

2023.07 Monthly Note

Phase 58.0.04

Deploy Item

구분	O/X	Deploy
JAR	O	9114d89
DB	X	-
CFG	X	-
ETC	X	-

Development Task

[KCP-123034] V3 회원 검증 API 관련 프로세스 개선

- 결합 카드 회원 가입을 위한 회원 검증 시, 탈퇴 상태 회원 조건 누락되어 다건 조회되면서 에러 발생
- V3 회원 검증 API 조회 시, excludeStatus 조건에 의해 DELETED 상태 조건 제외될 수 있도록 개선
- POST /api/v3/user/registration/verify 관련 수정

Phase 58.2.00

Deploy Item

구분	O/X	Deploy
JAR	O	8a93d05
DB	O	Before: 1개
CFG	X	-
ETC	X	-

DB Query

Before / After	File Name
Before	20230719_PHASE_58.2.00_김지윤_MAP_01_ALTER_WALLET_USER.sql

Development Task

[KCP-123647] 회원 원장 DB 테이블 영문 성/이름 신규 Column 추가

- 영문 성/이름 분리된 신규 Column 추가

- ENG_FSTN : 영문명 이름
- ENG_LSTN : 영문명 성
- 신규 Column 추가 대상 테이블
 - WALLET_USER
 - WALLET_USER_HISTORY
 - DORMANCY_USER
 - WITHDRAWAL_USER
- 20230719_PHASE_58.2.00_김지윤_MAP_01_ALTER_WALLET_USER.sql DB 기안문 작성
- 영문명 관련 신규 Embeddable 방식 EngNameInfo 클래스 생성

[KCP-123649] 회원 정보 변경 API 영문 성/이름 변경 요청 필드 추가

- 회원의 영문 성/이름 정보를 변경하기 위해 요청 필드 추가 및 회원 정보 변경 프로세스 적용
 - engFirstName : 영문명 이름
 - englastName : 영문명 성
- 회원 정보 변경 API 목록
 - POST /api/v3/user/update
 - POST /api/user/changeUserInfo

[KCP-123650] 회원 정보 조회 API 영문 성/이름 조회 응답 필드 추가

- 회원의 영문 성/이름 정보 제공을 위한 DB 조회 프로세스 적용하여 회원 정보 조회 API 응답 필드 추가
 - engFirstName : 영문명 이름
 - englastName : 영문명 성
- 회원 정보 조회 API 목록
 - POST /api/v3/user
 - POST /api/users/{userId}
 - POST /api/users//mpaId/{mpaId}

[KCP-123663] 회원 영문 성/이름 신규 Column 파기 프로세스 적용

- WALLET_USER 테이블에 신규 추가된 회원 영문 성/이름 Column 을 파기할 수 있도록 Batch 프로세스에 적용
- 회원 정보 파기 Batch 목록
 - POST /api/v3/batch/dormancy/{batchJob}
 - POST /api/v3/batch/withdrawal/{batchJob}/{batchType}

[KCP-123203] 삼성카드 회원 정보 암호화 Key 설정

- 삼성카드 매입 대행 프로세스 관련 회원 정보 암호화 Key 설정
- ENCRYPT_KEY_CONFIG DB 테이블에 삼성카드 암호화 Key 정보 저장
 - 개발계 Key: 42304645413141453446313633323735
- EncryptKeyUsageEnum 클래스 SAMCARD 프로퍼티 추가

Phase 59.0

Deploy Item

구분	O/X	Deploy
JAR	O	de8e3e7

구분	O/X	Deploy
DB	X	-
CFG	O	map.properties 수정
ETC	X	-

Config Properties

map.properties 수정

```
# 신규 white-list 회원 DI 추가 (MONA 서비스 테스트 관리자 4건 추가)
# Insert map.white-list.di
map.white-list.di=
nchDIImomQYH8X2T/TqtvumTHXPe6h1khjgcKsZeiemA+gs5z5LiVMSdKvxcAq3Gq0hq6gW//IirUuxQU2AbQ==,\ # 박아영
9sn5dXw9+sk0znH28v9uNX6t7EZxwWME4tL59TgEdFrLLmSQqfIyYvEUQN/nkLcZKJR81pSH0o/KrZNCzhcxkw==,\ # 기수영
3acPNw9nPavSJKaaBB1G8CmocJoyFersvk0ph7J72hd/Zd+b2X3eg0gsYGW/asyBoF+aLqYLX1a3saaY0+jUw==,\ # 허진석
RaTkW0BdIhhzNfsvBrEiq0/5fCv9J/7MY5fbXVAo9cbCIQmtaIvCbos5HzmxuL87oAIXu1o+9MnrsncU3A5vSg== # 김충길
```

Development Task

[KCP-122643] 부정사용자 목록 조회 및 관리 API 개선

- ISMS-P 정책에 따라 분리 보관된 탈퇴 회원 정보의 영향으로 부정사용자 목록 조회 API 개선 필요
- 기존 회원 로그인 ID or 휴대폰 번호로 조회 조건이 WALLET_USER 테이블이 아닌 WITHDRAWAL_USER 테이블 컬럼 조건으로 변경
- WalletUserDetailsDaoImpl.getBlackedWalletUserWithDetails() 함수를 QueryDSL 적용하여 변환 (line: 563 참고)
- GET /api/users/paginated-blacklisted-user API 관련 수정

[KCP-122644] 부정사용자 등록된 회원 재가입 시도 휴대폰 번호 저장 프로세스 신규 추가

- 부정사용자가 재가입 시도하여 서비스 진입한 경우, 운영 대응을 위한 휴대폰 본인 인증 당시의 휴대폰 번호를 별도 캐싱 처리하여 조회할 수 있도록 프로세스 신규 추가
- 부정사용자 Cache 정보

```
{
  "blacklist:{mobileNumber}:{aspId}": "{hashId}"
}
```

- Cache 설정
 - 최초 1회만 캐싱 처리
 - 생성된 Cache 정보는 7일 뒤 삭제 처리
- 해당 Cache 정보를 조회/수정/삭제 할 수 있도록 API 추가 개발 필요
- AbstractWalletUserServiceImpl.verifyMembershipPolicy() 함수 처리 시, 부정사용자 확인 대상 회원인 경우 휴대폰 번호 저장 처리할 수 있도록 수정 (line: 594 참고)
 - POST /api/v3/user/registration/verify API 관련 수정
- BlacklistedUserServiceImpl.removeUser() 함수 처리 시, 부정사용자 Cache 정보 삭제 처리 추가
 - POST /api/user/blacklist/remove API 관련 수정

[KCP-122645] 부정사용자 목록 조회 API 재가입 시도 휴대폰 번호 Cache 정보 조회 개선

- 부정사용자 목록 조회 처리 시, 재가입 시도하여 저장된 휴대폰 번호 Cache 정보도 포함하여 조회될 수 있도록 개선
- V3BlacklistUserFindService.findAll() 함수 처리 시, 부정사용자 Cache 정보 조회 처리 추가
 - GET /api/users/paginated-blacklisted-user API 관련 수정

[KCP-122645] V3 회원 정보 조회 API 요청시 Empty String 인 경우 조회 조건 제외 처리 추가

- V3 회원 정보 조회 API 에서 List 형식의 요청 받는 필드에 대해 Empty String 인 경우, 조건 제외될 수 있도록 null 처리 추가
- POST /api/v3/user API 수정

[KCP-123327] 휴면 해제 처리 API 프로세스 OpenAPI 관련 개선

- OpenAPI 휴면 해제 요청인 경우, CI 기준으로 휴면 회원 정보 조회할 수 있도록 수정
- POST /api/user/dormancy/change-active API 관련 수정

[KCP-122371] 부가서비스 등록 회원 목록 조회 API 신규 개발

- 특정 부가서비스에 등록된 요청 STATUS 회원 상태 조건과 일치한 회원 목록 조회 API 신규 개발
- status 요청은 , COMMA 구분 문자열 목록
- GET /api/v3/user/addition/service/{additionServiceId}/users?status={status}&page={page}&size={size} API 신규 추가

Spring Data Redis 사용 후기**👉 :phantom Cache Key 에 대해서.. 👉**

- Spring Data Redis 자체적으로 저장된 Cache Key 정보가 Expiry 만료가 되는 시점에 별도의 Event 이벤트를 받기 위해 {cache key}:phantom 방식으로 복사본을 저장한다.
- {cache key} 정보가 만료되는 시점에 {cache key}:phantom 를 통해서 Application 에 정의된 Event 처리를 수행할 수 있게 된다.

```
@EnableRedisRepositories(enableKeyspaceEvents = RedisKeyValueAdapter.EnableKeyspaceEvents.ON_STARTUP)
```

- 위 설정을 통해서 Keysapce Event 에 대한 처리를 Application 에서 수행하게 된다.
 - !! 해당 설정이 없다면, Key 가 만료되더라도 :phantom 를 포함한 다른 Cache 정보가 삭제되지 않는다 !!

👉 Secondary Indexing 에 대해서.. 👉

- Secondary Indexing**: key 조건으로 검색 외의 Index 로 지정한 정보에 대해서 검색할 수 있도록 새로운 SET Cache 정보를 저장한다.
- Redis 의 Secondary Indexing 처리를 위해 SpringDataRedis 에서는 @Indexed Annotation 을 통해서 간단하게 구현이 가능하다.

```
@RedisHash(value = "blacklist")
data class BlacklistUserCacheModel(
    @Id
    val id: String,
    val aspId: String,
    @Indexed
    val hashId: String? = null,
    @TimeToLive(unit = TimeUnit.SECONDS)
    val ttl: Long? = 604800
)
```

SET	blacklist:01020310001:0001770000000000:idx	No limit	-
SET	blacklist:hashId:6D173B65DA1943BF369FB590A3F4D1D5EBCB12E14164CAF13AB4F43C...	No limit	-
HASH	blacklist:01020310001:0001770000000000	25 s	-
HASH	blacklist:01020310001:0001770000000000:phantom	5 min	-

- blacklist:01020310001:0001770000000000 Cache Key 로 hashId 저장
- blacklist:hashId:6D173B65DA1943BF369FB590A3F4D1D5EBCB12E14164CAF13AB4F43C1DFE8B58 Cache Key 로 원본 Cache 정보를 Index 처리하여 새로운 Cache 정보로 저장

! Secondary Indexing 설정할 때 주의 사항 !?

Keyspace 설정 확인 !

- BlacklistUserCacheModel 클래스의 @RedisHash 를 보면 blacklist 란 키워드로 Keyspace 를 설정한 것을 확인할 수 있다.
 - 보통 Redis 에서 Key 생성 시, : 구분자로 Keyspace 를 구성한다.
- Keyspace 가 만약 blacklist 가 아닌 map:blacklist 와 같이 : 가 들어가면, 만료 Event 발생하였을 때 Event 처리가 되지 않을 수 있다.
 - RedisKeyExpiredEvent 클래스에서 Event 를 수신하여 처리하지만, 설정된 Keyspace 의 문자열에서 : 로 잘라서 Key 만료 처리를 한다.
 - ¶ 이 때, 잘린 Keyspace 명이 일치하지 않아 정상적으로 Index Cache 정보가 삭제되지 않게 된다 ¶