Things you should know about React Native

Neo @ JSConf China 2017

About Me

- Neo, frontend developer@autodesk
- Email: nihgwu@live.com
- Github: https://github.com/nihgwu
- Backend / Frontend / App
- Ionic -> React Native

whosyourdaddy

I’m your father

A Brief History

- Mar 27, 2015 - Initial public release
- Sep 15, 2015 - React Native for Android

16,777 downloads in the last day

111,129 downloads in the last week

477,962 downloads in the last month

1125 open issues on GitHub

254 open pull requests on GitHub

react-native

6,325 downloads in the last day

35,800 downloads in the last week

148,509 downloads in the last month

95 open issues on GitHub

4 open pull requests on GitHub

ionic-native

Props & Cons

- ~~Hot patch / Hot update~~
- Bring modern web techniques to mobile
- Cross platforms

Props & Cons

- ~~Breaking changes (0.26 / 0.29 / 0.40 / 0.47)~~
- Docs
- Navigation

Keep Improving

- Android performance
- Nodes
- FlatList / SectionList
- yoga, metro-bundler
- Road map, Monthly meeting

Native Animation

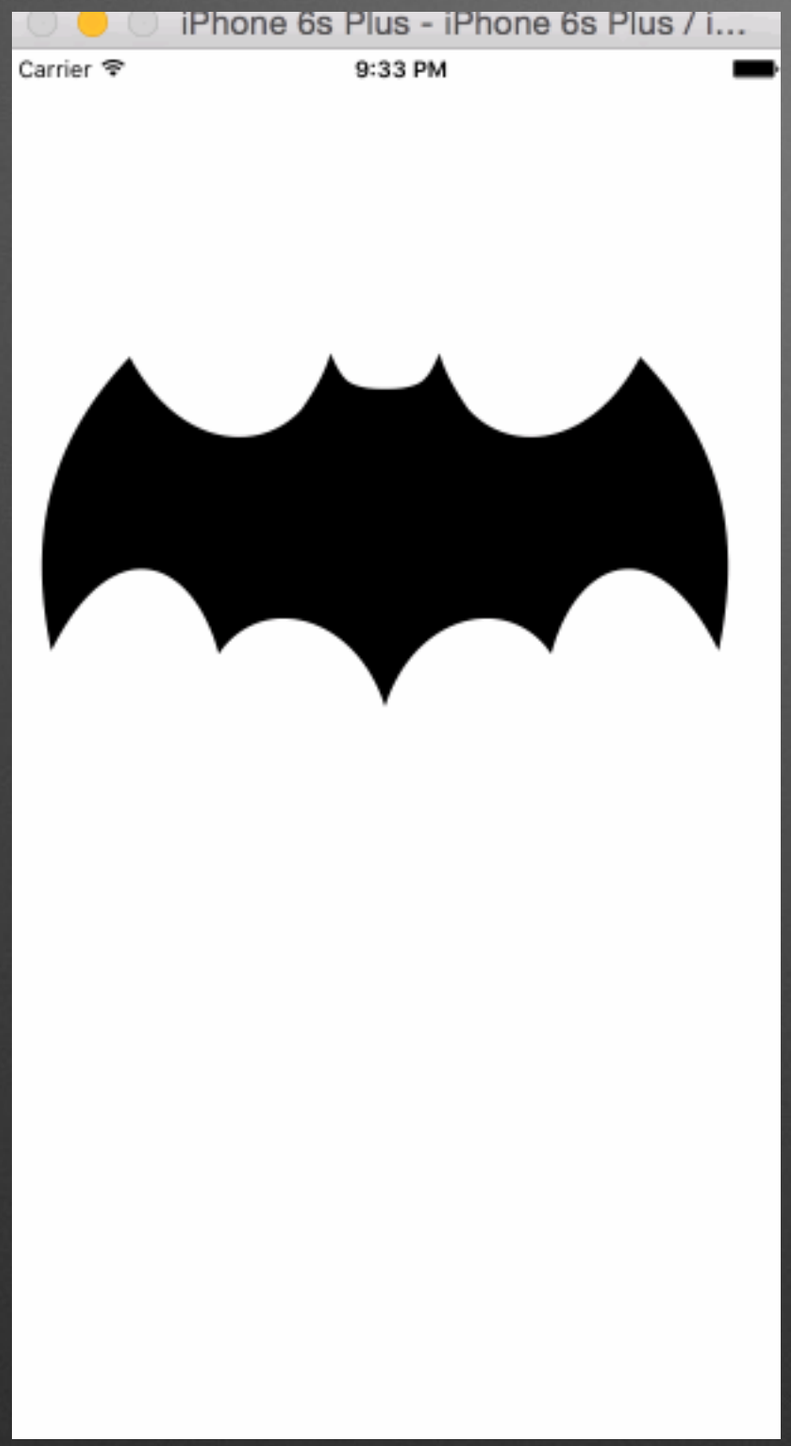
Animation in real world

Navigation

Parallax

Game

requestAnimationFrame + setState



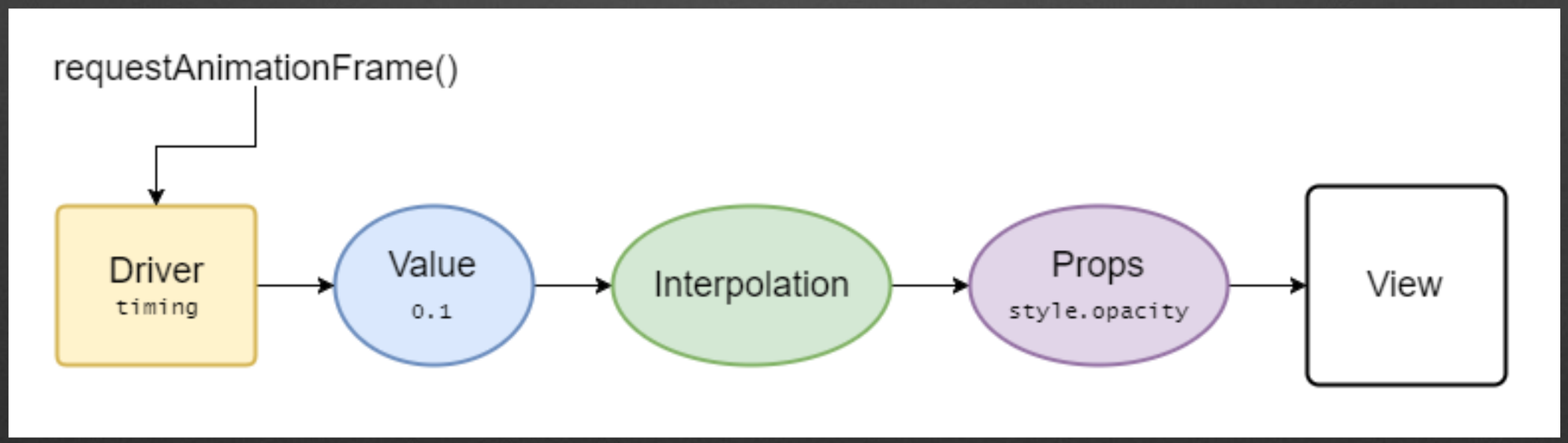

```
componentWillMount: function() {
  this._current = 1;
},
componentDidMount: function() {
  this.animate(null, this.nextAnimation)
},
nextAnimation: function() {
  this._current += 1;
  if (this._current >= BatmanLogoPaths.length) return;

  this.setState({
      transition:  Morph.Tween(BatmanLogoPaths[this._current - 1], BatmanLogoPaths[this._current])
  }, () => this.animate(null, this.nextAnimation))
},
animate: function(start, cb) {
  requestAnimationFrame((timestamp) => {
    if (!start) start = timestamp;
    var delta = (timestamp - start) / 1000;

    if (delta > 1) return cb();

    this.state.transition.tween(delta);
    this.setState(this.state);
    this.animate(start, cb);
  })
},
```

http://browniefed.com/blog/react-native-morphing-svg-paths-with-react-art/

Animated

requestAnimation + setNativeProps

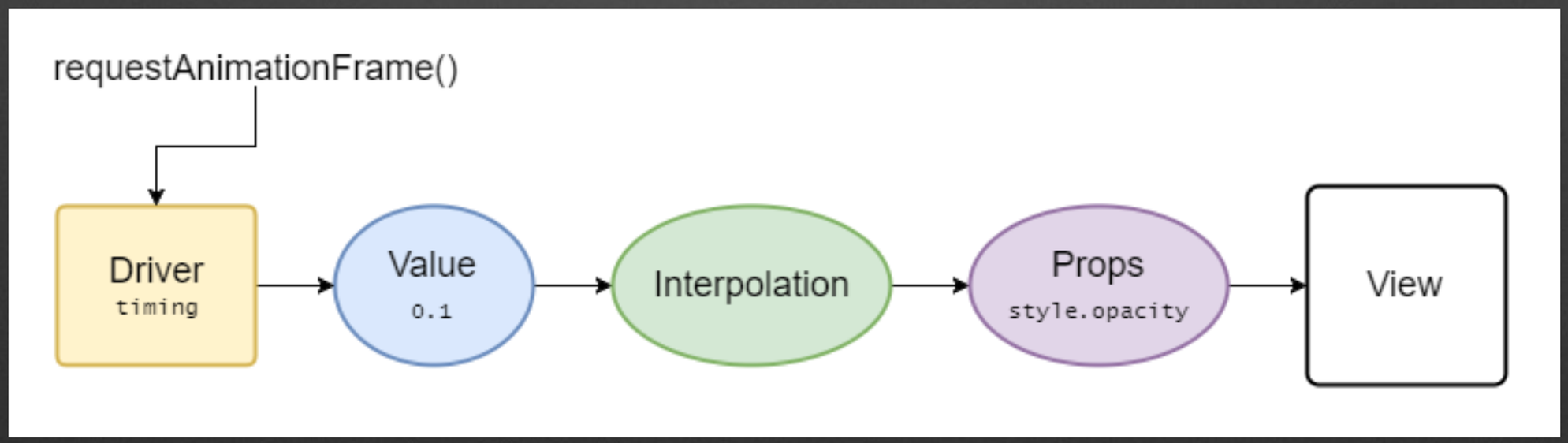
```
this.state.opacity.setValue(0)
Animated.timing(this.state.opacity, {
  toValue,
  duration: 1000,
}).start()
```



Native Animated

~~requestAnimation + setNativeProps~~

```
this.state.opacity.setValue(0)
Animated.timing(this.state.opacity, {
  toValue,
  duration: 1000,
}).start()
```



useNativeDriver: true

```
this.state.opacity.setValue(0)
Animated.timing(this.state.opacity, {
  toValue,
  duration: 1000,
}).start()
```

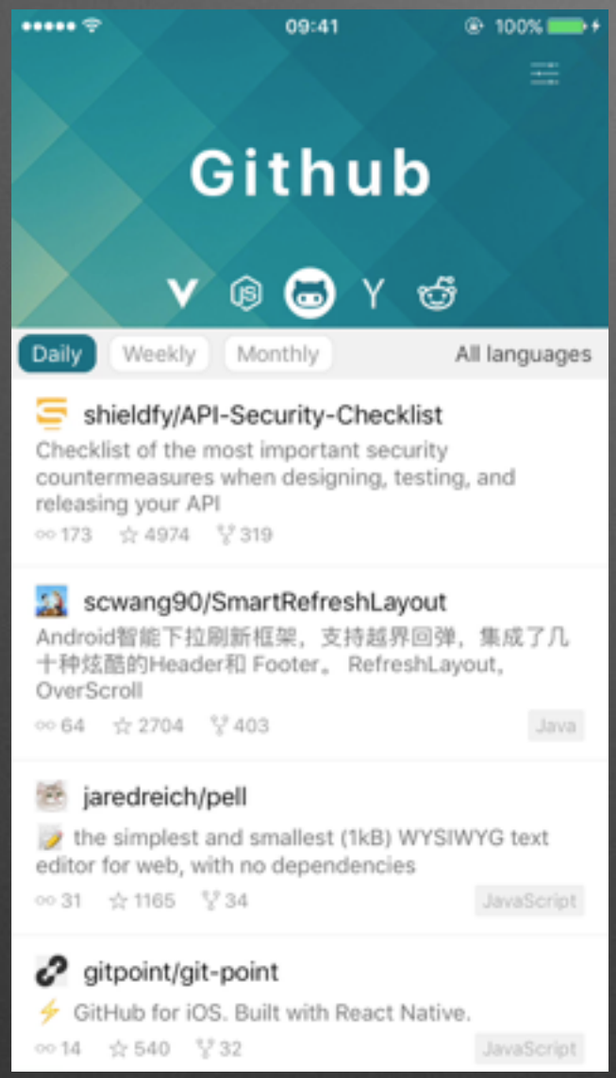
```
this.state.opacity.setValue(0)
Animated.timing(this.state.opacity, {
  toValue,
  duration: 1000,
  useNativeDriver: true,
}).start()
```

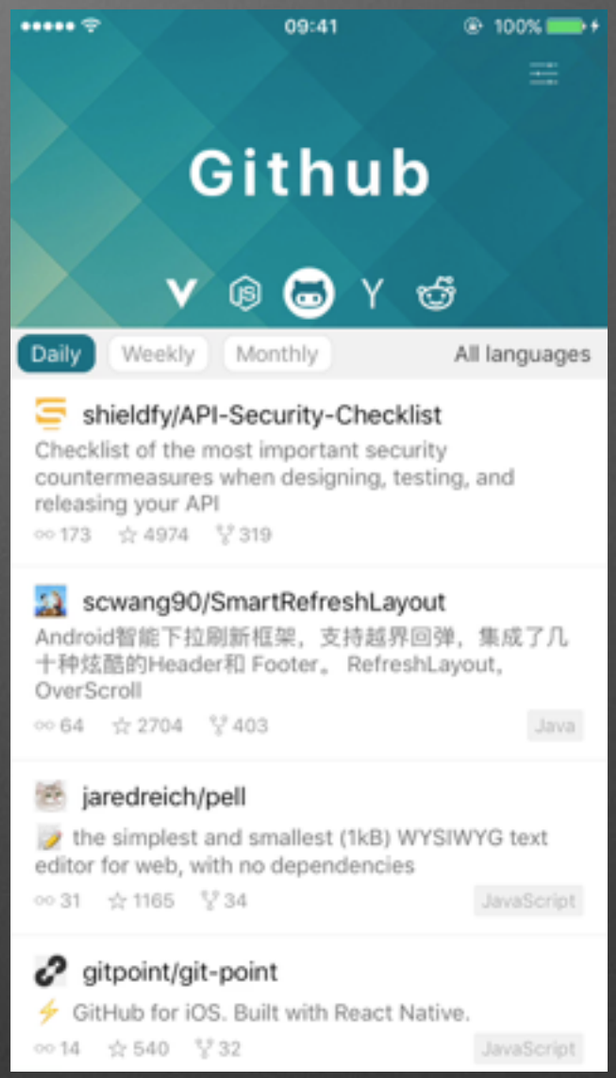
```
onScroll={Animated.event(
  [{ nativeEvent: { contentOffset: { y: this.state.scrollY } } }]
)}
```

```
onScroll={Animated.event(
  [{ nativeEvent: { contentOffset: { y: this.state.scrollY } } }],
  { useNativeDriver: true }
)}
```

https://facebook.github.io/react-native/blog/2017/02/14/using-native-driver-for-animated.html

Any differences?





Limitations

- Only support transform, opacity
- ~~flexbox, position properties~~

Recipes

- layout -> transform
- LayoutAnimation
- ~~runAfterInteractions~~

“Use Native Animated everywhere possible”

Navigation

- Navigator / react-native-custom-components
- NavigationExperimental
- react-navigation (747 issues, 64 prs)
- react-native-flux-router
- native-navigation
- react-native-navigation

Debug

- Chrome Dev Tools
- React Dev Tools
- React Native Debugger

Inspector

Autoselect instances

filter...

Commit

v2ex/fetchSuccess	+00:00.33
cnode/fetchSuccess	+00:00.17
toutiao/fetchSuccess	+00:00.13
github/updateTab	+00:00.15
reddit/fetchSuccess	+00:00.01
hackernews/fetchData	+00:00.20
app/fetchStart	+00:00.00
github/fetchSuccess	+00:00.86
hackernews/fetchSuccess	+00:00.11
app/fetchEnd	+00:00.36
hackernews/fetchSuccess	+00:00.02

Action

Action | State | Diff | Test

Tree | Chart | Raw

```
type (pin): "hackernews/fetchSuccess"
▼ payload (pin)
  ▶ data (pin): [{…}, {…}, {…}, {…}, …]
  ▶ list (pin): [14664104, 14660998, 14659775, 14662892, …]
    page (pin): 1
```

1x

Pause recording | Lock changes | Dispatcher | Slider | Import | Export | Print

Highlight Updates | Highlight Search

Connect

```
▼<Connect(HackerNewsList) key="hackernews" index=5 anim=AnimatedValue{…} refreshControl=null…>
 ▼<HackerNewsList index=5 anim=AnimatedValue{…} refreshControl=null…>
  ▼<List ref=ref() title="Hacker News" icon=6 index=5…>
   ▼<AnimatedComponent ref="listView" showsVerticalScrollIndicator=false automaticallyAdjustContentInsets=false scrollEventThrottle=
    ▼<ListView ref=bound _setComponentRef() showsVerticalScrollIndicator=false automaticallyAdjustContentInsets=false scrollEventTh
     ▼<ScrollView ref=bound _setScrollComponentRef() showsVerticalScrollIndicator=false automaticallyAdjustContentInsets=false scr
      ▼<RCTScrollView ref=bound _setScrollViewRef() showsVerticalScrollIndicator=false automaticallyAdjustContentInsets=false scro
       ▶<RCTScrollContentView ref=bound _setInnerViewRef() onLayout=bound _handleContentOnLayout() style=[null, null] removeClippe
       </RCTScrollView>
      </ScrollView>
     </ListView>
    </AnimatedComponent>
   </List>
  </HackerNewsList>
```

AppContainer View RCTView View RCTView Unknown Provider Connect(Router) Router NavigationContainer Navigator Unknown
CardStackTransitioner Transitioner View RCTView CardStack View RCTView View RCTView Container Card AnimatedComponent View
RCTView SceneView NavigationContainer Navigator Unknown CardStackTransitioner Transitioner View RCTView CardStack View RCTView

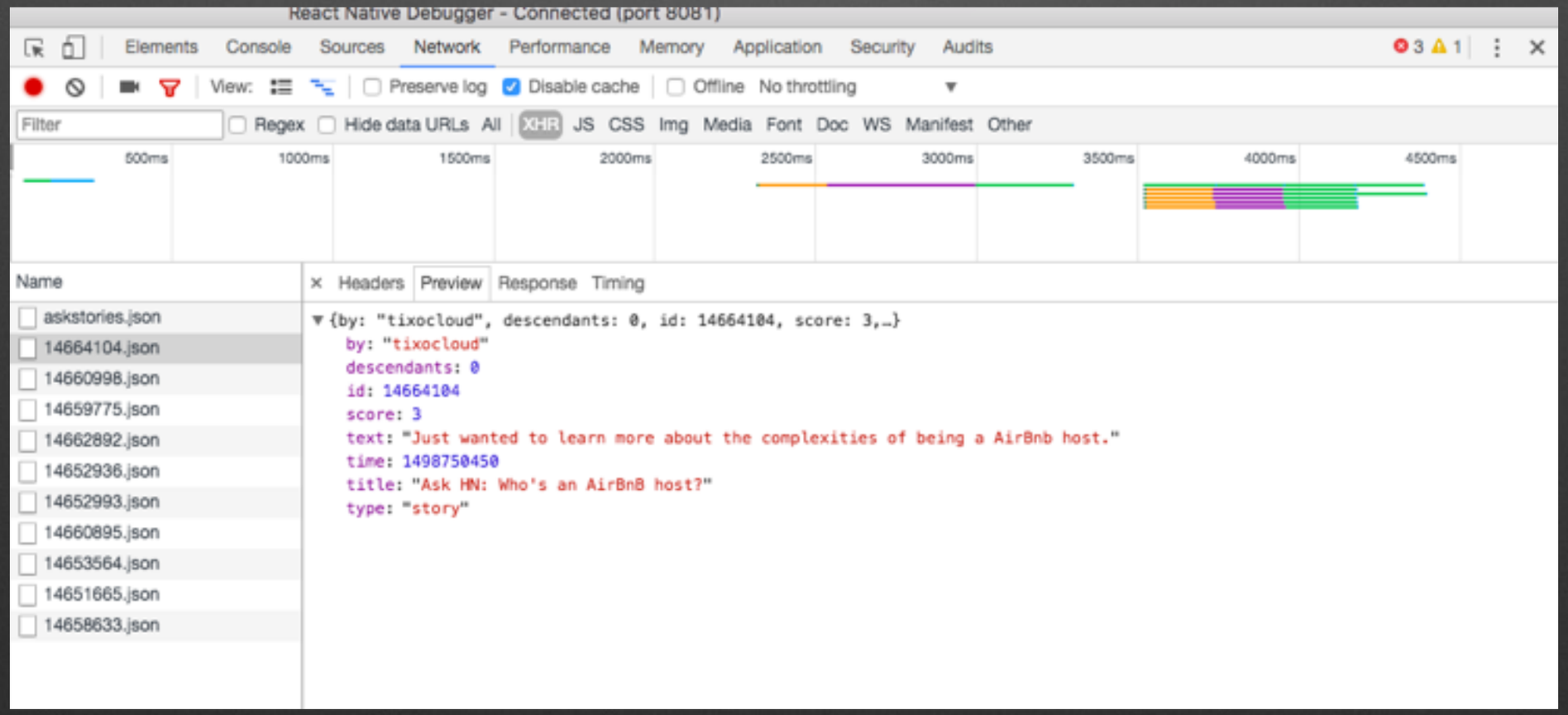
Ref bound _setScrollViewRef()

Props read-only

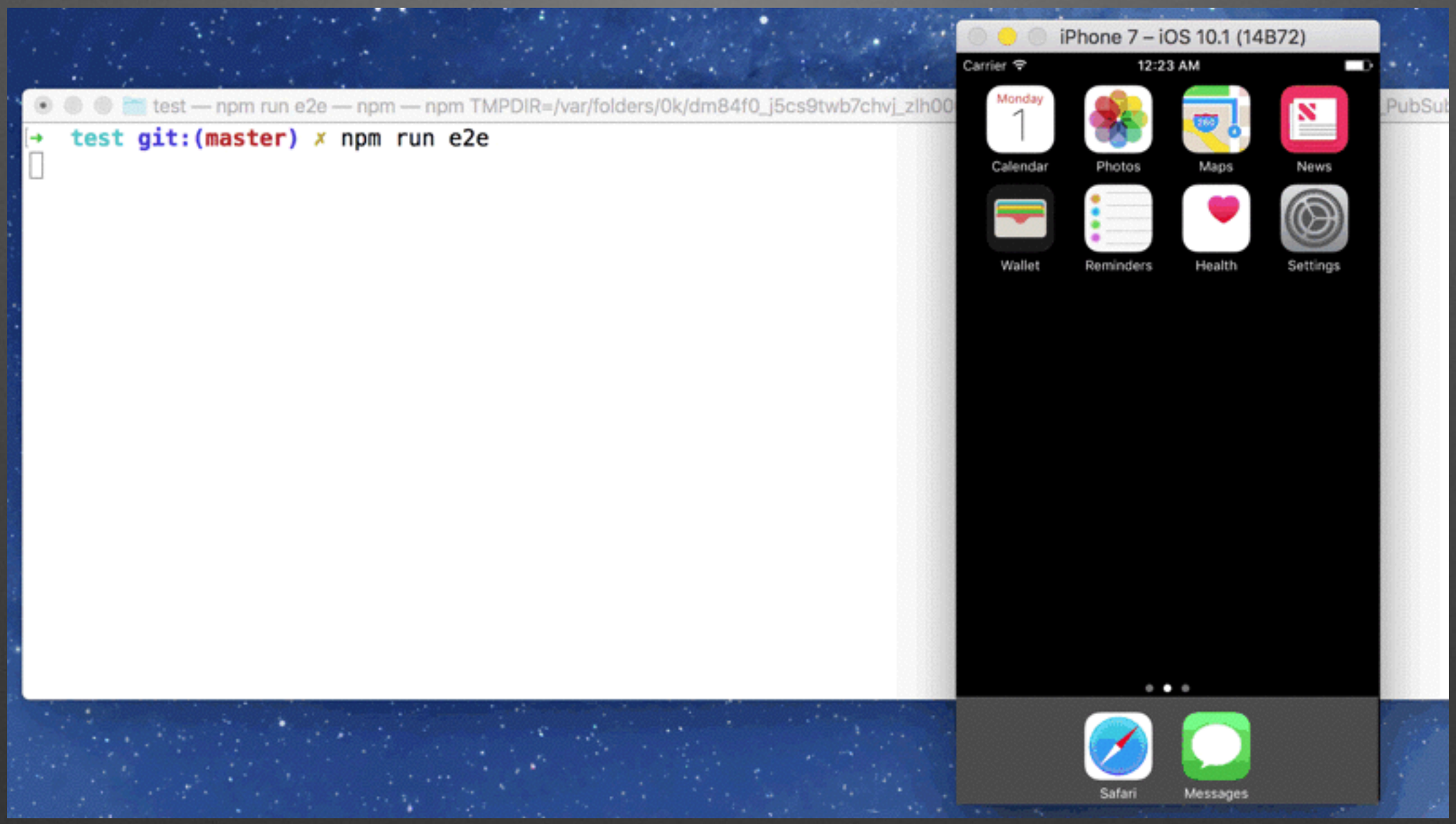
```
  alwaysBounceVertical: true
  automaticallyAdjustContentIns
▶ children: {…}
  collapsable: false
▶ contentInset: {…}
▶ data: Array[10]
▶ dataSource: ListViewDataSourc
▶ dispatch: fn()
  enableEmptySections: true
  fetching: false
  icon: 6
  index: 5
  initialListSize: 10
▶ list: Array[86]
```

Network Inspector

```
if (__DEV__) {
  global.XMLHttpRequest = global.originalXMLHttpRequest
    ? global.originalXMLHttpRequest
    : global.XMLHttpRequest
  global.FormData = global.originalFormData
    ? global.originalFormData
    : global.FormData
}
```



Visual Testing



https://github.com/wix/detox

State Management

- Built-in state
- Redux
- Mobx
- Mobx State Tree
- dva

Tips

- Always use the latest version
- Don’t use any ui frameworks
- F8app is not for beginners

Open source Apps

- Guoqi https://github.com/yuche/gouqi
- Psnine https://github.com/smallpath/psnine
- iShiWuPai https://github.com/ljunb/react-native-iShiWuPai

Resources

- Official site
- Source code
- JS.Coach
- Awesome list

Thanks