DIN\_블록체인 파트

개발 문서

안병헌

**목차**

1. **소개**
2. **블록체인 js 라이브러리**
   1. caver
   2. 로그인
   3. Transaction 사용
3. **기타 기능 라이브러리**
   1. 계정 생성
   2. 계약 배포
4. 소개

이 문서는 DIN Team의 do it news 개발 파트 중 Solidity 및 caver에 대한 라이브러리를 제작하고 이에 대한 정보를 적고자 작성하는 문서입니다.

개발되기로 되어있던 것은 블록체인 라이브러리이지만 해당 라이브러리를 비슷한 환경에서 잘 작동하는지 확인하기 위하여 간단한 구성의 express를 주로 이용한 웹페이지로 제작되어 있습니다.

1. 블록체인 js 라이브러리

자바스크립트로 블록체인 관련 기능들을 사용하기 편하도록 여러 기능들을 만들어 둔 라이브러리입니다. 기사를 블록체인에 업로드하거나 가져올 때 필요한 핵심 기능을 포함하고 있습니다.

Solidity로 작성한 기능들을 사용하기 위해선 먼저 블록체인에 private key를 통해 wallet을 받아온 후 wallet을 등록시켜야 합니다.

* 1. Caver #ref. caver.js

블록체인과 관련된 기능을 사용하기 위해선 가장 먼저 caver 객체를 가져와야 합니다. lib/caver.js 에서 해당 객체를 추출하고 있습니다.

이 caver 객체는 en노드를 운영하는 컴퓨터에서 가져올 수 있습니다. en노드를 운영하는 것은 비용이 드는데 kas(klaytn api service)를 이용하면 en노드를 따로 꾸리지 않아도 부분 무료로 caver node api를 사용할 수 있습니다.

Kas를 이용하기 위해선 인증절차를 거쳐야 합니다. Kas console([링크](https://console.klaytnapi.com/))에서 인증에 필요한 데이터를 받을 수 있습니다. Kas console 웹페이지에서 security 메뉴 중 credential 이 있습니다. 이 credential을 클릭해 들어가면 AccessKey를 생성하고 삭제할 수 있습니다. 인증을 하기 위해선 이 AccessKey가 필요합니다.

lib/caver.js 에서는 저장시켜 놓은 AccessKey를 읽어와 인증에 사용합니다. 이 키가 kas console에서 삭제되거나 정보를 잃어버리거나 문제가 생기면 인증을 하지못해 기능을 사용하지 못하고 유출되면 피해가 발생할 수 있으니 키 관리를 주의해야 합니다.

* 1. 로그인 #ref. caver.js , BlockcahinLogin.js

계정생성시 얻을 수 있는 keystore file과 생성시 만든 비밀번호, 이 두 정보로 블록체인에 로그인을 할 수 있습니다.

getPrivateKey함수는 keystore파일과 비밀번호를 받아 해당 계정의 privateKey를 받아옵니다.

getWalletInstance함수는 privateKey를 받아 wallet Instance를 받아옵니다.

addWallet함수는 wallet Instance를 받아 클레이튼(블록체인 네트워크)에 추가(등록)시킬 수 있습니다.

login함수에서는 keystore과 비밀번호를 받은 후 앞선 로그인 관련기능들을 한번에 실행시켜주고 그 후 wallet Instance를 반환하도록 하였습니다.

일반적으로는 keystore과 password를 받으면 간편하게 login함수를 통해 계정 인증을 진행하길 추천합니다. 다만 wallet 등록(addWallet 에서 행하는 기능)은 가끔 초기화되기도 하기 때문에 login할 때 반환된 wallet Instance를 따로 저장해두고 기능을 사용하기전 가능한 addWallet함수를 실행하는 편이 좋습니다.

* 1. Transaction 사용 #ref. caver.js , BlockchainArticle.js ,BlockchainAccount.js

기본적인 세팅이 다 이루어졌다면 이제 자유로이 기능들을 사용할 수 있습니다.

-----BlockchainArticle.js-----

BlockchainArticle.js에서 기사를 가져오거나 업로드하는 기능을 주로 제공합니다. 기사들은 배열의 형태로 저장되어 있어 낮은 인덱스부터 차례로 저장됩니다.

getArticle(index)는 기사를 가져오는 함수입니다. 인덱스를 받고 그에 해당하는 기사를 반환하는 함수입니다. 즉, 기사를 가져오는 함수입니다.

uploadArticle(html,address)는 기사를 업로드하는 함수입니다. 입력된 address를 transaction을 보내는 주체로 설정하고 html(기사)를 업로드하는 함수입니다. Address에 해당하는 계정은 이 함수를 사용하기전에 인증이 완료되어야 합니다.

getArticlesCount()는 현재 저장된 기사의 개수를 가져오는 함수입니다.

getArticles()는 저장된 모든 기사를 가져오는 함수입니다. 배열형태로 반환합니다.

getArticleOwner(articleIndex)는 인덱스에 해당하는 기사의 작성자 address를 가져오는 함수입니다.

-----BlockchainAccount.js-----

BlockchainAccount.js는 기사와 관련된 기능은 아니고 블록체인 계정과 관련된 기능들을 담고있다.

getBalance(address)는 address(클레이튼 계정 주소)에 해당하는 계좌가 얼마나 클레이를 가지고 있는지에 대한 값을 가져오는 함수입니다.

Transfer(fromAddress,toAddress,klay)는 fromAddress 에서 toAddress로 klay에 전달된 값만큼의 클레이를 클레이를 송금하는 기능입니다. 송금자 측에서 약간의 수수료를 부담합니다. 송금하는 양의 클레이 이상으로 약간의 여유가 있어야 송금을 할 수 있습니다.

1. 기타 기능 라이브러리

이 라이브러리는 주요 기능 외 정상적인 기능 사용을 위해 필요한 부차적인 기능들을 담고 있다. Node로 바로 실행시켜 이용할 수 있다.

* 1. 계정 생성 #ref. createAccount.js

Solidity로 작성한 스마트 컨트랙트는 배포를 블록체인에 배포를 한 뒤에 사용할 수 있다. 이 배포시엔 수수료가 든다. 이 수수료는 kas에 만들어진 계정으로 지불할 수 있다. 이 계정 만드는 것을 kasLib/createAccout.js 을 node로 실행시켜 할 수 있다. 다만, 미리 만들어 놓은 계정 하나를 쓰기로 가정 해놓고 개발해 두었기에 일단은 필요하지 않은 기능이다.

* 1. 계약 배포 #ref. deploy.js

Solidity로 작성한 스마트 컨트랙트는 배포를 블록체인에 배포를 한 뒤에 사용할 수 있다. 이 deploy 를 노드로 실행시키면 배포할 수 있다. 그리고 배포된 컨트랙트를 사용하기 위해선 그 컨트랙트가 배포된 주소와 배포된 컨트랙트의 abi, 이 2개가 필요한데 배포하면서 이를 출력하도록 만들었다. 이 address와 abi가 없다면 스마트 컨트랙트를 이용할 수 없으니 이를 주의하여야 한다.