

Front End III

Por que Hooks, se com as classes estávamos bem?

Um breve contexto

Como já sabemos, as tecnologias da Internet vêm se desenvolvendo exponencialmente nos últimos anos, por conta do incontrolável sucesso das FAANG¹. Por exemplo, o caso do Facebook, que passou de —na sua fase inicial (2003)— um website chamado FaceMatch (criado pelo Mark Zuckerberg para o ambiente da Universidade de Harvard, no qual rostos universitários eram qualificados) para, na atualidade (após quase 20 anos), uma das maiores corporações de alcance quase global: proprietária do Instagram, Whatsapp, Oculus, criadora da Internet.org, entre outras milhares de iniciativas. O avanço constante dessas corporações ocorre no contexto de uma concorrência **constante para a criação de uma Internet que atenda às necessidades** e defina as formas de comunicação mais usadas no Ocidente.

É esse o contexto no qual o React surge, assim como o Angular (Google): é a busca pelo controle do rumo da internet e, com isso, por conta da necessidade implícita de **aprimoramento e otimização permanente das tecnologias da Internet**, a necessidade de melhoria contínua em termos desses frameworks/bibliotecas.

Mas como se chegou aos Hooks?

¹ FAANG faz referência às cinco empresas de tecnologia norte-americanas proeminentes: Facebook, Amazon, Apple, Netflix e Alphabet (GOOG). Em: https://es.wikipedia.org/wiki/Gigantes_tecnol%C3%B3gicos#FAANG.

O lançamento do React da mão do Facebook, por volta de 2013, teve um grande sucesso. O seu enfoque baseado em uma arquitetura de dados fluindo para baixo, composição em componentes e uma atualização do DOM de forma declarativa super rápida (por conta do uso do Virtual DOM), tornou-se viral no mundo front-end. Foi uma estratégia bem-sucedida, e continua sendo.

Problemas

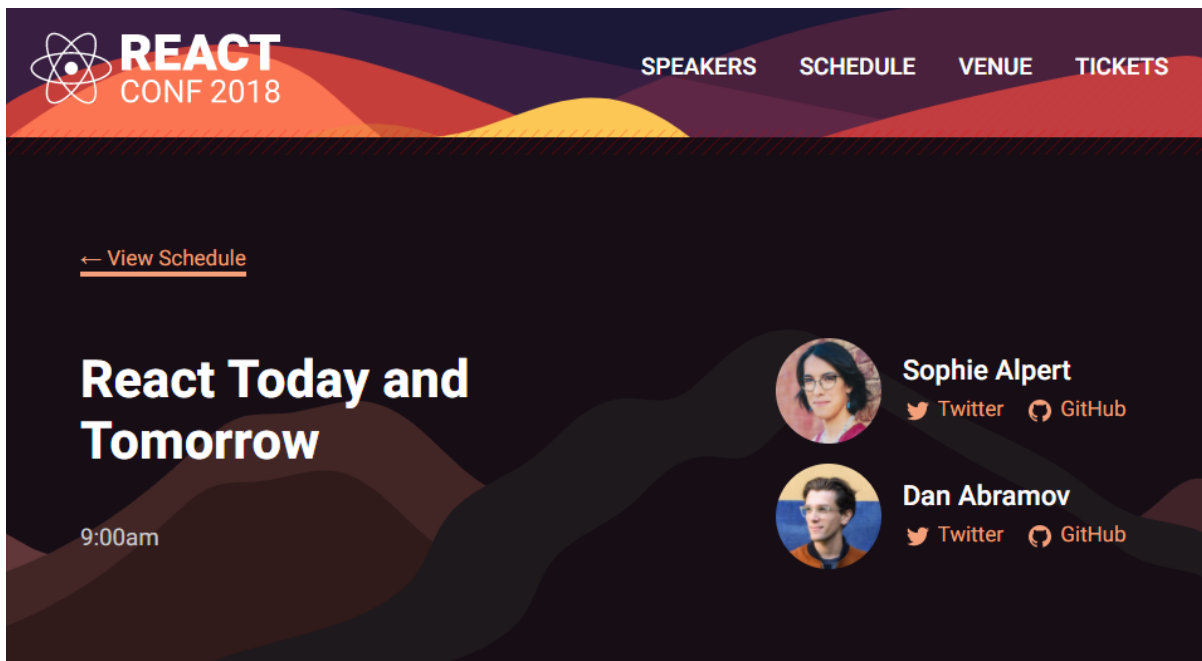
Apesar disso, a história do React não esteve isenta de problemas e adequações, sobretudo a respeito da implementação dos seus componentes, sua parte mestre. Quando o React foi lançado para a comunidade, JavaScript ainda não dispunha de um sistema de classes, portanto o React optou por criar a API **React.createClassAPI** para criação de componentes. Foi um caminho muito eficiente.

Contudo, a felicidade durou pouco. Com o lançamento da versão 6 do ECMAScript, em 2015, JavaScript introduz oficialmente a palavra reservada `Class` e, diante do dilema de seguir seu próprio caminho ou se adaptar ao ECMAScript para não perder seguidores, **o React optou por adotar classes JavaScript.**

Ainda assim, as particularidades do funcionamento das classes em JavaScript geraram novos problemas, pois esse novo caminho exigiu a criação de subclasses estendidas do `React.Component`, a declaração do estado dentro de um construtor, o uso de `this`, `bind` e `super(props)`. De qualquer maneira, poderíamos dizer que, embora seja um pouco entediante, o uso de `bind` e `super(props)` não eram problemas em si, mas ninguém gosta de programar durante anos com certo desconforto.

Os Hooks nascem

Ao reconhecer o tédio na comunidade front-end, o React trabalhou na procura de soluções. No mês de outubro de 2018 (na React Conf, realizada em Henderson, Nevada, EUA), Sophie Alpert e DAN Abramov, integrantes da equipe do Facebook responsável pelo desenvolvimento do React (naquele momento) apresentaram uma nova característica revolucionária para o mundo do front-end: React Hooks. Mais tarde, em fevereiro de 2019, essa proposta se tornou realidade e o React liberou a sua versão 16.8, incluindo a API de Hooks.



E isso é só o começo!