

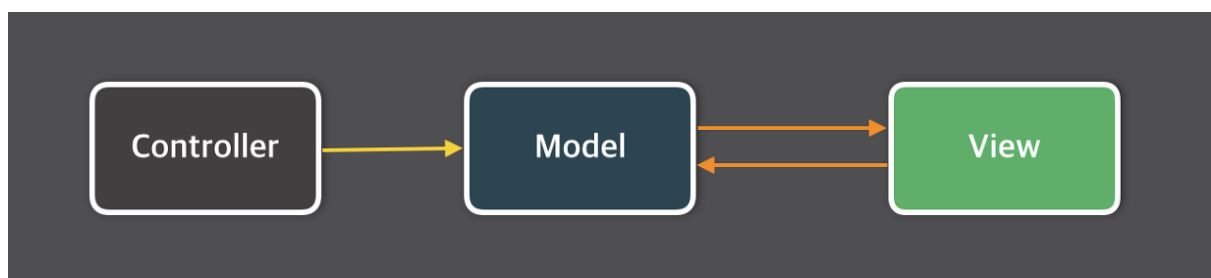
Flux Pattern

/* Flux Pattern */

React

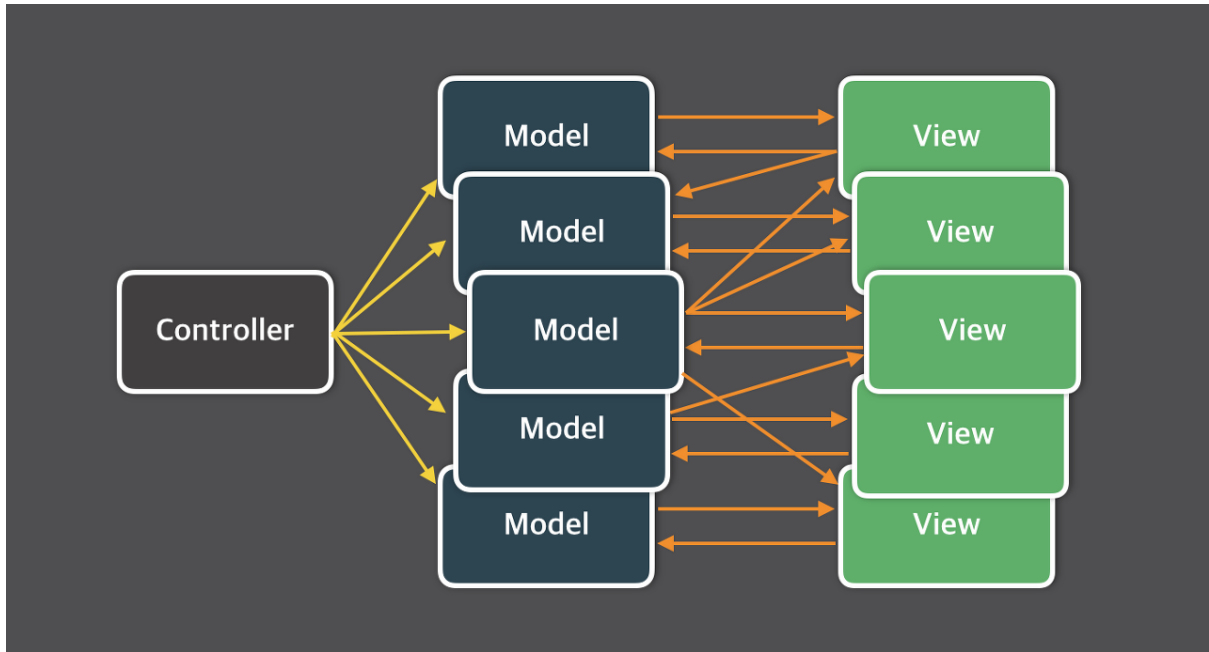
Flux 패턴의 등장 배경

- Flux 패턴은 페이스북(*현 메타*)에서 MVC 모델의 단점을 보완하기 위해 발표한 애플리케이션 아키텍처이다.



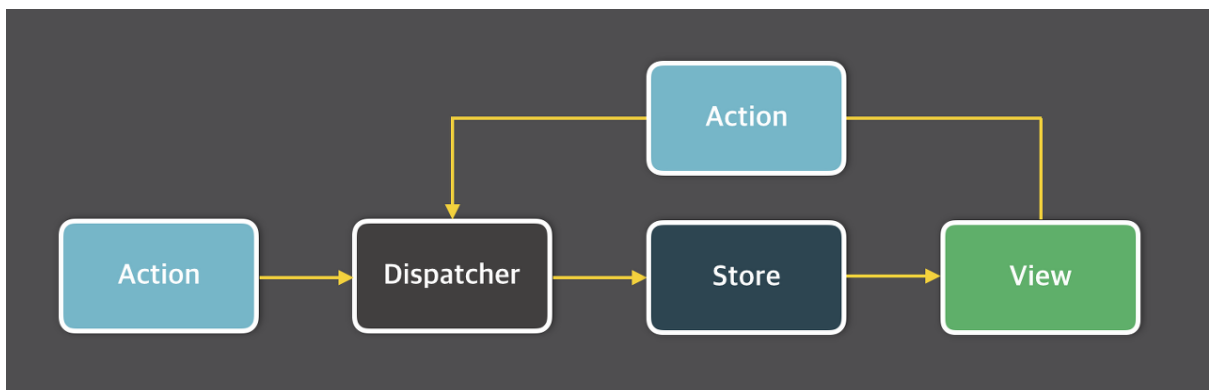
MVC 모델

- 페이스북은 기존에 MVC 모델을 사용하였다.



- 의존성이 복잡한 애플리케이션은 시스템의 복잡도가 높아질 수 있다.
- MVC 모델에서는 View가 다양한 상호작용을 위해 여러 개의 Model을 동시에 업데이트 하고 Model 역시 여러 개의 View를 업데이트할 수 있는데, 새로운 기능이 추가되거나 규모가 커질수록 복잡한 데이터 흐름을 가지게 되어 예측 불가능한 코드를 만들게 되고 버그가 발생하게 되었다.
- 따라서 페이스북은 대규모 애플리케이션에서 **데이터의 흐름을 일관성 있게 관리하여 프로그램의 예측 가능성을 높일 수 있도록, 단방향 데이터 흐름을 가진 Flux를 고안하게 된다.**

Flux 패턴이란?



Flux 패턴의 데이터 흐름

- Flux 패턴은 **단방향 데이터 흐름**을 유지하는 아키텍처 디자인 패턴이다.
- **사용자의 입력을 기반으로 Action을 만들고, Action을 Dispatcher에 전달하여 Store의 데이터를 변경한 뒤 View에 반영하게 된다.**

Action

- 애플리케이션에서 일어나는 이벤트를 나타내는 객체이다.
- Action creator 메서드는 새로 발생한 action의 type과 새로운 데이터(payload)를 묶어 Dispatcher에게 전달한다.

Dispatcher

- Action을 받아서 Store에 전달하는 역할을 한다.
- 즉, 모든 데이터의 흐름을 관리하는 중앙 허브이다.
- Store의 데이터를 조작하는 것은 오직 Dispatcher만 가능하다.

Store

- 애플리케이션의 모든 상태와 로직을 가지고 있다.
- Store의 state 변경이 완료되면 변경된 상태를 View에 전달한다.

View

- Flux의 View는 MVC의 View와는 달리 Store에서 데이터를 가져오는 한편 데이터를 자식 View로 전달하기도 하는 일종의 View-Controller로 보아야 한다.
- 리액트의 컴포넌트라고 생각하면 된다.

아키텍처 디자인과 디자인 패턴

- 지금까지 자주 접한 단어인 아키텍처 디자인과 디자인 패턴은 둘 다 디자인이라는 단어가 들어간다. 그렇다면 같은 의미인 걸까? 정답은 아니오! ❌

- 정확히 말하자면 MVC는 애플리케이션의 구성 요소를 세 가지 역할(Model, View, Controller)로 나눈 **아키텍처 디자인 패턴**이고, Flux는 단방향 데이터 흐름을 구현하는데 사용되는 **디자인 패턴**이다.
- 아키텍처 디자인은 **전체 시스템의 구조와 구성 요소 간의 관계를 설계**하는 것을 의미하며, 대표적인 예로는 MVC 등이 있다.
- 디자인 패턴은 **소프트웨어 개발 시 특정 상황에서 자주 발생하는 문제를 해결하는 패턴을 쉽게 재사용할 수 있도록 추상화**한 것이다. 대표적인 예로는 Singleton, Observer 등이 있다.

Flux 패턴과 React

- Flux 패턴은 페이스북에서 React를 개발할 때 사용한 아키텍처 디자인 중 하나이다.
- 단방향 데이터 흐름을 유지하므로, **React와 같은 뷰 라이브러리와 함께 사용될 때 일관성 있는 데이터 흐름을 구현하는데 매우 유용**하다.

Flux 패턴과 Redux

- **Redux는 Flux 아키텍처 디자인 패턴을 기반으로 한 라이브러리**이다.
- Redux는 애플리케이션의 상태를 하나의 Store에 중앙 집중적으로 관리하고, 이 Store에서 상태를 변경하는 모든 작업은 Action을 통해 수행된다. 변경된 상태는 Store에서 구독하고 있는 모든 구독자에게 전달되어 UI가 업데이트된다.
- Flux 패턴과 Redux는 모두 **단방향 데이터 흐름을 사용하여 데이터의 일관성과 예측 가능성을 유지하며, 모든 상태 변경이 불변성을 유지하도록 구현**된다. 이러한 구조는 **React와 함께 사용될 때, 웹 애플리케이션의 복잡성을 낮추고 유지 보수성을 높이는 데 도움**이 된다!