프로젝트 개발 일지

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 일 시 | 2017년 4 월 6 일 | | | |
| 학 번 | 30113 | | 성 명 | 유승우 |
| 프로젝트 주제 | react-native 를 이용한 애니 편성표 어플리케이션 제작 | | | |
| 학습한 내용 및 활동 결과 | | | | |
| React 와 자주 함께 사용하는 아키텍쳐인 flux의 흐름 | | - MVC 패턴에서 Model 과 View 의 수가 커지고 데이터의 흐름이 양방향으로 이루어질 수록 복잡도는 더욱 증가하고 디버깅 및 코드를 이해하기 어려워짐 => 규모가 큰 어플리케이션에는 적합하지 않음  - react 와 flux 에서는 데이터가 단방향으로만 움직이게 하여 MVC의 단점을 보완함  - 액션 생성자(the action creator) : 모든 변경사항과 사용자와의 상호작용이 거쳐가야 하는 액션의 생성을 담당  - 디스패쳐(dispatcher) : 콜백(callback)이 등록되어 있는 곳. 액션 생성자로부터 액션이 넘어오면 여러 스토어에 액션을 보냄  - 스토어(store) : 애플리케이션 내의 모든 상태와 그와 관련된 로직을 가지고 있음  - 컨트롤러 뷰(the controller view) : 상태가 변경되었을 때 스토어가 그 사실을 컨트롤러 뷰에게 알려주면, 컨트롤러 뷰는 자신의 아래에 있는 모든 뷰에게 새로운 상태를 넘김  - 뷰(the view) : 받은 데이터를 처리해서 사람들이 이해할 수 있는 포맷(HTML)으로 바꿈 | | |
| 고찰 및 추후 계발 계획 : react와 flux 는 거의 항상 함께 사용된다. 다음에는 이런 flux를 react 에 맞게 구현한 라이브러리인 redux에 대해 공부할 것이다. | | | | |
| ㄴㅇㄹㄴㅇ | | | | |