React Native – An Introduction

08.08.2017

CocoaHeads LE

Motivation

- creating an app for different platforms with native development toolchains does not scale
- native UI libraries for iOS & Android are not that declarative
- building custom UI components should be smooth

Semi live coding

Building blocks

JSX

- a syntax for embedding XML within JavaScript
- lets you write your markup language inside code
- e.g. `<Text>` represents a React component

Components

- describe what should be rendered by providing a `render` function
- have `props` and `state`

Props

- parameters used for customization on component creation
- set by the parent
- fixed throughout the lifetime of a component
- e.g.: `Image` has a property `source`

State

- used for data that is going to change throughout the component lifetime
- initialize `state` in the constructor, call `setState` for changes
- using `setState` re-renders the component

Style

- application is styled using JavaScript
- core components accept a prop named `style`
- pass an array of styles - last style in the array has precedence (style inheritance)

Conclusion

The Good

- model domain logic only once
- cross platform development with native GUI
- declarative views
- live reload programming
- GUI component based thinking from the ground up

The Good

- easy to read/review GUI code
- Flex Box instead of Auto Layout (subjective)
- strong backing by Facebook
- large (largest?) developer community on the planet
- well documented

The Bad

- weakly typed JavaScript (if you prefer strongly typed languages)
  - What about Flow?
- Performance issues? Start up time?
  - no experience with that until now

The Bad

- some custom components must be written with native languages still
  - unsure which ones
  - learning another technology stack
  - another layer of technology that can fail

Links

https://facebook.github.io/react-native

thanks 🤗