

# 한 번에 끝내는 블록체인 개발 A to Z

---

## Chapter 4

나만의 Uniswap v1 만들어보기(오픈소스 활용)

Chapter 4

나만의 Uniswap v1 만들어보기(오픈소스 활용)

# Uniswap v1 Front 연동(1)

# GOAL

- 유니스왑v1 sdk에 대해서 학습한다.
- 유니스왑v1 소스코드를 받아서 실행해본다.

# uniswap-sdk

- The SDK helps developers interact with the Uniswap protocol in any javascript environment.
- Uses range from [returning chain data](#) for off-chain modeling, to [building](#) customized transactions which [will be broadcast to the EVM](#).
- JavaScript 환경의 프론트엔드에서 유니스왑과 연동을 쉽게 하기 위한 라이브러리이다.
- 지난 강의에서 프론트엔드 연동 할 때 직접 밸런스 조회 및 계산하는 로직을 구현 했었지만 sdk를 사용하여 컨트랙트의 데이터를 쉽게 읽어 올 수 있다.
- EVM에 직접적으로 트랜잭션을 보내는 역할은 하지 않는다. SendTransaction을 위해 컨트랙트를 호출하는 부분은 직접 구현해야 한다.
  - 호출 하는 컨트랙트 함수의 파라미터를 sdk를 통해 쉽게 얻어 올 수 있다.

# uniswap-sdk

```
import { getTokenReserves, tradeEthForExactTokensWithData } from '@uniswap/v1-sdk'

function App() {

  async function getToken() {
    const tokenAddress = '0x89d24A6b4CcB1B6fAA2625fE562bDD9a23260359';
    const networkId = 1;
    const amount = '1000000000000000000';
    const reserves = await getTokenReserves(tokenAddress, networkId);
    console.log(await tradeEthForExactTokensWithData(reserves, amount));
  }

  return (
    <div className="App">
      <button onClick={getToken}>Get</button>
    </div>
  );
}
```

# Uniswap v1 UI

- git clone <https://github.com/Uniswap/v1-interface>
- yarn
- `rm yarn.lock`
- yarn start

# Frontend 개발 설정

1. React App Create
  - yarn create react-app gray-uniswap --template typescript
2. package.json 업데이트
  - web3-react, material-ui, ethers, react-router-dom 설치
3. 폴더 구조 생성
  - components, pages, constants, functions, utils 폴더 생성
4. Solidity 빌드 파일 복사
  - constants/typachatin-types 복사
5. 각 페이지 및 기능 구현

# Uniswap v1 UI

- constants/index.js에 FACTORY\_ADDRESSES 수정
- components/Header/index.js에서 아이콘 수정



# 다음 강의

- Uniswap v1 interface에서 유동성 공급, 스왑 등 주요한 구현 사항에 대해서 파악해본다.