**1. Redux**

The main problem that Redux tries to solve is that data changes are triggered by different parts of an app, and are directed at different parts of an app. As such, controlling this data flow and ensuring that all data is being updated at the right time and in the right places that don’t disrupt the user experience is hard.

Redux is a Javascript library that can be used with frameworks such as React, and is used to create a central data store that contains the “state” of an application. This “state” is the source of truth to keep components (e.g. the Model, View, Controller) synchronized, as opposed to synchronizing each component separately.

[Descriptive link](https://medium.com/codingthesmartway-com-blog/learn-redux-introduction-to-state-management-with-react-b87bc570b12a)

**2. REST API**

REST APIs are a type of API commonly used to extract data from servers. Broadly speaking, you make a “call” (or request) to an API with certain parameters and expect to receive a response that consists of some type of data in a certain shape.

[Descriptive link](https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/)

**3. Twitter Bootstrap**

Bootstrap is an open source framework for front-end development of websites and apps using HTML, CSS and Javascript. One of its primary use cases is to build mobile websites that can auto-scale based on the device it’s being used on (aka responsive), though can be used for desktop applications as well.

**4. Express**

Express is a Node.js framework to help organize an app into the MVC architecture on the server side. It allows devs to create things like REST APIs much more quickly than if they were simply using core js modules.

[Descriptive link](https://stackoverflow.com/questions/12616153/what-is-express-js)

**5. Behavior-driven development (BDD)**

An evolution of TDD. In general, both TDD and BDD are processes for writing and running tests that evaluate the functionality of your code. The general philosophy of TDD is that you should first write a test, and then write code that passes that test.

BDD expands upon TDD by defining how to write more useful tests (vs. TDD is more concerned with WHEN you write tests and less on how).

[Descriptive link](https://codeutopia.net/blog/2015/03/01/unit-testing-tdd-and-bdd/)

**6. Webpack / Gulp / Grunt**

These are “task runners,” which is to say that they perform specific functionality to, for example, optimize images, running unit tests, or refreshing a browser when a file is saved. These tasks are common repetitive tasks that devs need to do as part of their workflow, and which Webpack / Gulp / Grunt help to automate.

**7. Mocha**

Mocha is a JS test framework for Node that is commonly used in TDD. Mocha is a tool often compared to Jest, Jasmine, Karma, etc.

**8. Middleware**

Middleware is software that makes it easier for applications to interact with each other, often by doing some type of operation on some type of data. Examples of this include making it easy to request and send data, combining data sources, or providing access to a service as though it were local. An example of Middleware would be an API.

**9. Apache Kafka**

Kafka allows you to store large amounts of data in a way so that it can be both read/written quickly, and be stored in a reliable way to reduce the impact of downtime on a given “node.”

[Descriptive link](https://hackernoon.com/thorough-introduction-to-apache-kafka-6fbf2989bbc1)

**10. Angular.js / backbone.js / ember.js**

These are front-end app frameworks used primarily for front-end development. Like most other frameworks, they are geared toward simplifying the process of creating applications using some derivative of the MVC framework. Backbone favors the Model-View-Presenter design paradigm, Ember.js favors the Model-View-Viewmodel, while Angular is known as MVW (model-view-whatever)