

“운전자격확인시스템(RIMS) 구축”

# 사용자/API 서버 간 통신규약

Ver. 1.6



2025. 10. 21

## <제목 차례>

|  |   |
|--|---|
| 1 개요                                   | 1 |
| 1.1 목적                                 | 1 |
| 2 목록                                   | 1 |
| 2.1 용어정의                               | 1 |
| 2.2 기능정의                               | 1 |
| 3 공통 설계                                | 1 |
| 3.1 송신 데이터의 형식                         | 1 |
| 3.2 보안 방법                              | 1 |
| 3.2.1 암호화 알고리즘                         | 1 |
| 3.2.2 암호화 키 정보                         | 1 |
| 3.2.3 암호화 알고리즘 제공 방식                   | 2 |
| 3.3 렌트카 운전자격확인 API 사용 시 차량 정보          | 2 |
| 3.4 운전자격확인 API 와 대여확정 API 분리           | 2 |
| 3.5 데이터 Request 목록                     | 3 |
| 3.5.1 request 목록                       | 3 |
| 4 서비스별 API                             | 4 |
| 4.1 공통                                 | 4 |
| 4.1.1 토큰발급                             | 4 |
| 4.1.1.1 인증 URL                         | 4 |
| 4.1.1.2 토큰 발급 헤더                       | 4 |
| 4.1.2 사용자검증                            | 5 |
| 4.1.2.1 API 호출                         | 5 |
| 4.1.2.2 사용자검증 URL                      | 5 |
| 4.1.2.3 사용자검증 헤더                       | 5 |
| 4.1.2.4 사용자검증 응답 예시                    | 6 |
| 4.1.2.5 사용자 검증 조회 응답(response) Body 목록 | 6 |
| 4.1.3 비대면 운전자격확인 딥링크                   | 7 |

|   |    |
|---|----|
| 4.1.3.1 API 호출 .....                                    | 7  |
| 4.1.3.2 비대면 운전자격확인 딥링크 URL .....                        | 7  |
| 4.1.3.3 비대면 운전자격확인 딥링크 예시 .....                         | 7  |
| 4.2 렌트카 .....   | 8  |
| 4.2.1 렌트카 사업자 운전자격확인 단건 조회 .....                        | 8  |
| 4.2.1.1 API 호출 .....                                    | 8  |
| 4.2.1.2 렌트카 사업자 운전자격확인 단건 조회 URL .....                  | 8  |
| 4.2.1.3 렌트카 사업자 운전자격확인 단건 조회 헤더 .....                   | 8  |
| 4.2.1.4 렌트카 사업자 운전자격확인 단건 조회 바디 .....                   | 8  |
| 4.2.1.5 렌트카 사업자 단건 조회 예시 .....                          | 9  |
| 4.2.1.6 렌트카 사업자 운전자격확인 단건 조회 응답(response) Body 목록 ..... | 10 |
| 4.2.1.6.1 Header .....                                  | 10 |
| 4.2.1.6.2 Body .....                                    | 10 |
| 4.2.2 렌트카 사업자 운전자격확인 배치 조회 .....                        | 11 |
| 4.2.2.1 API 호출 .....                                    | 11 |
| 4.2.2.2 렌트카 사업자 운전자격확인 배치 URL .....                     | 11 |
| 4.2.2.3 렌트카 사업자 운전자격확인 배치 조회 헤더 .....                   | 11 |
| 4.2.2.4 렌트카 사업자 운전자격확인 배치 조회 바디 .....                   | 11 |
| 4.2.2.5 렌트카 사업자 배치 조회 예시 .....                          | 12 |
| 4.2.2.6 렌트카 사업자 운전자격확인 배치 조회 응답(response) Body 목록 ..... | 13 |
| 4.2.2.6.1 Header .....                                  | 13 |
| 4.2.2.6.2 Body .....                                    | 14 |
| 4.2.3 렌트카 사업자 대여확정 .....                                | 14 |
| 4.2.3.1 API 호출 .....                                    | 14 |
| 4.2.3.2 렌트카 사업자 대여확정 URL .....                          | 14 |
| 4.2.3.3 렌트카 사업자 대여확정 헤더 .....                           | 14 |
| 4.2.3.4 렌트카 사업자 대여확정 바디 .....                           | 15 |
| 4.2.3.5 렌트카 사업자 대여확정 예시 .....                           | 15 |
| 4.2.3.6 렌트카 사업자 대여확정 응답(response) Body 목록 .....         | 17 |

|   |    |
|---|----|
| 4.2.3.6.1 Header .....                                      | 17 |
| 4.2.3.6.2 Body .....  | 17 |
| 4.2.4 렌트카 사업자 반복예약 .....                                    | 17 |
| 4.2.4.1 API 호출 .....  | 17 |
| 4.2.4.2 렌트카 사업자 반복예약 URL .....                              | 17 |
| 4.2.4.3 렌트카 사업자 반복예약 헤더 .....                               | 17 |
| 4.2.4.4 렌트카 사업자 반복예약 바디 .....                               | 18 |
| 4.2.4.5 렌트카 사업자 반복예약 예시 .....                               | 18 |
| 4.2.4.6 렌트카 사업자 반복예약 응답(response) Body 목록 .....             | 20 |
| 4.2.4.6.1 Header .....                                      | 20 |
| 4.2.4.6.2 Body .....  | 20 |
| 4.2.5 렌트카 사업자 대여자 대여정보 이상여부 확인 .....                        | 21 |
| 4.2.5.1 API 호출 .....  | 21 |
| 4.2.5.2 렌트카 사업자 대여자 대여정보 이상여부 확인 URL .....                  | 21 |
| 4.2.5.3 렌트카 사업자 대여자 대여정보 이상여부 확인 헤더 .....                   | 21 |
| 4.2.5.4 렌트카 사업자 대여자 대여정보 이상여부 확인 바디 .....                   | 21 |
| 4.2.5.5 렌트카 사업자 대여자 대여정보 이상여부 확인 예시 .....                   | 22 |
| 4.2.5.6 렌트카 사업자 대여자 대여정보 이상여부 확인 응답(response) Body 목록 ..... | 22 |
| 4.2.5.6.1 결과 .....  | 22 |
| 4.2.6 렌트카 사업자 대여번호 생성 .....                                 | 23 |
| 4.2.6.1 API 호출 .....  | 23 |
| 4.2.6.2 렌트카 사업자 대여번호 생성 URL .....                           | 23 |
| 4.2.6.3 렌트카 사업자 대여번호 생성 헤더 .....                            | 23 |
| 4.2.6.4 렌트카 사업자 대여번호 생성 바디 .....                            | 23 |
| 4.2.6.5 렌트카 사업자 대여번호 생성 예시 .....                            | 24 |
| 4.2.6.6 렌트카 사업자 대여번호 생성 응답(response) Body 목록 .....          | 24 |
| 4.2.6.6.1 결과 .....  | 24 |
| 4.2.7 렌트카 사업자 비대면 인증 결과 확인 .....                            | 25 |
| 4.2.7.1 API 호출 .....  | 25 |

|   |    |
|---|----|
| 4.2.7.2 렌트카 사업자 비대면 인증 결과 확인 URL                  | 25 |
| 4.2.7.3 렌트카 사업자 비대면 인증 결과 확인 헤더                   | 25 |
| 4.2.7.4 렌트카 사업자 비대면 인증 결과 확인 바디                   | 25 |
| 4.2.7.5 렌트카 사업자 비대면 인증 결과 확인 예시                   | 26 |
| 4.2.7.6 렌트카 사업자 비대면 인증 결과 확인 응답(response) Body 목록 | 27 |
| 4.2.7.6.1 Header                                  | 27 |
| 4.2.7.6.2 Body                                    | 27 |
| 4.2.8 렌트카 사업자 업체별 대여이력 조회                         | 27 |
| 4.2.8.1 API 호출                                    | 27 |
| 4.2.8.2 렌트카 사업자 업체별 대여이력 조회 URL                   | 27 |
| 4.2.8.3 렌트카 사업자 업체별 대여이력 조회 헤더                    | 28 |
| 4.2.8.4 렌트카 사업자 업체별 대여이력 조회 바디                    | 28 |
| 4.2.8.5 렌트카 사업자 업체별 대여이력 조회 예시                    | 29 |
| 4.2.8.6 렌트카 사업자 업체별 대여이력 조회(response) Body 목록     | 30 |
| 4.2.8.6.1 Body                                    | 30 |
| 4.2.9 렌트카 사업자 업체별 반복예약 결과 조회                      | 30 |
| 4.2.9.1 API 호출                                    | 31 |
| 4.2.9.2 렌트카 사업자 업체별 반복예약 결과 조회 URL                | 31 |
| 4.2.9.3 렌트카 사업자 업체별 반복예약 결과 조회 헤더                 | 31 |
| 4.2.9.4 렌트카 사업자 업체별 반복예약 결과 조회 바디                 | 31 |
| 4.2.9.5 렌트카 사업자 업체별 반복예약 결과 조회 예시                 | 32 |
| 4.2.9.6 렌트카 사업자 업체별 반복예약 결과 조회(response) Body 목록  | 33 |
| 4.2.9.6.1 Body                                    | 33 |
| 4.3 소화물배송대행서비스                                    | 34 |
| 4.3.1 소화물배송대행서비스 사업자 운전자격확인 단건 조회                 | 34 |
| 4.3.1.1 API 호출                                    | 34 |
| 4.3.1.2 소화물배송대행서비스 사업자 운전자격확인 단건 조회 URL           | 35 |
| 4.3.1.3 소화물배송대행서비스 사업자 운전자격확인 헤더                  | 35 |
| 4.3.1.4 소화물배송대행서비스 사업자 운전자격확인 바디                  | 35 |

|  |    |
|--|----|
| 4.3.1.5 소화물배송대행서비스 사업자 단건 조회 예시                          | 36 |
| 4.3.1.6 소화물배송대행서비스 사업자 운전자격확인 단건 조회 응답(response) Body 목록 | 37 |
| 4.3.1.6.1 Header   | 37 |
| 4.3.1.6.2 Body   | 38 |
| 4.3.2 소화물배송대행서비스 사업자 운전자격확인 배치 조회                        | 38 |
| 4.3.2.1 API 호출   | 38 |
| 4.3.2.2 소화물배송대행서비스 사업자 운전자격확인 배치 조회 URL                  | 38 |
| 4.3.2.3 소화물배송대행서비스 사업자 운전자격확인 배치 조회 헤더                   | 38 |
| 4.3.2.4 소화물배송대행서비스 사업자 배치 조회 바디                          | 39 |
| 4.3.2.5 소화물배송대행서비스 사업자 배치 조회 예시                          | 39 |
| 4.3.2.6 소화물배송대행서비스 사업자 운전자격확인 배치 조회 응답(response) Body 목록 | 41 |
| 4.3.2.6.1 Header   | 41 |
| 4.3.2.6.2 Body   | 41 |
| 4.3.3 소화물배송대행서비스 사업자 검증번호 생성                             | 41 |
| 4.3.3.1 API 호출   | 41 |
| 4.3.3.2 소화물배송대행서비스 사업자 검증번호 생성 URL                       | 41 |
| 4.3.3.3 소화물배송대행서비스 사업자 검증번호 생성 헤더                        | 42 |
| 4.3.3.4 소화물배송대행서비스 사업자 검증번호 생성 바디                        | 42 |
| 4.3.3.5 소화물배송대행서비스 사업자 검증번호 생성 예시                        | 42 |
| 4.3.3.6 소화물배송대행서비스 사업자 검증번호 생성 응답(response) Body 목록      | 43 |
| 4.3.3.6.1 결과   | 43 |
| 4.3.4 소화물배송대행서비스 사업자 비대면 인증 결과 확인                        | 43 |
| 4.3.4.1 API 호출   | 43 |
| 4.3.4.2 소화물배송대행서비스 사업자 비대면 인증 결과 확인 URL                  | 43 |
| 4.3.4.3 소화물배송대행서비스 사업자 비대면 인증 결과 확인 헤더                   | 43 |
| 4.3.4.4 소화물배송대행서비스 사업자 비대면 인증 결과 확인 바디                   | 44 |
| 4.3.4.5 소화물배송대행서비스 사업자 비대면 인증 결과 확인 예시                   | 44 |
| 4.3.5 소화물배송대행서비스 사업자 비대면 인증 결과 확인 응답(response) Body 목록   | 45 |
| 4.3.5.1.1 Header   | 45 |

|   |    |
|---|----|
| 4.3.5.1.2 Body .....  | 45 |
| 4.4 개인형이동장치 .....   | 46 |
| 4.4.1 개인형이동장치 운전자격확인 단건 조회 .....                                | 46 |
| 4.4.1.1 API 호출 .....  | 46 |
| 4.4.1.2 개인형이동장치 서비스 사업자 운전자격확인 조회 URL .....                     | 46 |
| 4.4.1.3 개인형이동장치 서비스 사업자 운전자격확인 단건 조회 헤더 .....                   | 47 |
| 4.4.1.4 개인형이동장치 서비스 사업자 운전자격확인 단건 조회 바디 .....                   | 47 |
| 4.4.1.5 개인형이동장치 서비스 사업자 운전자격확인 조회 예시 .....                      | 47 |
| 4.4.1.6 개인형이동장치 운전자격확인 조회 응답(response) Body 목록 .....            | 48 |
| 4.4.1.6.1 Header .....  | 48 |
| 4.4.1.6.2 Body .....  | 49 |
| 4.4.2 개인형이동장치서비스 사업자 운전자격확인 배치 조회 .....                         | 49 |
| 4.4.2.1 API 호출 .....  | 49 |
| 4.4.2.2 개인형이동장치 서비스 사업자 운전자격확인 배치 URL .....                     | 49 |
| 4.4.2.3 개인형이동장치 서비스 사업자 운전자격확인 헤더 .....                         | 49 |
| 4.4.2.4 개인형이동장치 서비스 사업자 운전자격확인 바디 .....                         | 50 |
| 4.4.2.5 개인형이동장치 서비스 사업자 배치 조회 예시 .....                          | 51 |
| 4.4.2.6 개인형이동장치 서비스 사업자 운전자격확인 배치 조회 응답(response) Body 목록 ..... | 52 |
| 4.4.2.6.1 Header .....  | 52 |
| 4.4.2.6.2 Body .....  | 52 |
| 4.4.3 개인형이동장치 서비스 사업자 대여번호 생성 .....                             | 52 |
| 4.4.3.1 API 호출 .....  | 52 |
| 4.4.3.2 개인형이동장치 서비스 사업자 대여번호 생성 URL .....                       | 53 |
| 4.4.3.3 개인형이동장치 서비스 사업자 대여번호 생성 헤더 .....                        | 53 |
| 4.4.3.4 개인형이동장치 서비스 사업자 대여번호 생성 바디 .....                        | 53 |
| 4.4.3.5 개인형이동장치 서비스 사업자 대여번호 생성 예시 .....                        | 53 |
| 4.4.3.6 개인형이동장치 서비스 사업자 대여번호 생성 응답(response) Body 목록 .....      | 54 |
| 4.4.3.6.1 결과 .....  | 54 |
| 4.4.4 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 .....                        | 54 |

|   |    |
|---|----|
| 4.4.4.1 API 호출 .....  | 54 |
| 4.4.4.2 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 URL .....                  | 54 |
| 4.4.4.3 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 헤더 .....                   | 55 |
| 4.4.4.4 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버) .....        | 55 |
| 4.4.4.5 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 바디 .....                   | 55 |
| 4.4.4.6 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 예시 .....                   | 55 |
| 4.4.4.7 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 응답(response) Body 목록 ..... | 56 |
| 4.4.4.7.1 Header .....  | 56 |
| 4.4.4.7.2 Body .....  | 57 |
| 4.5 대리운전서비스 .....   | 57 |
| 4.5.1 대리운전 서비스 사업자 운전자격확인 단건 조회 .....                           | 57 |
| 4.5.1.1 API 호출 .....  | 57 |
| 4.5.1.2 대리운전 서비스 사업자 운전자격확인 단건 조회 URL .....                     | 58 |
| 4.5.1.3 대리운전 서비스 사업자 운전자격확인 단건 조회 헤더 .....                      | 58 |
| 4.5.1.4 대리운전 서비스 사업자 운전자격확인 바디 .....                            | 58 |
| 4.5.1.5 대리운전 서비스 사업자 단건 조회 예시 .....                             | 59 |
| 4.5.1.6 대리운전 서비스 사업자 운전자격확인 단건 조회 응답(response) Body 목록 .....    | 60 |
| 4.5.1.6.1 Header .....  | 60 |
| 4.5.1.6.2 Body .....  | 61 |
| 4.5.2 대리운전 서비스 사업자 운전자격확인 배치 조회 .....                           | 61 |
| 4.5.2.1 API 호출 .....  | 61 |
| 4.5.2.2 대리운전 서비스 사업자 운전자격확인 배치 URL .....                        | 61 |
| 4.5.2.3 대리운전 서비스 사업자 운전자격확인 헤더 .....                            | 61 |
| 4.5.2.4 대리운전 서비스 사업자 운전자격확인 바디 .....                            | 62 |
| 4.5.2.5 대리운전 서비스 사업자 배치 조회 예시 .....                             | 63 |
| 4.5.2.6 대리운전 서비스 사업자 운전자격확인 배치 조회 응답(response) Body 목록 .....    | 64 |
| 4.5.2.6.1 Header .....  | 64 |
| 4.5.2.6.2 Body .....  | 64 |
| 4.5.2.7 대리운전 서비스 사업자 검증번호 생성 .....                              | 65 |

|  |    |
|--|----|
| 4.5.2.8 API 호출                                       | 65 |
| 4.5.2.9 대리운전 서비스 사업자 검증번호 생성 URL                     | 65 |
| 4.5.2.10 대리운전 서비스 사업자 검증번호 생성 헤더                     | 65 |
| 4.5.2.11 대리운전 서비스 사업자 검증번호 생성 바디                     | 66 |
| 4.5.2.12 대리운전 서비스 사업자 검증번호 생성 예시                     | 66 |
| 4.5.2.13 대리운전 서비스 사업자 검증번호 생성 응답(response) Body 목록   | 67 |
| 4.5.2.13.1 결과  | 67 |
| 4.5.3 대리운전 서비스 사업자 비대면 인증 결과 확인                      | 67 |
| 4.5.3.1 API 호출                                       | 67 |
| 4.5.3.2 대리운전 서비스 사업자 비대면 인증 결과 확인 URL                | 67 |
| 4.5.3.3 대리운전 서비스 사업자 비대면 인증 결과 확인 헤더                 | 67 |
| 4.5.3.4 대리운전 서비스 사업자 비대면 인증 결과 확인 바디                 | 67 |
| 4.5.3.5 대리운전 서비스 사업자 비대면 인증 결과 확인 예시                 | 68 |
| 4.5.4 대리운전 서비스 사업자 비대면 인증 결과 확인 응답(response) Body 목록 | 69 |
| 4.5.4.1.1 Header                                     | 69 |
| 4.5.4.1.2 Body                                       | 69 |
| 4.6 범용   | 70 |
| 4.6.1 범용 운전자격확인 단건 조회                                | 70 |
| 4.6.1.1 API 호출                                       | 70 |
| 4.6.1.2 범용 서비스 사업자 운전자격확인 단건 조회 URL                  | 70 |
| 4.6.1.3 범용 운전자격확인 단건 조회 헤더                           | 70 |
| 4.6.1.4 범용 운전자격확인 바디                                 | 70 |
| 4.6.1.5 범용 단건 조회 예시                                  | 71 |
| 4.6.1.6 범용 운전자격확인 단건 조회 응답(response) Body 목록         | 72 |
| 4.6.1.6.1 Header                                     | 72 |
| 4.6.1.6.2 Body                                       | 73 |
| 4.6.2 범용 운전자격확인 배치 조회                                | 73 |
| 4.6.2.1 API 호출                                       | 73 |
| 4.6.2.2 범용 운전자격확인 배치 URL                             | 73 |

|  |    |
|--|----|
| 4.6.2.3 범용 운전자격확인 헤더 .....                         | 73 |
| 4.6.2.4 범용 운전자격확인 바디 .....                         | 74 |
| 4.6.2.5 범용 배치 조회 예시 .....                          | 75 |
| 4.6.2.6 범용 운전자격확인 배치 조회 응답(response) Body 목록 ..... | 76 |
| 4.6.2.6.1 Header .....                             | 76 |
| 4.6.2.6.2 Body .....                               | 76 |
| 4.6.3 범용 검증번호 생성 .....                             | 77 |
| 4.6.3.1 API 호출 .....                               | 77 |
| 4.6.3.2 범용 검증번호 생성 URL .....                       | 77 |
| 4.6.3.3 범용 검증번호 생성 헤더 .....                        | 77 |
| 4.6.3.4 범용 검증번호 생성 바디 .....                        | 77 |
| 4.6.3.5 범용 검증번호 생성 예시 .....                        | 78 |
| 4.6.3.6 범용 검증번호 생성 응답(response) Body 목록 .....      | 79 |
| 4.6.3.6.1 결과 .....                                 | 79 |
| 4.6.4 범용 비대면 인증 결과 확인 .....                        | 79 |
| 4.6.4.1 API 호출 .....                               | 79 |
| 4.6.4.2 범용 비대면 인증 결과 확인 URL .....                  | 79 |
| 4.6.4.3 범용 비대면 인증 결과 확인 헤더 .....                   | 79 |
| 4.6.4.4 범용 비대면 인증 결과 확인 바디 .....                   | 80 |
| 4.6.4.5 범용 비대면 인증 결과 확인 예시 .....                   | 80 |
| 4.6.5 범용 비대면 인증 결과 확인 응답(response) Body 목록 .....   | 81 |
| 4.6.5.1.1 Header .....                             | 81 |
| 4.6.5.1.2 Body .....                               | 81 |
| 4.7 공통 API 오류 응답 예시 .....                          | 82 |
| 5 첨부표 .....  | 82 |
| 5.1 공통 API 오류 응답 코드표 .....                         | 83 |
| 5.2 면허정보 응답 코드표 .....                              | 84 |
| 5.3 요청처리 응답 코드표 .....                              | 85 |
| 5.4 면허 종별 코드표 .....                                | 86 |

|                          |    |
|--------------------------|----|
| 5.5 지역 코드표 .....         | 86 |
| 5.6 계정 상태 코드표 .....      | 87 |
| 5.7 승인 상태 코드표 .....      | 87 |
| 5.8 얼굴 활성도 판정 기준표 .....  | 87 |
| 5.9 얼굴 유사도 판정 기준표 .....  | 88 |
| 5.10 반복 주기 코드 .....      | 88 |
| 5.11 면허 사진 검증 코드표 .....  | 88 |
| 5.12 범용 API 유형 코드표 ..... | 89 |
| 5.13 검색유형 코드표 .....      | 90 |
| 6 개발 예시 .....            | 90 |

# 1 개요

## 1.1 목적

- 운전자격확인시스템은 대여사업자, 플랫폼사, 소화물 배송대행 서비스 사업자 및 지자체 담당자를 주 대상으로 하며, 대여 전 운전자의 자격을 확인하는 기능을 제공한다. 본 문서는 운전자격확인 API를 호출할 때의 통신규약을 정의 한다.

# 2 목록

## 2.1 용어정의

- 본 문서에서 사용하는 용어는 다음과 같다.

| 용어   | 설명          |
|------|-------------|
| RIMS | - 운전자격확인시스템 |

## 2.2 기능정의

- 사용자는 인증키와 Secret Key를 시스템(<https://rims.kotsa.or.kr>)의 OpenAPI 메뉴에서 발급받으며, 이때 개발용과 운영용 인증키가 함께 발급된다.
- 개발키는 즉시 사용할 수 있으며, 운영키는 별도의 승인 절차를 거친 후 사용 가능하다.
- “3.5. 데이터 Request 목록” 기준으로 키 구분별 API 사용이 제한된다.
- 사용자는 인증키를 이용해 인증 토큰을 발급받는다.
- 토큰의 유효 기간은 3시간이다.
- 인증 토큰은 발급 시마다 새롭게 생성되며, 유효 기간이 초기화된다.
- 사용자는 발급받은 토큰을 첨부하여 운전자격확인 API를 호출한다.
- 시스템은 사용자의 호출 정보를 기반으로 한국도로교통공단의 운전자격확인 API를 호출해 결과를 받는다.
- 시스템은 이력을 저장하고, 검증 결과를 사용자에게 반환한다.

# 3 공통 설계

## 3.1 송신 데이터의 형식

- 데이터는 암호화 후 Base64로 인코딩하여 “encryptedData” 를 key로 Body에 JSON타입으로 구성한다.
- 데이터 송수신은 HTTP 1.1 방식을 따르며, 몇 가지 추가적인 HTTP 헤더를 정의하여 사용한다.

## 3.2 보안 방법

### 3.2.1 암호화 알고리즘

- AES (Advanced Encryption Standard) 암호화 알고리즘을 사용하여 암호화하고 데이터는 Base64 인코딩한다.
- 이때, 표준 옵션인 ECB 모드와 PKCS5 패딩으로 진행해야 한다.

### 3.2.2 암호화 키 정보

- 시스템에서 발급받은 Secret Key 정보를 통해 데이터를 암호화 한다.

### 3.2.3 암호화 알고리즘 제공 방식

- 각 언어의 표준 라이브러리 또는 서드파티 라이브러리 문서를 참조한다.

## 3.3 렌트카 운전자격확인 API 사용 시 차량 정보

- 자격 확인 시 차량번호는 필수 항목이며, 차량 배치나 차량번호가 미정인 경우 임시 차량번호 “99임9999”를 입력한다.
- 전산(국토교통부 자동차관리 전산망)에 등록된 차량의 경우, 차량 확인 코드(vhcl\_idnty\_cd)는 1로 반환되며, 이는 차량이 확인되었음을 의미한다.
- 전산에 등록된 차량은 요청 시 bizinfo 항목이 첨부된 경우 해당 사업자로 등록된 차량을 기준으로 하며, 첨부되지 않은 경우 요청자의 사업자 차량을 기준으로 한다.
- 차량 확인 코드(vhcl\_idnty\_cd)가 2(미확인)로 반환된 경우, 해당 차량의 확인 유예 기간은 한 달로 적용된다.
- 대여확정 API에서는 임시 차량 번호 “99임9999”를 사용할 수 없다.

## 3.4 운전자격확인 API와 대여확정 API 분리

- 기존 운전자격확인 단건 및 배치 API 호출 시 운전자격 확인 이력과 대여 이력이 함께 저장되던 방식을 별도의 API로 분리한다.
- 대여확정 API 호출 시 운전자격 확인과 대여 이력을 저장하며, RIMS 홈페이지의 “운전자격 확인 이력 조회” 메뉴와 “대여정보 관리” 메뉴에서 각각 확인할 수 있다.
- 대여확정 API에서는 차량번호가 필수 항목이며, “99임9999” 임시 번호는 사용 할 수 없다.
- 상세 요청 및 응답 항목은 대여확정 API 문서를 참고한다.

## 3.5 데이터 Request 목록

### 3.5.1 request 목록

| 구분       | 인터페이스명           | Request uri                                  | 구분    | 내용  |
|----------|------------------|--|-------|---|
| 공통       | токن 발급          | /col/oauth2                                  | 개발/운영 | токن 발급을 위한 인터페이스                             |
|          | 사용자검증            | /col/isUserInDatabase                        | 개발/운영 | 시스템에 등록된 사용자인지 검증에 대한 인터페이스                   |
| 렌트카      | 운전자격확인 단건        | /licenseVerification                         | 개발/운영 | 렌터카 사업자용 운전자격확인조회를 위한 인터페이스                   |
|          | 운전자격확인 배치        | /licenseVerificationBatch                    | 개발/운영 | 렌터카 사업자용 복수의 운전자에 대한 운전자격확인조회를 위한 인터페이스       |
|          | 대여확정             | /rentalConfirm                               | 운영    | 렌터카 사업자용 대여 확정에 대한 인터페이스                      |
|          | 반복예약             | /licenseVerificationSchedule                 | 운영    | 렌터카 사업자용 운전자격확인조회의 반복 예약을 위한 인터페이스            |
|          | 대여자 대여정보 이상여부 확인 | /rental/anomaly                              | 운영    | 렌터카 사업자용 대여자의 최근 7일간 대여 건수를 확인하기 위한 인터페이스     |
|          | 대여번호 생성          | /generateRentNo                              | 운영    | 렌터카 사업자용 비대면 인증 대여번호를 생성하기 위한 인터페이스           |
|          | 비대면 인증 결과 확인     | /licenseVerificationResult                   | 운영    | 렌터카 사업자용 비대면 인증 결과 확인을 위한 인터페이스               |
|          | 업체별 대여이력 조회      | /company/rentals/history                     | 개발/운영 | 렌터카 사업자용 업체별 대여이력 조회를 위한 인터페이스                |
|          | 업체별 반복예약 결과 조회   | /company/licenseVerificationSchedule/history | 개발/운영 | 렌터카 사업자용 업체별 반복예약 결과 조회를 위한 인터페이스             |
| 소화물배송 대행 | 운전자격확인 단건        | /dapi/licenseVerification                    | 개발/운영 | 소화물배송대행서비스사업자용 운전자격확인조회를 위한 인터페이스             |
|          | 운전자격확인 배치        | /dapi/licenseVerificationBatch               | 개발/운영 | 소화물배송대행서비스사업자용 복수의 운전자에 대한 운전자격확인조회를 위한 인터페이스 |
|          | 검증번호 생성          | /dapi/generateVrfcNo                         | 운영    | 소화물배송대행서비스사업자용 비대면 인증 검증번호 생성하기 위한 인터페이스      |
|          | 비대면 인증 결과 확인     | /dapi/licenseVerificationResult              | 운영    | 소화물배송대행서비스사업자용 비대면 인증 결과 확인을 위한 인터페이스         |
| 개인형이동장치  | 운전자격확인 단건        | /pm/licenseVerification                      | 개발/운영 | 개인형이동장치사업자용 운전자격확인조회를 위한 인터페이스                |
|          | 운전자격확인 배치        | /pm/licenseVerification                      | 개발/운  | 개인형이동장치사업자용                                   |

|              |                                |       |   |
|--------------|--------------------------------|-------|---|
|              | Batch                          | 영     | 복수의 운전자에 대한 운전자격확인조회를 위한 인터페이스          |
| 대여번호 생성      | /pm/generateRentNo             | 운영    | 개인형이동장치사업자용 비대면 인증 대여번호를 생성하기 위한 인터페이스  |
| 비대면 인증 결과 확인 | /pm/licenseVerification Result | 운영    | 개인형이동장치사업자용 비대면 인증 결과 확인을 위한 인터페이스      |
| 대리운전         | 운전자격확인 단건                      | 개발/운영 | 대리운전사업자용 운전자에 대한 운전자격확인조회를 위한 인터페이스     |
|              | 운전자격확인 배치                      | 개발/운영 | 대리운전사업자용 복수의 운전자에 대한 운전자격확인조회를 위한 인터페이스 |
|              | 검증번호 생성                        | 운영    | 대리운전사업자용 비대면 인증 검증번호 생성하기 위한 인터페이스      |
|              | 비대면 인증 결과 확인                   | 운영    | 대리운전사업자용 비대면 인증 결과 확인을 위한 인터페이스         |
| 범용           | 운전자격확인 단건                      | 개발/운영 | 범용(기타) 운전자에 대한 운전자격확인조회를 위한 인터페이스       |
|              | 운전자격확인 배치                      | 개발/운영 | 범용(기타) 복수의 운전자에 대한 운전자격확인조회를 위한 인터페이스   |
|              | 검증번호 생성                        | 운영    | 범용(기타) 비대면 인증 검증번호 생성하기 위한 인터페이스        |
|              | 비대면 인증 결과 확인                   | 운영    | 범용(기타) 비대면 인증 결과 확인을 위한 인터페이스           |

## 4 서비스별 API

### 4.1 공통

#### 4.1.1 토큰발급

- OAuth2 인증 방식에 따라 제공된 인증키로 인증 토큰을 발급받고, 해당 토큰을 사용하여 운전자격확인 API를 호출한다.

##### 4.1.1.1 인증 URL

| 환경 | API명     | URL   |
|----|----------|---|
| 운영 | токен 발급 | <a href="https://rims.kotsa.or.kr:8114/col/oauth2">https://rims.kotsa.or.kr:8114/col/oauth2</a> |

##### 4.1.1.2 토큰 발급 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고             |
|---------------|-----|-----------|------|----------------|
| host          | Req | 웹서버 정보    | 생략가능 |                |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Basic " + 인증키 |

- 토큰 발급 시 HTTP Request (클라이언트 → 인증서버)

```
GET /col/oauth2?grantType=password
Host: rims.kotsa.or.kr
Authorization: "Basic " + Base64 인코딩(인증키)
ex) Basic YTFjO...
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 위 표의 빨간색 글꼴이 있는 부분은 파라미터로 포함한다.
- 위 표의 파란색 글꼴이 있는 부분은 인증키를 Base64로 인코딩한다.
- 요청 본문은 사용하지 않는다.
- 토큰 발급 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

Authorization: Bearer 토큰 값

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.

## 4.1.2 사용자검증

### 4.1.2.1 API 호출

- 사용자ID와 사업자등록번호로 시스템에 등록된 사용자인지 검증한다.
- 플랫폼, 대행사, 조합 등 API인증키의 사용자와 차량 소유주가 일치하지 않을 경우 사용자정보(bizinfo) 값의 정합성 체크를 위한 API

### 4.1.2.2 사용자검증 URL

| 환경 | API명  | URL  |
|----|-------|--|
| 운영 | 사용자검증 | https://rims.kotsa.or.kr:8114/col/isUserInDatabase |

### 4.1.2.3 사용자검증 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 사용자검증 시 HTTP Request (클라이언트 → 인증서버)

```
GET /col/isUserInDatabase?user_id=test&brno=1234567890
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 위 표의 빨간색 글꼴이 있는 부분은 파라미터로 포함한다.
- 위 표의 파란색 글꼴이 있는 부분은 사용자 ID, 사업자등록번호 순으로 포함한다.
- 사용자검증 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.1.2.4 사용자검증 응답 예시

응답샘플

HTTP/1.1 200 OK

등록

```
{
  "aprvt_stts_cd": "2",
  "acnt_stts_cd": "1",
  "user_id": "test",
  "registration_status": "registered",
  "brno": "1234567890"
}
```

미등록

```
{
  "user_id": "test",
  "registration_status": "unregistered",
  "brno": "1234567890"
}
```

#### 4.1.2.5 사용자 검증 조회 응답(response) Body 목록

| 항목명     | 항목 Key              | 항목 형식  | 내용                                |
|---------|---------------------|--------|-----------------------------------|
| 승인상태코드  | aprv_stts_cd        | String | 승인 상태 코드표 참조                      |
| 계정상태코드  | acnt_stts_cd        | String | 계정 상태 코드표 참조                      |
| 사용자ID   | user_id             | String | 요청 사용자ID                          |
| 등록여부    | registration_status | String | unregistered(미등록), registered(등록) |
| 사업자등록번호 | brno                | String | 요청 사업자등록번호                        |

- 응답은 “4.1.2.4 사용자검증 응답 예시”를 참고한다.
- 계정상태코드는 “5.6 계정 상태 코드표”를 참고한다.
- 승인상태코드는 “5.7 승인 상태 코드표”를 참고한다.

### 4.1.3 비대면 운전자격확인 딥링크

#### 4.1.3.1 API 호출

- 발급받은 Token을 통해 검증번호 또는 대여번호를 생성한다. (도메인별 검증 번호 또는 대여번호 생성 참조)

#### 4.1.3.2 비대면 운전자격확인 딥링크 URL

| 환경 | API명              | URL   |
|----|-------------------|---|
| 운영 | 비대면 운전자격확인<br>딥링크 | <a href="https://rims.kotsa.or.kr/ma/openMobileWeb?mwverf={대여번호}">https://rims.kotsa.or.kr/ma/openMobileWeb?mwverf={대여번호}</a> |

#### 4.1.3.3 비대면 운전자격확인 딥링크 예시

<https://rims.kotsa.or.kr/ma/openMobileWeb?mwverf=R25-123456789>

대여번호 예시 : (V(소화물),R(렌트카),S(대리운전),P(PM))25-123456789

- 생성한 비대면 운전자격확인 딥링크 URL을 사용자에게 전달하여 비대면 운전자격확인을 진행한다.

## 4.2 렌트카

### 4.2.1 렌트카 사업자 운전자격확인 단건 조회

#### 4.2.1.1 API 호출

- 발급받은 Token을 통해 대여자의 운전 자격을 확인한다.

#### 4.2.1.2 렌트카 사업자 운전자격확인 단건 조회 URL

| 환경 | API명                 | URL   |
|----|----------------------|---|
| 운영 | 렌트카 사업자 운전자격확인 단건 조회 | <a href="https://rims.kotsa.or.kr:8114/licenseVerification">https://rims.kotsa.or.kr:8114/licenseVerification</a> |

#### 4.2.1.3 렌트카 사업자 운전자격확인 단건 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /licenseVerification  
Host: rims.kotsa.or.kr  
Authorization: "Bearer " + 토큰  
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)  
HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.2.1.4 렌트카 사업자 운전자격확인 단건 조회 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 사용자정보         | bizinfo         | String | 32 | 필수여부 | API 호출 사용자와 일치 할 경우 생략 가능                           |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 필수   | 임시차량의 경우<br>“99임9999”                               |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.

#### 4.2.1.5 렌트카 사업자 단건 조회 예시

```
POST https://rims.kotsa.or.kr:8114/licenseVerification
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_license_no": "221212121212",
  "f_resident_name": "홍길동",
  "f_licn_con_code": "12",
  "f_from_date": "20241023",
  "f_to_date": "20241023",
  "vhcl_reg_no": "67라6767"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header": {
    "f rtn msg": "",
    "f request date": "20231010",
    "f rtn cd": "0",
    "f send cnt": "1"
  },
  "body": {
    "vhcl idnty cd": "1",
    "f rtn code": "00",
    "f license no": "2*1*1*1*2*2"
  }
}
```

#### 4.2.1.6 렌트카 사업자 운전자격확인 단건 조회(response) Body 목록

##### 4.2.1.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn msg      | String | 요청처리실패메시지            |

##### 4.2.1.6.2 Body

| 항목명    | 항목 Key        | 항목 형식  | 내용                             |
|--------|---------------|--------|--------------------------------|
| 차량확인코드 | vhcl idnty cd | String | (1:확인, 2:미확인)                  |
| 검증결과코드 | f rtn_code    | String | 검증결과코드 (적격 : 00, 부적격 : 01~)    |
| 운전면허번호 | f license_no  | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.2.1.5 렌트카 사업자 단건 조회 예시”를 참고한다.
- “4.2.1.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.2.1.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.2.1.6.2 Body”는 생략된다.

## 4.2.2 렌트카 사업자 운전자격확인 배치 조회

### 4.2.2.1 API 호출

- 발급받은 Token을 통해 복수의 면허소유자를 검증한다.

### 4.2.2.2 렌트카 사업자 운전자격확인 배치 URL

| 환경 | API명                 | URL  |
|----|----------------------|--|
| 운영 | 렌트카 사업자 운전자격확인 배치 조회 | https://rims.kotsa.or.kr:8114/licenseVerificationBatch |

### 4.2.2.3 렌트카 사업자 운전자격확인 배치 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /licenseVerificationBatch
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

### 4.2.2.4 렌트카 사업자 운전자격확인 배치 조회 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 요청건수          | f_send_cnt      | String | 4  | 필수   | 리스트 인덱스와 일치(최대 1,000개)                              |
| 사용자정보         | bizinfo         | String | 32 | 필수   | API 호출 사용자와 일치 할 경우 생략 가능                           |
| 요청리스트         | requestList     | String |    | 필수   | 복수 요청의 키  |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 필수   | 임시차량의 경우<br>“99임9999”                               |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- 요청건수는 요청리스트의 인덱스와 일치해야 한다.

#### 4.2.2.5 렌트카 사업자 배치 조회 예시

```
POST https://rims.kotsa.or.kr:8114/licenseVerification
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "f_send_cnt": "2",
  "bizinfo": "asdf12341....",
  "requestList": [
    {
      "f_license_no": "221212121212",
      "f_resident_name": "홍길동",
      "f_licn_con_code": "12",
      "f_from_date": "20241023",
      "f_to_date": "20241023",
      "vhcl_reg_no": "67라6767"
    }
  ]
}
```

```

},
{
    "f_license_no": "221212121212",
    "f_resident_name": "홍길동",
    "f_licn_con_code": "12",
    "f_from_date": "20241023",
    "f_to_date": "20241023"
    "vhcl_reg_no": "67라6767"
}]
}

```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "header" : {
        "f_send_cnt" : "2",
        "f_request_date" : "20220922122321",
        "f rtn_cd" : "00",
        "f rtn_msg" : ""
    }
    "body" : [
        {
            "vhcl_idnty_cd": "1",
            "f_license_no" : "121234561212",
            "f rtn_code" : "25"
        },
        {
            "vhcl_idnty_cd": "1",
            "f_license_no" : "121234561212",
            "f rtn_code" : "00"
        }
    ]
}
```

#### 4.2.2.6 렌트카 사업자 운전자격확인 배치 조회 응답(response) Body 목록

##### 4.2.2.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.2.2.6.2 Body

| 항목명    | 항목 Key        | 항목 형식  | 내용                             |
|--------|---------------|--------|--------------------------------|
| 차량확인코드 | vhcl_idnty_cd | String | (1:확인, 2:미확인)                  |
| 검증결과코드 | f rtn_code    | String | 검증결과코드 (직격 : 00, 부직격 : 01~)    |
| 운전면허번호 | f_license_no  | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.2.2.5 배치 조회 예시”를 참고한다.
- “4.2.2.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.2.2.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.2.2.6.2 Body”는 생략된다.

### 4.2.3 렌트카 사업자 대여확정

#### 4.2.3.1 API 호출

- 발급받은 Token을 통해 차량의 대여를 확정한다.
- API 호출 시 운전자격 검증을 진행하고 검증 실패 시 검증 이력은 저장되지만, 대여 이력은 저장되지 않는다.

#### 4.2.3.2 렌트카 사업자 대여확정 URL

| 환경 | API명            | URL   |
|----|-----------------|---|
| 운영 | 렌트카 사업자<br>대여확정 | <a href="https://rims.kotsa.or.kr:8114/rentalConfirm">https://rims.kotsa.or.kr:8114/rentalConfirm</a> |

#### 4.2.3.3 렌트카 사업자 대여확정 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 대여확정 시 HTTP Request (클라이언트 → 인증서버)

POST /rentalConfirm

```

Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr

```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 대여확정 시 인증서버의 HTTP Response (인증서버 → 클라이언트)  
HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.2.3.4 렌트카 사업자 대여확정 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부                      | 내용  |
|---------------|-----------------|--------|----|---------------------------|---|
| 사용자정보         | bizinfo         | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID<br>( Y : 편도,<br>N : 왕복,<br>생략시 null)          |
| 편도 여부         | one_way_yn      | String | 1  | 생략가능                      |   |
| 운전면허번호        | f_license_no    | String | 12 | 필수                        |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수                        |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수                        |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수                        | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수                        |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 필수                        | 임시차량번호 불가   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.

#### 4.2.3.5 렌트카 사업자 대여확정 예시

```

POST https://rims.kotsa.or.kr:8114/rentalConfirm
Authorization: Bearer ...

```

```
Content-Type: application/json  
Content-Length: ...
```

암호화 전

```
{  
    "bizinfo": "asdf12341...",  
    "f_license_no": "221212121212",  
    "f_resident_name": "홍길동",  
    "f_licn_con_code": "12",  
    "f_from_date": "20241023",  
    "f_to_date": "20241023",  
    "vhcl_reg_no": "67라6767",  
    "one_way_yn": "Y"  
}
```

암호화 후

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "header": {  
        "f rtn msg": "",  
        "f request date": "20231010",  
        "f rtn cd": "0",  
        "f send cnt": "1"  
    },  
    "body": {  
        "f rtn code": "00",  
        "f license no": "2*1*1*1*2*2"  
    }  
}
```

### 4.2.3.6 렌트카 사업자 대여확정 응답(response) Body 목록

#### 4.2.3.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.2.3.6.2 Body

| 항목명    | 항목 Key        | 항목 형식  | 내용                             |
|--------|---------------|--------|--------------------------------|
| 차량확인코드 | vhcl_idnty_cd | String | (1:확인, 2:미확인)                  |
| 검증결과코드 | f rtn_code    | String | 검증결과코드 (적격 : 00, 부적격 : 01~)    |
| 운전면허번호 | f license_no  | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.2.3.5 렌트카 대여확정 예시”를 참고한다.
- “4.2.3.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.2.3.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.2.3.6.2 Body”는 생략된다.

### 4.2.4 렌트카 사업자 반복예약

#### 4.2.4.1 API 호출

- 발급받은 Token을 통해 렌터카 사업자용 운전자격확인조회를 반복 예약 한다.

#### 4.2.4.2 렌트카 사업자 반복예약 URL

| 환경 | API명            | URL   |
|----|-----------------|---|
| 운영 | 렌트카 사업자<br>반복예약 | <a href="https://rims.kotsa.or.kr:8114/licenseVerificationSchedule">https://rims.kotsa.or.kr:8114/licenseVerificationSchedule</a> |

#### 4.2.4.3 렌트카 사업자 반복예약 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 반복예약 시 HTTP Request (클라이언트 → 인증서버)

```
POST /licenseVerificationSchedule
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.

- 반복예약 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.2.4.4 렌트카 사업자 반복예약 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 사용자정보         | bizinfo         | String | 32 | 필수여부 | API 호출 사용자와 일치 할 경우 생략 가능                           |
| 반복주기          | repeat_period   | String | 1  | 필수   | 반복주기 코드(1: 일주일, 2:보름, 3: 한 달)                       |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 필수   | 임시차량번호 불가   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.

#### 4.2.4.5 렌트카 사업자 반복예약 예시

```
POST https://rims.kotsa.or.kr:8114/licenseVerificationSchedule
Authorization: Bearer ...
Content-Type: application/json
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "repeat_period": "1",
  "f_license_no": "221212121212",
  "f_resident_name": "홍길동",
  "f_licn_con_code": "12",
  "f_from_date": "20241023",
  "f_to_date": "20241023",
  "vhcl_reg_no": "67라6767"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header": {
    "f rtn msg": "",
    "f request date": "20231010",
    "f rtn cd": "0",
    "f send cnt": "1"
  }
}
```

```

},
"body": {
    "f rtn code": "00",
    "f license no": "2*1*1*1*2*2"
},
"repeat status": "success"
}

```

#### 4.2.4.6 렌트카 사업자 반복예약 응답(response) Body 목록

##### 4.2.4.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn msg      | String | 요청처리실패메시지            |

##### 4.2.4.6.2 Body

| 항목명    | 항목 Key        | 항목 형식  | 내용                             |
|--------|---------------|--------|--------------------------------|
| 차량확인코드 | vhcl_idnty_cd | String | (1:확인, 2:미확인)                  |
| 검증결과코드 | f rtn_code    | String | 검증결과코드 (적격 : 00,<br>부적격 : 01~) |
| 운전면허번호 | f license_no  | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |
| 예약결과상태 | repeat status | String | (success: 성공,<br>failed : 실패)  |

- 응답은 “4.2.4.5 렌트카 사업자 반복예약 예시”를 참고한다.
- “4.2.4.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.2.4.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.2.4.6.2 Body”는 생략된다.

## 4.2.5 렌트카 사업자 대여자 대여정보 이상여부 확인

### 4.2.5.1 API 호출

- 발급받은 Token을 통해 대여자 대여정보 이상여부를 확인

### 4.2.5.2 렌트카 사업자 대여자 대여정보 이상여부 확인 URL

| 환경 | API명                     | URL   |
|----|--------------------------|---|
| 운영 | 렌트카 사업자 대여자 대여정보 이상여부 확인 | <a href="https://rims.kotsa.or.kr:8114/rental/anomaly">https://rims.kotsa.or.kr:8114/rental/anomaly</a> |

### 4.2.5.3 렌트카 사업자 대여자 대여정보 이상여부 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 대여자 대여정보 이상여부 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /rental/anomaly
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 대여자 대여정보 이상여부 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

### 4.2.5.4 렌트카 사업자 대여자 대여정보 이상여부 확인 바디

| 항목명    | 항목 Key       | 항목 형식  | 길이 | 필수여부 | 내용 |
|--------|--------------|--------|----|------|----|
| 운전면허번호 | f_license_no | String | 12 | 필수   |    |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역명(서울, 인천 등)이 포함된 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.

#### 4.2.5.5 렌트카 사업자 대여자 대여정보 이상여부 확인 예시

```
POST https://rims.kotsa.or.kr:8114/rental/anomaly  
Authorization: Bearer ...  
Content-Type: application/json  
Content-Length: ...
```

암호화 전

```
{  
    "f_license_no": "123456789123"  
}
```

암호화 후

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "result" : 11  
}
```

#### 4.2.5.6 렌트카 사업자 대여자 대여정보 이상여부 확인 응답(response) Body 목록

##### 4.2.5.6.1 결과

| 항목명 | 항목 Key | 항목 형식   | 내용    |
|-----|--------|---------|-------|
| 결과  | result | Integer | 대여 횟수 |

- 응답은 “4.2.5.5 렌트카 사업자 대여자 대여정보 이상여부 확인 예시”를 참

고한다.

#### 4.2.6 렌트카 사업자 대여번호 생성

##### 4.2.6.1 API 호출

- 발급받은 Token을 통해 대여번호를 생성한다.

##### 4.2.6.2 렌트카 사업자 대여번호 생성 URL

| 환경 | API명    | URL   |
|----|---------|---|
| 운영 | 대여번호 생성 | <a href="https://rims.kotsa.or.kr:8114/generateRentNo">https://rims.kotsa.or.kr:8114/generateRentNo</a> |

##### 4.2.6.3 렌트카 사업자 대여번호 생성 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 대여번호 생성 시 HTTP Request (클라이언트 → 인증서버)

```
POST /generateRentNo
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 대여번호 생성 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

##### 4.2.6.4 렌트카 사업자 대여번호 생성 바디

| 항목명    | 항목 Key       | 항목 형식  | 길이 | 필수여부                      | 내용                                     |
|--------|--------------|--------|----|---------------------------|--|
| 사용자정보  | bizinfo      | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID                                  |
| 대여시작일  | rent_bgng_dt | String | 8  | 필수                        | (yyyyMMdd)<br>대여 시작일은 오늘을 포함한 이후여야 한다. |
| 대여종료일  | rent_end_dt  | String | 8  | 필수                        |  |
| 차량등록번호 | vhcl_reg_no  | String | 16 | 필수                        | 임시차량번호 불가                              |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.

#### 4.2.6.5 렌트카 사업자 대여번호 생성 예시

```
POST https://rims.kotsa.or.kr:8114/generateRentNo
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
    "bizinfo": "asdf12341....",
    "vhcl_reg_no": "12합1234",
    "rent_bgng_dt": "20240926",
    "rent_end_dt": "20250925"
}
```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "rent_no": "R24-000924523"
}
```

#### 4.2.6.6 렌트카 사업자 대여번호 생성 응답(response) Body 목록

##### 4.2.6.6.1 결과

| 항목명  | 항목 Key  | 항목 형식  | 내용                   |
|------|---------|--------|----------------------|
| 대여번호 | rent_no | String | (형식 : R00-123456789) |

- 응답은 “4.2.6.5 렌트카 사업자 대여번호 생성 예시”를 참고한다.

## 4.2.7 렌트카 사업자 비대면 인증 결과 확인

### 4.2.7.1 API 호출

- 발급받은 Token을 통해 비대면 인증 결과를 조회한다.

### 4.2.7.2 렌트카 사업자 비대면 인증 결과 확인 URL

| 환경 | API명                 | URL   |
|----|----------------------|---|
| 운영 | 렌트카 사업자 비대면 인증 결과 확인 | <a href="https://rims.kotsa.or.kr:8114/licenseVerificationResult">https://rims.kotsa.or.kr:8114/licenseVerificationResult</a> |

### 4.2.7.3 렌트카 사업자 비대면 인증 결과 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /licenseVerificationResult
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

### 4.2.7.4 렌트카 사업자 비대면 인증 결과 확인 바디

| 항목명   | 항목 Key  | 항목 형식  | 길이 | 필수여부 | 내용                        |
|-------|---------|--------|----|------|---------------------------|
| 대여번호  | rent_no | String | 13 | 필수   | 대여번호                      |
| 사용자정보 | bizinfo | String | 32 | 필수   | API 호출 사용자와 일치 할 경우 생략 가능 |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 대여번호는 “4.2.6 렌트카 대여번호 생성” 호출 후 응답의 대여번호(rent\_no)를 사용한다.

#### 4.2.7.5 렌트카 사업자 비대면 인증 결과 확인 예시

```
POST https://rims.kotsa.or.kr:8114/licenseVerificationResult
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "rent_no": "R24-000924523"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header": {
    "rent_no": "R25-000000843",
    "f rtn_cd": "0",
    "f request_date": "20250908",
    "f send_cnt": "1"
  },
  "body": {
    "vhcl_idnty_cd": "2",
    "f rtn_code": "00",
    "face_similarity": 99.0,
    "face_liveness": 99.0,
    "original_reliability": 90.0,
    "police_lcns_vrfc_cd": "1"
  }
}
```

}

#### 4.2.7.6 렌트카 사업자 비대면 인증 결과 확인 응답(response) Body 목록

##### 4.2.7.6.1 Header

| 항목명  | 항목 Key         | 항목 형식  | 내용                       |
|------|----------------|--------|--------------------------|
| 대여번호 | rent_no        | String | 대여번호(형식 : R00-123456789) |
| 요청건수 | f_send_cnt     | String | 요청건수(Default 1)          |
| 요청일시 | f_request_date | String | 요청일시 (형식 : YYYYMMDD)     |
| 요청처리 | f rtn_cd       | String | 요청처리코드                   |

##### 4.2.7.6.2 Body

| 항목명      | 항목 Key               | 항목 형식  | 내용                          |
|----------|----------------------|--------|-----------------------------|
| 차량확인코드   | vhcl_idnty_cd        | String | (1:확인, 2:미확인)               |
| 검증결과코드   | f rtn_code           | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 얼굴유사도    | face_similarity      | Double | 얼굴유사도 수치 (0~100)            |
| 얼굴활성도    | face_liveness        | Double | 얼굴활성도 수치 (0~100)            |
| 원본신뢰도    | original_reliability | Double | 원본신뢰도 수치 (0~100)            |
| 면허사진검증코드 | police_lcns_vrfc_cd  | String | 면허사진검증코드(적격 : 1)            |

- 응답은 “4.2.7.5 렌트카 비대면 인증 결과 확인 예시”를 참고한다.
- “4.2.7.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.2.7.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.2.7.6.2 Body”는 생략된다.
- “4.2.7.6.2 Body”의 f rtn\_code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “4.2.7.6.2 Body”의 face\_similarity는 “5.9 얼굴 유사도 판정 기준표”를 참고하며 활용 시 12단계 이상을 권장한다.
- “4.2.7.6.2 Body”의 face\_liveness는 “5.8 얼굴 활성도 판정 기준표”를 참고하며 활용 시 4단계 이상을 권장한다.
- “4.2.7.6.2 Body”의 police\_lcns\_vrfc\_cd는 “5.11 면허 사진 검증 코드표”를 참고한다.

#### 4.2.8 렌트카 사업자 업체별 대여이력 조회

##### 4.2.8.1 API 호출

- 발급받은 Token을 통해 업체별 대여이력을 조회한다.

##### 4.2.8.2 렌트카 사업자 업체별 대여이력 조회 URL

| 환경 | API명                | URL   |
|----|---------------------|---|
| 운영 | 렌트카 사업자 업체별 대여이력 조회 | https://rims.kotsa.or.kr:8114/company/rentals/history |

#### 4.2.8.3 렌트카 사업자 업체별 대여이력 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /company/rentals/history
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.2.8.4 렌트카 사업자 업체별 대여이력 조회 바디

| 항목명   | 항목 Key      | 항목 형식   | 길이 | 필수여부                      | 내용     |
|-------|-------------|---------|----|---------------------------|--------|
| 시작일시  | f_from_date | String  | 8  | 필수                        | 조회시작일시 |
| 종료일시  | f_to_date   | String  | 8  | 필수                        | 조회종료일시 |
| 차량번호  | vhcl_reg_no | String  | 16 | 선택                        | 차량번호   |
| 회사명   | co_nm       | String  | 80 | 선택                        | 회사명    |
| 페이지번호 | page_no     | Integer |    | 필수                        | 페이지번호  |
| 조회건수  | num_of_rows | Integer |    | 필수                        | 조회건수   |
| 검색유형  | search_type |         |    | 선택                        | 검색유형   |
| 사용자정보 | bizinfo     | String  | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID  |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.

- search\_type는 미기입시 대여시작일 기준이며 “5.13 검색유형 코드표”를 참고한다.
- f\_from\_date(시작일시)와 f\_to\_date(종료일시)의 간격은 최대 1개월(달 기준)까지만 허용한다.

#### 4.2.8.5 렌트카 사업자 업체별 대여이력 조회 예시

```
POST https://rims.kotsa.or.kr:8114/company/rentals/history
```

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "f_from_date": "20250915",
  "f_to_date": "20251001",
  "vhcl_reg_no": "01호1234",
  "co_nm": "테스트회사",
  "page_no": 1,
  "num_of_rows": 10,
  "search_type": "rent_create_date",
  "bizinfo": "s01"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "data": [
    {
      "co_nm": "(주)테스트",
      "rent_no": "R25-000000964",
      "vhcl_reg_no": "66호1234",
      "rent_bgng_dt": "2025-10-17",
      "rent_end_dt": "2025-10-29",
      "rent_stts_cd": "2",
      "reg_dt": "2025-09-27T19:14:23.885",
      "lcns_idntf_nm": "일반면허",
      "verification_info": {
        "status": "PENDING",
        "message": "Verification pending. Please check back later."
      }
    }
  ]
}
```

```

        "vrfc_dmnd_dt": "2025-09-27 19:14:23.158",
        "f rtn_code": "00"
    }
}
],
"page_no": 1,
"num_of_rows": 1,
"total_count": 14
}

```

#### 4.2.8.6 렌트카 사업자 업체별 대여이력 조회(response) Body 목록

##### 4.2.8.6.1 Body

| 항목명    | 항목 Key        | 항목 형식   | 내용                                  |
|--------|---------------|---------|-------------------------------------|
| 회사명    | co_nm         | String  | 회사명                                 |
| 대여번호   | rent_no       | String  | (형식 : R00-123456789)                |
| 차량번호   | vhcl_reg_no   | String  | 차량번호                                |
| 대여시작일시 | rent_bgng_dt  | String  | 시작일시 (형식 : yyyy-MM-dd)              |
| 대여종료일시 | rent_end_dt   | String  | 종료일시 (형식 : yyyy-MM-dd)              |
| 대여상태정보 | rent_stts_cd  | String  | 대여상태코드(1: 미확정, 2: 대여확정, 9: 대여취소)    |
| 등록일시   | reg_dt        | String  | 등록일시 (형식 : yyyy-MM-dd HH:mm:ss.SSS) |
| 면허종류   | lcns_idntf_nm | String  | 일반면허 or 임시면허                        |
| 검증요청일시 | vrfc_dmnd_dt  | String  | 요청일시 (형식 : yyyy-MM-dd HH:mm:ss.SSS) |
| 검증결과코드 | f rtn_code    | String  | 검증결과코드 (적격 : 00, 부적격 : 01~)         |
| 페이지번호  | page_no       | Integer | 페이지번호                               |
| 조회건수   | num_of_rows   | Integer | 조회건수                                |
| 전체건수   | total_count   | Integer | 전체건수                                |

- 응답은 “4.2.8.5 렌트카 업체별 대여이력 조회 예시”를 참고한다.
- “4.2.8.6.1 Body”의 f rtn\_code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “num\_of\_rows”는 1,000건을 넘지 않는다.

#### 4.2.9 렌트카 사업자 업체별 반복예약 결과 조회

#### 4.2.9.1 API 호출

- 발급받은 Token을 통해 업체별 반복예약 결과 조회한다.

#### 4.2.9.2 렌트카 사업자 업체별 반복예약 결과 조회 URL

| 환경 | API명                   | URL   |
|----|------------------------|---|
| 운영 | 렌트카 사업자 업체별 반복예약 결과 조회 | https://rims.kotsa.or.kr:8114/company/licenseVerificationSchedule/history |

#### 4.2.9.3 렌트카 사업자 업체별 반복예약 결과 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /company/licenseVerificationSchedule/history
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.2.9.4 렌트카 사업자 업체별 반복예약 결과 조회 바디

| 항목명   | 항목 Key      | 항목 형식   | 길이 | 필수여부                          | 내용     |
|-------|-------------|---------|----|-------------------------------|--------|
| 시작일시  | f_from_date | String  | 8  | 필수                            | 조회시작일시 |
| 종료일시  | f_to_date   | String  | 8  | 필수                            | 조회종료일시 |
| 차량번호  | vhcl_reg_no | String  | 16 | 선택                            | 차량번호   |
| 회사명   | co_nm       | String  | 80 | 선택                            | 회사명    |
| 페이지번호 | page_no     | Integer |    | 필수                            | 페이지번호  |
| 조회건수  | num_of_rows | Integer |    | 필수                            | 조회건수   |
| 검색유형  | search_type |         |    | 선택                            | 검색유형   |
| 사용자정보 | bizinfo     | String  | 32 | API 호출 사용자와 일치 할 경우 생략 가능(선택) | 사용자ID  |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.

- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- search\_type은 미기입시 반복예약 시작일 기준이며 “5.13 검색유형 코드표”를 참고한다.
- “num\_of\_rows”는 1,000건을 넘지 않는다.
- f\_from\_date(시작일시)와 f\_to\_date(종료일시)의 간격은 최대 1개월(달 기준)까지만 허용한다.

#### 4.2.9.5 렌트카 사업자 업체별 반복예약 결과 조회 예시

```
POST https://rims.kotsa.or.kr:8114/company/licenseVerificationSchedule/history
```

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "f_from_date": "20251001",
  "f_to_date": "20251010",
  "vhcl_reg_no": "01호5303",
  "co_nm": "테스트회사",
  "page_no": "1",
  "num_of_rows": "10",
  "search_type": "reservation_start_date",
  "bizinfo": "s01"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "data": [
    {
      "co_nm": "테스트 본사",
      "rent_no": "R25-000000935",
      "vhcl_reg_no": "01호1234",
      "f_license_no": "2*1*0*1*0*1*",
      "f_licn_con_code": "12",
    }
  ]
}
```

```
        "repeat_period": "2",
        "rsvt_bgng_ymd": "2025-09-22",
        "rsvt_end_ymd": "2025-10-11",
        "last_vrfc_rslt": "21",
        "last_vrfc_ymd": "2025-10-07",
        "next_vrfc_ymd": "2025-10-22",
        "rsvt_nm": "예*자",
        "f_resident_name": "홍*동",
        "reg_dt": "2025-09-22 15:14:08.899"
    }
],
"page_no": 1,
"num_of_rows": 1,
"total_count": 1
}
```

#### 4.2.9.6 렌트카 사업자 업체별 반복예약 결과 조회(response) Body 목록

##### 4.2.9.6.1 Body

| 항목명      | 항목 Key          | 항목 형식   | 내용                                  |
|----------|-----------------|---------|-------------------------------------|
| 회사명      | co_nm           | String  | 회사명                                 |
| 대여번호     | rent_no         | String  | (형식 : R00-123456789)                |
| 차량번호     | vhcl_reg_no     | String  | 차량번호                                |
| 운전면허번호   | f_license_no    | String  | 면허번호는 1*1*3*5*1*1* 형태로 마스킹 처리       |
| 면허종별(코드) | f_licn_con_code | String  | 면허종별(코드)                            |
| 반복주기     | repeat_period   | String  | 반복주기코드(1: 일주일, 2:보름, 3: 한 달)        |
| 반복예약시작일자 | rsvt_bgng_ymd   | String  | 반복예약시작일자(형식 : yyyy-MM-dd)           |
| 반복예약종료일자 | rsvt_end_ymd    | String  | 반복예약종료일자(형식 : yyyy-MM-dd)           |
| 최종검증결과   | last_vrfc_rslt  | String  |                                     |
| 최종검증일자   | last_vrfc_ymd   | String  | 최종검증일자(형식 : yyyy-MM-dd)             |
| 다음검증일자   | next_vrfc_ymd   | String  | 다음검증일자(형식 : yyyy-MM-dd)             |
| 예약자명     | rsvt_nm         | String  | 반복예약 등록자명 (홍*동) 형태로 마스킹 처리          |
| 운전자명     | f_resident_name | String  | 면허검증 운전자명 (홍*동) 형태로 마스킹 처리          |
| 등록일자     | reg_dt          | String  | 등록일시 (형식 : yyyy-MM-dd HH:mm:ss.SSS) |
| 페이지번호    | page_no         | Integer | 페이지번호                               |
| 조회건수     | num_of_rows     | Integer | 조회건수                                |
| 전체건수     | total_count     | Integer | 전체건수                                |

- 응답은 “4.2.9.5 렌트카 업체별 반복예약 결과 조회 예시”를 참고한다.
- “4.2.9.6.1 Body”의 last\_vrfc\_rslt는 “5.2 면허정보 응답 코드표”를 참고한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고한다

## 4.3 소화물배송대행서비스

### 4.3.1 소화물배송대행서비스 사업자 운전자격확인 단건 조회

#### 4.3.1.1 API 호출

- 발급받은 Token을 통해 면허소유자를 검증한다.

#### 4.3.1.2 소화물배송대행서비스 사업자 운전자격확인 단건 조회 URL

| 환경 | API명                              | URL  |
|----|-----------------------------------|--|
| 운영 | 소화물배송대행서비스<br>사업자 운전자격확인<br>단건 조회 | https://rims.kotsa.or.kr:8114/dapi/licenseVerification |

#### 4.3.1.3 소화물배송대행서비스 사업자 운전자격확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /dapi/licenseVerification
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 상태코드를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.3.1.4 소화물배송대행서비스 사업자 운전자격확인 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 사용자정보         | bizinfo         | String | 32 | 선택   | 사용자ID   |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.

#### 4.3.1.5 소화물배송대행서비스 사업자 단건 조회 예시

POST https://rims.kotsa.or.kr:8114/dapi/licenseVerification

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_license_no": "221212121212",
  "f_resident_name": "홍길동",
  "f_licn_con_code": "12",
  "f_from_date": "20241023",
  "f_to_date": "20241023",
  "vhcl_reg_no": "67라6767"
}
```

암호화 후

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "header": {  
        "f rtn msg": "",  
        "f request date": "20231010",  
        "f rtn cd": "0",  
        "f send cnt": "1"  
    },  
    "body": {  
        "f rtn code": "00",  
        "f license no": "2*1*1*1*2*2"  
    }  
}
```

#### 4.3.1.6 소화물배송대행서비스 사업자 운전자격확인 단건 조회 응답 (response) Body 목록

##### 4.3.1.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn msg      | String | 요청처리실패메시지            |

#### 4.3.1.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                          |
|--------|--------------|--------|-----------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 운전면허번호 | f license_no | String | 번호는 1*1*3*5*1*1* 형태로 마스킹 처리 |

- 응답은 “4.3.1.5 소화물배송대행서비스 사업자 단건 조회 예시”를 참고한다.
- “4.3.1.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.3.1.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.3.1.6.2 Body”는 생략된다.

### 4.3.2 소화물배송대행서비스 사업자 운전자격확인 배치 조회

#### 4.3.2.1 API 호출

- 발급받은 Token을 통해 복수의 면허소유자를 검증한다.

#### 4.3.2.2 소화물배송대행서비스 사업자 운전자격확인 배치 조회 URL

| 환경 | API명                              | URL   |
|----|-----------------------------------|---|
| 운영 | 소화물배송대행서비스<br>사업자 운전자격확인<br>배치 조회 | https://rims.kotsa.or.kr:8114/dapi/licenseVerificationBatch |

#### 4.3.2.3 소화물배송대행서비스 사업자 운전자격확인 배치 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /dapi/licenseVerificationBatch
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.

- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.3.2.4 소화물배송대행서비스 사업자 배치 조회 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 요청건수          | f_send_cnt      | String | 4  | 필수   | 리스트 인덱스와 일치(최대 1,000개)                              |
| 사용자정보         | bizinfo         | String | 32 | 선택   | 사용자ID   |
| 요청리스트         | requestList     | String |    | 필수   | 복수 요청의 키  |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 필수   | (yyyyMMdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 필수   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- 요청건수는 요청리스트의 인덱스와 일치해야 한다.

#### 4.3.2.5 소화물배송대행서비스 사업자 배치 조회 예시

```
POST https://rims.kotsa.or.kr:8114/dapi/licenseVerificationBatch
Authorization: Bearer ...
Content-Type: application/json
Content-Length: ...
```

암호화 전

```
{
  "f_send_cnt": "2",
  "bizinfo": "asdf12341....",
  "requestList": [
    {
      "f_license_no": "221212121212",
```

```

        "f_resident_name": "홍길동",
        "f_licn_con_code": "12",
        "f_from_date": "20241023",
        "f_to_date": "20241023",
        "vhcl_reg_no": "67라6767"
    },
    {
        "f_license_no": "221212121212",
        "f_resident_name": "홍길동",
        "f_licn_con_code": "12",
        "f_from_date": "20241023",
        "f_to_date": "20241023"
        "vhcl_reg_no": "67라6767"
    }]
}

```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "header" : {
        "f_send_cnt" : "2",
        "f_request_date" : "20220922122321",
        "f rtn_cd" : "00",
        "f rtn_msg" : ""
    }
    "body" : [
        {
            "vhcl_idnty_cd": "1",
            "f_license_no" : "121234561212",
            "f rtn_code" : "25"
        },
        {
            "vhcl_idnty_cd": "1",

```

```

        "f_license_no" : "121234561212",
        "f rtn_code" : "00"
    }]
}

```

### 4.3.2.6 소화물배송대행서비스 사업자 운전자격확인 배치 조회 응답(response) Body 목록

#### 4.3.2.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.3.2.6.2 Body

| 항목명    | 항목 Key        | 항목 형식  | 내용                             |
|--------|---------------|--------|--------------------------------|
| 차량확인코드 | vhcl_idnty_cd | String | (1:확인, 2:미확인)                  |
| 검증결과코드 | f rtn_code    | String | 검증결과코드 (적격 : 00, 부적격 : 01~)    |
| 운전면허번호 | f_license_no  | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.3.2.5 소화물배송대행서비스 사업자 배치 조회 예시”를 참고한다.
- “4.3.2.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.3.2.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.3.2.6.2 Body”는 생략된다.

### 4.3.3 소화물배송대행서비스 사업자 검증번호 생성

#### 4.3.3.1 API 호출

- 발급받은 Token을 통해 검증번호를 생성한다.

#### 4.3.3.2 소화물배송대행서비스 사업자 검증번호 생성 URL

| 환경 | API명                      | URL   |
|----|---------------------------|---|
| 운영 | 소화물배송대행서비스<br>사업자 검증번호 생성 | <a href="https://rims.kotsa.or.kr:8114/dapi/generateVrfcNo">https://rims.kotsa.or.kr:8114/dapi/generateVrfcNo</a> |

#### 4.3.3.3 소화물배송대행서비스 사업자 검증번호 생성 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 검증번호 생성 시 HTTP Request (클라이언트 → 인증서버)

```
POST /dapi/generateVrfcNo
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.3.3.4 소화물배송대행서비스 사업자 검증번호 생성 바디

| 항목명           | 항목 Key      | 항목 형식  | 길이 | 필수여부                      | 내용  |
|---------------|-------------|--------|----|---------------------------|---|
| 사용자정보         | bizinfo     | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID   |
| 운전자격<br>검증시작일 | f_from_date | String | 8  | 선택                        | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date   | String | 8  | 선택                        |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.

#### 4.3.3.5 소화물배송대행서비스 사업자 검증번호 생성 예시

```
POST https://rims.kotsa.or.kr:8114/dapi/generateVrfcNo
```

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_from_date": "20240926",
```

```

    "f_to_date": "20250925"
}

암호화 후
{
    "encryptedData": "G+joSdT..."
}

응답샘플

HTTP/1.1 200 OK

{
    "vrfc_no": "V25-000000293"
}

```

#### 4.3.3.6 소화물배송대행서비스 사업자 검증번호 생성 응답(response) Body 목록

##### 4.3.3.6.1 결과

| 항목명  | 항목 Key  | 항목 형식  | 내용                   |
|------|---------|--------|----------------------|
| 검증번호 | vrfc_no | String | (형식 : V00-123456789) |

#### 4.3.4 소화물배송대행서비스 사업자 비대면 인증 결과 확인

##### 4.3.4.1 API 호출

- 발급받은 Token을 통해 비대면 인증 결과를 조회한다.

##### 4.3.4.2 소화물배송대행서비스 사업자 비대면 인증 결과 확인 URL

| 환경 | API명                       | URL   |
|----|----------------------------|---|
| 운영 | 소화물배송대행서비스사업자 비대면 인증 결과 확인 | <a href="https://rims.kotsa.or.kr:8114/dapi/licenseVerificationResult">https://rims.kotsa.or.kr:8114/dapi/licenseVerificationResult</a> |

##### 4.3.4.3 소화물배송대행서비스 사업자 비대면 인증 결과 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /dapi/licenseVerificationResult
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.3.4.4 소화물배송대행서비스 사업자 비대면 인증 결과 확인 바디

| 항목명   | 항목 Key  | 항목 형식  | 길이 | 필수여부                      | 내용    |
|-------|---------|--------|----|---------------------------|-------|
| 검증번호  | vrfc_no | String | 13 | 필수                        | 대여번호  |
| 사용자정보 | bizinfo | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 대여번호는 “4.3.3 검증번호 생성” 호출 후 응답의 검증번호(vrfc\_no)를 사용한다.

#### 4.3.4.5 소화물배송대행서비스 사업자 비대면 인증 결과 확인 예시

```
POST https://rims.kotsa.or.kr:8114/dapi/licenseVerificationResult
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "vrfc_no": "V25-000924523"
}
```

```

암호화 후
{
    "encryptedData": "G+joSdT..."
}

```

응답샘플

HTTP/1.1 200 OK

```

{
    "header": {
        "vrfc_no": "V25-000000014",
        "f_rtn_cd": "0",
        "f_request_date": "20250811",
        "f_send_cnt": "1"
    },
    "body": {
        "f rtn code": "00",
        "face_similarity": 100.0,
        "face_liveness": 90.68,
        "original_reliability": 100.0,
        "police_lcns_vrfc_cd": "1"
    }
}

```

#### 4.3.5 소화물배송대행서비스 사업자 비대면 인증 결과 확인 응답 (response) Body 목록

##### 4.3.5.1.1 Header

| 항목명  | 항목 Key         | 항목 형식  | 내용                       |
|------|----------------|--------|--------------------------|
| 대여번호 | vrfc_no        | String | 검증번호(형식 : V00-123456789) |
| 요청건수 | f_send_cnt     | String | 요청건수(Default 1)          |
| 요청일시 | f_request_date | String | 요청일시 (형식 : YYYYMMDD)     |
| 요청처리 | f_rtn_cd       | String | 요청처리코드                   |

##### 4.3.5.1.2 Body

| 항목명      | 항목 Key               | 항목 형식  | 내용                          |
|----------|----------------------|--------|-----------------------------|
| 검증결과코드   | f rtn_code           | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 얼굴유사도    | face_similarity      | Double | 얼굴유사도 수치 (0~100)            |
| 얼굴활성도    | face_liveness        | Double | 얼굴활성도 수치 (0~100)            |
| 원본신뢰도    | original_reliability | Double | 원본신뢰도 수치 (0~100)            |
| 면허사진검증코드 | police_lcns_vrfc_cd  | String | 면허사진검증코드(적격 : 1)            |

- 응답은 “4.3.4.5 소화물배송대행서비스 사업자 비대면 인증 결과 확인 예시”를 참고한다.
- “4.3.5.1.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.3.5.1.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.3.5.1.2 Body”는 생략된다.
- “4.3.5.1.2 Body”의 f rtn\_code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “4.3.5.1.2 Body”의 face\_similarity는 “5.9 얼굴 유사도 판정 기준표”를 참고하며 활용 시 12단계 이상을 권장한다.
- “4.3.5.1.2 Body”의 face\_liveness는 “5.8 얼굴 활성도 판정 기준표”를 참고하며 활용 시 4단계 이상을 권장한다.
- “4.3.5.1.2 Body”의 police\_lcns\_vrfc\_cd는 “5.11 면허 사진 검증 코드표”를 참고한다.
- 

## 4.4 개인형이동장치

### 4.4.1 개인형이동장치 운전자격확인 단건 조회

#### 4.4.1.1 API 호출

- 발급받은 Token을 통해 대여자의 운전 자격을 확인한다.

| 환경 | API명                 | URL   |
|----|----------------------|---|
| 운영 | 개인형이동장치 운전자격확인 단건 조회 | <a href="https://rims.kotsa.or.kr:8114/pm/licenseVerification">https://rims.kotsa.or.kr:8114/pm/licenseVerification</a> |

#### 4.4.1.2 개인형이동장치 서비스 사업자 운전자격확인 조회 URL

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

#### 4.4.1.3 개인형이동장치 서비스 사업자 운전자격확인 단건 조회 헤더

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /pm/licenseVerification
Host: rims.kotsa.or.kr
Authorization: "Bearer" + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.4.1.4 개인형이동장치 서비스 사업자 운전자격확인 단건 조회 바디

| 항목명         | 항목 Key          | 항목 형식  | 길이  | 필수여부             | 내용   |
|-------------|-----------------|--------|-----|------------------|--|
| 단말기 식별 고유정보 | vhcl_reg_no     | String | 200 | 필수               | 사업자의 단말기를 식별할 수 있는 고유 정보 기입                |
| 운전면허자명      | f_resident_name | String | 40  | 필수               |  |
| 면허종별(코드)    | f_licn_con_code | String | 2   | 필수               |  |
| 운전자격 검증시작일  | f_from_date     | String | 8   | 선택(장기 대여인 경우 기입) | (yyyymmdd)<br>운전자격 검증시작일은 오늘을 포함한 이후여야 한다. |
| 운전자격 검증종료일  | f_to_date       | String | 8   | 선택(장기 대여인 경우 기입) |  |
| 운전면허번호      | f_license_no    | String | 12  | 필수               |  |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역 (서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다. ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)

#### 4.4.1.5 개인형이동장치 서비스 사업자 운전자격확인 조회 예시

```
POST https://rims.kotsa.or.kr:8114/pm/licenseVerification
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json  
Content-Length: ...
```

암호화 전

```
{  
    "f_license_no": "123456789123",  
    "f_resident_name": "홍길동",  
    "f_licn_con_code": "12",  
    "f_from_date": "20250716",  
    "f_to_date": "20251229",  
    "vhcl_reg_no": "PM000001"  
}
```

암호화 후

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "header": {  
        "f rtn msg": "",  
        "f request date": "20250718",  
        "f rtn cd": "0",  
        "f send cnt": "1"  
    },  
    "body": {  
        "f rtn code": "00",  
        "f license no": "1*1*0*1*2*0"  
    }  
}
```

#### 4.4.1.6 개인형이동장치 운전자격확인 조회 응답(response) Body 목록

##### 4.4.1.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.4.1.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                          |
|--------|--------------|--------|-----------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 운전면허번호 | f license_no | String | 번호는 1*1*3*5*1*1* 형태로 마스킹 처리 |

- 응답은 “4.4.1.5 개인형이동장치 운전자격확인 조회 예시”를 참고한다.
- “4.4.1.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.4.1.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.4.1.6.2 Body”는 생략된다.

### 4.4.2 개인형이동장치서비스 사업자 운전자격확인 배치 조회

#### 4.4.2.1 API 호출

- 발급받은 Token을 통해 복수의 면허소유자를 검증한다.

#### 4.4.2.2 개인형이동장치 서비스 사업자 운전자격확인 배치 URL

| 환경 | API명                         | URL   |
|----|------------------------------|---|
| 운영 | 개인형이동장치<br>서비스 사업자<br>운전자격확인 | <a href="https://rims.kotsa.or.kr:8114/pm/licenseVerificationBatch">https://rims.kotsa.or.kr:8114/pm/licenseVerificationBatch</a> |

#### 4.4.2.3 개인형이동장치 서비스 사업자 운전자격확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```

POST /pm/licenseVerificationBatch
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr

```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.4.2.4 개인형이동장치 서비스 사업자 운전자격확인 바디

| 항목명            | 항목 Key          | 항목 형식  | 길이  | 필수여부                      | 내용  |
|----------------|-----------------|--------|-----|---------------------------|---|
| 요청건수           | f_send_cnt      | String | 4   | 필수                        | 리스트 인덱스와 일치(최대 1,000개)                              |
| 사용자정보          | bizinfo         | String | 32  | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID   |
| 요청리스트          | requestList     | String |     | 필수                        | 복수 요청의 키  |
| 운전면허번호         | f_license_no    | String | 12  | 필수                        |   |
| 운전면허자명         | f_resident_name | String | 40  | 필수                        |   |
| 면허종별(코드)       | f_licn_con_code | String | 2   | 필수                        |   |
| 운전자격<br>검증시작일  | f_from_date     | String | 8   | 미필수(장기 대여인 경우에만 기입)       | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일  | f_to_date       | String | 8   | 미필수(장기 대여인 경우에만 기입)       |   |
| 단말기 식별<br>고유정보 | vhcl_reg_no     | String | 200 | 필수                        | 사업자의 단말기를<br>식별할 수 있는<br>고유 정보 기입                   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- 요청건수는 요청리스트의 인덱스와 일치해야 한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다.ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)

#### 4.4.2.5 개인형이동장치 서비스 사업자 배치 조회 예시

```
POST https://rims.kotsa.or.kr:8114/pm/licenseVerificationBatch
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
    "f_send_cnt": "2",
    "bizinfo": "asdf12341....",
    "requestList": [
        {
            "f_license_no": "221212121212",
            "f_resident_name": "홍길동",
            "f_licn_con_code": "12",
            "f_from_date": "20241023",
            "f_to_date": "20241023",
            "vhcl_reg_no": "pm_id01"
        },
        {
            "f_license_no": "221212121212",
            "f_resident_name": "홍길동",
            "f_licn_con_code": "12",
            "f_from_date": "20241023",
            "f_to_date": "20241023"
            "vhcl_reg_no": "pm_id01"
        }
    ]
}
```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "header" : {
        "f_send_cnt" : "2",
        "f_request_date" : "20220922122321",
        "f_rtn_cd" : "00",
    }
}
```

```

        "f rtn msg" : ""
    }
    "body" : [
        {
            "f_license_no" : "121234561212",
            "f rtn code" : "25"
        },
        {
            "f_license_no" : "121234561212",
            "f rtn code" : "00"
        }
    ]
}

```

#### 4.4.2.6 개인형이동장치 서비스 사업자 운전자격확인 배치 조회 응답 (response) Body 목록

##### 4.4.2.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn msg      | String | 요청처리실패메시지            |

##### 4.4.2.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                             |
|--------|--------------|--------|--------------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (직격 : 00,<br>부적격 : 01~) |
| 운전면허번호 | f license_no | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.4.2.5 개인형이동장치 서비스 사업자 배치 조회 예시”를 참고한다.
- “4.4.2.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.4.2.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.4.2.6.2 Body”는 생략된다.

#### 4.4.3 개인형이동장치 서비스 사업자 대여번호 생성

##### 4.4.3.1 API 호출

- 발급받은 Token을 통해 대여번호를 생성한다.

#### 4.4.3.2 개인형이동장치 서비스 사업자 대여번호 생성 URL

| 환경 | API명    | URL   |
|----|---------|---|
| 운영 | 대여번호 생성 | https://rims.kotsa.or.kr:8114/pm/generateRentNo |

#### 4.4.3.3 개인형이동장치 서비스 사업자 대여번호 생성 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 대여번호 생성 시 HTTP Request (클라이언트 → 인증서버)

```
POST /pm/generateRentNo
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 대여번호 생성 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.4.3.4 개인형이동장치 서비스 사업자 대여번호 생성 바디

| 항목명         | 항목 Key      | 항목 형식  | 길이  | 필수여부                      | 내용                                     |
|-------------|-------------|--------|-----|---------------------------|--|
| 사용자정보       | bizinfo     | String | 32  | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID                                  |
| 대여시작일       | f_from_date | String | 8   | 미필수(장기 대여인 경우에만 기입)       | (yyyymmdd)<br>대여 시작일은 오늘을 포함한 이후여야 한다. |
| 대여종료일       | f_to_date   | String | 8   | 미필수(장기 대여인 경우에만 기입)       |  |
| 단말기 식별 고유정보 | vhcl_reg_no | String | 200 | 필수                        | 사업자의 단말기를 식별할 수 있는 고유 정보 기입            |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.

#### 4.4.3.5 개인형이동장치 서비스 사업자 대여번호 생성 예시

```
POST https://rims.kotsa.or.kr:8114/pm/generateRentNo
```

```
Authorization: Bearer ...
Content-Type: application/json
Content-Length: ...
```

암호화 전

```
{
    "bizinfo": "asdf12341....",
    "vhcl_reg_no": "pm_id3",
    "f_from_date": "20240926",
    "f_to_date": "20250925"
}
```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "rent_no": "P25-000000080"
}
```

#### 4.4.3.6 개인형이동장치 서비스 사업자 대여번호 생성 응답(response) Body 목록

##### 4.4.3.6.1 결과

| 항목명  | 항목 Key  | 항목 형식  | 내용                   |
|------|---------|--------|----------------------|
| 대여번호 | rent_no | String | (형식 : P00-123456789) |

- 응답은 “4.4.3.5 개인형이동장치 서비스 사업자 대여번호 생성 예시”를 참고한다.

#### 4.4.4 개인형이동장치 서비스 사업자 비대면 인증 결과 확인

##### 4.4.4.1 API 호출

- 발급받은 Token을 통해 비대면 인증 결과를 조회한다.

##### 4.4.4.2 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 URL

| 환경 | API명                               | URL  |
|----|------------------------------------|--|
| 운영 | 개인형이동장치 서비스<br>사업자 비대면 인증<br>결과 확인 | https://rims.kotsa.or.kr:8114/pm/licenseVerificationResult |

#### 4.4.4.3 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

#### 4.4.4.4 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /pm/licenseVerificationResult
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.4.4.5 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 바디

| 항목명   | 항목 Key  | 항목 형식  | 길이 | 필수여부                      | 내용    |
|-------|---------|--------|----|---------------------------|-------|
| 대여번호  | rent_no | String | 13 | 필수                        | 대여번호  |
| 사용자정보 | bizinfo | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 대여번호는 “4.4.3 개인형이동장치 서비스 사업자 대여번호 생성” 호출 후 응답의 대여번호(rent\_no)를 사용한다.

#### 4.4.4.6 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 예시

```
POST https://rims.kotsa.or.kr:8114/pm/licenseVerificationResult
Authorization: Bearer ...
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{  
    "bizinfo": "asdf12341....",  
    "rent_no": "P25-000000080"  
}
```

암호화 후

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "header": {  
        "rent_no": "P25-000000843",  
        "f rtn_cd": "0",  
        "f request_date": "20250908",  
        "f send_cnt": "1"  
    },  
    "body": {  
        "vhcl_idnty_cd": "2",  
        "f rtn_code": "00",  
        "face_similarity": 99.0,  
        "face_liveness": 99.0,  
        "original_reliability": 90.0,  
        "police_lcns_vrfc_cd": "1"  
    }  
}
```

#### 4.4.4.7 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 응답 (response) Body 목록

##### 4.4.4.7.1 Header

| 항목명  | 항목 Key         | 항목 형식  | 내용                       |
|------|----------------|--------|--------------------------|
| 대여번호 | rent_no        | String | 대여번호(형식 : R00-123456789) |
| 요청건수 | f_send_cnt     | String | 요청건수(Default 1)          |
| 요청일시 | f_request_date | String | 요청일시 (형식 : YYYYMMDD)     |
| 요청처리 | f rtn_cd       | String | 요청처리코드                   |

#### 4.4.4.7.2 Body

| 항목명      | 항목 Key               | 항목 형식  | 내용                          |
|----------|----------------------|--------|-----------------------------|
| 차량확인코드   | vhcl_idnty_cd        | String | (1:확인, 2:미확인)               |
| 검증결과코드   | f rtn_code           | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 얼굴유사도    | face_similarity      | Double | 얼굴유사도 수치 (0~100)            |
| 얼굴활성도    | face_liveness        | Double | 얼굴활성도 수치 (0~100)            |
| 원본신뢰도    | original_reliability | Double | 원본신뢰도 수치 (0~100)            |
| 면허사진검증코드 | police_lcns_vrfc_cd  | String | 면허사진검증코드(적격 : 1)            |

- 응답은 “4.4.4.6 개인형이동장치 서비스 사업자 비대면 인증 결과 확인 예시”를 참고한다.
- “4.4.4.7.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.4.4.7.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.4.4.7.2 Body”는 생략된다.
- “4.4.4.7.2 Body”의 f rtn\_code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “4.4.4.7.2 Body”의 face\_similarity는 “5.9 얼굴 유사도 판정 기준표”를 참고하며 활용 시 12단계 이상을 권장한다.
- “4.4.4.7.2 Body”의 face\_liveness는 “5.8 얼굴 활성도 판정 기준표”를 참고하며 활용 시 4단계 이상을 권장한다.
- “4.4.4.7.2 Body”의 police\_lcns\_vrfc\_cd는 “5.11 면허 사진 검증 코드표”를 참고한다.

## 4.5 대리운전서비스

### 4.5.1 대리운전 서비스 사업자 운전자격확인 단건 조회

#### 4.5.1.1 API 호출

- 발급받은 Token을 통해 면허소유자를 검증한다.

#### 4.5.1.2 대리운전 서비스 사업자 운전자격확인 단건 조회 URL

| 환경 | API명                          | URL   |
|----|-------------------------------|---|
| 운영 | 대리운전서비스사업자<br>운전자격확인 단건<br>조회 | <a href="https://rims.kotsa.or.kr:8114/sub/licenseVerification">https://rims.kotsa.or.kr:8114/sub/licenseVerification</a> |

#### 4.5.1.3 대리운전 서비스 사업자 운전자격확인 단건 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /sub/licenseVerification
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)
 

|             |
|-------------|
| HTTP 200 OK |
|-------------|
- 인증서버는 HTTP 상태코드를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.5.1.4 대리운전 서비스 사업자 운전자격확인 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 사용자정보         | bizinfo         | String | 32 | 선택   | 사용자ID   |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 선택   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 선택   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다.ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)

#### 4.5.1.5 대리운전 서비스 사업자 단건 조회 예시

POST https://rims.kotsa.or.kr:8114/sub/licenseVerification

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_license_no": "221212121212",
  "f_resident_name": "홍길동",
  "f_licn_con_code": "12",
  "f_from_date": "20241023",
  "f_to_date": "20241023",
```

```
        "vhcl_reg_no": "67라6767"  
    }  
  
암호화 후
```

```
{  
    "encryptedData": "G+joSdT..."  
}
```

응답샘플

HTTP/1.1 200 OK

```
{  
    "header": {  
        "f rtn msg": "",  
        "f request date": "20231010",  
        "f rtn cd": "0",  
        "f send cnt": "1"  
    },  
    "body": {  
        "f rtn code": "00",  
        "f license no": "2*1*1*1*2*2"  
    }  
}
```

#### 4.5.1.6 대리운전 서비스 사업자 운전자격확인 단건 조회 응답(response) Body 목록

##### 4.5.1.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.5.1.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                             |
|--------|--------------|--------|--------------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00,<br>부적격 : 01~) |
| 운전면허번호 | f license_no | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.5.1.5 대리운전 서비스 사업자 단건 조회 예시”를 참고한다.
- “4.5.1.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.5.1.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.5.1.6.2 Body”는 생략된다.

### 4.5.2 대리운전 서비스 사업자 운전자격확인 배치 조회

#### 4.5.2.1 API 호출

- 발급받은 Token을 통해 복수의 면허소유자를 검증한다.

#### 4.5.2.2 대리운전 서비스 사업자 운전자격확인 배치 URL

| 환경 | API명                          | URL  |
|----|-------------------------------|--|
| 운영 | 대리운전서비스사업자<br>운전자격확인 배치<br>조회 | https://rims.kotsa.or.kr:8114/sub/licenseVerificationB<br>atch |

#### 4.5.2.3 대리운전 서비스 사업자 운전자격확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```

POST /sub/licenseVerificationBatch
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr

```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)  
HTTP 200 OK
- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.5.2.4 대리운전 서비스 사업자 운전자격확인 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 요청건수          | f_send_cnt      | String | 4  | 필수   | 리스트 인덱스와 일치(최대 1,000개)                              |
| 사용자정보         | bizinfo         | String | 32 | 필수   | API 호출 사용자와 일치 할 경우 생략 가능                           |
| 요청리스트         | requestList     | String |    | 필수   | 복수 요청의 키  |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 선택   | (yyyyMMdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 선택   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역 (서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.

- 요청건수는 요청리스트의 인덱스와 일치해야한다.
- “운전자격검증 시작일(f\_from\_date)” 과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다.ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)

#### 4.5.2.5 대리운전 서비스 사업자 배치 조회 예시

```
POST https://rims.kotsa.or.kr:8114/sub/licenseVerificationBatch
Authorization: Bearer ...
Content-Type: application/json
Content-Length: ...
```

암호화 전

```
{
  "f_send_cnt": "2",
  "bizinfo": "asdf12341....",
  "requestList": [
    {
      "f_license_no": "221212121212",
      "f_resident_name": "홍길동",
      "f_licn_con_code": "12",
      "f_from_date": "20241023",
      "f_to_date": "20241023",
      "vhcl_reg_no": "67라6767"
    },
    {
      "f_license_no": "221212121212",
      "f_resident_name": "홍길동",
      "f_licn_con_code": "12",
      "f_from_date": "20241023",
      "f_to_date": "20241023",
      "vhcl_reg_no": "67라6767"
    }
  ]
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

```

HTTP/1.1 200 OK

{
    "header" : {
        "f_send_cnt" : "2",
        "f_request_date" : "20220922122321",
        "f rtn_cd" : "00",
        "f rtn_msg" : ""
    }
    "body" : [
        {
            "f_license_no" : "121234561212",
            "f rtn_code" : "25"
        },
        {
            "f_license_no" : "121234561212",
            "f rtn_code" : "00"
        }
    ]
}

```

#### 4.5.2.6 대리운전 서비스 사업자 운전자격확인 배치 조회 응답(response) Body 목록

##### 4.5.2.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

##### 4.5.2.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                          |
|--------|--------------|--------|-----------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 운전면허번호 | f_license_no | String | 번호는 1*1*3*5*1*1* 형태로 마스킹 처리 |

- 응답은 “4.5.2.5 대리운전 서비스 사업자 배치 조회 예시”를 참고한다.
- “4.5.2.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.5.2.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.5.2.6.2 Body”는 생략된다.

#### 4.5.2.7 대리운전 서비스 사업자 검증번호 생성

##### 4.5.2.8 API 호출

- 발급받은 Token을 통해 검증번호를 생성한다.

##### 4.5.2.9 대리운전 서비스 사업자 검증번호 생성 URL

| 환경 | API명                  | URL   |
|----|-----------------------|---|
| 운영 | 대리운전서비스사업자<br>검증번호 생성 | <a href="https://rims.kotsa.or.kr:8114/sub/generateVrfcNo">https://rims.kotsa.or.kr:8114/sub/generateVrfcNo</a> |

##### 4.5.2.10 대리운전 서비스 사업자 검증번호 생성 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /sub/generateVrfcNo
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

|             |
|-------------|
| HTTP 200 OK |
|-------------|

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.5.2.11 대리운전 서비스 사업자 검증번호 생성 바디

| 항목명           | 항목 Key      | 항목 형식  | 길이 | 필수여부                      | 내용  |
|---------------|-------------|--------|----|---------------------------|---|
| 사용자정보         | bizinfo     | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID   |
| 운전자격<br>검증시작일 | f_from_date | String | 8  | 선택                        | (yyyyMMdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date   | String | 8  | 선택                        |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다.ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)

#### 4.5.2.12 대리운전 서비스 사업자 검증번호 생성 예시

```
POST https://rims.kotsa.or.kr:8114/sub/generateVrfcNo
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_from_date": "20240926",
  "f_to_date": "20250925"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "vrfc_no": "D25-000000293"
}
```

#### 4.5.2.13 대리운전 서비스 사업자 검증번호 생성 응답(response) Body 목록

##### 4.5.2.13.1 결과

| 항목명  | 항목 Key  | 항목 형식  | 내용                   |
|------|---------|--------|----------------------|
| 검증번호 | vrfc_no | String | (형식 : D00-123456789) |

- 응답은 “4.5.2.12 대리운전 서비스 사업자 검증번호 생성 예시”를 참고한다.

#### 4.5.3 대리운전 서비스 사업자 비대면 인증 결과 확인

##### 4.5.3.1 API 호출

- 발급받은 Token을 통해 비대면 인증 결과를 조회한다.

##### 4.5.3.2 대리운전 서비스 사업자 비대면 인증 결과 확인 URL

| 환경 | API명                    | URL   |
|----|-------------------------|---|
| 운영 | 대리운전서비스사업자 비대면 인증 결과 확인 | <a href="https://rims.kotsa.or.kr:8114/sub/licenseVerificationResult">https://rims.kotsa.or.kr:8114/sub/licenseVerificationResult</a> |

##### 4.5.3.3 대리운전 서비스 사업자 비대면 인증 결과 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /sub/licenseVerificationResult  
Host: rims.kotsa.or.kr  
Authorization: "Bearer " + 토큰  
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

##### 4.5.3.4 대리운전 서비스 사업자 비대면 인증 결과 확인 바디

| 항목명   | 항목 Key  | 항목 형식  | 길이 | 필수여부                      | 내용    |
|-------|---------|--------|----|---------------------------|-------|
| 검증번호  | vrfc_no | String | 13 | 필수                        | 대여번호  |
| 사용자정보 | bizinfo | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 검증번호는 “4.5.2.7 대리운전 사업자 검증번호 생성” 호출 후 응답의 검증 번호(vrfc\_no)를 사용한다.

#### 4.5.3.5 대리운전 서비스 사업자 비대면 인증 결과 확인 예시

```
POST https://rims.kotsa.or.kr:8114/sub/licenseVerificationResult
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "vrfc_no": "D25-000000014"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header": {
    "vrfc_no": "D25-000000014",
    "f rtn_cd": "0",
    "f request_date": "20250811",
    "f send_cnt": "1"
  },
  "body": {
    "f rtn_code": "00",
    "f rtn_desc": "Success"
  }
}
```

```

        "face_similarity": 100.0,
        "face_liveness": 90.68,
        "original_reliability": 100.0,
        "police_lcns_vrfc_cd": "1"
    }
}

```

#### 4.5.4 대리운전 서비스 사업자 비대면 인증 결과 확인 응답(response) Body 목록

##### 4.5.4.1.1 Header

| 항목명  | 항목 Key         | 항목 형식  | 내용                       |
|------|----------------|--------|--------------------------|
| 대여번호 | vrfc_no        | String | 검증번호(형식 : D00-123456789) |
| 요청건수 | f_send_cnt     | String | 요청건수(Default 1)          |
| 요청일시 | f_request_date | String | 요청일시 (형식 : YYYYMMDD)     |
| 요청처리 | f rtn_cd       | String | 요청처리코드                   |

##### 4.5.4.1.2 Body

| 항목명      | 항목 Key               | 항목 형식  | 내용                          |
|----------|----------------------|--------|-----------------------------|
| 검증결과코드   | f rtn_code           | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 얼굴유사도    | face_similarity      | Double | 얼굴유사도 수치 (0~100)            |
| 얼굴활성도    | face_liveness        | Double | 얼굴활성도 수치 (0~100)            |
| 원본신뢰도    | original_reliability | Double | 원본신뢰도 수치 (0~100)            |
| 면허사진검증코드 | police_lcns_vrfc_cd  | String | 면허사진검증코드(적격 : 1)            |

- 응답은 “4.5.3.5 대리운전 사업자 비대면 인증 결과 확인 예시”를 참고한다.
- “4.5.4.1.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.5.4.1.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.5.4.1.2 Body”는 생략된다.
- “4.5.4.1.2 Body”의 f rtn\_code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “4.5.4.1.2 Body”의 face\_similarity는 “5.9 얼굴 유사도 판정 기준표”를 참고하며 활용 시 12단계 이상을 권장한다.
- “4.5.4.1.2 Body”의 face\_liveness는 “5.8 얼굴 활성도 판정 기준표”를 참고하며 활용 시 4단계 이상을 권장한다.
- “4.5.4.1.2 Body”의 police\_lcns\_vrfc\_cd는 “5.11 면허 사진 검증 코드표”를 참고한다.

## 4.6 범용

### 4.6.1 범용 운전자격확인 단건 조회

#### 4.6.1.1 API 호출

- 발급받은 Token을 통해 면허소유자를 검증한다.

#### 4.6.1.2 범용 서비스 사업자 운전자격확인 단건 조회 URL

| 환경 | API명               | URL   |
|----|--------------------|---|
| 운영 | 범용 운전자격확인<br>단건 조회 | <a href="https://rims.kotsa.or.kr:8114/cmm/licenseVerification">https://rims.kotsa.or.kr:8114/cmm/licenseVerification</a> |

#### 4.6.1.3 범용 운전자격확인 단건 조회 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /cmm/licenseVerification
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 상태코드를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.6.1.4 범용 운전자격확인 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 사용자정보         | bizinfo         | String | 32 | 선택   | 사용자ID   |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 선택   | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 선택   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |
| 범용API유형       | com_api_type    | String | 3  | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.
- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다(ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)
- 범용API유형(com\_api\_type)은 총 3자리이며, “5.12 범용 API 유형 코드표”를 참고하여 전달한다.

#### 4.6.1.5 범용 단건 조회 예시

```
POST https://rims.kotsa.or.kr:8114/cmm/licenseVerification
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_license_no": "221212121212",
```

```
"f_resident_name": "홍길동",
"f_licn_con_code": "12",
"f_from_date": "20241023",
"f_to_date": "20241023",
"vhcl_reg_no": "67라6767",
"com_api_type": "S00"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header": {
    "f rtn msg": "",
    "f request date": "20231010",
    "f rtn cd": "0",
    "f send cnt": "1"
  },
  "body": {
    "f rtn code": "00",
    "f license no": "2*1*1*1*2*2"
  }
}
```

#### 4.6.1.6 범용 운전자격확인 단건 조회 응답(response) Body 목록

##### 4.6.1.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

#### 4.6.1.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                             |
|--------|--------------|--------|--------------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00, 부적격 : 01~)    |
| 운전면허번호 | f license_no | String | 번호는 1*1*3*5*1*1* 형태로<br>마스킹 처리 |

- 응답은 “4.6.1.5 범용 단건 조회 예시”를 참고한다.
- “4.6.1.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.6.1.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.6.1.6.2 Body”는 생략된다.

## 4.6.2 범용 운전자격확인 배치 조회

### 4.6.2.1 API 호출

- 발급받은 Token을 통해 복수의 면허소유자를 검증한다.

### 4.6.2.2 범용 운전자격확인 배치 URL

| 환경 | API명            | URL   |
|----|-----------------|---|
| 운영 | 범용 운전자격확인 배치 조회 | <a href="https://rims.kotsa.or.kr:8114/cmm/licenseVerificationBatch">https://rims.kotsa.or.kr:8114/cmm/licenseVerificationBatch</a> |

### 4.6.2.3 범용 운전자격확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```

POST /cmm/licenseVerificationBatch
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr

```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)  
HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.6.2.4 범용 운전자격확인 바디

| 항목명           | 항목 Key          | 항목 형식  | 길이 | 필수여부 | 내용  |
|---------------|-----------------|--------|----|------|---|
| 요청건수          | f_send_cnt      | String | 4  | 필수   | 리스트 인덱스와 일치(최대 1,000개)                              |
| 사용자정보         | bizinfo         | String | 32 | 필수   | API 호출 사용자와 일치 할 경우 생략 가능                           |
| 요청리스트         | requestList     | String |    | 필수   | 복수 요청의 키  |
| 운전면허번호        | f_license_no    | String | 12 | 필수   |   |
| 운전면허자명        | f_resident_name | String | 40 | 필수   |   |
| 면허종별(코드)      | f_licn_con_code | String | 2  | 필수   |   |
| 운전자격<br>검증시작일 | f_from_date     | String | 8  | 선택   | (yyyyMMdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date       | String | 8  | 선택   |   |
| 차량등록번호        | vhcl_reg_no     | String | 16 | 선택   |   |
| 범용API유형       | com_api_type    | String | 3  | 선택   |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 “5.5 지역 코드표”를 참고하여 지역명을 숫자로 치환하여 전달한다.

- 면허종별(f\_licn\_con\_code)은 총 2자리이며, “5.4 면허 종별 코드표”를 참고하여 숫자로 치환하여 전달한다.
- 요청건수는 요청리스트의 인덱스와 일치해야한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다.ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)
- 범용API유형(com\_api\_type)은 총 3자리이며, “5.12 범용 API 유형 코드표”를 참고하여 전달한다.

#### 4.6.2.5 범용 배치 조회 예시

```
POST https://rims.kotsa.or.kr:8114/cmm/licenseVerificationBatch
```

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
  "f_send_cnt": "2",
  "bizinfo": "asdf12341....",
  "com_api_type": "B00",
  "requestList": [
    {
      "f_license_no": "221212121212",
      "f_resident_name": "홍길동",
      "f_licn_con_code": "12",
      "f_from_date": "20241023",
      "f_to_date": "20241023",
      "vhcl_reg_no": "67라6767"
    },
    {
      "f_license_no": "221212121212",
      "f_resident_name": "홍길동",
      "f_licn_con_code": "12",
      "f_from_date": "20241023",
      "f_to_date": "20241023",
      "vhcl_reg_no": "67라6767"
    }
  ]
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
```

```
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "header" : {
    "f_send_cnt" : "2",
    "f_request_date" : "20220922122321",
    "f rtn_cd" : "00",
    "f rtn_msg" : ""
  }
  "body" : [
    {
      "f_license_no" : "121234561212",
      "f rtn_code" : "25"
    },
    {
      "f_license_no" : "121234561212",
      "f rtn_code" : "00"
    }
  ]
}
```

#### 4.6.2.6 범용 운전자격확인 배치 조회 응답(response) Body 목록

##### 4.6.2.6.1 Header

| 항목명           | 항목 Key         | 항목 형식  | 내용                   |
|---------------|----------------|--------|----------------------|
| 요청건수          | f_send_cnt     | String | 요청건수(Default 1)      |
| 요청일시          | f_request_date | String | 요청일시 (형식 : YYYYMMDD) |
| 요청처리          | f rtn_cd       | String | 요청처리코드               |
| 요청처리실패<br>메시지 | f rtn_msg      | String | 요청처리실패메시지            |

##### 4.6.2.6.2 Body

| 항목명    | 항목 Key       | 항목 형식  | 내용                          |
|--------|--------------|--------|-----------------------------|
| 검증결과코드 | f rtn_code   | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 운전면허번호 | f_license_no | String | 번호는 1*1*3*5*1*1* 형태로 마스킹 처리 |

- 응답은 “4.6.2.5 범용 배치 조회 예시”를 참고한다.
- “4.6.2.6.1 Header”의 f rtn\_cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.6.2.6.1 Header”의 f rtn\_cd가 0이 아닐 경우 “4.5.2.6.2 Body”는 생략된다.

### 4.6.3 범용 검증번호 생성

#### 4.6.3.1 API 호출

- 발급받은 Token을 통해 검증번호를 생성한다.

#### 4.6.3.2 범용 검증번호 생성 URL

| 환경 | API명       | URL  |
|----|------------|--|
| 운영 | 범용 검증번호 생성 | https://rims.kotsa.or.kr:8114/cmm/generateVrfcNo |

#### 4.6.3.3 범용 검증번호 생성 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 운전자격확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /cmm/generateVrfcNo
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 운전자격확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

HTTP 200 OK

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.6.3.4 범용 검증번호 생성 바디

| 항목명           | 항목 Key       | 항목 형식  | 길이 | 필수여부                      | 내용  |
|---------------|--------------|--------|----|---------------------------|---|
| 사용자정보         | bizinfo      | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID   |
| 운전자격<br>검증시작일 | f_from_date  | String | 8  | 선택                        | (yyyymmdd)<br>운전자격<br>검증시작일은<br>오늘을 포함한<br>이후여야 한다. |
| 운전자격<br>검증종료일 | f_to_date    | String | 8  | 선택                        |   |
| 범용API유형       | com_api_type | String | 3  | 선택                        |   |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- “운전자격검증 시작일(f\_from\_date)”과 “운전자격검증 종료일(f\_to\_date)”은 미기입시 요청일 기준으로 다음날 자정(00:00)으로 지정한다. ex) 2025.07.18. 검증요청시 20250718(f\_from\_date), 검증종료일 20250719(f\_to\_date)
- 범용API유형(com\_api\_type)은 총 3자리이며, “5.12 범용 API 유형 코드표”를 참고하여 전달한다.

#### 4.6.3.5 범용 검증번호 생성 예시

```
POST https://rims.kotsa.or.kr:8114/cmm/generateVrfcNo
```

```
Authorization: Bearer ...
```

```
Content-Type: application/json
```

```
Content-Length: ...
```

```
com_api_type: G00
```

암호화 전

```
{
  "bizinfo": "asdf12341....",
  "f_from_date": "20240926",
  "f_to_date": "20250925",
  "com_api_type": "G00"
}
```

암호화 후

```
{
  "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
  "vrfc_no": "C25-000000293"
```

```
}
```

#### 4.6.3.6 범용 검증번호 생성 응답(response) Body 목록

##### 4.6.3.6.1 결과

| 항목명  | 항목 Key  | 항목 형식  | 내용                   |
|------|---------|--------|----------------------|
| 검증번호 | vrfc_no | String | (형식 : C00-123456789) |

- 응답은 “4.6.2.12 범용 검증번호 생성 예시”를 참고한다.

#### 4.6.4 범용 비대면 인증 결과 확인

##### 4.6.4.1 API 호출

- 발급받은 Token을 통해 비대면 인증 결과를 조회한다.

##### 4.6.4.2 범용 비대면 인증 결과 확인 URL

| 환경 | API명            | URL   |
|----|-----------------|---|
| 운영 | 범용 비대면 인증 결과 확인 | <a href="https://rims.kotsa.or.kr:8114/cmm/licenseVerificationResult">https://rims.kotsa.or.kr:8114/cmm/licenseVerificationResult</a> |

##### 4.6.4.3 범용 비대면 인증 결과 확인 헤더

| 필드명           | 상태  | 내용        | 필수여부 | 비고                |
|---------------|-----|-----------|------|-------------------|
| host          | Req | 웹서버 정보    | 생략가능 |                   |
| Authorization | Req | 유효한 토큰 정보 | 필수   | "Bearer " + token |

- 비대면 인증 결과 확인 시 HTTP Request (클라이언트 → 인증서버)

```
POST /cmm/licenseVerificationResult
Host: rims.kotsa.or.kr
Authorization: "Bearer " + 토큰
ex) Bearer e03ee90d-83b8-4b1d-ae80-84e3d2a27ytr
```

- 요청을 전송할 때는 위와 같은 형태로 전송하여야 한다.
- 비대면 인증 결과 확인 시 인증서버의 HTTP Response (인증서버 → 클라이언트)

```
HTTP 200 OK
```

- 인증서버는 HTTP 응답메시지를 위의 표와 같이 클라이언트에게 전송한다. 위의 표는 응답코드(HTTP Status code)가 200일 경우의 예이다.
- 토큰 검증 실패 시 401 Unauthorized를 반환한다.

#### 4.6.4.4 범용 비대면 인증 결과 확인 바디

| 항목명   | 항목 Key  | 항목 형식  | 길이 | 필수여부                      | 내용    |
|-------|---------|--------|----|---------------------------|-------|
| 검증번호  | vrfc_no | String | 13 | 필수                        | 대여번호  |
| 사용자정보 | bizinfo | String | 32 | API 호출 사용자와 일치 할 경우 생략 가능 | 사용자ID |

- Body 정보는 “3.2 보안방법”과 같이 암호화한다.
- API 호출 인증키의 사용자와 검증요청 차량의 소유자가 일치하지 않는 플랫폼, 대행사, 조합 등의 경우 사용자정보(bizinfo)를 차량 소유자의 사용자 ID를 입력한다.
- 암호화한 데이터는 “3.1 송신 데이터의 형식”에 맞게 본문을 구성한다.
- 검증번호는 “4.6.2.7 범용 검증번호 생성” 호출 후 응답의 검증번호(vrfc\_no)를 사용한다.

#### 4.6.4.5 범용 비대면 인증 결과 확인 예시

```
gPOST https://rims.kotsa.or.kr:8114/cmm/licenseVerificationResult
```

Authorization: Bearer ...

Content-Type: application/json

Content-Length: ...

암호화 전

```
{
    "bizinfo": "asdf12341....",
    "vrfc_no": "C25-000000014"
}
```

암호화 후

```
{
    "encryptedData": "G+joSdT..."
}
```

응답샘플

HTTP/1.1 200 OK

```
{
    "header": {
        "vrfc_no": "C25-000000014",
        "f rtn_cd": "0",
        "f request_date": "20250811",
        "f send_cnt": "1"
    },
    "body": {
        "f rtn_code": "00",
        "f rtn_desc": "Success"
    }
}
```

```

        "face_similarity": 100.0,
        "face_liveness": 90.68,
        "original_reliability": 100.0,
        "police_lcns_vrfc_cd": "1"
    }
}

```

## 4.6.5 범용 비대면 인증 결과 확인 응답(response) Body 목록

### 4.6.5.1.1 Header

| 항목명  | 항목 Key         | 항목 형식  | 내용                       |
|------|----------------|--------|--------------------------|
| 검증번호 | vrfc_no        | String | 대여번호(형식 : C00-123456789) |
| 요청건수 | f_send_cnt     | String | 요청건수(Default 1)          |
| 요청일시 | f_request_date | String | 요청일시 (형식 : YYYYMMDD)     |
| 요청처리 | f rtn cd       | String | 요청처리코드                   |

### 4.6.5.1.2 Body

| 항목명      | 항목 Key               | 항목 형식  | 내용                          |
|----------|----------------------|--------|-----------------------------|
| 검증결과코드   | f rtn code           | String | 검증결과코드 (적격 : 00, 부적격 : 01~) |
| 얼굴유사도    | face_similarity      | Double | 얼굴유사도 수치 (0~100)            |
| 얼굴활성도    | face_liveness        | Double | 얼굴활성도 수치 (0~100)            |
| 원본신뢰도    | original_reliability | Double | 원본신뢰도 수치 (0~100)            |
| 면허사진검증코드 | police_lcns_vrfc_cd  | String | 면허사진검증코드(적격 : 1)            |

- 응답은 “4.6.3.5 범용 비대면 인증 결과 확인 예시”를 참고한다.
- “4.6.4.1.1 Header”의 f rtn cd는 “5.3 요청처리 응답 코드표”를 참고한다.
- “4.6.4.1.1 Header”의 f rtn cd가 0이 아닐 경우 “4.6.4.1.2 Body”는 생략된다.
- “4.6.4.1.2 Body”의 f rtn code는 “5.2 면허정보 응답 코드표”를 참고한다.
- “4.6.4.1.2 Body”의 face\_similarity는 “5.9 얼굴 유사도 판정 기준표”를 참고하며 활용 시 12단계 이상을 권장한다.
- “4.6.4.1.2 Body”의 face\_liveness는 “5.8 얼굴 활성도 판정 기준표”를 참고하며 활용 시 4단계 이상을 권장한다.
- “4.6.4.1.2 Body”의 police\_lcns\_vrfc\_cd는 “5.11 면허 사진 검증 코드표”를 참고한다.
-

## 4.7 공통 API 오류 응답 예시

응답샘플

HTTP/1.1 200 OK

요청한 API 가 제공되지 않을 경우

```
{  
    "respCode": -100,  
    "errorMsg": "API를 찾을 수 없습니다."  
}
```

bizinfo에 첨부된 아이디가 유효하지 않을 경우

```
{  
    "respCode": -130,  
    "errorMsg": "bizinfo에 첨부된 아이디가 유효하지 않습니다."  
}
```

이하 생략

...

- 응답은 “5.1 공통 API 오류 응답 코드표” 를 참고한다.

## 5 첨부표

## 5.1 공통 API 오류 응답 코드표

| 코드명                           | 코드 값 | 내용   |
|-------------------------------|------|--|
| SERVER_CONNECT_ERROR          | -90  | 연계 서버와 연결 실패                                 |
| NOT_FOUND_API                 | -100 | 요청한 API 가 제공되지 않을 경우                         |
| RIMS_DATABASE_ERROR           | -110 | 운전자격확인시스템 데이터베이스 오류                          |
| BIZINFO_ID_INVALID            | -130 | bizinfo에 첨부된 아이디가 유효하지 않을 경우                 |
| CLIENT_AUTHENTICATION_FAILURE | -140 | 클라이언트 인증 실패                                  |
| TOKEN_GENERATION_ERROR        | -150 | 토큰 생성 중 오류 발생                                |
| INVALID_PARAMETER             | -160 | 유효하지 않은 파라미터                                 |
| MISSING_PARAMETER             | -170 | 파라미터 누락                                      |
| INVALID_ENCRYPTED_DATA        | -180 | 규약 위반: encryptedData 필드가 존재하지 않거나 문자열이 아닌 경우 |
| DECRYPTION_FAILURE            | -190 | 복호화 실패: 데이터 복호화에 실패한 경우                      |
| INVALID_ARRAY_SIZE_MESSAGE    | -200 | 배열 크기와 요청 건수 불일치                             |
| KOROAD_API_CALL_FAILURE       | -210 | 도로교통공단 API 호출 실패                             |
| VEHICLE_MATCH_FAILURE         | -220 | 차량 매칭에 실패한 경우                                |
| INVALID_RENT_NO               | -230 | 유효하지 않은 대여번호인 경우                             |
| WORKING_VERIFICATION          | -240 | 자격 검증 작업이 끝나지 않은 경우                          |
| INVALID_VRFC_NO               | -260 | 유효하지 않은 검증번호인 경우                             |

## 5.2 면허정보 응답 코드표

| 코드명              | 코드 값 | 내용            |
|------------------|------|---------------|
| 검증결과코드(코드 값과 동일) | 00   | 정상            |
| 검증결과코드(코드 값과 동일) | 01   | 면허번호 없음       |
| 검증결과코드(코드 값과 동일) | 02   | 재발급된 면허       |
| 검증결과코드(코드 값과 동일) | 03   | 분실 된 면허       |
| 검증결과코드(코드 값과 동일) | 04   | 사망 취소된 면허     |
| 검증결과코드(코드 값과 동일) | 11   | 취소 된 면허       |
| 검증결과코드(코드 값과 동일) | 12   | 정지 된 면허       |
| 검증결과코드(코드 값과 동일) | 13   | 기간 중 취소 면허    |
| 검증결과코드(코드 값과 동일) | 14   | 기간 중 정지 면허    |
| 검증결과코드(코드 값과 동일) | 21   | 정보불일치(이름)     |
| 검증결과코드(코드 값과 동일) | 22   | 정보불일치(생년월일)   |
| 검증결과코드(코드 값과 동일) | 23   | 정보불일치(암호일련번호) |
| 검증결과코드(코드 값과 동일) | 24   | 정보불일치(종별)     |
| 검증결과코드(코드 값과 동일) | 25   | 필수값 누락(대여기간)  |
| 검증결과코드(코드 값과 동일) | 31   | 암호화 안 된 면허    |

### 5.3 요청처리 응답 코드표

| 코드명            | 코드 값 | 내용                                       |
|----------------|------|--|
| 요청처리(코드 값과 동일) | 0    | 처리 완료                                    |
| 요청처리(코드 값과 동일) | 1    | 인증 정보 없음 (인증 토큰 없음)                      |
| 요청처리(코드 값과 동일) | 2    | 잘못된 인증 정보 (인증 토큰 값 오류)                   |
| 요청처리(코드 값과 동일) | 3    | 인증 실패                                    |
| 요청처리(코드 값과 동일) | 4    | 만료된 토큰 정보                                |
| 요청처리(코드 값과 동일) | 10   | 잘못된 경로를 통한 접근 (등록된 IP와 다름)               |
| 요청처리(코드 값과 동일) | 20   | 복호화 키 정보 없음 (사용자 정보 중 복호화에 사용될 비밀 값이 없음) |
| 요청처리(코드 값과 동일) | 21   | 메시지 복호화 실패                               |
| 요청처리(코드 값과 동일) | 22   | 메시지 암호화 실패                               |
| 요청처리(코드 값과 동일) | 40   | 수수료 결제 정보 없음                             |
| 요청처리(코드 값과 동일) | 41   | 수수료 결제 중 오류                              |
| 요청처리(코드 값과 동일) | 97   | 자동 검증 시스템 작업 장애                          |
| 요청처리(코드 값과 동일) | 98   | 경찰청 운전 면허 조회 장애                          |
| 요청처리(코드 값과 동일) | 99   | 자동 검증 시스템 장애                             |

#### 5.4 면허 종별 코드표

| 코드 | 코드명   |
|----|-------|
| 11 | 1종대형  |
| 12 | 1종보통  |
| 13 | 1종소형  |
| 14 | 대형견인차 |
| 15 | 구난차   |
| 16 | 소형견인차 |
| 32 | 2종보통  |
| 33 | 2종소형  |
| 38 | 2종원자  |

#### 5.5 지역 코드표

| 코드 | 코드명  |
|----|------|
| 11 | 서울   |
| 12 | 부산   |
| 13 | 경기   |
| 13 | 경기남부 |
| 14 | 강원   |
| 15 | 충북   |
| 16 | 충남   |
| 17 | 전북   |
| 18 | 전남   |
| 19 | 경북   |
| 20 | 경남   |
| 21 | 제주   |
| 22 | 대구   |
| 23 | 인천   |
| 24 | 광주   |
| 25 | 대전   |
| 26 | 울산   |
| 28 | 경기북부 |

## 5.6 계정 상태 코드표

| 코드 | 코드명 |
|----|-----|
| 1  | 정상  |
| 2  | 잠김  |
| 3  | 휴면  |
| 4  | 정지  |
| 5  | 삭제  |
| 6  | 탈퇴  |

## 5.7 승인 상태 코드표

| 코드 | 코드명 |
|----|-----|
| 1  | 요청  |
| 2  | 승인  |
| 3  | 반려  |

## 5.8 얼굴 활성도 판정 기준표

- 오거부율 : 실물 얼굴임에도 불구하고 거부가 발생하는 정도

| 판정 기준 | 민감도    | 오거부율       |
|-------|--------|------------|
| 1단계   | 61.07% | 0.48 / 100 |
| 2단계   | 88.93% | 0.8 / 100  |
| 3단계   | 93.97% | 1.04 / 100 |
| 4단계   | 96.84% | 1.4 / 100  |
| 5단계   | 97.50% | 1.55 / 100 |
| 6단계   | 97.74% | 1.62 / 100 |
| 7단계   | 98.81% | 2.15 / 100 |

## 5.9 얼굴 유사도 판정 기준표

- 타인 수락 발생률 : 타인임에도 불구하고 수락이 발생하는 정도

| 판정 기준 | 민감도    | 타인 수락 발생률   |
|-------|--------|-------------|
| 1단계   | 97.64% | 1 / 100     |
| 2단계   | 98.72% | 5 / 1,000   |
| 3단계   | 99.72% | 1 / 1,000   |
| 4단계   | 99.74% | 9 / 10,000  |
| 5단계   | 99.77% | 8 / 10,000  |
| 6단계   | 99.80% | 7 / 10,000  |
| 7단계   | 99.83% | 6 / 10,000  |
| 8단계   | 99.85% | 5 / 10,000  |
| 9단계   | 99.87% | 4 / 10,000  |
| 10단계  | 99.90% | 3 / 10,000  |
| 11단계  | 99.92% | 2 / 10,000  |
| 12단계  | 99.96% | 1 / 10,000  |
| 13단계  | 99.97% | 7 / 100,000 |
| 14단계  | 99.98% | 5 / 100,000 |
| 15단계  | 99.99% | 3 / 100,000 |
| 16단계  | 100%   | 1 / 100,000 |

## 5.10 반복 주기 코드

| 코드명 | 코드 값 | 내용          |
|-----|------|-------------|
| 매주  | 1    | 매주 월요일마다 실행 |
| 보름  | 2    | 15일 마다 실행   |
| 매달  | 3    | 매달 1일마다 실행  |

## 5.11 면허 사진 검증 코드표

| 코드명              | 코드 값 | 내용                |
|------------------|------|-------------------|
| 검증결과코드(코드 값과 동일) | 1    | 정상                |
| 검증결과코드(코드 값과 동일) | -1   | 면허 사진 특징점 형식 불일치  |
| 검증결과코드(코드 값과 동일) | 11   | 정보불일치(면허 사진)      |
| 검증결과코드(코드 값과 동일) | 13   | 정보불일치(면허 사진)      |
| 검증결과코드(코드 값과 동일) | 19   | 면허 사진 특징점 누락      |
| 검증결과코드(코드 값과 동일) | 31   | 정보불일치(주민등록번호)     |
| 검증결과코드(코드 값과 동일) | 32   | 정보불일치(이름)         |
| 검증결과코드(코드 값과 동일) | 33   | 정보불일치(발급일자)       |
| 검증결과코드(코드 값과 동일) | 39   | 정보불일치(면허번호)       |
| 검증결과코드(코드 값과 동일) | 42   | 정보불일치(면허번호)       |
| 검증결과코드(코드 값과 동일) | 88   | 정보 형식 불일치         |
| 검증결과코드(코드 값과 동일) | -500 | 신분증 진위여부 서버 통신 실패 |

## 5.12 범용 API 유형 코드표

| 분류            | 코드  | 코드명    | 내용                       |
|---------------|-----|--------|--------------------------|
| 운전자격확인 단 건 조회 | S00 | 범용     | 범용 운전자격확인 단 건 조회 API     |
|               | S01 | 트럭커    | 트럭커 운전자격확인 단 건 조회 API    |
|               | S02 | 애완동물택시 | 애완동물택시 운전자격확인 단 건 조회 API |
| 운전자격확인 배치 조회  | B00 | 범용     | 범용 운전자격확인 배치 조회 API      |
|               | B01 | 트럭커    | 트럭커 운전자격확인 배치 조회 API     |
|               | B02 | 애완동물택시 | 애완동물택시 운전자격확인 배치 조회 API  |
| 검증번호생성        | G00 | 범용     | 범용 검증번호 생성 API           |
|               | G01 | 트럭커    | 트럭커 검증번호 생성 API          |
|               | G02 | 애완동물택시 | 애완동물택시 검증번호 생성 API       |

### 5.13 검색유형 코드표

| 구분  | API명        | 코드명    | 코드 값                    | 내용               |
|-----|-------------|--------|-------------------------|------------------|
| 렌트카 | 업체별 대여이력 조회 | 검색유형코드 | rent_start_date         | 대여 시작일 기준(기본값)   |
|     |             | 검색유형코드 | rent_end_date           | 대여 종료일 기준        |
|     |             | 검색유형코드 | rent_create_date        | 대여 등록일 기준        |
|     | 반복예약 결과 조회  | 검색유형코드 | reservation_start_date  | 반복예약 시작일 기준(기본값) |
|     |             | 검색유형코드 | reservation_end_date    | 반복예약 종료일 기준      |
|     |             | 검색유형코드 | reservation_create_date | 반복예약 등록일 기준      |

## 6 개발 예시

- 아래 예시는 Java로 작성한 참고용 예시이며 환경과 사용하는 라이브러리에 따라 코드는 달라질 수 있음.

```
public class ExService {

    private final WebClient webClient;

    public ExService(WebClient webClient) {
        this.webClient = webClient;
    }

    private static final int AES_KEY_LENGTH = 256;
    private static final String INSTANCE_TYPE = "AES/ECB/PKCS5Padding";
    private static final String CLIENT_SEC = "1234b0842f2848c090a0835c6e7ac599";

    // 토큰 발생
    public void getToken() throws UnsupportedEncodingException {
        String certKey = "사용자 인증키";
        String uri = "http://서버도메인/col/oauth2?grantType=password";
        String authHeader = "Basic " +
            Base64.getEncoder().encodeToString(certKey.getBytes("UTF-8"));
        webClient.get()
            .uri(uri)
            .header(HttpHeaders.AUTHORIZATION, authHeader)
            .exchange()
            .flatMap(response -> {

```

```

        HttpHeaders headers = response.headers().asHttpHeaders();
        List<String> authHeaderValues =
headers.get(HttpHeaders.AUTHORIZATION);
        String authorizationHeaderValue = authHeaderValues.get(0);
        log.info("토큰 : {}", authorizationHeaderValue);
        return Mono.empty();
    }
    .doOnError(error -> {
        log.error("에러 : {}", error.getMessage());
    }).subscribe();
}

public static String encrypt(String plaintext, String secretKey) throws Exception {
    SecretKeySpec keySpec = generateKey(secretKey);
    Cipher cipher = Cipher.getInstance(INSTANCE_TYPE);
    cipher.init(Cipher.ENCRYPT_MODE, keySpec);
    byte[] encryptedBytes = cipher.doFinal(plaintext.getBytes(StandardCharsets.UTF_8));
    return Base64.getEncoder().encodeToString(encryptedBytes);
}

private static SecretKeySpec generateKey(String secretKey) {
    byte[] keyData = Arrays.copyOf(secretKey.getBytes(StandardCharsets.UTF_8),
AES_KEY_LENGTH / 8);
    return new SecretKeySpec(keyData, "AES");
}

public static String single = "{\n" +
    " \"f_license_no\": \"123456789012\",\\n" +
    " \"f_resident_name\": \"홍길동\",\\n" +
    " \"f_licn_con_code\": \"12\",\\n" +
    " \"f_from_date\": \"20231106\",\\n" +
    " \"f_to_date\": \"20231106\",\\n" +
    " \"vhcl_reg_no\": \"99임9999\"\\n" +
"}";

public static String batch = "{\n" +
    " \"f_send_cnt\": \"2\",\\n" +
    " \"bizinfo\": \"asdf12341....\",\\n" +
    " \"requestList\": [\n" +
    "     {\n" +
    "         \"f_license_no\": \"123456789012\",\\n" +
    "         \"f_resident_name\": \"홍길동\",\\n" +
    "         \"f_licn_con_code\": \"12\",\\n" +
    "         \"f_from_date\": \"20231106\",\\n" +
    "         \"f_to_date\": \"20231106\",\\n" +
    "         \"vhcl_reg_no\": \"99임9999\"\\n" +
}

```

```

    "    },\n" +
    "    {\n" +
    "        \"f_license_no\": \"123456789012\",\\n" +
    "        \"f_resident_name\": \"홍길동\",\\n" +
    "        \"f_licn_con_code\": \"12\",\\n" +
    "        \"f_from_date\": \"20231106\",\\n" +
    "        \"f_to_date\": \"20231106\",\\n" +
    "        \"vhcl_reg_no\": \"99임9999\"\\n" +
    "    }\n" +
    "}\n" +
"}";

```

// 운전자격검증 API 호출(소화물배송대행서비스사업자 API와 동일)

```

public void callApi() {
    String singleUri = "http://서버도메인/licenseVerification"; // 단건 조회
    String batchUri = "http://서버도메인/licenseVerificationBatch"; // 배치 조회
    String token = "Bearer 026f1744-1234-4bcb-b0a6-990f9da9a745"; // 발급한 토큰
    JSONObject body = new JSONObject();
    try {
        String encryptedStr = encrypt(single, CLIENT_SEC); // 목적에 따라 단건, 또는
        배치 uri 선택 및 본문을 구성한다.
        body.put("encryptedData", encryptedStr);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    webClient.post()
        .uri(singleUri)
        .header(HttpHeaders.AUTHORIZATION, token)
        .contentType(MediaType.APPLICATION_JSON)
        .bodyValue(body)
        .retrieve()
        .bodyToMono(String.class)
        .flatMap(response -> {
            log.info("호출 결과 확인 : {}", response);
            return Mono.empty();
        })
        .onErrorResume(e -> {
            log.error("에러", e);
            return Mono.error(RuntimeException::new);
        }).subscribe();
}

```

// 사용자 검증 API 호출

```

public void checkBusinessRegistration() {
    String token = "Bearer 026f1744-e3e0-4bcb-b0a6-990f9da9a745"; // 발급한 토큰
    String userCheckUri =

```

```
"http://서버도메인/col/isUserInDatabase?user_id=사용자ID&brno=사업자등록번호";
    webClient.get()
        .uri(userCheckUri)
        .header(HttpHeaders.AUTHORIZATION, token)
        .retrieve()
        .bodyToMono(String.class)
        .flatMap(response -> {
            log.info("호출 결과 확인 : {}", response);
            return Mono.empty();
        })
        .onErrorResume(e -> {
            log.error("에러", e);
            return Mono.error(RuntimeException::new);
        }).subscribe();
}
```