



2022/06/03

시나리오에 따라 적용할 함수

nft발행 시나리오

CardMinting.js의 내의 함수들 이용

서버에서 24시간마다 saveAccessToken()을 이용하여 액세스 키 갱신

서버에서 앨범마다 CardImageUpload(imageLocation)을 이용하여 앨범 템플릿을 만들고 저장

유저가 nft발행 버튼 클릭시 서버로 앨범정보와 유저지갑과 함께 post요청

서버에서 받은 앨범정보를 이용하여 imageUrl(템플릿정보), nfcKey, album_Id를 디비에서 불러옴

KlipCardMinting(imageurl, userAddress,nfcKey,album_Id)함수를 이용하여 nft 발행

nft조회 시나리오

getWallet()으로 로그인하여 유저지갑 정보 얻어옴

```
async getWallet(): Promise<String> {  
  if (this.userAccountAddress !== undefined) {  
    return this.userAccountAddress;  
  } else {  
    return await this.loginKlip();  
  }  
}
```

getAccountTokenBalance()을 이용하여 유저가 보유한 총 nft갯수 가져옴

```
async getAccountTokenBalance(): Promise<Number> {  
  return this.tokenContract.methods.balanceOf(await this.getWallet()).call();  
}
```

```
}
```

getTokenOfOwnerByIndex(index:Number)을 이용하면 유저가 보유한 토큰들의 인덱스를 이용하여 token id를 가져올수있음 ex)유저가 보유한 2번째 토큰아이디=getTokenOfOwnerByIndex(2)

```
async getTokenOfOwnerByIndex(index: Number): Promise<Number> {  
    return this.tokenContract.methods.tokenOfOwnerByIndex(await this.getWallet(), index).call();  
}
```

getAllTokenOfOwner(balance:Number)위의 getAccountTokenBalance와 getTokenOfOwnerByIndex를 이용하여 유저가 보유한 전체 nft 각각의 token id 리스트를 가져옴

```
async getAllTokenOfOwner(balance:Number): Promise<Number[]> {  
    let list=[];  
    for (var i = 0; i < balance; i++) {  
  
        var tokenId = await this.getTokenOfOwnerByIndex(  
            i  
        );  
        list.push(tokenId);  
        console.log("TestKaikas testGetAllTokenOfOwner", tokenId);  
  
    }  
    return list;  
}
```

getTokeninfoURL(tokenId: Number)을 이용하여 각 토큰마다의 정보 얻어옴(앨범이미지url, 발행할때 설정한 nfckey, album_id등의 attribute)

```
async getTokeninfoURL(tokenId: Number): Promise<String> {  
    return this.tokenContract.methods.tokenURI(tokenId).call();  
}
```

nft 조회시나리오에 사용되는 함수들은 KlipNftTransaction.ts에 KlipNftTransaction클래스로 구현되어 있음 사용시

```
const nftTransaction=new KlipNftTransaction();  
//await nftTransaction.getWallet();  
console.log(await nftTransaction.getAccountTokenBalance());
```

처럼 클래스를 생성후 메서드를 사용하면됨

KlipNftTransaction.ts는 타입스크립트로 구현되어 있으므로 js로 변환후 사용해야함

tsc KlipNftTransaction.ts 명령어를 사용하여 변환후 생성된 KlipNftTransaction.js 파일을 사용하면 됨

혹은 webpack build에 명령어를 넣어 컴파일시 자동 변환을 할 수 있음

nft마트 시나리오

```
approve(): Promise<Boolean>;  
setSaleOnCont(tokenId: Number, price: Number): Promise<Boolean>;  
getTokenPrice(tokenId: Number): Promise<Number>;  
buyToken(tokenId: Number, price: Number): Promise<Boolean>;
```