

Social Login

Code dispatch

```
const KakaoAuthHandler = (props) => {  
  const dispatch = useDispatch();  
  const code = new URL(window.location.href).searchParams.get("code");  
  
  //console.log('!!! oauth 페이지 code : ', code)  
  
  useEffect(()=>{ //백엔드로 쿠키 토큰 전송  
    dispatch(__userOauthKakao(code))  
  },[])  
  
  return (  
    <>  
      <Wrap>  
        <StLoadingText>사용자 인증하는 중 🔒</StLoadingText>  
        <StInfiniteRotatingLogo>⌛</StInfiniteRotatingLogo>  
      </Wrap>  
    </>  
  )  
};
```











































































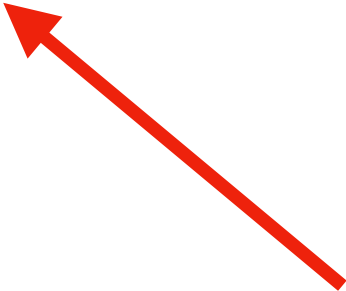
















cloncodingfinal.vercel.app/oauth?code=KSISNlcKMMPh_DIoHrvdnxrrCw-d_TAqhfZNj4XLBTeGAcffnYEL6YFASo1d5bAvgyJrgo9c-wAAAAGGDA04oQ





소셜 관리 페이지에 등록된 URL로 로그인 요청을 하면
인가코드가 URL의 Params로 반환 됨

Social Login

Code dispatch

```
const [KakaoAuthHandler, {params}] = useAuthHandlers();
const dispatch = useDispatch();
const code = new URL(window.location.href).searchParams.get("code");

//console.log('!!! oauth 페이지 code : ', code)

useEffect(() => { // 백엔드로 쿠키 토큰 전송
  dispatch(userOAuthKakao(code))
}, [])

return (
  <>
    <Wrap>
      <StLoadingText>사용자 인증하는 중 🔒</StLoadingText>
      <StInfiniteRotatingLogo>⌚</StInfiniteRotatingLogo>
    </Wrap>
  </>
)
```

소셜 관리 페이지에 등록된 URL로 로그인 요청을 하면
인가코드가 URL의 Params로 반환 됨

Social Login

Response Header

```
export const __userOauthKakao = createAsyncThunk(
  "oauth/USER_OAUTH_KAKAO",
  async (payload, thunkAPI) => {
    try{
      const data = await api.post(`auth/kakao`, {payload})
      .then((res)=>{
        const accessToken = res.headers.authorization;
        const refreshToken = res.headers.refreshToken;
        const nickname = res.data.nickname;

        //유저 토큰 + 닉네임이 있다면 가져온 후 세팅
        if(accessToken && refreshToken && nickname){
          localStorage.setItem("token", accessToken);
          localStorage.setItem("refreshToken", refreshToken);
          localStorage.setItem("nickname", nickname);
        }else{
          alert('인증 오류! 다시 시도해주세요!')
          return window.location.assign("/");
        }
      })
      //유저 토큰 + 닉네임 가져오기
      const accessTokenGet = localStorage.getItem("token");
      const refreshTokenGet = localStorage.getItem("refreshToken");
      const nicknameGet = localStorage.getItem("nickname");
      if(accessTokenGet && refreshTokenGet && nicknameGet){
        alert('소셜로그인 인증 완료!')
        window.location.assign("/main");
      }else{
        alert('연결 오류! 다시 시도해주세요!')
        return window.location.assign("/");
      }
    }
    return res
  })
  return thunkAPI.fulfillWithValue(data)
}catch(error){
  window.location.assign("/");
  return thunkAPI.rejectWithValue(error)
}
};
```