Flux is an architecture for unidirectional data flow.

Can be embedded in JS framework.

Used by redux library.

Redux is a ***state management library*** for JavaScript apps.

Redux can be used react, angular, vue and vanilla js.

Redux is a**predictable state container for JavaScript applications**.

Redux derives its ideas form the Flux architecture. It is basically a flux-like approach to React applications.

Redux doesn’t necessarily have to be used with React only, you can use it for AngularJS or JS too. But Redux works well with React.

To change the state of an app. A new object is created.

Single Immutable object which stores all the state of an application.

Like a client-side database

Diagram

Description automatically generated

Redux is a pattern and library for managing and updating application state using events called actions. It serves as centralized store for state that needs to be used across your entire application, with rules ensuring that the state can only be updated in a predictable fashion.

REDUX main topics are

**ACTION (what to do)**

Action are plain JavaScript object that have a type field.

**REDUCER (how to do)**

Reducer are functions that take the current state and an action as arguments, and return a new state result.

**STORE (object which holds the state of the application)**

The Redux store brings together the state, actions and reducers that make up you app.

It’s important to note that you’ll only have a single store in a redux application.

Every Redux store has a single root reducer function.

**FUNCTIONS associated with store (createStore(), dispatch(action), getState())**

REDUX Principle

**Single Source of Truth**- The global state of your application is stored as an object inside a single store.

**State is read-only-** The only way to change the state is to dispatch an action

**Immutability, one-way data flow, Predictability of outcome**

**Changes are made with pure Reducer Functions**