1. [javascript - React component initialize state from props - Stack Overflow](https://stackoverflow.com/questions/40063468/react-component-initialize-state-from-props)

TouchableOpacity 에서, navigate 하면서 type이라는 props를 전달하고, AuthForm 컴포넌트에서 state 초기화를 할 때 type을 props로 받았다.

1. React-Navigation 의 Drawer Navigator를 쓰려고 디펜던시로 React Native Reanimated2를 받았는데, 에러가 떴다. 해결방법은 이유는 모르겠으나 npx react-native start --reset-cache 를 이용해서 시작하자 에러가 사라졌다. 물론 링크와 같은 작업들도 해줘야 한다.

[Installation | React Native Reanimated (swmansion.com)](https://docs.swmansion.com/react-native-reanimated/docs/fundamentals/installation)

1. 2단계 인증 화면은 일단 하나의 스크린에서 state업댓으로 관리하는걸로 생각해두려고 한다.

22/01/07

필요에 의해 커스텀 네비게이터를 만들려고 한다. 사실, 스택 네비게이터와 동일한 느낌으로 동작하되 사이드 메뉴 바가 필요한 상황이다.

[[JavaScript] 자주 사용하는 ES6 문법 정리 (velog.io)](https://velog.io/@kimhscom/JavaScript-%EC%9E%90%EC%A3%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EB%8A%94-ES6-%EB%AC%B8%EB%B2%95-%EC%A0%95%EB%A6%AC)  
진작에 이렇게 링크 걸어놨어야 했는데…

**BLE**

프로젝트 방향성 상 블루투스가 더욱 중요 해졌다. 따라서 이번엔 BLE에 대한 내용을 정리한 뒤, react-native-ble-plx를 제대로 짚어보려고 한다.

기존 Classic Bluetooth와 가장 큰 차이점은, 페어링 방식에 있다.

Classic은 master가 주변 slave에 요청을 날리고 slave가 응답하는 방식으로 페어링을 한다.

BLE는 peripheral(slave)가 마치 비컨처럼, 주기적으로 advertise를 날리고 central(master)이 스캔에 성공함으로써 페어링을 한다.

BLE는 두가지 모드가 존재한다.

Advertise Mode : 주로 존재 자체를 알리거나 적은 양의 데이터를 보낼 때 사용.

advertiser(peripheral)이 주기적으로 advertise를 하고, observer(central)이 advertiser에게 신호를 받기 위해 주기적으로 scanning을 한다.

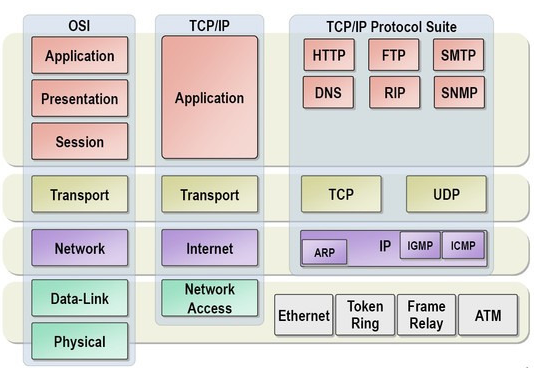
Connection Mode : 양방향 통신이거나, advertise만으로는 부족할 때 사용한다. 1대1 통신이다.

용어

OSI 7 Layer

목적 : 이 모델은 프로토콜을 기능별로 나눈 것이다. 각 계층은 하위 계층의 기능만을 이용하고, 상위 계층에게 기능을 제공한다. '프로토콜 스택' 혹은 '스택'은 이러한 계층들로 구성되는 프로토콜 시스템이 구현된 시스템을 가리키는데, 프로토콜 스택은 하드웨어나 소프트웨어 혹은 둘의 혼합으로 구현될 수 있다. 일반적으로 하위 계층들은 하드웨어로, 상위 계층들은 소프트웨어로 구현된다.

1. Physical Layer : 어떤 데이터 인지, 에러의 유무 등은 신경 쓰지 않고 데이터를 전기적인 신호로 변환해 주고 받기만 함.  
   EX) 통신 케이블, 리피터, 허브
2. Data Link Layer : 1) 에서 전달된 데이터의 오류와 흐름을 관리하여 데이터를 전달. Point To Point 간 신뢰성 있는 데이터 전송을 보장하기 위한 계층으로, CRC (Cyclic Redundancy Check) 기반 오류 제어 및 흐름 제어가 필요. 프레임 (본 계층에서 전송 데이터 단위)에 MAC 주소 (Media Access Control Address)를 부여한다.  
   EX) 브리지, 스위치  
   [MAC 주소 - 위키백과, 우리 모두의 백과사전 (wikipedia.org)](https://ko.wikipedia.org/wiki/MAC_%EC%A3%BC%EC%86%8C)
3. Network Layer : 경로를 선택하고, 주소를 정한 뒤 경로에 따라 패킷을 전달해주는 역할이다. 라우팅, 흐름제어, segmentation, 오류제어, internetworking 등을 수행한다. 네트워크 관리자가 직접 주소를 할당하는 구조, 즉 논리적인 주소 구조 (IP)를 갖는다.  
   3.1) IP Layer : IP ([IP 주소 - 위키백과, 우리 모두의 백과사전 (wikipedia.org)](https://ko.wikipedia.org/wiki/IP_%EC%A3%BC%EC%86%8C)) 주소를 정의하고, IP 패킷의 전달 및 라우팅을 담당하는 계층.  
   3.2) IP Protocol : IP Layer 에서 IP 패킷의 라우팅 대상이 됨. IP 주소 지정.
4. Transport Layer : 보통 TCP 프로토콜을 이용하고, 포트를 열어 응용프로그램들이 전송을 할 수 있게 함. End To End 사용자들이 신뢰성 있는 데이터를 주고 받을 수 있게 함. 특정 연결의 유효성을 제어하고, 일부는 stateful ([스테이트풀(Stateful)과 스테이트리스(Stateless) 차이 및 비교 (redhat.com)](https://www.redhat.com/ko/topics/cloud-native-apps/stateful-vs-stateless)) 하고, connection oriented 이다. 즉 패킷 (Assembly/Sequencing/Deassembly/Error Detection/Request repeat/Flow Control)들의 전송이 유효한지 확인하고 전송 실패한 패킷들을 다시 전송한다는 것이다.  
   4.1) TCP Protocol : Reliable, Connection-Oriented 가 그 특징이다. 패킷의 손실, 중복, 순서 바뀜이 없도록 보장 해주고, 느슨한 연결로 connection-oriented라고 함.  
   4.2) UDP Protocol : 비연결성, 낮은 신뢰성, 순서화되지 않은 Datagram 제공. 빠른 요청과 응답이 필요한 실시간 응용에 적합하다.
5. Session Layer : 데이터가 통신하기 위한 논리적인 연결. End To End 응용 프로세스가 통신 관리를 하기 위한 방법 제공.
6. Presentation Layer : 코드 간 번역을 담당하고, 인코딩 암호화 등의 동작이 이루어짐.
7. Application Layer : HTTP, FTP, SMTP, POP3, IMAP, Telnet 과 같은 프로토콜이 존재함. 브라우저 등을 쉽게 사용할 수 있도록 해주는 것. 즉 모든 통신의 양 끝단은 HTTP 와 같은 프로토콜이지, 응용프로그램이 아니다.



아주 정리를 잘 해 놓은 링크를 남겨둔다.

[OSI 7 계층이란?, OSI 7 계층을 나눈 이유 :: effortDev (tistory.com)](https://shlee0882.tistory.com/110)

GAP (Generic Access Profile) : 서로 다른 제조사가 만든 디바이스 간 호환 규격 제공. 여기서 advertiser / observer를 선택해 ble 네트워크 이용.

GATT (Generic Attrribute Profile) :  
-Profile : Service 묶음.  
-Service : 데이터를 논리적 단위로 묶어놓은 그룹. Characteristic 이라는 단위를 하나 이상 포함.  
-Characteristic : UUID 포함 ([UUID (Universally Unique IDentifier)란? (tistory.com)](https://mattmk.tistory.com/31))