

운전면허정보 자동검증시스템

# 오픈 API 연동가이드

(V 1.6)

2021.04



# 문서 정보

## 문서개요

이 문서는 도로교통공단에서 제공하는 운전면허정보 자동검증 연동을 위한 API 기능과 개발 가이드를 제공합니다.

## 문의사항

이 문서의 내용에 오류가 있거나 내용과 관련한 문의 사항이 있으면 운전면허정보 자동검증시스템 고객센터로 문의합니다.

Tel : 033-749-5118

E-mail : dlv\_master@koroad.or.kr

## 변경내역

버전		이력사항	승인자
0.9	2017.07.28	초안작성	이은승
1.0	2017.08.10	송수신 파라메터 명 변경 (경찰청 협의 완료), 응답코드표 수정, 운전면허 종별 코드 추가, 수수료결제 정보 추가, 암호알고리즘 제공방식 추가	이은승
1.1	2017.08.21	처리 응답코드 추가	이은승
1.2	2018.01.29	당일 검증 면허정보 수수료 미처리 추가 지역코드 추가 reflush_token 삭제	이은승
1.3	2019.10.28	본문 암호화 알고리즘 변경	이은승
1.6	2021.04.30	면허진위여부 조회 간소화	고요한

# 문서 설명

## 소스 코드 표기

이 문서에서 소스 코드는 회색 바탕에 검정색 글씨로 표기합니다.

```
public static void main(String args[]) {  
    System.out.println("HelloWorld!");  
}
```

## 예시 표기

이 문서에서 예시를 표현하는 내용은 회색의 이탤릭체로 표시합니다.

*예) 예시를 표현하는 내용은 회색 이탤릭체*

## 샘플 표기

이 문서에서 샘플은 회색 박스 안에 검정색 글씨로 표기합니다.

```
{  
    name:홍길동,  
    age:24,  
    nickname:바람,  
    email:admin@galaxia.co.kr  
}
```

# - 목차 -

1. 개요.....	1
1.1. 소개.....	1
1.2. 통신방법.....	1
1.2.1. HTTP Protocol .....	1
1.2.2. HTTP Method .....	1
1.2.3. 데이터 포맷 .....	1
1.3. 개발을 위한 오픈 API KEY 발급 방법 .....	1
1.3.1. 테스트 서버 개발 시.....	1
1.3.2. 운영서버 서버 적용 시 .....	1
1.4. 보안 방법 .....	2
1.4.1. 암호화 알고리즘 .....	2
1.4.2. 암호화 키 정보.....	2
1.4.3. 암호화 알고리즘 제공 방식.....	2
2. 메시지 명세 .....	6
2.1. 파라미터.....	6
2.1.1. 요청 메시지 .....	6
2.1.2 응답 메시지 .....	6
3. 서버정보 및 사용자 인증.....	7
3.1. 오픈 API 서버정보 .....	7
3.2. 인증.....	7
3.2.1. OAuth2 .....	7
3.2.2. 인증 URL .....	7

3.2.3. 인증 키 정보 .....	7
3.2.4. 인증 키 (Token) 유효기간 .....	8
4. API 목록 .....	9
4.1. API 변경 설명 .....	9
4.2. 처리 프로세스 .....	10
5. API 정의 .....	11
5.1. 면허정보 단건조회 .....	11
5.1.1. 프로토콜 상세 .....	11
5.1.2 요청 변수(Request Parameters).....	11
5.1.3. 응답 변수(Response Parameters) .....	13
5.2. 면허정보 배치조회 .....	14
5.2.1. 프로토콜 상세 .....	14
5.2.2. 요청 변수(Request Parameters).....	14
5.2.3. 응답 변수(Response Parameters) .....	15
5.3. 면허정보 단건조회( 중복과금 제거).....	17
5.3.1. 프로토콜 상세 .....	17
5.3.2 요청 변수(Request Parameters).....	17
5.3.3. 응답 변수(Response Parameters) .....	19
5.4. 면허정보 배치조회(중복과금 제거) .....	20
5.4.1. 프로토콜 상세 .....	20
5.4.2. 요청 변수(Request Parameters).....	20
5.4.3. 응답 변수(Response Parameters) .....	21
5.5. 면허정보 단건조회( 알고리즘 변경).....	23
5.5.1. 프로토콜 상세 .....	23

5.5.2 요청 변수(Request Parameters).....	23
5.5.3. 응답 변수(Response Parameters) .....	25
5.6. 면허정보 배치조회(암호알고리즘 변경).....	26
5.6.1. 프로토콜 상세 .....	26
5.6.2. 요청 변수(Request Parameters).....	26
5.6.3. 응답 변수(Response Parameters) .....	27
5.7. 면허정보 단건조회(중복과금 제거, 암호 알고리즘 변경).....	29
5.7.1. 프로토콜 상세 .....	29
5.7.2 요청 변수(Request Parameters).....	29
5.7.3. 응답 변수(Response Parameters) .....	31
5.8. 면허정보 배치조회(중복과금 제거, 암호 알고리즘 변경).....	32
5.8.1. 프로토콜 상세 .....	32
5.8.2. 요청 변수(Request Parameters).....	32
5.8.3. 응답 변수(Response Parameters) .....	33
6. 사용예제 (자바 코드만 제공 – httpclient 라이브러리 사용) .....	35
6.1. 토큰 발생 .....	35
6.2. API 호출 .....	36
7. Appendix A ( 면허정보 응답 코드표).....	38
8. Appendix B (면허 종별 코드표).....	38
9. Appendix C (요청처리 응답 코드표) .....	39
10. Appendix D (지역 코드표) .....	40

# 1. 개요

## 1.1. 소개

본 서비스는 도로교통공단 운전면허정보 자동검증시스템과 회원사 시스템의 연동을 위한 오픈 API 서비스를 정의하고 설명합니다.

## 1.2. 통신방법

### 1.2.1. HTTP Protocol

- 운영서버 : HTTPS (SSL 통신)
- 개발서버 : HTTP

### 1.2.2. HTTP Method

- Post

### 1.2.3. 데이터 포맷

- JSON

## 1.3. 개발을 위한 오픈 API KEY 발급 방법

### 1.3.1. 테스트 서버 개발 시

- API 사용 신청서 작성 후 제출 시 고객센터에서 API KEY (Client\_ID, Client\_Secret) 제공.  
(제공되는 Client\_secret은 Base64 인코딩한 값임)

### 1.3.2. 운영서버 서버 적용 시

- dlv.koroad.or.kr 사이트에 회원 가입 후 오픈 API 메뉴를 통해 직접 발급 받음

## 1.4. 보안 방법

### 1.4.1. 암호화 알고리즘

- Aria Block 암호화 알고리즘을 사용하여 암호화하고 데이터는 Base64 인코딩한다.

### 1.4.2. 암호화 키 정보

- 테스트 서버: 제공되는 API KEY 값 중 Client\_Secret 값 (Base64 인코딩한 값임)
- 운영서버 서버 : dlv.koroad.or.kr 사이트에서 발급 받은 오픈 API KEY 값 중 Client\_Secret 값

### 1.4.3. 암호화 알고리즘 제공 방식

암호화 알고리즘은 라이브러리 제공 또는 암호화 웹 서비스 제공 등 2가지 방식으로 제공됩니다.

사업자의 시스템이 JAVA로 구성된 경우 제공되는 라이브러리를 사용하여 암호화 알고리즘을 적용하면 되고, 사업자 시스템이 JAVA가 아닌 경우 (PHP, ASP 등) 제공되는 암호화 웹 서비스를 사용하여 암호화 알고리즘을 적용하여 개발하면 됩니다.

## A. 라이브러리 제공 (dlv\_koroad.jar)

Sample (java 코드 기준, 추후 중단 예정)

```
String client_secret = "8117a5d75e9909eb7858b5638803d72c707fb744";
String data = "[";
for(int i=0;i<10;i++) {
    if (i != 0)
        data += ",\n";
    data += "{\"f_license_no\" : \"121234561212\", "+
        "\"f_resident_name\" : \"홍길동\", "+
        "\"f_resident_date\" : \"123456\", "+
        "\"f_seq_no\" : \"A23456\", "+
        "\"f_licn_con_code\" : \"11\", "+
        "\"f_from_date\" : \"20170728\", "+
        "\"f_to_date\" : \"20170728\"}";
}
data += "]";
ARIACipher ac = new ARIACipher(Base64.encode(client_secret.getBytes()));
byte[] encBody = data.getBytes();
```



```
byte[] encBody = ac.encrypt(encBody);
String encStr = new String(encBody, "EUC-KR");
```

### Sample 2 (암호 알고리즘 변경, java 코드 기준 ARIACipher → ARIACipher256으로 변경)

```
String client_secret = "8117a5d75e9909eb7858b5638803d72c707fb744";
String data = "[";
for(int i=0;i<10;i++) {
    if (i != 0)
        data += ",\n";
    data += "{\"f_license_no\" : \"121234561212\", "+
        "\"f_resident_name\" : \"홍길동\", "+
        "\"f_resident_date\" : \"123456\", "+
        "\"f_seq_no\" : \"A23456\", "+
        "\"f_licn_con_code\" : \"11\", "+
        "\"f_from_date\" : \"20170728\", "+
        "\"f_to_date\" : \"20170728\"}";
}
data += "]";
ARIACipher256 ac = new ARIACipher256(Base64.encode(client_secret.getBytes()));
byte[] encBody = data.getBytes();
byte[] encBody = ac.encrypt(encBody);
String encStr = new String(encBody, "EUC-KR");
```

## B. 암호화 웹 서비스 제공 (실행 모듈 제공)

### Sample (javascript 코드 기준, 추후 중단 예정)

```
var dataTemp;

$.ajax({
    url: "http://localhost:9028/encrypt.do",
    method: "post",
    type: "json",
    data: JSON.stringify(form),
    contentType: "application/json",
    success: function(data) {
```

```
        console.log(data);
        dataTemp = data.encText;
    }
});

function decodeCall() {
    form1.decText = dataTemp;

    $.ajax({
        url: "http://localhost:9028/decrypt.do",
        method: "post",
        type: "json",
        data: JSON.stringify(form1),
        contentType: "application/json",
        success: function(data) {
            console.log(data);
            Base64.decode(data.decText)
        }
    });
}
```

**Sample 2 (javascript 코드 기준, encrypt.do → encryptv2.do 호출URL 변경)**

```
var dataTemp;

$.ajax({
    url: "http://localhost:9028/encryptv2.do",
    method: "post",
    type: "json",
    data: JSON.stringify(form),
    contentType: "application/json",
    success: function(data) {
        console.log(data);
        dataTemp = data.encText;
    }
});

function decodeCall() {
    form1.decText = dataTemp;
```

```
$.ajax({
  url: "http://localhost:9028/decryptv2.do",
  method: "post",
  type: "json",
  data: JSON.stringify(form1),
  contentType: "application/json",
  success: function(data) {
    console.log(data);
    Base64.decode(data.decText)
  }
});
}
```

## 2. 메시지 명세

메시지는 HTTP (HTTPS) 프로토콜을 사용하여 JSON 형식의 응답으로 메시지를 주고 받습니다.

응답 값의 인코딩은 "UTF-8" 입니다.

### 2.1. 파라미터

#### 2.1.1. 요청 메시지

요청 변수의 데이터 형식은 HTTP Method에 따라 아래와 같이 입력합니다.

HTTP Method	데이터 형식
POST	HTTP 요청의 Content에 JSON 형식으로 전달 <pre>{   "seq": 123 }</pre>

#### 2.1.2 응답 메시지

처리된 결과는 JSON 형식으로 리턴 됩니다.

## 3. 서버정보 및 사용자 인증

### 3.1. 오픈 API 서버정보

환경	API URL	IP	Port
개발	없음	203.243.39.19	80
운영	https://dlv.koroad.or.kr	없음.	443

### 3.2. 인증

#### 3.2.1. OAuth2

OAuth2 인증방식에 따라 제공되는 Client\_ID, Client\_Secret로 인증 토큰을 발급받아 Token을 통해 API 호출을 한다.

#### 3.2.2. 인증 URL

##### A. 테스트 서버

API명	METHOD	URL	설명
토큰 발급	GET	http://개발서버/oauth2/token	인증토큰 발급. 추후 중단 예정
토큰 발급	GET	http://개발서버/oauth2/token2	인증토큰 발급. 암호 알고리즘 변경

##### B. 운영서버

API명	METHOD	URL	설명
토큰 발급	GET	http://운영서버/oauth2/token	인증토큰 발급. 추후 중단 예정
토큰 발급	GET	http://운영서버/oauth2/token2	인증토큰 발급. 암호 알고리즘 변경

#### 3.2.3. 인증 키 정보

##### A. 테스트 서버 (예시)

- API 사용 신청서 작성 후 제출 시 고객센터에서 API KEY (Client\_ID, Client\_Secret) 제공

- Client\_ID : OTk4MDIyOGEtMWZkOC00NTAxLWJlNzctY2U4ZTk4ZWVzMThj  
(Base64 인코딩된 값임)
- 인증 토큰 : return URL 을 통해 발급 (Base64 인코딩된 값임)

### B. 운영서버

- dlvr.koroad.or.kr 사이트에 회원 가입 후 오픈 API 메뉴를 통해 직접 발급 받음

### 3.2.4. 인증 키 (Token) 유효기간

- 토큰 생성 후 3시간

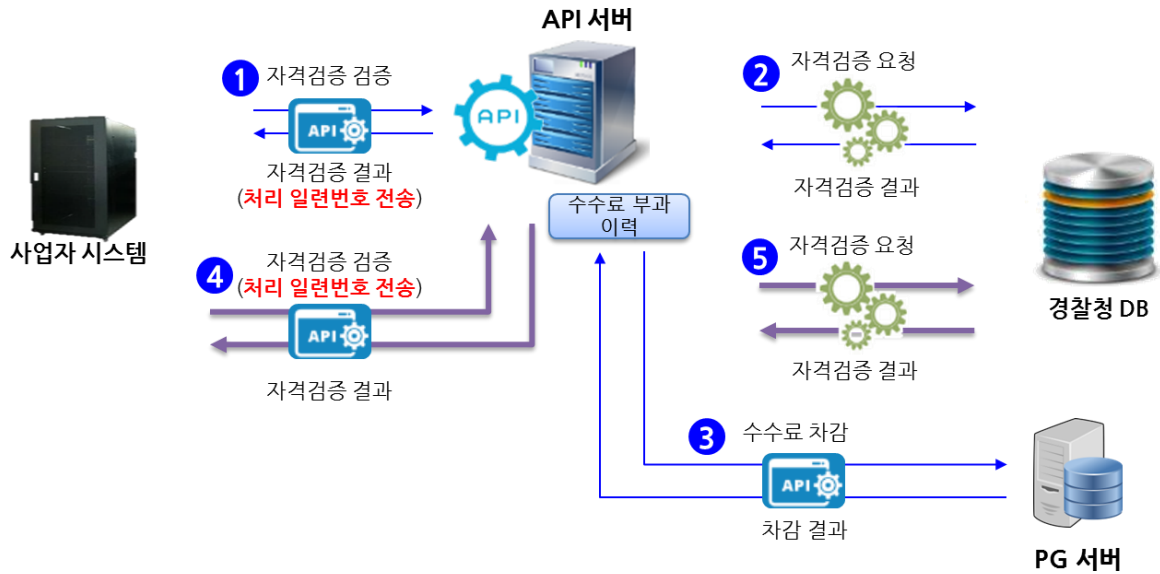
## 4. API 목록

구분	API명	METHOD	설명
기존	면허정보 단건조회	POST	한 명의 운전자에 대한 면허 정보를 검증합니다. 추후 중단 예정
	면허정보 배치조회	POST	복수 운전자에 대한 면허 정보를 검증합니다. 추후 중단 예정
	면허정보 단건조회(v1)	POST	한 명의 운전자에 대한 면허 정보를 검증합니다. (당일 동일 면허자에 한해 1회만 수수료 과금) 추후 중단 예정
	면허정보 배치조회(v1)	POST	복수 운전자에 대한 면허 정보를 검증합니다. (당일 동일 면허자에 한해 1회만 수수료 과금) 추후 중단 예정
신규	면허정보 단건조회(v2)	POST	한 명의 운전자에 대한 면허 정보를 검증합니다. 암호화 알고리즘 변경
	면허정보 배치조회(v2)	POST	복수 운전자에 대한 면허 정보를 검증합니다. 암호화 알고리즘 변경
	면허정보 단건조회(v3)	POST	한 명의 운전자에 대한 면허 정보를 검증합니다. 암호화 알고리즘 변경 (당일 동일 면허자에 한해 1회만 수수료 과금)
	면허정보 배치조회(v3)	POST	복수 운전자에 대한 면허 정보를 검증합니다. 암호화 알고리즘 변경 (당일 동일 면허자에 한해 1회만 수수료 과금)

### 4.1. API 변경 설명

기본 제공 API는 단건과 배치 조회와 중복 수수료 과금 면제를 위한 면허정보 단건조회(v1) 및 배치조회(v1)는 당일 동일 면허정보 검증 건에 한해 1회만 수수료를 과금 하도록 하는 API임, 추후 암호 알고리즘 취약점에 따라 중단 예정임.

신규로 암호 알고리즘 변경에 따른 단건,배치 조회 API(v2)와 중복수수료 면제 단건, 배치조회(v3) API를 추가함.



## 4.2. 처리 프로세스

- ① 최초 A 면허의 자격 검증 요청 (처리 일련번호 없음)
- ② 자격 검증 처리
- ③ 수수료 차감 (처리일련번호 생성), 사업자 시스템에 저장
- ④ 자격 검증 처리 결과와 함께 처리 일련번호 전달
- ⑤ 당일 A 면허의 자격 검증 재 요청 (처리 일련번호 있음)
- ⑥ 자격 검증 처리
- ⑦ 처리 번호 확인 후 처리 번호가 당일 생성 처리 번호일 경우 수수료 차감하지 않음



## 5. API 정의

### 5.1. 면허정보 단건조회

한 명의 운전자 면허에 대한 검증 조회를 요청합니다.

#### 5.1.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/onevalidator.do
Charset	UTF-8
Content-Type	application/json
Method	POST

#### 5.1.2 요청 변수(Request Parameters)

##### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

##### Body

파라미터	자료형	길이	필수	설명
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

## Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 있는 경우

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "123456",
    "f_seq_no" : "1P0456",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

## Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 없는 경우

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "",
    "f_seq_no" : "",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

### 5.1.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9. Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7. Appendix A (면허정보 응답 코드표) 참조

#### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "1",
    "f_request_date" : "20170728122321",
    "f_rtn_cd" : "00",
    "f_rtn_msg" : ""
  }
  "body" : {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  }
}
```

## 5.2. 면허정보 배치조회

2명 이상의 운전 면허정보를 조회합니다.

### 5.2.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/batchvalidator.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.2.2. 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

Sample (body 암호화 생략) – 암호일련번호, 생년월일이 있는 경우

```
{
  "header" : {
    "f_send_cnt" : "2",
```

```

        "f_request_date" : "20170728",
        "f_pin_info" : "1234567890:12345678"
    },
    "body" : [{
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "",
        "f_seq_no" : "",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    },
    {
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "123456",
        "f_seq_no" : "123456",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    }
    ]
}

```

### 5.2.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9.Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7,Appendix A (면허정보 응답 코드표) 참조

### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "1",
    "f_request_date" : "20170728122321",
    "f_rtn_yn" : "00",
    "f_rtn_msg" : ""
  }
  "body" : [{
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  },
  {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  }]
}
```

### 5.3. 면허정보 단건조회( 중복과금 제거)

한명의 운전자에 대한 면허에 대한 검증정보를 조회합니다.

#### 5.3.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/onevalidatorv1.do
Charset	UTF-8
Content-Type	application/json
Method	POST

#### 5.3.2 요청 변수(Request Parameters)

##### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

##### Body

파라미터	자료형	길이	필수	설명
f_vaild_key	String	150	X	중복과금 방지를 위한 과금 번호
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

필수가 아닌 파라미터도 원형은 유지되어야 함.

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 있는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "123456",
    "f_seq_no" : "123456",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 없는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "",
    "f_seq_no" : "",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```



```
}
}
```

### 5.3.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9. Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호 (번호는 1*1*3*5*1*1* 형태로 마스킹 처리함.)
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)
f_vaild_key	String	150	중복과금 방지를 위한 과금 번호

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7. Appendix A (면허정보 응답 코드표) 참조

#### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "1",
    "f_request_date" : "20170728122321",
    "f_rtn_cd" : "00",
    "f_rtn_msg" : ""
  }
  "body" : {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" :
"abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"
  }
}
```

## 5.4. 면허정보 배치조회(중복과금 제거)

복수의 운전자에 대한 면허에 대한 검증정보를 조회합니다.

### 5.4.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/batchvalidator.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.4.2. 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_vaild_key	String	150	X	중복과금 방지를 위한 과금 번호
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

#### Sample (body 암호화 생략)

```
{
```

```

"header" : {
    "f_send_cnt" : "2",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
},
"body" : [{
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "123456",
    "f_seq_no" : "123456",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
},
{
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "",
    "f_seq_no" : "",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
}]
}

```

### 5.4.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9.Appendix C (요청처리 응답코드표)를 참조

**Body**

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)
f_vaild_key	String	150	중복과금 방지를 위한 과금 번호

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7,Appendex A (면허정보 응답 코드표) 참조

**Sample (body 암호화 생략)**

```
{
  "header" : {
    "f_send_cnt " : "2",
    "f_request_date" : "20170728122321",
    "f_rtn_yn" : "00",
    "f_rtn_msg" : ""
  }
  "body" : [{
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" : "00"
  },
  {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop"
  }
  ]
}
```

## 5.5. 면허정보 단건조회( 알고리즘 변경)

한명의 운전자에게 대한 면허에 대한 검증정보를 조회합니다.

### 5.5.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/onevalidatorv2.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.5.2 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

필수가 아닌 파라미터도 원형은 유지되어야 함.

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 있는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "123456",
    "f_seq_no" : "123456",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 없는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "",
    "f_seq_no" : "",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

### 5.5.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9.Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호 (번호는 1*1*3*5*1*1* 형태로 마스킹 처리함.)
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7,Appendix A (면허정보 응답 코드표) 참조

#### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "1",
    "f_request_date" : "20170728122321",
    "f_rtn_cd" : "00",
    "f_rtn_msg" : ""
  }
  "body" : {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  }
}
```

## 5.6. 면허정보 배치조회(암호알고리즘 변경)

복수의 운전자에 대한 면허에 대한 검증정보를 조회합니다.

### 5.6.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/batchvalidatorv2.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.6.2. 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

#### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt" : "2",
```



```

        "f_request_date" : "20170728",
        "f_pin_info" : "1234567890:12345678"
    },
    "body" : [{
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "123456",
        "f_seq_no" : "123456",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    },
    {
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "123456",
        "f_seq_no" : "",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    }
    ]
}

```

### 5.6.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9.Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7,Appendex A (면허정보 응답 코드표) 참조

Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "2",
    "f_request_date" : "20170728122321",
    "f_rtn_yn" : "00",
    "f_rtn_msg" : ""
  }
  "body" : [{
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  },
  {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00"
  }]
}
```

## 5.7. 면허정보 단건조회(중복과금 제거, 암호 알고리즘 변경)

한명의 운전자에 대한 면허에 대한 검증정보를 조회합니다.

### 5.7.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/onevalidatorv3.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.7.2 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_vaild_key	String	150	X	중복과금 방지를 위한 과금 번호
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

필수가 아닌 파라미터도 원형은 유지되어야 함.

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 있는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "123456",
    "f_seq_no" : "123456",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

**Sample (body 암호화 생략) – 암호 일련번호, 생년월일이 없는 경우**

```
{
  "header" : {
    "f_send_cnt" : "1",
    "f_request_date" : "20170728",
    "f_pin_info" : "1234567890:12345678"
  },
  "body" : {
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
    "f_license_no" : "121234561212",
    "f_resident_name" : "홍길동",
    "f_resident_date" : "",
    "f_seq_no" : "",
    "f_licn_con_code" : "11",
    "f_from_date" : "20170728",
    "f_to_date" : "20170728"
  }
}
```

```
}
}
```

### 5.7.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9. Appendix C (요청처리 응답코드표)를 참조

#### Body

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호 (번호는 1*1*3*5*1*1* 형태로 마스킹 처리함.)
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)
f_vaild_key	String	150	중복과금 방지를 위한 과금 번호

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7. Appendix A (면허정보 응답 코드표) 참조

#### Sample (body 암호화 생략)

```
{
  "header" : {
    "f_send_cnt " : "1",
    "f_request_date" : "20170728122321",
    "f_rtn_cd" : "00",
    "f_rtn_msg" : ""
  }
  "body" : {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" :
    "abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz"
  }
}
```

## 5.8. 면허정보 배치조회(중복과금 제거, 암호 알고리즘 변경)

복수의 운전자에 대한 면허에 대한 검증정보를 조회합니다.

### 5.8.1. 프로토콜 상세

항목	내용
HTTP type	HTTPS
Request URI	[OpenAPI 서버]/api/batchvalidator.do
Charset	UTF-8
Content-Type	application/json
Method	POST

### 5.8.2. 요청 변수(Request Parameters)

#### Header

파라미터	자료형	길이	필수	설명
f_send_cnt	String	4	X	요청건수(Default 1)
f_request_date	String	8	O	요청일시 (형식 : YYYYMMDD)
f_pin_info	String	32	O	수수료 결제정보(pin번호:pin비밀번호 Base64 인코딩)

#### Body

파라미터	자료형	길이	필수	설명
f_vaild_key	String	150	X	중복과금 방지를 위한 과금 번호
f_license_no	String	12	O	운전면허번호
f_resident_name	String	32	O	운전면허자명
f_resident_date	String	6	X	생년월일
f_seq_no	String	6	X	암호일련번호
f_licn_con_code	String	2	O	면허종별(코드)
f_from_date	String	8	O	대여기간시작
f_to_date	String	8	O	대여기간종료

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ 운전면허번호(f\_license\_no)은 총 12자리이며, 면허증 운전면허번호 앞에 지역(서울, 인천 등) 이름이 쓰여진 경우 10. Appendix D (지역 코드표)를 참고하여 지역 명을 숫자로 치환하여 전달

#### Sample (body 암호화 생략)

```
{
```

```

    "header" : {
        "f_send_cnt" : "2",
        "f_request_date" : "20170728",
        "f_pin_info" : "1234567890:12345678"
    },
    "body" : [{
        "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "123456",
        "f_seq_no" : "123456",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    },
    {
        "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop",
        "f_license_no" : "121234561212",
        "f_resident_name" : "홍길동",
        "f_resident_date" : "",
        "f_seq_no" : "",
        "f_licn_con_code" : "11",
        "f_from_date" : "20170728",
        "f_to_date" : "20170728"
    }
  ]
}

```

### 5.8.3. 응답 변수(Response Parameters)

#### Header

파라미터	자료형	길이	설명
f_send_cnt	String	4	요청건수(Default 1)
f_request_date	String	8	요청일시 (형식 : YYYYMMDD)
f_rtn_cd	String	5	요청처리성공여부(0/요청처리 Error Code)
f_rtn_msg	String	128	요청처리실패메세지

※ f\_rtn\_cd 값은 9.Appendix C (요청처리 응답코드표)를 참조

**Body**

파라미터	자료형	길이	설명
f_license_no	String	12	운전면허번호
f_rtn_code	String	2	검증결과코드 (적격 : 00, 부적격 : 01~)
f_vaild_key	String	150	중복과금 방지를 위한 과금 번호

※ Body 정보는 암호화알고리즘을 통해 암호화 후 Base64로 인코딩하여 전달함.

※ f\_rtn\_code는 7,Appendex A (면허정보 응답 코드표) 참조

**Sample (body 암호화 생략)**

```
{
  "header" : {
    "f_send_cnt " : "2",
    "f_request_date" : "20170728122321",
    "f_rtn_yn" : "00",
    "f_rtn_msg" : ""
  }
  "body" : [{
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" : "00"
  },
  {
    "f_license_no" : "121234561212",
    "f_rtn_code" : "00",
    "f_vaild_key" :
"abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop"
  }
  ]
}
```



## 6. 사용예제 (자바 코드만 제공 – httpClient 라이브러리 사용)

### 6.1. 토큰 발생

```
String client_id = "회원사 Client_id";
String client_secret = "회원사 Client_secret";
String authHeader = OAuth2ClientUtil.generateBasicAuthHeaderString(client_id,
        client_secret);
HttpClient client = new HttpClient();
System.out.println(authHeader);
BufferedReader br = null;
GetMethod method = new GetMethod("http://서버아이피/oauth2/token2");
method.setQueryString(new NameValuePair[] {
        new NameValuePair("grantType", "password")
});
Header header = new Header();
header.setName("Authorization");
header.setValue(authHeader);
method.setRequestHeader(header);
try{
    int returnCode = client.executeMethod(method);
    String rtnMsg = "";
    if(returnCode == HttpStatus.SC_OK) {
        rtnMsg = method.getResponseBodyAsString();
        System.out.println(rtnMsg);
    }
} catch (Exception e) {
    e.printStackTrace();
} finally
{ method.releaseConnection();
  if(br != null) try { br.close(); } catch (Exception fe) {}
}
```

## 6.2. API 호출

```

HttpClient client1 = new HttpClient();
BufferedReader br = null;
String lno = "123456789012"
String lnm = "홍길동"
String bno = ""
String sno = ""
String sdt = "20170823"
String edt = "20170823"
String ltno = "12"
String valkey = ""
authHeader = OAuth2ClientUtil.generateBearerTokenHeaderString(rtnMsg);
String jsonBody = "{\"f_license_no\" : \""+lno+"\", "+
    "\"f_resident_name\" : \""+lnm+"\", "+
    "\"f_resident_date\" : \""+bno+"\", "+
    "\"f_seq_no\" : \""+sno+"\", "+
    "\"f_vaild_key\" : \""+valkey+"\", "+
    "\"f_licn_con_code\" : \""+ltno+"\", "+
    "\"f_from_date\" : \""+sdt+"\", "+
    "\"f_to_date\" : \""+edt+"\"}";
System.out.println(jsonBody);
ARIACipher256 ac = new ARIACipher256(Base64.encode(client_secret.getBytes()));
byte[] encBody = ac.encrypt(jsonBody.getBytes("UTF-8"));
String encStr = Base64.encode(encBody);
PostMethod post = new PostMethod(API URL);
post.setRequestHeader("Authorization", authHeader);
String reqStr = "{\"header\" : {\"f_send_cnt\" : \"1\", \"f_request_date\" : \"20170823\", \"f_pin_info\" : \""+Base64.encode("전자민원캐쉬에서 발급된 pinnum: 전자민원캐쉬에서 발급된 pinpwd".getBytes())+"\"}, \"body\" : \""+encStr+"\"}";
System.out.println(reqStr);
StringRequestEntity postingString = new
StringRequestEntity(reqStr, "application/json", "UTF-8");
post.setRequestEntity(postingString);
try {

    returnCode = client1.executeMethod(post);

```

```

        if(returnCode == HttpStatus.SC_OK)
            {br = new BufferedReader(new
InputStreamReader(post.getResponseBodyAsStream()));
            String readLine;
            StringBuffer st = new StringBuffer();
            while(((readLine = br.readLine()) != null)) {
                st.append(readLine);
            }
            System.out.println(st.toString());
            JSONParser parser = new JSONParser();
            JSONObject jsonObj = (JSONObject) parser.parse(st.toString());
            System.out.println(jsonObj.toJSONString());
            System.out.println(((String)jsonObj.get("header")));
            if (!"0".equals((String)((JSONObject)
parser.parse((String)jsonObj.get("header"))).get("f_rtn_cd"))) {
                out.println(" header :"+jsonObj.toJSONString());
            } else {
                byte[] encBody1 =
Base64.decode((String)jsonObj.get("body"));
                byte[] decBody = ac.decrypt(encBody1);
                JSONObject bodys = (JSONObject) parser.parse(new
String(decBody,"UTF-8"));
                JSONObject headers = (JSONObject)jsonObj.get("header");
                out.println(" header :"+headers.toJSONString());
                out.println(" body :"+bodys.toJSONString());
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally
    { method.releaseConnection();
      if(br != null) try { br.close(); } catch (Exception fe) {}
    }
}

```

## 7. Appendix A ( 면허정보 응답 코드표)

응답코드	응답메시지
00	정상
01	면허번호 없음.
02	재발급된 면허
03	분실된 면허
04	사망취소된 면허
11	취소된 면허
12	정지된 면허
13	기간중취소면허
14	기간중정지면허
21	정보불일치(이름)
22	정보불일치(생년월일)
23	정보불일치(암호일련번호)
24	정보불일치(종별)
31	암호화안된 면허

## 8. Appendix B (면허 종별 코드표)

코드	코드명
11	1 종대형
12	1 종보통
13	1 종소형
14	대형견인차(트레일러)
15	구난차(레커)
16	소형견인차
32	2 종보통
33	2 종소형
38	2 종원자

괄호안 이름은 면허종별 이름이 바뀌기 전의 이름으로 운전면허증에 트레일러, 레커로 표기되어진 것이 있음.

## 9. Appendix C (요청처리 응답 코드표)

코드	코드명
0	처리완료
1	인증정보없음(인증Token 없음)
2	잘못된 인증정보(인증Token 값이상)
3	인증실패
4	만료된 토큰 정보
10	잘못된 경로를 통한 접근(등록된 IP 와다름)
20	복호화키 정보없음(사용자정보중 복호화에 사용될 Secret 값이 없음)
21	메세지 복호화실패
22	메세지 암호화실패
40	수수료결제 정보없음.
41	수수료결제 중 오류
97	자동검증시스템 작업 장애
98	경찰청 운전면허 조회 장애
99	자동검증시스템 장애

## 10. Appendix D (지역 코드표)

코드	지역명
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