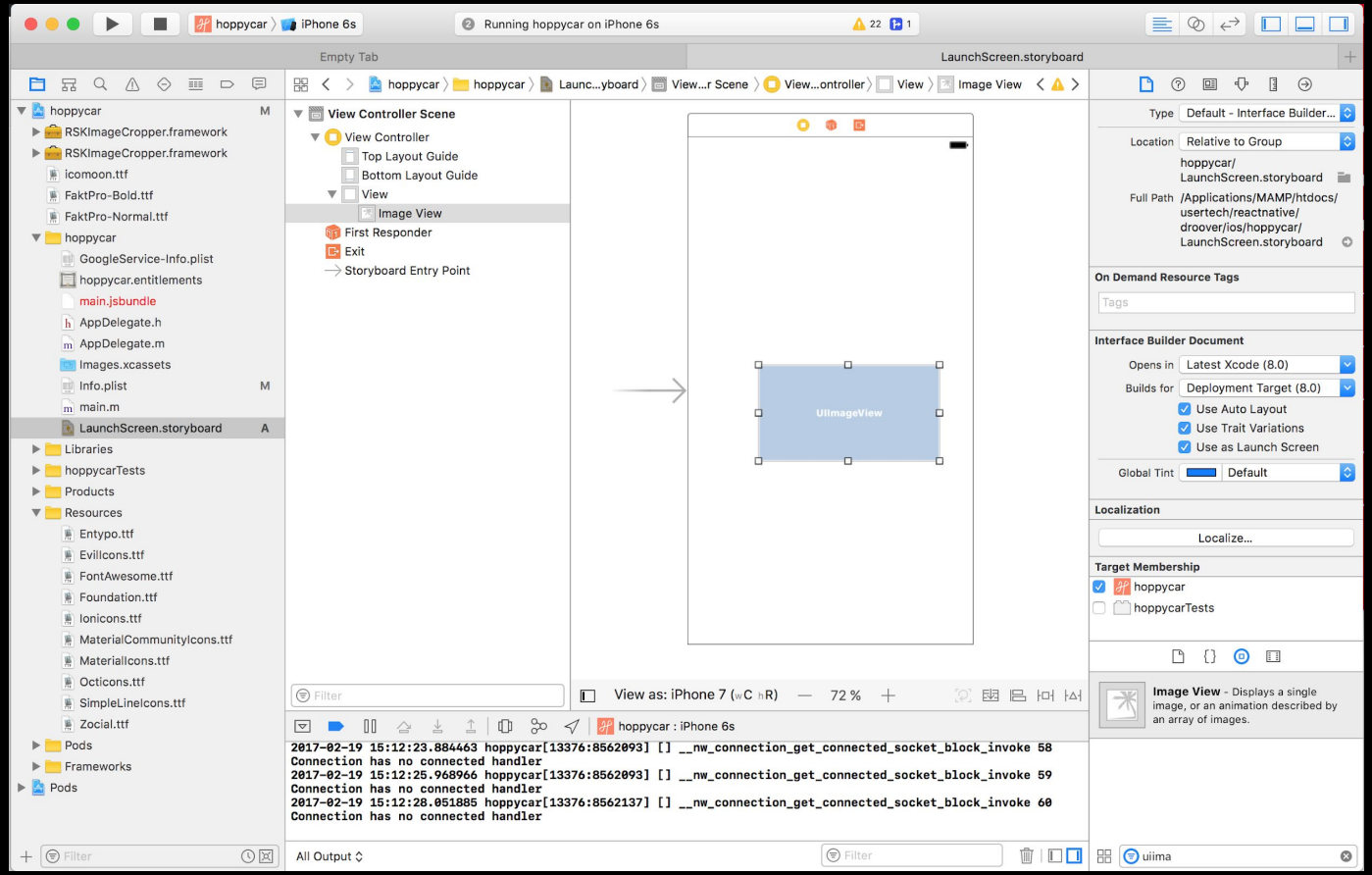
Choose storyboard as Launch screen



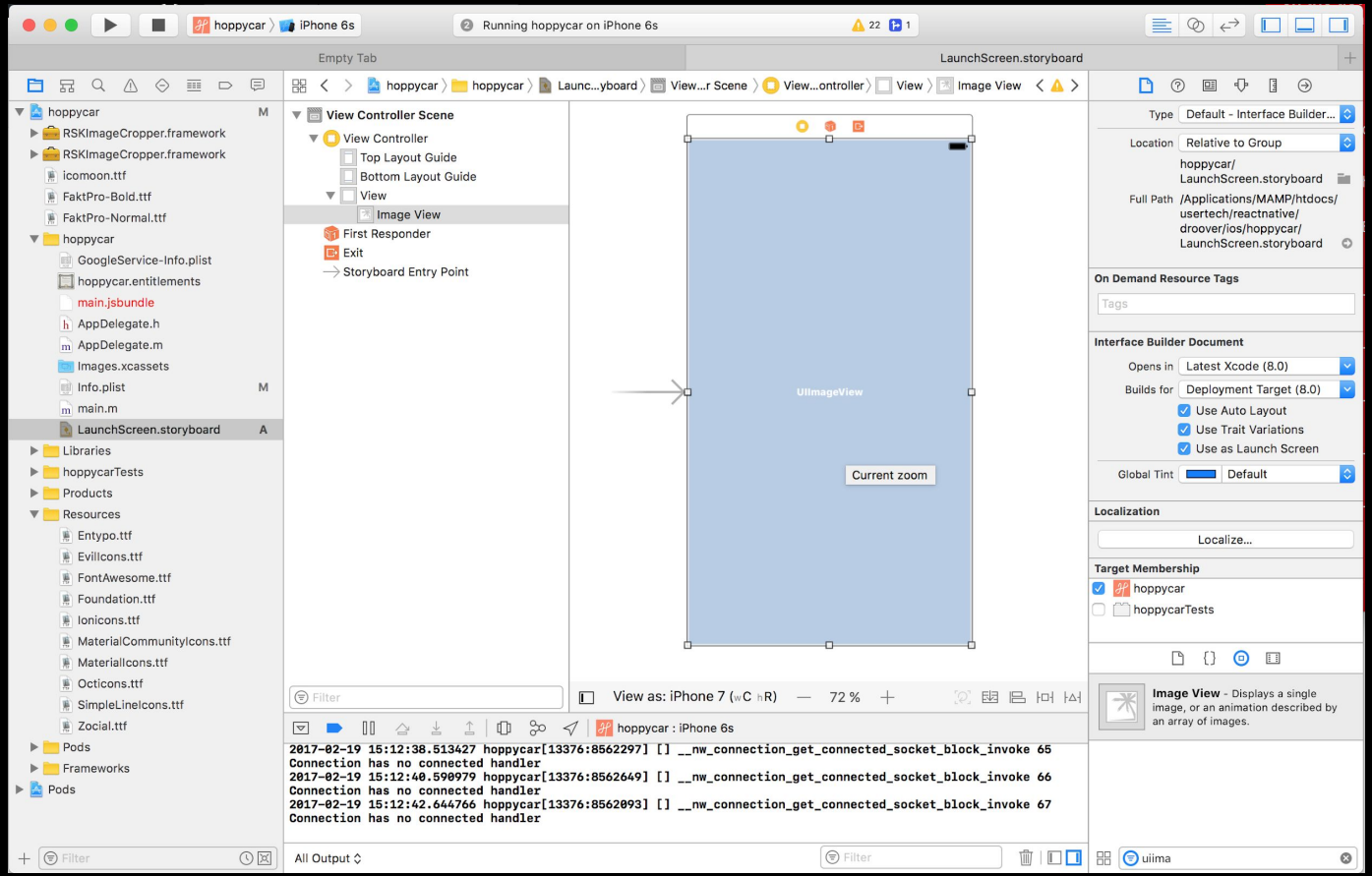
Storyboard


hoppycar
iPhone 6s
Running hoppycar on iPhone 6s
Empty Tab
LaunchScreen.storyboard
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First Responder
Exit
Storyboard Entry Point
Type
Default - Interface Builder...
Location
Relative to Group
Full Path
On Demand Resource Tags
Interface Builder Document
Opens in
Latest Xcode (8.0)
Builds for
Deployment Target (8.0)
Use Auto Layout
Use Trait Variations
Use as Launch Screen
Global Tint
Localization
Localize...
Target Membership
Image View - Displays a single image, or an animation described by an array of images.
2017-02-19 15:12:13.558302 hoppycar[13376:8562649] [] __nw_connection_get_connected_socket_block_invoke 53
Connection has no connected handler
2017-02-19 15:12:15.634409 hoppycar[13376:8562297] [] __nw_connection_get_connected_socket_block_invoke 54
Connection has no connected handler
2017-02-19 15:12:17.695188 hoppycar[13376:8562297] [] __nw_connection_get_connected_socket_block_invoke 55
Connection has no connected handler

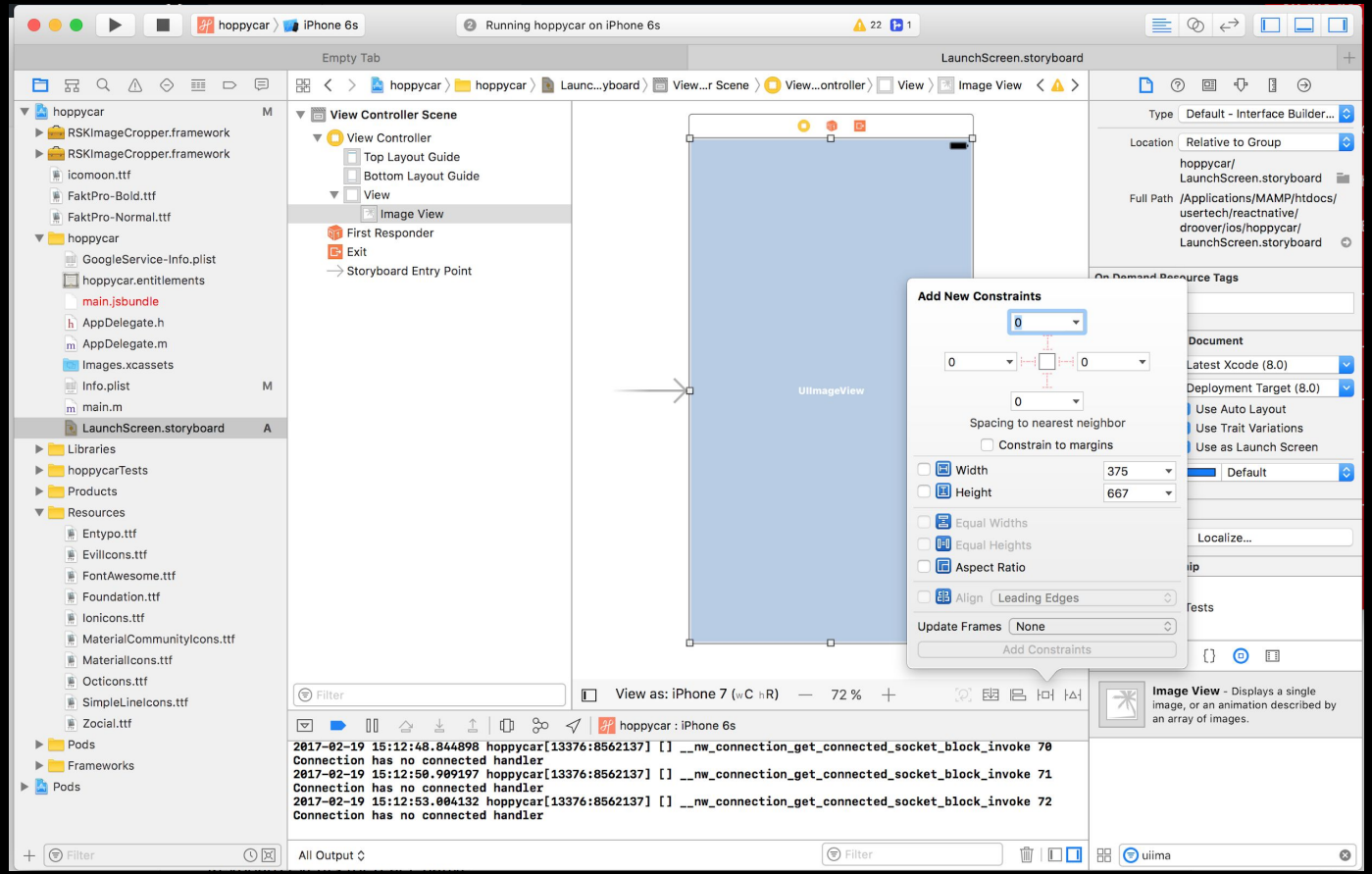
Add UIImageView



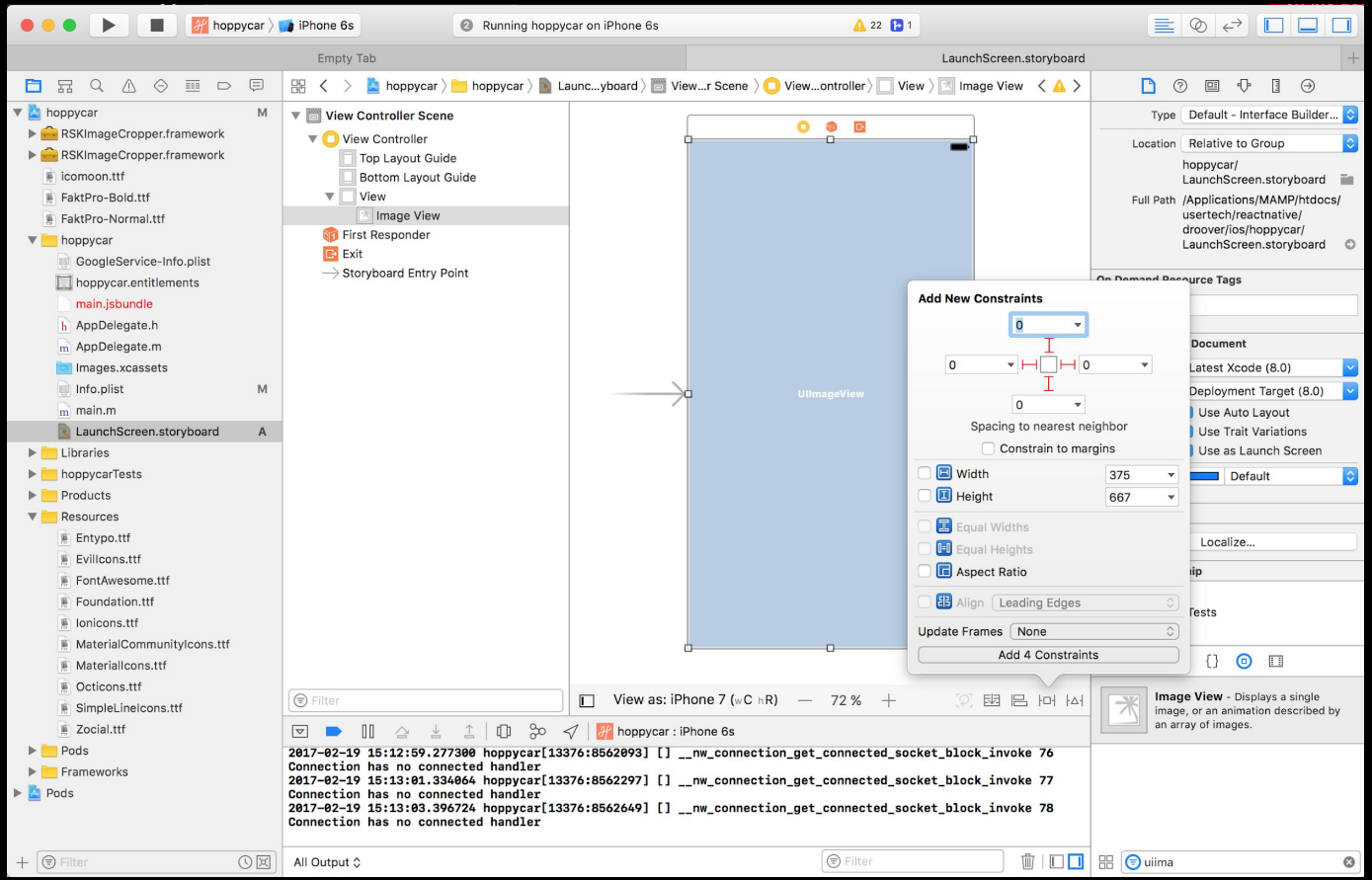
Resize UIImageView



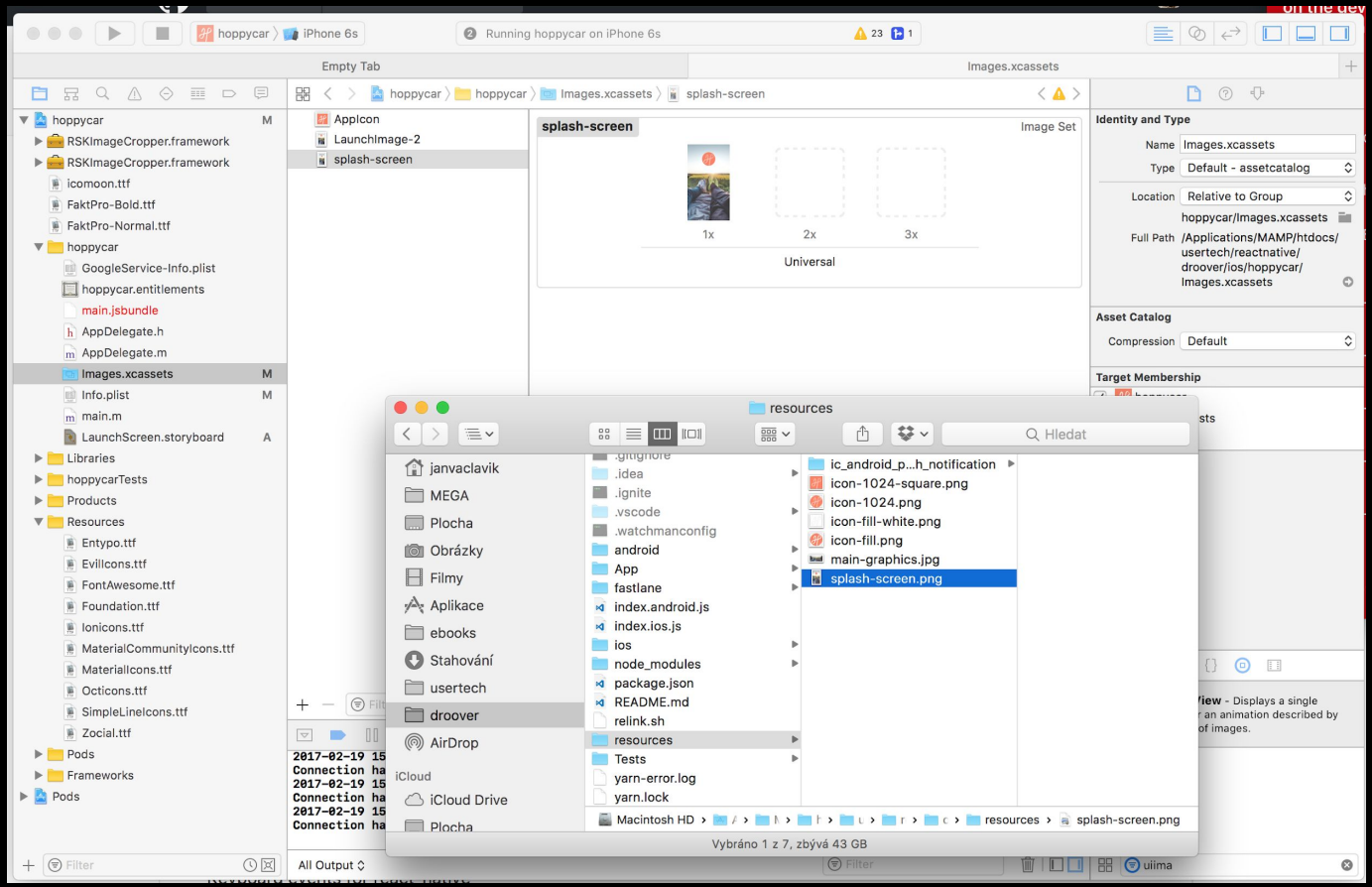
Add constraints



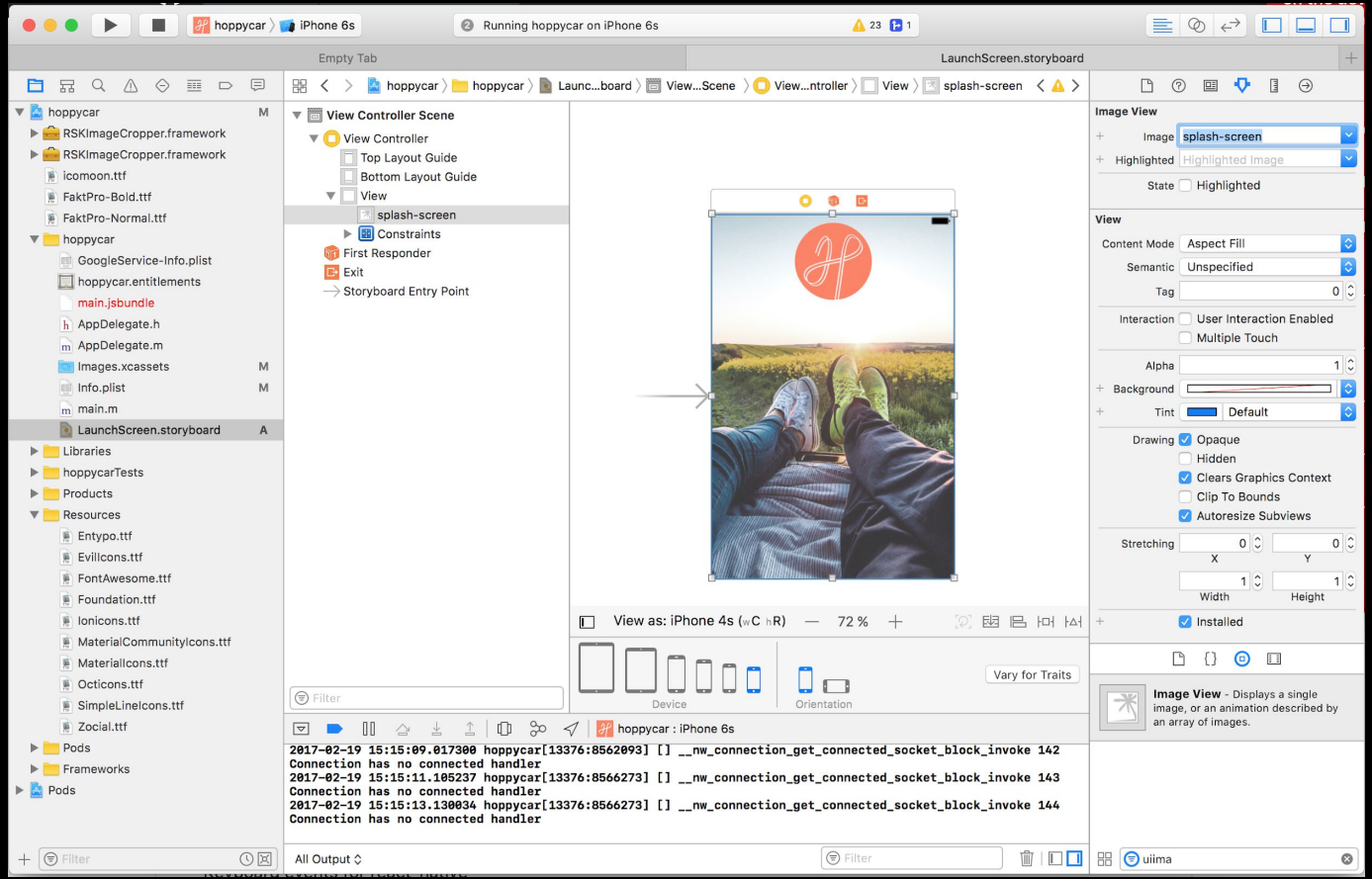
Set constraints for fullscreen image



Add image to asset catalog



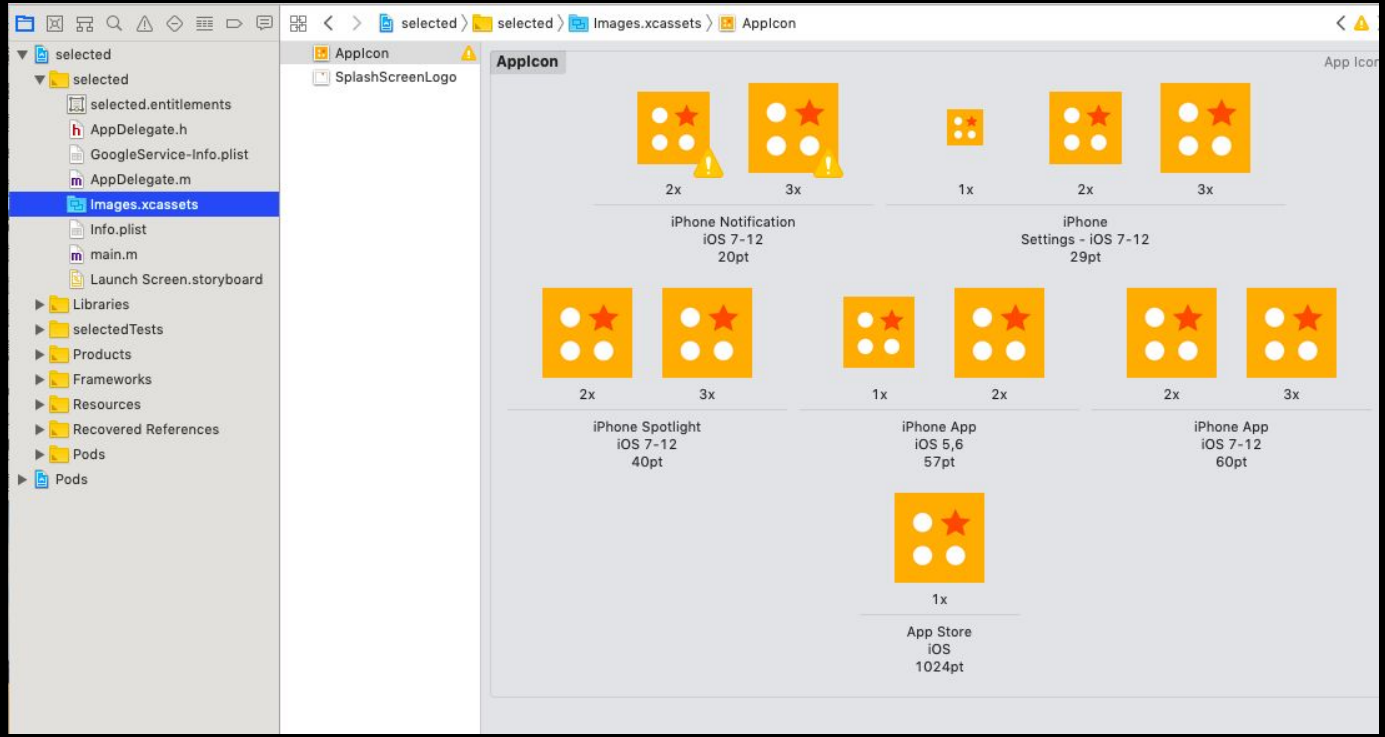
Set image


hoppycar
iPhone 6s
Running hoppycar on iPhone 6s
Empty Tab
LaunchScreen.storyboard
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
splash-screen
Constraints
First Responder
Exit
Storyboard Entry Point
Image View
Image
splash-screen
Highlighted
State
Highlighted
View
Content Mode
Aspect Fill
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip To Bounds
Autoresize Subviews
Stretching
Installed
Image View - Displays a single image, or an animation described by an array of images.
View as: iPhone 4s
72 %
Vary for Traits
Device
Orientation
2017-02-19 15:15:09.017300 hoppycar[13376:8562093] [] __nw_connection_get_connected_socket_block_invoke 142
Connection has no connected handler
2017-02-19 15:15:11.105237 hoppycar[13376:8566273] [] __nw_connection_get_connected_socket_block_invoke 143
Connection has no connected handler
2017-02-19 15:15:13.130034 hoppycar[13376:8566273] [] __nw_connection_get_connected_socket_block_invoke 144
Connection has no connected handler
All Output
uiima

Icons

- Icons are dummy
- You need many sizes of icon
- Use app icon generators
  - E. g. https://appicon.co

Icons


selected
selected
selected.entitlements
AppDelegate.h
GoogleService-Info.plist
AppDelegate.m
Images.xcassets
Info.plist
main.m
Launch Screen.storyboard
Libraries
selectedTests
Products
Frameworks
Resources
Recovered References
Pods
Pods
selected
selected
Images.xcassets
AppIcon
AppIcon
SplashScreenLogo
AppIcon
App Icon
2x
3x
iPhone Notification
iOS 7-12
20pt
1x
2x
3x
iPhone
Settings - iOS 7-12
29pt
2x
3x
iPhone Spotlight
iOS 7-12
40pt
1x
2x
iPhone App
iOS 5,6
57pt
2x
3x
iPhone App
iOS 7-12
60pt
1x
App Store
iOS
1024pt

Detox

Detox

- Automation framework for testing react native apps
- End to end testing, like a real user clicking through the app
- Also possible to run in cloud

Questions?

Sources

-