

업체

코인

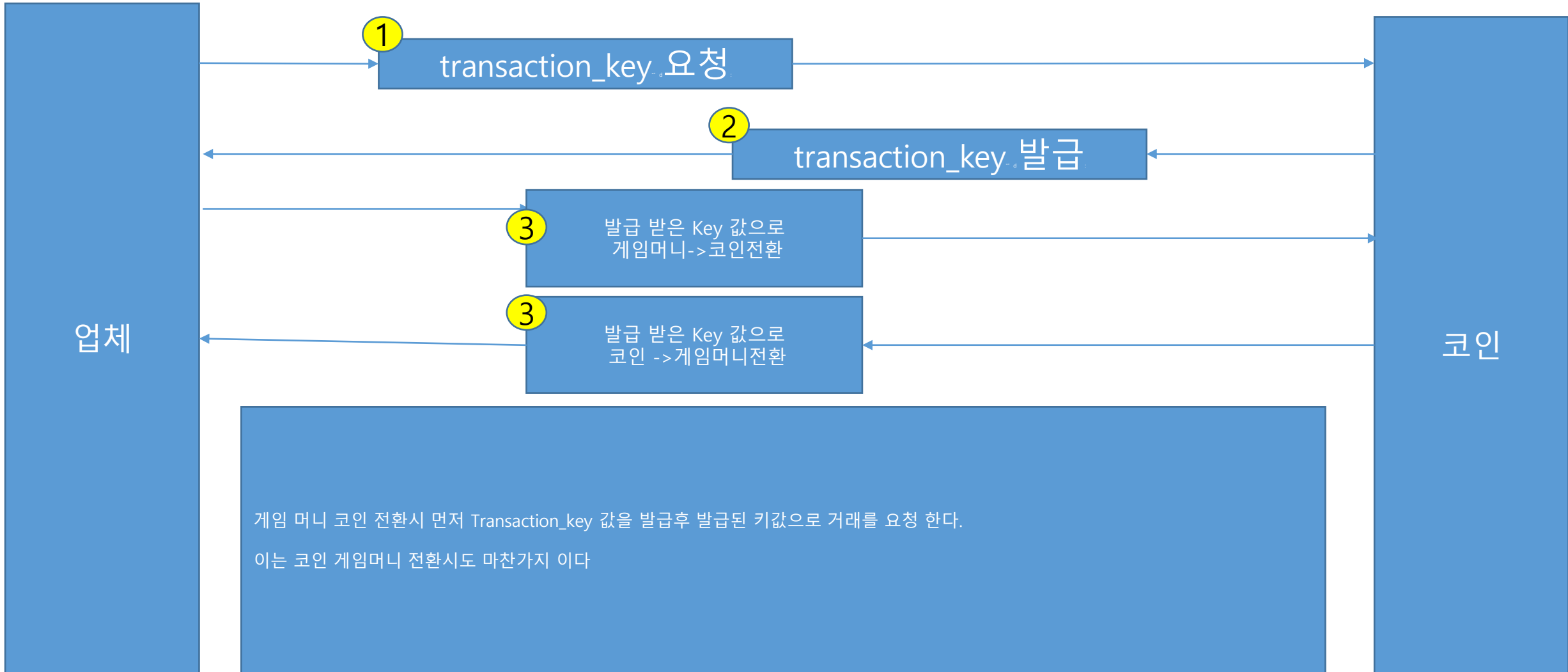
* 1번 창은 반드시 필요
: 코인 사이트의 비밀번호를 저장 여부는 업체에서 판단.

* 코인사이트의 비밀번호를 저장 할때 는 2번 창이 불필요

- 코인사이트의 비밀번호를 저장 안할때는 2번 창이 필요

* 업체에서 유저의 코인 사이트 비밀번호는 코인 가져오거나 기타 api 사용 할때 필요하므로 항상 저장 하고 있는 편이 편하다.

게임머니 -> 코인 전환 절차
코인->게임머니 전환 절차



Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

가입유무조회
/api/partner/account/check/registered

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX"  
  "userAccount" : "anirime",  
  "ref_Code" : "청주322"  
}
```

없는 유저시 :
{ "result":false,"is_registered":false,"msg":"등록되지 않은 사용자
입니다."}

Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

코인->게임머니변환
/api/partner/gameplayer/withdraw

Header

```
{  
  "Content-Type" : "application/json;  
  charset=UTF-8";  
  "Authorization" : Bearer (로그인시 얻은 토큰  
값 )  
}
```

Body:

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX",  
  "transaction_key" : "CCF1BEF92EA3",  
  "withdraw_amount" : 2000  
}
```

```
{  
  "result":true,"msg":"포인트가 인출 되었습니다."  
  "withdraw_amount":2000.0  
}
```

Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

게임머니 -> 코인 변환
/api/partner/gameplayer/deposit

Header

```
{  
  "Content-Type" : "application/json;  
  charset=UTF-8";  
  "Authorization" : Bearer (로그인시 얻은 토큰  
값 )  
}
```

Body:

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX",  
  "transaction_key" : "CCF1BEF92EA3",  
  "deposit_amount" : 20000  
}
```

결과

```
{"result":true,"msg":"변환 되었습니다."}  
또는  
{"result":false,"msg": "error code"}
```

Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

잔액조회
/api/partner/gameplayer/balance

Header

```
{  
  "Content-Type" : "application/json;  
  charset=UTF-8";  
  "Authorization" : Bearer (로그인시 얻은 토큰  
값 )  
}
```

Body :

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX"  
}
```

결과

```
{  
  "result":true,  
  "msg":"","current_balance":7158989.0  
}
```


Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

트랜잭션 키 발급

/api/partner/gameplayer/getTransactionKey

Header

```
{  
  "Content-Type" : "application/json;  
  charset=UTF-8";  
  "Authorization" : Bearer (로그인시 얻은 토큰  
  값 )  
}
```

Body :

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX"  
}
```

결과

```
{  
  "result":true,  
  "msg":"발급 완료",  
  "transaction_key":"E20E02DDB281"  
}
```

Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

트랜잭션 키 발급

/api/partner/admin/getTransactionKey

Header

```
{  
  "Content-Type" : "application/json;  
  charset=UTF-8";  
  "Authorization" : Bearer (로그인시 얻은 토큰  
  값 )  
}
```

Body :

```
{  
  "api_key" : "SSR-XXXXXXXXXXXX"  
}
```

결과

```
{  
  "result":true,  
  "msg":"발급 완료",  
  "transaction_key":"E20E02DDB281"  
}
```

Api 주소 : <https://api.neo-dex.com/> (POST)
TEST API KEY : SSR-XXXXXXXXXXXX (추후 업체별로 변경 예정)

발급된 트랜잭션 키값으로 특정 사용자의 네오를 대리점 으로부터 입금 시킴.(일종의 환급)
/api/partner/admin/refund

Body :

```
{  
  "api_key": "api 값",  
  "transaction_key": "키값",  
  "userAccount": "사용자 아이디",  
  "ref_code": "대리점 매장코드",  
  "refund_amount": 금액  
}
```

결과

```
{  
  
}
```

※ 입금 순서

1. 게임내 충전 진행 ->
2. login api 를 이용한 회원 연동
3. 유저가 코인사이트에서 코인구매 신청
4. 트레이더 (랜덤 계좌)가 코인 판매
5. withdraw api를 이용하여 구매된코인 > 코인사이트의 매장관리자 계정으로 코인이동
6. 이동된 코인 만큼 유저가 입금신청 > 승인 > 게임사이트에 충전
7. 코인사이트의 매장관리자 계정에서 입금 승인된 코인만큼 변동 확인

※ 출금순서

1. 게임내 보유금중 출금신청
2. login api 를 이용한 회원 연동
3. 유저가 보유 코인을 판매신청
 - 코인사이트의 매장관리자계정에 보유 코인이 충족시
 - > deposit api를 이용하여 코인거래소의 매장관리자 계정에서 유저 계정으로 코인이동
 - 코인사이트의 매장관리자계정에 보유 코인이 부족시
 - > 코인사이트의 전체관리자에게 충전요청 입금후 승인 (매장에서 직접 코인구입)
 - > deposit api를 이용하여 코인거래소의 매장관리자 계정에서 유저 계정으로 코인이동
4. 트레이더 (랜덤 계좌)가 코인 구매
5. 코인이전 확인 > 미승인시 강제출금

TEST

te.neo-dex.com -> 테스트 서버 url

te-api.neo-dex.com -> 테스트 api url

te-ad.neo-dex.com -> 테스트 admin page url

REAL

neo-dex.com

www.neo-dex.com -> 실서버 url

api.neo-dex.com -> 실서버 api url

ad.neo-dex.com -> 실서버 admin page