What is flux

Flux is a JavaScript architecture or pattern for UI which runs on a unidirectional data flow and has a centralized dispatcher. It is useful when your project has dynamic data and you need to keep the data updated in an effective manner. It was created by Facebook, and complements React as view. This model is used to ease maintenance. It has three primary components: Views, Stores, and Dispatcher. As the MVC application grows, we find many numbers of views as models, which are all talking to each other, making it difficult to manage and debug.

Action: Action describes user interaction that occurred in a component, such as a user clicking a button.

Dispatcher: Dispatcher methods are invoked by an action component. This emits an event with data that needs to go to a store, which is a singleton registry.

Store: The store listens to certain events from the dispatcher. When it listens to an event from the dispatcher, it will then modify its internal data and emit a different event for views.

View: Views are typically a react component. Views get data from the store and set up a listener to refresh itself when the store emits any changed events.

Advantage of Flux:

It is a unidirectional data flow model which is easy to understand.

It is open source and more of a design pattern than a formal framework like MVC architecture.

The flux application is easier to maintain.

The flux application parts are decoupled.

MVC Vs. Flux

|  |  |  |
| --- | --- | --- |
| SN | MVC | FLUX |
| 1. | It was introduced in 1976. | It was introduced just a few years ago. |
| 2. | It supports Bi-directional data Flow model. | It supports Uni-directional data flow model. |
| 3. | In this, data binding is the key. | In this, events or actions are the keys. |
| 4. | It is synchronous. | It is asynchronous. |
| 5. | Here, controllers handle everything(logic). | Here, stores handle all logic. |
| 6. | It is hard to debug. | It is easy to debug because it has common initiating point: Dispatcher. |
| 7. | It is difficult to understand as the project size increases. | It is easy to understand. |
| 8. | Its maintainability is difficult as the project scope goes huge. | Its maintainability is easy and reduces runtime errors. |
| 9. | Testing of application is difficult. | Testing of application is easy. |
| 10. | Scalability is complex. | It can be easily scalable. |