Javascript Modules

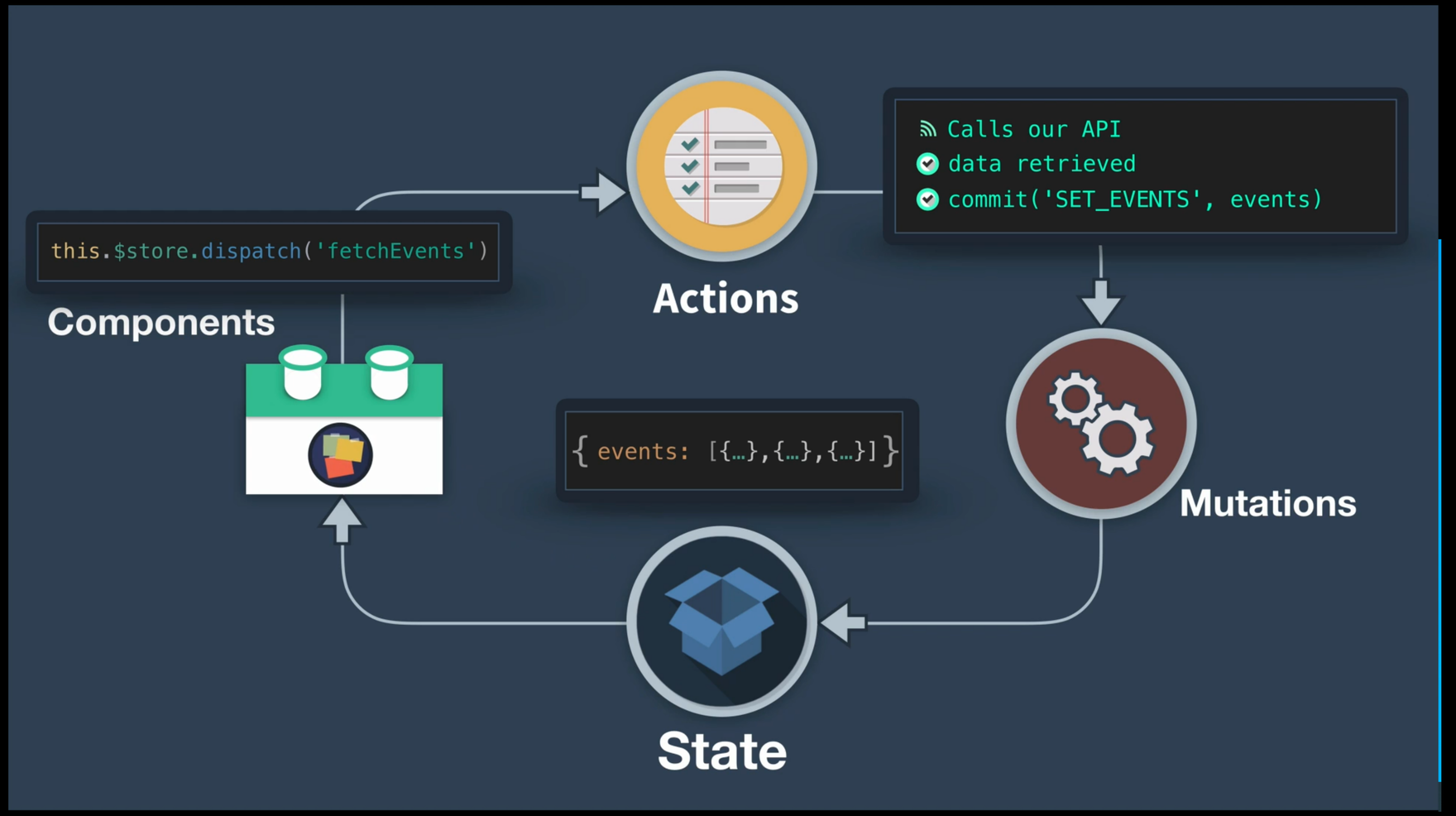
Javascript History & Modules

Created by Aleksander Apostolidis
PHP Fullstack course, Code Academy 2020

Event List

Vuex

- Event List (dispatch)
- Action (API Call)
- Mutation
- State
- Event List (update)



A (VERY) SHORT HISTORY OF JAVASCRIPT

- **1996:** Changed from LiveScript to JavaScript to attract Java developers. **JavaScript has almost nothing to do with Java** ☝️
- **1997:** ES1 (ECMAScript 1) became the first version of the JavaScript language standard:
  - ECMAScript: The language standard;
  - JavaScript: The language in practice.
- **2009:** ES5 (ECMAScript 5) was released with lots of new features.
- **2015:** ES6/ES2015 (ECMAScript 2015) was released: **the biggest update to the language ever!**
- **2015:** Changed to an **annual release cycle** 🙏

JAVASCRIPT TODAY: WHICH VERSION TO USE?

ES5

- Fully supported in all browsers;
- Ready to be used today 👍

http://kangax.github.io/compat-table

ES6/ES2015

ES7/ES2016

ES8/ES2017

- Well supported in all **modern** browsers
- No support in older browsers;
- Can use **most** features in production with transpiling and polyfilling (converting to ES5) 😀

ES9/ES2018

ES10/ES2019

- Future versions, together called ESNext;
- Some features supported in modern browsers;
- Can already use **some** features in production with transpiling and polyfilling 😊

(As of mid-2018)

Javascript today
ECMAScript 2020 - ES2020

- Code bundling or bundle - the main script file which can include other dependencies or modules. They don’t need to be loaded individually any more the main file can load. Bundling often means an increase in file size.
- Tree shaking - When a muddle is included the whole package gets added to the bundle, even if only part of it is used. Javascript tools can eliminate this not used “Dead code” and the process is often called “Tree shaking”. For example Webpack can make “tree shaking” - it can shake the dead code out of the “tree”, or bundle.
- Transpiling - subset of compiling where the source code of one language is converted into other language or in different version of same language
- Polyfilling is a way to include functionality which is not present natively (currently)
- Tools: Babel, Webpack, Browserify and others

Module

A reusable block of code whose existence does not accidentally impact other code .

Modules before ES2015

Before ES2015 release, not supported modules, so there were used third-part solutions. There were at least 3 major modules competing standards:

- Asynchronous Module Definition (AMD)
- RequireJS Modules
- CommonJS Modules

ES6 Modules (ESM)

- Everything inside the an ES6 module is private by default, and runs in strict mode.
- Public variables, functions and classes are exposed using export
- Exposed modules are called into other modules using import
- Modules are deferred, and only run after a document is loaded
- There are named and default exports
- Default exports are one per module