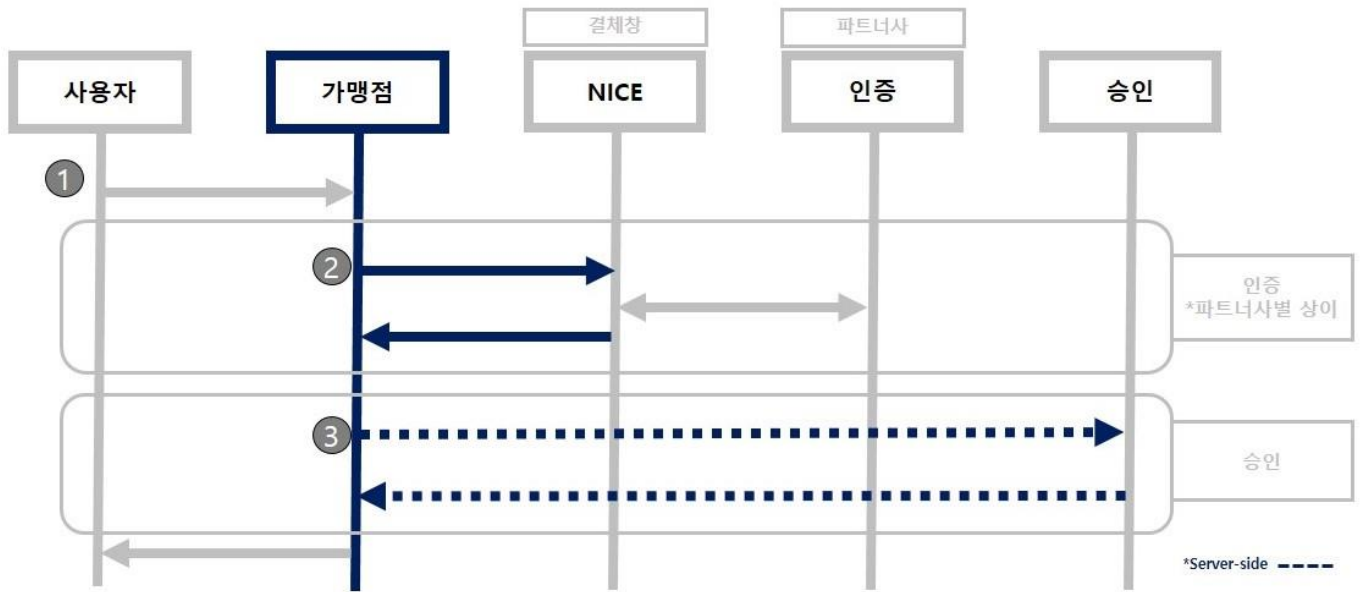


## 1. Over-view



[그림 1-1] 인증결제 flow

### Flow over-view

#### 1. 인증 결제 정보 입력

- 사용자는 가맹점이 구성한 웹 페이지에서 결제 정보를 입력합니다.

#### 2. 인증 요청, 응답 프로세스

- Target : (PC) <https://web.nicepay.co.kr/v3/webstd/js/nicepay-3.0.js>, 인코딩 : euc-kr  
(Mobile) <https://web.nicepay.co.kr/v3/v3Payment.jsp>, 인코딩 : euc-kr
- 가맹점은 사용자로부터 전달받은 결제 정보와 가맹점 정보를 가공하여 PC인 경우 nicepay-3.0.js로, Mobile인 경우 v3Payment.jsp 로 인증을 요청합니다.
- 사용자는 선택한 결제수단에 따라 파트너사(카드사, 금융결제원(뱅크페이) 등)의 인증 프로세스를 진행합니다.
- 가맹점에서는 사용자의 인증 완료 여부를 확인한 후 승인 프로세스를 진행합니다.

#### 3. 승인 요청, 응답

- Target : [https://webapi.nicepay.co.kr/webapi/pay\\_process.jsp](https://webapi.nicepay.co.kr/webapi/pay_process.jsp), 인코딩 : euc-kr
- 가맹점은 Server-side로 인증결과 값을 가공하여 pay\_process.jsp로 승인을 요청합니다.
- 승인 응답 값을 가공하여 이후 가맹점 프로세스를 진행합니다.
- PG사의 응답데이터는 JSON 기본 포맷으로 응답되며, 기능 추가에 따라 응답 필드가 추가될 수 있습니다. 이에 따라 가맹점에서 응답필드가 추가 될 수 있음을 고려해야 합니다.
- 응답데이터의 유효성 검증은 Signature 필드를 이용하여 검증해야 합니다.

## 주의 사항

- \* 제공되는 파라미터 이외 값은 보장 하지 않습니다. 사용 가능한 결제 수단은 MID(가맹점 ID) 설정에 따라 변경 될 수 있습니다. 별도 Test 계정이 필요한 경우 영업팀을 통해 지원 요청 합니다.
- \* iframe, 혹은 별도 레이어-팝업 내 결제창을 처리하는 경우 작동이 원활하지 않을 수 있으며, 권장 하지 않습니다.
- \* 결제창 내 Script 변경 혹은 event 처리를 매뉴얼에서 제공하지 않는 방법으로 처리 하는 경우 지원이 어렵습니다.
- \* 나이스에서 제공하는 테스트 계정으로 가상개좌 테스트 시 가상계좌 발급까지만 권장합니다.
- 입금 및 입금통보, 환불 테스트를 원하는 경우 영업담당자를 통해 별도의 계정을 발급받은 후 테스트해야 합니다
- \*PG 사의 응답데이터는 JSON 기본 포맷으로 응답되며, 기능 추가에 따라 응답 필드가 추가될 수 있습니다. 이에 따라 가맹점에서 응답필드가 추가 될 수 있음을 고려해야 합니다.

## 2. 방화벽 설정

### 2.1 방화벽 연결

	API 결제	입금 통보
연결대상	121.133.126.56 (webapi.nicepay.co.kr)	121.133.126.10 121.133.126.11 211.33.136.39
포트	443	
프로토콜	HTTPS	
연결방향	OUTBOUND	INBOUND

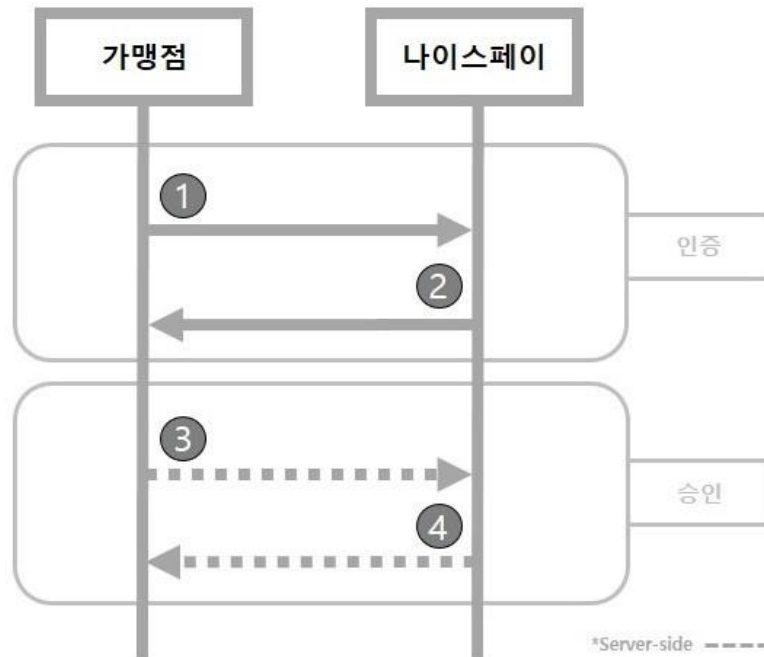
[표 2-1] 방화벽 설정 표

### 2.2 Timeout 정보

Connection (sec)	Receive(Read) (sec)
5	30

[표 2-2] Timeout 정보

### 3. 예외처리



[그림 3-1] 예외처리 사항

#### 1. 인증 요청 시 주의 및 오류 사항

##### A. 결제정보 암호화

- i. 인증 요청 시 결제정보를 반드시 암호화하여 나이스페이로 전달합니다.

SignData 암호화

<SignData>

암호화 항목 :

전문생성일시+상점아이디+금액+상점키

- ii. 금액, 상점아이디, 상점 키 등 결제 정보가 변경되는 경우 SignData를 재구성하여야 합니다.  
(변경 후 미적용 시 SIGNDATA 검증 실패 오류가 발생합니다.)

- B. PC는 Script(js), Mobile은 웹 페이지(jsp)를 각각 호출하며, 결제 기기에 맞지 않는 결제창을 호출하는 경우 결제 진행이 제한됩니다.

#### 2. 인증 응답 시 주의 및 오류 사항

- A. 사용자 결제수단의 유효성 문제가 있는 경우 파트너사에서 전달하는 오류코드 및 메시지를 그대로 전달합니다.  
해당 경우, 파트너사를 통해 사용자가 직접 결제를 시도한 결제수단에 대하여 문의해야 합니다.
- B. 가맹점 내부 인증 오류의 상세 확인이 필요한 경우,  
인증 시 사용한 Moid, MID, EdiDate 등을 정확하게 기재하여 [it@nicepay.co.kr](mailto:it@nicepay.co.kr)로 보내어 확인할 수 있습니다.

### 3. 승인 요청 시 주의 및 오류 사항

- A. 인증이 성공한 경우에만 파라미터를 가공하여 승인 요청을 진행하시기 바랍니다.
- B. 결제정보 암호화
  - i. 인증 요청 시 결제정보를 반드시 암호화하여 나이스페이로 전달합니다.

SignData 암호화

<SignData>

암호화 항목 :

인증토큰+상점아이디+금액+주문번호+상점키

- C. 호출 오류
  - i. 방화벽 설정이 정상인지 체크 합니다.

IP: 121.133.126.56 / OUTBOUND

URL: <https://webapi.nicepay.co.kr>

포트: 443 / HTTPS

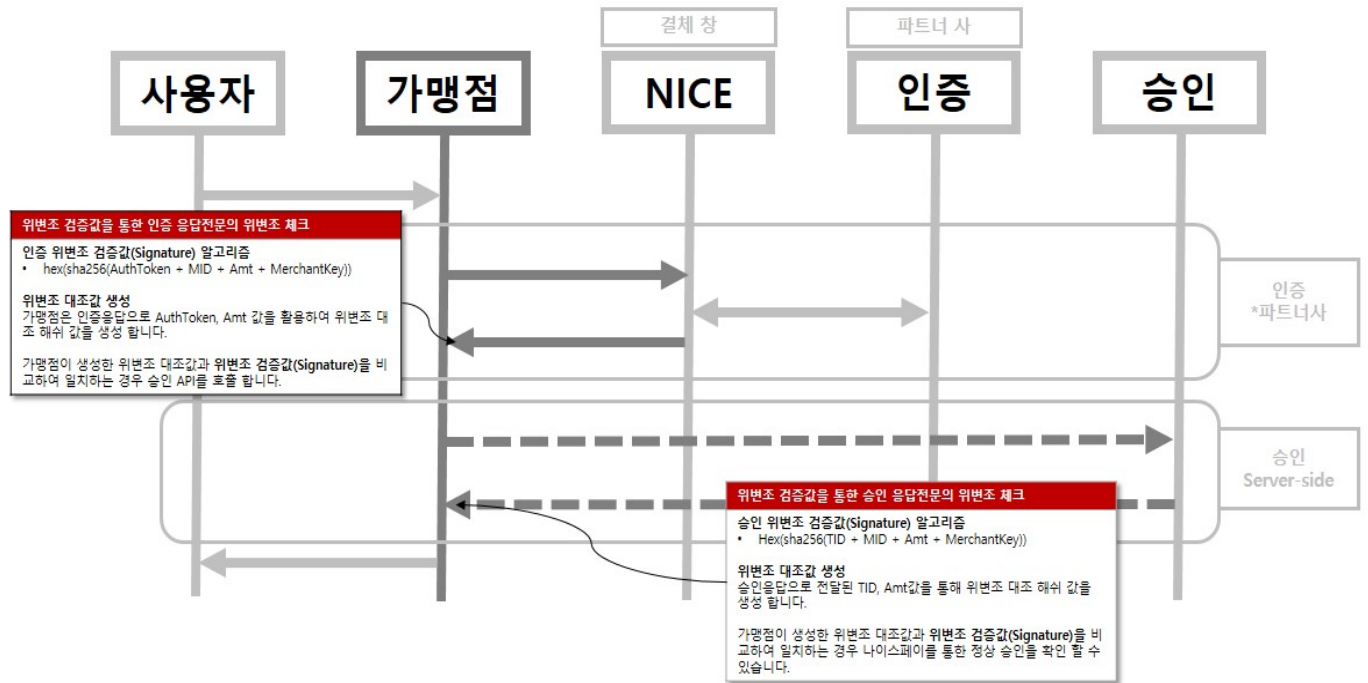
- D. 호출이 되지 않는 경우(Connection Timeout 등) 망취소 프로세스를 적용하여 거래를 취소할 수 있습니다.

### 4. 승인 응답 중 주의 및 오류 사항

- A. 승인 요청-응답 데이터가 외부로 노출되지 않도록 주의가 필요합니다.
- B. 승인 시점에 문제가 발생하는 경우 망취소 프로세스를 적용하여 거래를 취소할 수 있습니다.

## 4. 보안처리

### 4.1 인증결제 위변조 검증



[그림 4-1] 인증결제 위변조 검증

#### 4.1.1 주의사항

**Signature** 파라미터는 요청 데이터에 대한 무결성 검증을 위해 전달하는 파라미터로, 허위 결제 요청 등 결제 및 보안 관련 이슈가 발생할 만한 요소를 방지하기 위해 연동 시 사용하시기 바랍니다.

#### 4.1.2 위변조 검증 처리 방법

##### A. 위변조 검증값을 통한 인증 응답전문의 위변조체크

- 인증 요청시 응답으로 위변조 검증 값인 Signature를 전달 받습니다.
- 가맹점은 인증응답으로 전달된 AuthToken, Amt 값을 활용하여 위변조 대조 해쉬값을 생성합니다. 해당 규칙으로 생성한 대조값으로 전달 받은 Signature 값과 대조를 진행합니다.

**인증 위변조 대조값, Signature 생성 규칙**

$\text{Hex}(\text{sha256}(\text{AuthToken} + \text{MID} + \text{Amt} + \text{MerchantKey}))$

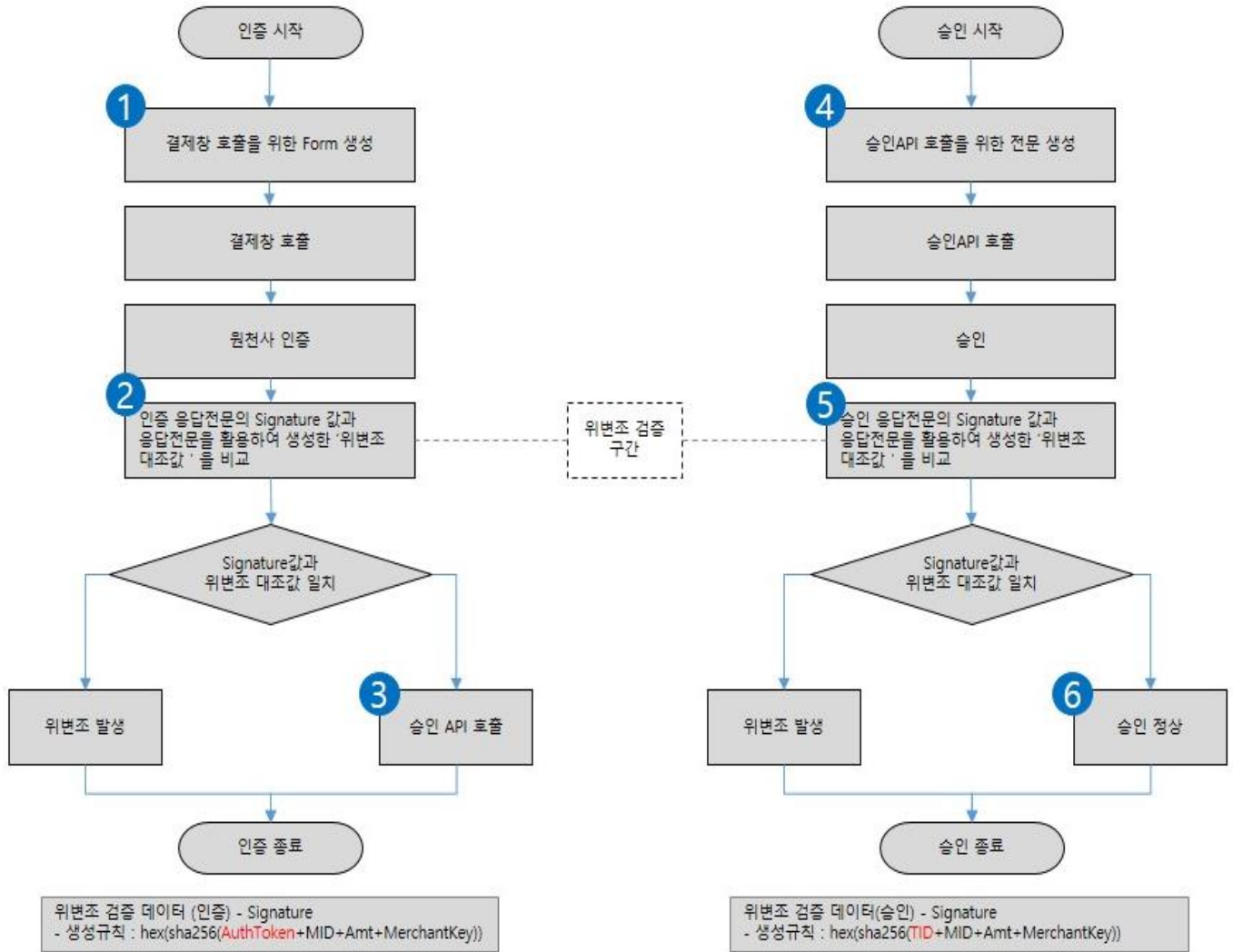
##### B. 위변조 검증값을 통한 승인 응답전문의 위변조체크

- 승인 요청시 응답으로 위변조 검증 값인 Signature를 전달 받습니다.
- 가맹점은 승인응답으로 전달된 TID, Amt 값을 활용하여 위변조 대조 해쉬값을 생성합니다. 해당 규칙으로 생성한 대조값으로 전달 받은 Signature 값과 대조를 진행합니다. 해당 대조가 정상적으로 일치할 경우 나이스페이를 통한 정상승인을 진행합니다.

**승인 위변조 대조값, Signature 생성 규칙**

$\text{Hex}(\text{sha256}(\text{TID} + \text{MID} + \text{Amt} + \text{MerchantKey}))$

## 4.2 위변조 검증 진행한 결제 Flow



[그림 4-2] 위변조 검증 진행한 결제 Flow

### 4.2.1 위변조 검증 진행 과정 상세 설명

1. 인증 시작시 결제창 호출을 위한 Form이 생성됩니다.
2. 인증 응답 전문 중 Signature 값과 응답전문을 활용하여 생성한 위변조 대조값으로 비교합니다.
3. 위변조 대조값과 Signature 값이 일치할 경우 승인 API 호출합니다.
4. 승인 API 호출을 위한 전문을 생성합니다.
5. 승인 응답 전문 중 Signature 값과 응답전문을 활용하여 생성한 위변조 대조값으로 비교합니다.
6. 위변조 대조값과 Signature 값이 일치할 경우 승인을 정상 처리합니다.

## 5. 인증 요청-응답 파라미터

### 5.1 Request URL

PC : <https://web.nicepay.co.kr/v3/webstd/js/nicepay-3.0.js>

Mobile : <https://web.nicepay.co.kr/v3/v3Payment.jsp>

### 5.2 Request Info

Protocol	Method	Encoding	Content-Type
HTTP/1.1	POST	EUC-KR	application/x-www-form-urlencoded;

[표 5-1] 요청 정보

### 5.3 인증 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
GoodsName	40	O	상품명 ("등 특수기호 사용시 별도 문의 필요)
Amt	12	O	결제 금액. (특수기호 사용금지, 결제 최소금액은 기준정보로 관리)
MID	10	O	상점 아이디
EdiDate	30	O	전문생성일시 (YYYYMMDDHHMISS)
Moid	64	O	상점 주문번호, 상점 고유의 식별값 (NICEPAY 가공 없음)
SignData	500	O	위변조 검증 데이터 - 생성규칙 : hex(sha256(EdiDate + MID + Amt + MerchantKey)) <u>*10. 암호화 항목 참조</u>
BuyerName	30		구매자 이름
BuyerTel	40		구매자 전화번호
ReturnURL	500		인증 처리 후 redirect url ( <b>PC Not use, Mobile Only</b> )
PayMethod	10		아래 값들중 하나로 설정. CARD: 신용카드(SSG페이포함) / BANK: 계좌이체 / VBANK: 가상계좌 / CELLPHONE: 휴대폰결제 / SSG_BANK: SSG 은행계좌 / GIFT_SSG: SSG머니 (모바일 Only) / GIFT_CULT: 문화상품권 / CMS_BANK: CMS계좌간편결제 - 다중값 사용을 위해서는 콤마로 구분 ex) CARD,BANK : 카드와 계좌이체 이용
ReqReserved	500		상점 정보 전달용 예비필드 (NICEPAY 가공없음) 전달한 값이 그대로 반환됨. 단, 큰따옴표("")는 이용불가
BuyerEmail	60		구매자 이메일
SupplyAmt	12		별도 공급가액 설정 시 사용
GoodsVat	12		별도 부가세 설정 시 사용
ServiceAmt	12		별도 봉사료 설정 시 사용
TaxFreeAmt	12		별도 면세 금액 설정 시 사용
CurrencyCode	3		KRW: 원화(Default) / USD: 달러 / CNY: 위안화
TransType	1		0: 일반거래 / 1: 에스프로 거래
Period	17		결제창에 노출될 제공기간 FORMAT YYYYMMDD:YYYYMMDD 예시) 20190901:20191001

LogoImage	100		로고 이미지의 full URL 전달 * 채널별 사이즈 (단위: pixel) 모바일 : width 50 X height 50 웹표준 : width 94 X height 25
NpLang	2		EN: 영문 / CN: 중문 / KO: 국문(Default) 예시) NpLang=EN -> 결제창 영문표기 NpLang=CN -> 결제창 중문표기
CharSet	10		가맹점 서버의 encoding 방식 전달 예시) utf-8 / euc-kr(Default)
MallUserID	20		상점에서 사용하는 사용자 아이디 (결제 이후 가맹점관리자페이지 내 “사용자ID” 항목으로 검색 가능) <b>계좌간편결제, 문화상품권 결제 시 필수(5.3.4 / 5.3.5 참조)</b>
NPDisableScroll	1		결제창 스크롤 미사용 여부 (PC Only, Y: 미사용 / N(default): 사용)
WapUrl	500		가맹점 독립 APP 을 이용하여 개발시 (Webview 연동, Mobile Only) ISP 혹은 계좌이체 연동 시 인증 완료 후 결과에 대한 focus 가 가맹점 APP 으로 돌아오기 위해 설정하는 가맹점 APP 의 scheme 값을 설정 예시) 가맹점 App scheme 이 nicepaysample 인 경우 WapUrl=nicepaysample://
IspCancelUrl	500		가맹점 독립 APP을 이용하여 개발시 (Webview 연동) ISP APP 에서 취소 버튼 터치 시 리턴되는 APP scheme 설정 예시) nicepaysample://ISPCancel

[표 5-2] 인증 요청 파라미터(공통)



### 5.3.1 신용카드 추가 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
SelectQuota	2		할부개월 설정 옵션 - ','를 구분자로 할부개월 나열 - 2 자리의 할부기간 설정 (일시불:00, 2 개월:02 ,3 개월:03 ...으로 설정) Ex) SelectQuota=03 (3 개월 할부) 최소금액 : 50,000 원 이상 <b>모바일 : SelectCardCode &amp; SelectQuota 가 함께 요청된 경우에만 SelectQuota 가 유효함</b> <b>웹표준 : SelectQuota 단독 처리.</b>
SelectCardCode	2		특정카드사 선택옵션 - ','를 구분자로 카드코드 나열 예시) SelectCardCode=02 (국민 카드만 이용할 수 있도록 제한) SelectCardCode=02,04,07 (국민, 삼성, 현대 세 카드사만을 이용할 수 있도록 제한) <b>*SSG페이, SSG페이+SSG머니 복합결제의 경우 아래와 같이 필수 적용</b> SelectCardCode=SSG (SSG페이), SelectCardCode=SSGPAY_MONEY(SSG 복합결제)
ShopInterest	1		상점무이자 사용여부 (0:미사용, 1:사용, 미입력(null):MID에 등록된 상점 무이자 전체 이용)
QuotaInterest	-		상점무이자 카드 정보, 기준정보에 등록된 무이자 중 사용할 정보 설정 - ShopInterest=1 인 경우 적용됨 - ','를 구분자로 하여 카드코드 나열 - ':'를 구분자로 하여 카드코드와 할부개월 구분 FORMAT - CardCode:할부개월,할부개월 CardCode:할부개월,할부개월 예시) - 무이자정보 : 01:06,07 02:06,07 설명) 비씨카드 6,7개월 무이자과 국민카드 6,7개월 무이자 사용 설정

[표 5-3] 결제수단이 신용카드인 경우 옵션 파라미터

### 5.3.2 가상계좌 추가 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
VbankExpDate	12	○	가상계좌 입금기한, 8 자리(YYYYMMDD) 또는 12 자리(YYYYMMDDHHMMI) 예시) 20201117 또는 201901171415

[표 5-4] 결제수단이 가상계좌인 경우 옵션 파라미터

### 5.3.3 휴대폰 소액결제 추가 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
GoodsCl	1	○	휴대폰 상품구분, <b>휴대폰 소액결제 시 필수</b> (0:컨텐츠, 1:현물)

[표 5-5] 결제수단이 휴대폰 소액결제인 경우 옵션 파라미터

### 5.3.4 문화상품권 추가 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
MallUserID	20	O	상점에서 사용하는 사용자 아이디, <b>문화상품권 결제 시 필수</b>

[표 5-6] 결제수단이 문화상품권인 경우 옵션 파라미터

### 5.3.5 CMS계좌간편결제 추가 요청 파라미터

파라미터	사이즈(Byte)	필수	설명
MallUserID	20	O	상점에서 사용하는 사용자 아이디, <b>CMS 계좌간편결제 시 필수</b>
BankPayMemID	16		사전 발급받은 CMS 계좌간편결제 회원 ID
BankPayDesc	12		CMS 계좌간편결제 출금통장 인자 (인출된 계좌에 표기될 정보)
BankPayMemNotiURL	255	O	CMS 계좌간편결제 가입 시 간편결제 회원의 상태 및 정보를 실시간 통보하는 URL (비동기), <b>CMS 계좌간편결제 이용 시 필수</b>
DirectShowOpt	100	O	인증사 바로 호출 옵션("CMS_BANK" 고정), <b>CMS 계좌간편결제 이용 시 필수</b>

[표 5-7] 결제수단이 문화상품권, CMS 계좌간편결제인 경우 옵션 파라미터

## 5.4 인증 응답 파라미터

파라미터	사이즈(Byte)	설명
AuthResultCode	4	인증 결과 코드 (0000 : 성공 / 그외 실패) (0000 인 경우에만 NextAppURL 을 호출하여 승인 요청 하도록 함) <u>*11. 결과코드 항목 참조</u>
AuthResultMsg	2000	인증결과 메시지 예시) 인증성공
AuthToken	40	인증 TOKEN, Unique Key
PayMethod	10	인증한 지불수단 코드
MID	10	상점 아이디
Moid	64	상점 주문번호, 상점 고유의 식별값(NICEPAY 가공 없음)
Amt	12	결제 금액
Signature	500	위변조 검증 데이터 - 생성규칙 : hex(sha256(AuthToken + MID + Amt + MerchantKey)) - 데이터 유효성 검증을 위해, 가맹점 수준에서 비교하는 로직 구현 필수 <u>*10. 암호화 항목 참조</u>
ReqReserved	500	상점 정보 전달용 예비 필드(NICEPAY 가공 없음)
TxTid	30	거래 ID, 승인거래 매핑에 사용(인증 성공시 리턴)
NextAppURL	255	승인 요청 URL 최종 결제를 위해, 승인 요청할 URL (인증결과를 포함하여, NextAppURL로 POST) - 인증 성공한경우 리턴됨 <a href="https://webapi.nicepay.co.kr/webapi/pay_process.jsp">https://webapi.nicepay.co.kr/webapi/pay_process.jsp</a>
NetCancelURL	255	망취소 요청 URL 승인요청 후 기타오류(Network 지연 또는 상점 내부 오류) 발생시, 취소 요청할 URL - 인증 성공한경우 리턴됨 <a href="https://webapi.nicepay.co.kr/webapi/cancel_process.jsp">https://webapi.nicepay.co.kr/webapi/cancel_process.jsp</a>

[표 5-8] 인증 응답 파라미터

## 6. 승인 요청-응답 파라미터

### 6.1 Request URL

[https://webapi.nicepay.co.kr/webapi/pay\\_process.jsp](https://webapi.nicepay.co.kr/webapi/pay_process.jsp)

### 6.2 Request Info

Protocol	Method	Encoding	Content-Type
HTTP/1.1	POST	EUC-KR	application/x-www-form-urlencoded;

[표 6-1] 요청 정보

### 6.3 요청 파라미터

파라미터	Byte	필수	설명
TID	30	O	거래번호, 인증결과로 받은 TxTid 파라미터 설정 예시) nicepay00m03011911140919215275
AuthToken	40	O	인증 TOKEN, 인증결과로 받은 AuthToken 파라미터 설정 예시) NICETOKNF435F661A2D54ED799BFB9F4B3F7E369
MID	10	O	상점 ID
Amt	12	O	결제금액
EdiDate	14	O	전문생성일시 (YYYYMMDDHHMISS)
SignData	256	O	위변조 검증 Data 생성규칙 : hex(sha256(AuthToken + MID + Amt + EdiDate + MerchantKey)) <b>*10. 암호화 항목 참조</b>
CharSet	10		응답파라미터 인코딩 방식
EdiType	10		응답전문 유형 (default(미설정): JSON / KV(설정): Key=Value형식 응답) 예시) EdiType=KV 요청한 경우. ResultCode=0000&ResultMsg=정상처리되었습니다.&TID=nicetest00m01011912311450329275.....
MallReserved	500		상점 정보 전달용 예비필드 (Nicepay 가공없음) 전달한 값이 그대로 반환 됩니다.

[표 6-2] 승인 요청 파라미터

#### 6.4 응답 파라미터

파라미터	Byte	설명
ResultCode	4	결제 결과 코드 (결제수단별 성공코드) 신용카드: 3001, 계좌이체: 4000, 가상계좌: 4100, 휴대폰: A000, SSG은행계좌: 0000, SSG복합결제: 0000, CMS계좌간편결제: 0000 <a href="#">*11. 결과코드 항목 참조</a>
ResultMsg	100	결제 결과 메시지 예시) 카드 결제 성공
Amt	12	거래금액 예시) 000000001004
MID	10	상점 아이디 예시) nicepay00m
Moid	64	상점 주문번호 예시) mnoid1234567890
Signature	500	위변조 검증 데이터 - 생성규칙 : hex( $\text{sha256}(\text{TID} + \text{MID} + \text{Amt} + \text{MerchantKey})$ ) - 데이터 유효성 검증을 위해, 가맹점 수준에서 비교하는 로직 구현 필수 <a href="#">*10. 암호화 항목 참조</a>
BuyerEmail	60	구매자이메일
BuyerTel	40	구매자전화번호
BuyerName	30	구매자명
GoodsName	40	상품명
TID	30	거래번호, 거래를 구분하는 transaction ID 예시) nicepay00m03011911140919215275
AuthCode	30	승인번호 (신용카드, 계좌이체, 휴대폰)
AuthDate	12	승인날짜 (YYMMDDHHMMSS) 예시) 190419165200
PayMethod	10	결제 수단 코드 신용카드: CARD, 계좌이체: BANK, 가상계좌: VBANK, 핸드폰: CELLPHONE, SSG은행계좌: SSG_BANK, CMS계좌간편결제: CMS_BANK
CartData	4000	장바구니 데이터
MallReserved	500	상점 정보 전달용 예비필드 (승인 요청 시 전달한 값 그대로 반환, Nicepay 가공없음)

[표 6-3] 승인 응답 파라미터(공통)

#### 6.4.1 신용카드 추가 응답 파라미터

파라미터	Byte	설명
CardCode	3	결제 카드사 코드 *12. 파트너사 코드 항목 참조 예시) 04
CardName	20	결제 카드사명 예시) 삼성
CardNo	20	카드번호 예시) 53611234****1234
CardQuota	2	할부개월 예시) 00 (일시불), 03(3개월 할부)
CardInterest	1	상점분담 무이자 적용 여부 (0:일반, 1:무이자)
AcquCardCode	3	매입카드사코드 예시) 04
AcquCardName	100	매입카드사명 예시) 삼성
CardCl	1	카드 구분 (0: 신용, 1: 체크)
CcPartCl	1	부분취소 가능 여부 (0: 불가능, 1: 가능)
ClickpayCl	2	간편결제 종류 예시) 6 : SKPAY / 8 : SAMSUNGPAY / 15 : PAYCO / 16 : KAKAOPAY
CouponAmt	12	쿠폰금액
CouponMinAmt	12	쿠폰최소기준금액
PointAppAmt	12	포인트승인금액
MultiCl	1	(페이코, 카카오 결제 시에만 응답) 복합결제 여부, 1:복합결제 사용, 0:복합결제 미사용
MultiCardAcquAmt	12	(페이코, 카카오 결제 시에만 응답) 복합 신용카드 금액, 반드시 숫자로만 입력, 예시) 1000 원인 경우 -> 1000
MultiPointAmt	12	(페이코, 카카오 결제 시에만 응답) 복합 포인트 금액, 반드시 숫자로만 입력, 예시) 5000 원인 경우 -> 5000
MultiCouponAmt	12	(페이코, 카카오 결제 시에만 응답) 복합 쿠폰 금액, 반드시 숫자로만 입력, 예시) 5000원인 경우 -> 5000
RcptType	1	(네이버페이-포인트 결제 시에만 응답) 현금영수증 타입, 예시) 1:소득공제, 2:지출증빙, 그외:발행안함
RcptTID	30	(네이버페이-포인트 결제 시에만 응답) 현금영수증 TID, 현금영수증 거래인 경우 필수
RcptAuthCode	30	(네이버페이-포인트 결제 시에만 응답) 현금영수증 승인번호, 현금영수증 거래인 경우 필수

[표 6-4] 신용카드 결제 추가 응답 파라미터

#### 6.4.2 계좌이체 추가 응답 파라미터

파라미터	Byte	설명
BankCode	3	결제은행 코드 *12. 파트너사 코드 항목 참조
BankName	20	결제은행 명
RcptType	1	현금영수증 타입 (0:발행안함, 1:소득공제, 2:지출증빙)
RcptTID	30	현금영수증 TID
RcptAuthCode	30	현금영수증 승인번호

[표 6-5] 계좌이체 결제 추가 응답 파라미터

#### 6.4.3 가상계좌 추가 응답 파라미터

파라미터	Byte	설명
VbankBankCode	3	결제은행 코드 *12. 파트너사 코드 항목 참조
VbankBankName	20	결제은행명 예시) SC제일은행
VbankNum	20	가상계좌 번호
VbankExpDate	8	가상계좌 입금 만료일(yyyyMMdd) 예시) 20190101
VbankExpTime	6	가상계좌 입금 만료시간(HH:mm:ss) 예시) 235959

[표 6-6] 가상계좌 결제 추가 응답 파라미터

#### 6.4.4 CMS 계좌간편결제 추가 응답 파라미터

파라미터	Byte	설명
EasyBankCode	3	결제은행 코드 *12. 파트너사 코드 항목 참조
EasyBankName	20	결제은행명 예시) 기업은행
EasyBankUserNo	20	간편결제 회원 일련번호 예시) 3000004005555555
EasyBankAccount	20	결제계좌 예시) *****01012
RcptResultCode	4	현금영수증 결과코드 예시) 7001
RcptResultMsg	100	현금영수증 결과 메시지 예시) 현금영수증 처리 성공
RcptType	1	현금영수증 타입 (0: 발행안함, 1: 소득공제, 2: 지출증빙)
RcptTID	30	현금영수증 TID 예시) nicetest00m04011104191651325596
RcptAuthCode	30	현금영수증 승인번호 예시) I74685858

[표 6-7] CMS 계좌간편결제 추가 응답 파라미터

## 7. 망취소 요청-응답 파라미터

### 7.1 Request URL

https://webapi.nicepay.co.kr/webapi/cancel\_process.jsp

### 7.2 Request Info

Protocol	Method	Encoding	Content-Type
HTTP/1.1	POST	EUC-KR	application/x-www-form-urlencoded;

[표 7-1] 요청 정보

### 7.3 요청 파라미터

파라미터	Byte	필수	설명
TID	30	O	거래번호, 인증결과로 받은 TxTid 파라미터 설정 예시) nicepay00m03011911140919215275
AuthToken	40	O	인증 TOKEN, 인증결과로 받은 AuthToken 파라미터 설정 예시) NICETOKNF435F661A2D54ED799BFB9F4B3F7E369
MID	10	O	상점 ID
Amt	12	O	결제금액
EdiDate	14	O	전문생성일시 (YYYYMMDDHHMISS)
NetCancel	1	O	망취소여부, 망취소시 "1" 설정
SignData	256	O	위변조 검증 Data 생성규칙 : hex(sha256(AuthToken + MID + Amt + EdiDate + MerchantKey)) <a href="#">*10. 암호화 항목 참조</a>
CharSet	10		응답파라미터 인코딩 방식
EdiType	10		응답전문 유형 (default(미설정): JSON / KV(설정): Key=Value형식 응답) 예시) EdiType=KV 요청한 경우. ResultCode=0000&ResultMsg=정상처리되었습니다.&TID=nicetest00m01011912311450329275.....

[표 7-2] 망취소 요청 파라미터

## 7.4 응답 파라미터

파라미터	Byte	설명
ResultCode	4	취소 결과 코드 (성공 : 2001 / 그외 실패) <u>*11. 결과코드 항목 참조</u>
ResultMsg	100	취소 결과 메시지 예시) 취소 성공
ErrorCD	5	에러코드 예) 2015
ErrorMsg	97	에러메시지 예) 해당거래 취소실패(기취소성공) : 전화 문의(1661-0808)
CancelAmt	12	금액 예시) 000000001004
MID	10	상점 ID
Moid	64	주문번호 예시) mnoid1234567890
Signature	500	위변조 검증 데이터 - 생성규칙 : hex(sha256(TID + MID + CancelAmt + MerchantKey)) - 데이터 유효성 검증을 위해, 가맹점 수준에서 비교하는 로직 구현 필수 <u>*10. 암호화 항목 참조</u>
PayMethod	10	결제수단 코드 (신용카드: CARD, 계좌이체: BANK, 가상계좌: VBANK, 핸드폰 :CELLPHONE)
TID	30	거래번호, 거래를 구분하는 transaction ID 예시) nictest00m01011912311450329275
CancelDate	8	취소일자, YYYYMMDD 예시) 20191231
CancelTime	6	취소시간, HHmmss 예시) 152210
CancelNum	8	취소번호 예시) 47145651
RemainAmt	12	취소 후 잔액 예시) 잔액이 1000원인 경우 -> 000000001000

[표 7-3] 망취소 응답 공통 파라미터

### 7.4.1 신용카드 추가 응답 파라미터

파라미터	Byte	설명
CouponAmt	12	쿠폰금액, 예시) 1000원인 경우 -> 000000001000
ClickpayCl	2	결제은행 명 (6: SKPAY, 8: SAMSUNGPAY, 15: PAYCO, 16: KAKAOPAY)
MultiCardAcquAmt	12	(페이코, 카카오 결제 시에만 응답) 간편결제 복합 신용카드 금액
MultiPointAmt	12	(페이코, 카카오 결제 시에만 응답) 간편결제 복합 포인트 금액
MultiCouponAmt	12	(페이코, 카카오 결제 시에만 응답) 간편결제 복합 쿠폰 금액

[표 7-4] 신용카드 결제 요청 건 망취소 응답 파라미터



## 8. 승인 취소 요청-응답 파라미터

### 8.1 Request URL

[https://webapi.nicepay.co.kr/webapi/cancel\\_process.jsp](https://webapi.nicepay.co.kr/webapi/cancel_process.jsp)

### 8.2 Request Info

Protocol	Method	Encoding	Content-Type
HTTP/1.1	POST	EUC-KR	application/x-www-form-urlencoded;

[표 8-1] 요청 정보

### 8.3 요청 파라미터

파라미터	Byte	필수	설명
TID	30	O	거래 ID 예시) nictest00m01011912311450329275
MID	10	O	상점 ID
Moid	64	O	상점에서 부여한 주문번호 (Unique하게 구성)
CancelAmt	12	O	취소 금액
CancelMsg	100	O	취소 사유
PartialCancelCode	1	O	부분취소 여부(전체취소 : 0 / 부분취소 : 1) <a href="#">*13.1. 부분취소 항목 참조</a>
EdiDate	14	O	전문생성일시 (YYYYMMDDHHMISS)
SignData	256	O	위변조 검증 Data, Hex(SHA256(MID + CancelAmt + EdiDate + 상점키)) <a href="#">*10. 암호화 항목 참조</a>
SupplyAmt	12		별도 공급가액 설정 시 사용
GoodsVat	12		별도 부가세 설정 시 사용
ServiceAmt	12		별도 봉사료 설정 시 사용
TaxFreeAmt	12		별도 면세금액 설정 시 사용
CharSet	10		응답파라미터 인코딩 방식 (utf-8 / euc-kr(default))
CartType	10		장바구니 결제 유형 (장바구니 결제: 1 / 그 외:0 )
EdiType	10		응답전문 유형 (default(미설정): JSON / KV(설정): Key=Value형식 응답) 예시) EdiType=KV 요청한 경우. ResultCode=0000&ResultMsg=정상처리되었습니다.&TID= nictest00m01011912311450329275.....
RefundAcctNo	16		환불계좌번호
RefundBankCd	3		환불계좌 은행코드
RefundAcctNm	10		환불계좌주명

[표 8-2] 취소 요청 파라미터

## 8.4 응답 파라미터

파라미터	Byte	설명
ResultCode	4	취소 결과 코드 (성공 : 2001 / 그외 실패) <b>*11. 결과코드 항목 참조</b>
ResultMsg	100	취소 결과 메시지 예시) 카드 결제 성공
ErrorCD	5	오류코드 예) 2015
ErrorMsg	97	오류메시지 예) 해당거래 취소실패(기취소성공) : 전화 문의(1661-0808)
CancelAmt	12	금액 예시) 000000001004
MID	10	상점 ID
Moid	64	주문번호 예시) mnoid1234567890
Signature	500	위변조 검증 데이터 - 생성규칙 : hex(sha256(TID + MID + Amt + MerchantKey)) - 데이터 유효성 검증을 위해, 가맹점 수준에서 비교하는 로직 구현 필수 <b>*10. 암호화 항목 참조</b>
PayMethod	10	결제수단 코드 (신용카드: CARD, 계좌이체: BANK, 가상계좌: VBANK, 핸드폰 :CELLPHONE)
TID	30	거래번호, 거래를 구분하는 transaction ID 예시) nictest00m01011912311450329275
OTID	30	<b>휴대폰 부분취소/부분환불일 경우 리턴됨.</b> <b>2회차 이상의 휴대폰 부분취소/부분환불 요청시, 응답전문내 OTID 로 취소요청 되어야함.</b>
CancelDate	8	취소일자, YYYYMMDD 예시) 20191231
CancelTime	6	취소시간, HHmmss 예시) 152210
CancelNum	8	취소번호 예시) 47145651
RemainAmt	12	취소 후 잔액 예시) 잔액이 1000원인 경우 -> 000000001000

[표 8-3] 취소 응답 공통 파라미터

### 8.4.1 신용카드 추가 응답 파라미터

파라미터	Byte	설명
CouponAmt	12	쿠폰금액, 예시) 1000원인 경우 -> 000000001000
ClickpayCl	2	결제은행 명 (6: SKPAY, 8: SAMSUNGPAY, 15: PAYCO, 16: KAKAOPAY)
MultiCardAcquAmt	12	(페이코, 카카오� 결제 시에만 응답) 간편결제 복합 신용카드 금액
MultiPointAmt	12	(페이코, 카카오� 결제 시에만 응답) 간편결제 복합 포인트 금액
MultiCouponAmt	12	(페이코, 카카오� 결제 시에만 응답) 간편결제 복합 쿠폰 금액

[표 8-4] 신용카드 결제 건 취소 응답 파라미터

## 9. TID 생성 규칙

### 9.1 주의 사항

모든 결제건은 규칙에 따라 유니크한 TID 를 생성합니다.

만약 결제 시 TID 를 전달받지 못하였다면, 인증이 완료되지 않았거나, 인증 완료 후 TxTid 값을 전달받지 못한 경우이므로, 결제창 호출 이후 승인 API URL 을 호출하기 직전까지의 구간 중 오류가 있는지 확인해보아야 합니다.

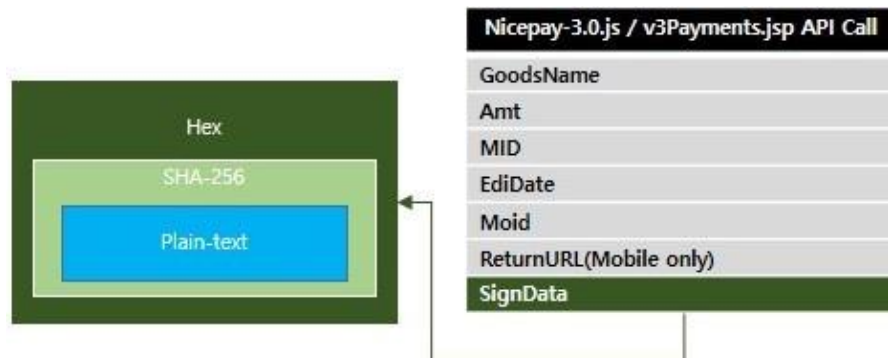
### 9.2 생성 규칙 예시

TID(30byte)		nicetest00m01011912191404041136		
MID(10byte)	지불수단(2byte)	매체구분(2byte)	시간정보(yyMMddHHmmss, 12byte)	랜덤(4byte)
nicetest00m	01 (신용카드) 02 (계좌이체) 03 (가상계좌) 05 (휴대폰) 21 (SSG머니)	01(일반) 07(SSG페이+머니 복합결제)	191219140404	1136

[표 9-1] TID 생성 규칙 예시

## 10. 암호화

### 10.1 인증 요청 시 암호화 구성



[그림 10-1] 인증 요청 시 암호화 이미지

인터페이스	파라미터	암호화 방식	Plain-text
nicepay-3.0.js v3Payment.jsp	SignData	Hex(Sha-256)	EdiDate + MID + Amt + 상점키(merchantKey)

[표 10-1] 인증 요청 시 SignData 암호화 처리방법

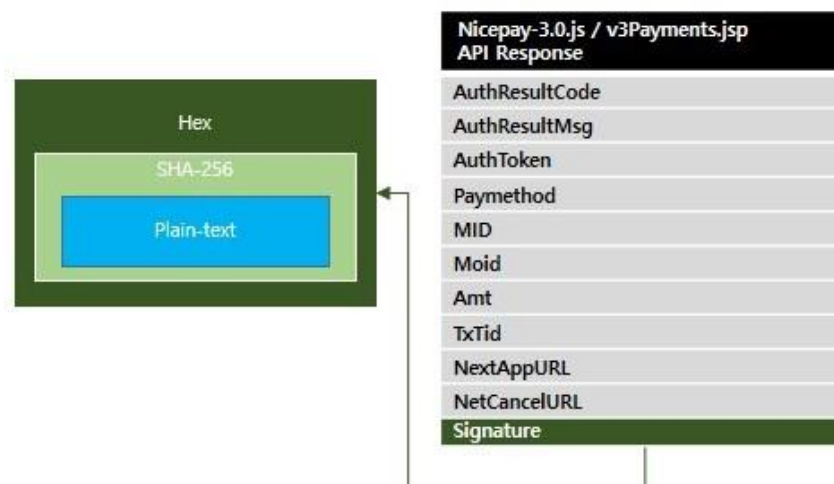
```
String ediDate = "20200622131021";
String mid = "nicepay00m";
String amt = "1004";
String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==";

//PlainText
20200622131021nicepay00m1004EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==

<암호화값>
475979a5628498711052598c6d4a73a17f0e3f0ef45960eed2e1b1776e3146bd
```

[예시 10-1] 인증 요청 시 SignData 암호화 구성 예시

## 10.2 인증 응답 시 암호화 구성



[그림 10-2] 인증 응답 시 암호화 이미지

인터페이스	파라미터	암호화 방식	Plain-text
nicepay-3.0.js v3Payment.jsp	Signature	Hex(Sha-256)	AuthToken + MID + Amt + 상점키(MerchantKey)

[표 10-2] 인증 응답 시 Signature 암호화 처리방법

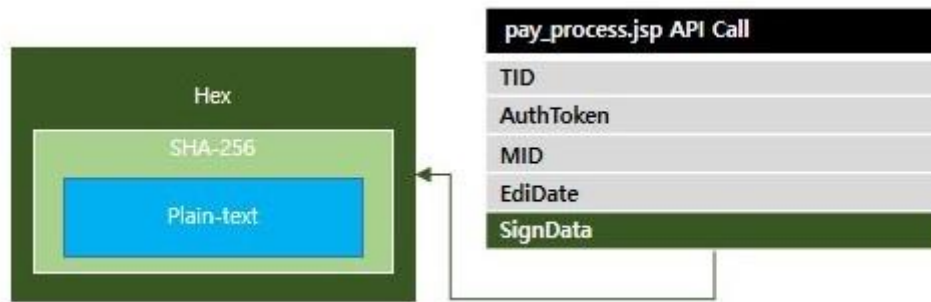
```
String authToken = "NICETOKNF435F661A2D54ED799BFB9F4B3F7E369";
String mid = "nicepay00m";
String amt = "1004";
String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFYoRNC6FkrzVjceg==";

//PlainText
NICETOKNF435F661A2D54ED799BFB9F4B3F7E369nicepay00m1004EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLua
UFYoRNC6FkrzVjceg==
```

<암호화값>  
cc94db193780ffb83d79845bb001b26da397cb5855dd285a3b85a4acc1fa55fe

[예시 10-2] 인증 응답 시 Signature 암호화 구성 예시

### 10.3 승인 요청 시 암호화 구성



[그림 10-3] 승인 요청 시 암호화 이미지

인터페이스	파라미터	암호화 방식	Plain-text
pay_process.jsp	SignData	Hex(Sha-256)	AuthToken + MID + Amt + EdiDate + 상점키(merchantKey)

[표 10-3] 승인 요청 시 SignData 암호화 처리방법

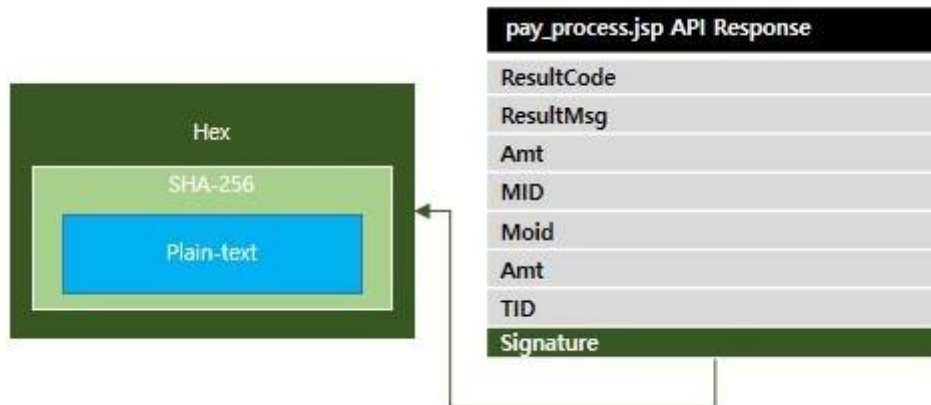
```
String authToken = "NICETOKNF435F661A2D54ED799BFB9F4B3F7E369";
String mid = "nicepay00m";
String amt = "1004";
String ediDate = "20200622131021";
String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFYoRNC6FkrzVjceg==";

//PlainText
NICETOKNF435F661A2D54ED799BFB9F4B3F7E369nicepay00m100420200622131021EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ
7Z1AoqQnXI3eLuaUFYoRNC6FkrzVjceg==
```

<암호화값>  
4916540b27cd60f3b6e37aa2e1257c7e4fb9faa0b4a84a1d85fac91461d00204

[예시 10-3] 승인 요청 시 SignData 암호화 구성 예시

## 10.4 승인 응답 시 암호화 구성



[그림 10-4] 승인 응답 시 암호화 이미지

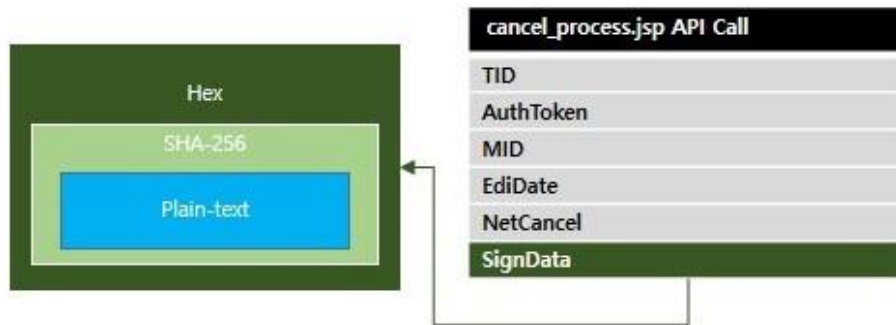
인터페이스	파라미터	암호화 방식	Plain-text
pay_process.jsp	Signature	Hex(Sha-256)	TID + MID + Amt + 상점키(merchantKey)

[표 10-4] 승인 응답 시 Signature 암호화 처리방법

<pre>String tid = "nicepay00m01012006221311045107"; String mid = "nicepay00m"; String amt = "1004"; String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AooQnXI3eLuaUFyoRNC6FkrzVjceg==";  //PlainText nicepay00m01012006221311045107nicepay00m1004EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AooQnXI3eLuaUFyoRNC6FkrzVjceg==</pre>
<p>&lt;암호화값&gt;</p> <p>9439b21e792ee41d1411d7b5e7e34a2062f0f42f19fa3f875f35c388954d4efd</p>

[예시 10-4] 승인 응답 시 Signature 암호화 구성 예시

## 10.5 망취소 요청 시 암호화 구성



[그림 10-5] 망취소 요청 시 암호화 이미지

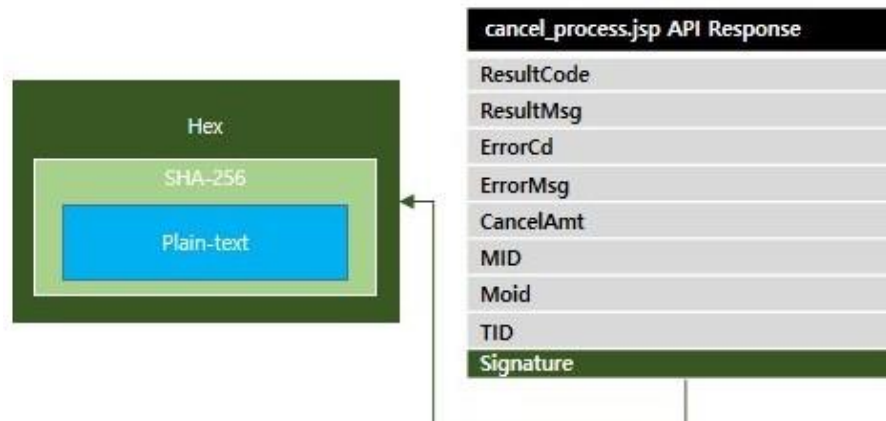
인터페이스	파라미터	암호화 방식	Plain-text
cancel_process.jsp	SignData	Hex(Sha-256)	AuthToken + MID + Amt + EdiDate + 상점키(merchantKey)

[표 10-5] 망취소 요청 시 SignData 암호화 처리방법

<pre>String authToken = "NICETOKNF435F661A2D54ED799BFB9F4B3F7E369"; String mid = "nicepay00m"; String amt = "1004"; String ediDate = "20200622131021"; String merchantKey = "EYzu8jGGMfqDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==";  //PlainText NICETOKNF435F661A2D54ED799BFB9F4B3F7E369nicepay00m100420200622131021EYzu8jGGMfqDEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==</pre>
<p>&lt;암호화값&gt;</p> <p>4916540b27cd60f3b6e37aa2e1257c7e4fb9faa0b4a84a1d85fac91461d00204</p>

[예시 10-5] 망취소 요청 시 SignData 암호화 구성 예시

## 10.6 망취소 응답 시 암호화 구성



[그림 10-6] 망취소 응답 시 암호화 이미지

인터페이스	파라미터	암호화 방식	Plain-text
cancel_process.jsp	Signature	Hex(Sha-256)	TID + MID + CancelAmt + 상점키(merchantKey)

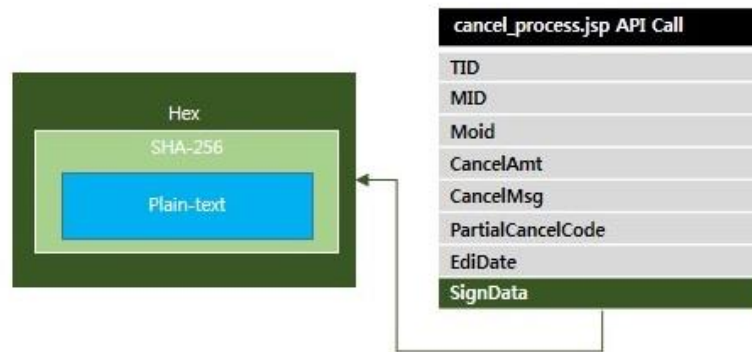
[표 10-6] 망취소 응답 시 SignData 암호화 처리방법

<pre>String tid = "nicepay00m01012006221311045107"; String mid = "nicepay00m"; String cancelAmt = "1004"; String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==";  //PlainText nicepay00m01012006221311045107nicepay00m1004EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AoqQnXI3eLuaUFyoRNC6FkrzVjceg==</pre>
<p>&lt;암호화값&gt;</p> <p>9439b21e792ee41d1411d7b5e7e34a2062f0f42f19fa3f875f35c388954d4efd</p>

[예시 10-6] 망취소 응답 시 Signature 암호화 구성 예시



## 10.7 승인 취소 요청 시 암호화 구성



[그림 10-7] 결제 취소 시 암호화 이미지

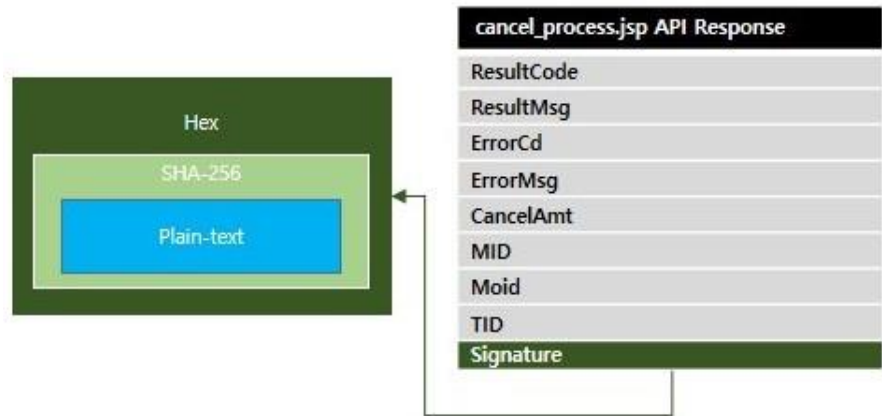
인터페이스	파라미터	암호화 방식	Plain-text
cancel_process.jsp	SignData	Hex(Sha-256)	MID + CancelAmt + EdiDate + 상점키(merchantKey)

[표 10-7] 결제 취소 시 SignData 암호화 처리방법

<pre>String mid = "nicepay00m"; String cancelAmt = "1004"; String ediDate = "20191219133357"; String merchantKey = "EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AooQnXI3eLuaUFyoRNC6FkrzVjceg==";  //PlainText nictest04m100420191219133357EYzu8jGGMfqaDEp76gSckuvnaHHu+bC4opsSN6lHv3b2lurNYkVXrZ7Z1AooQnXI3eLuaUFyoRNC6FkrzVjceg==</pre>
<p>&lt;암호화값&gt;</p> <p>3367c62414f8eb823819cffe4ed4e3900cb26fdf4787fa9512b3b9d8b7d8ed84</p>

[예시 10-7] 결제 취소 시 SignData 암호화 구성 예시

10.8 승인 취소 응답 시 암호화 구성



[그림 10-8] 승인 취소 응답 시 암호화 이미지

인터페이스	파라미터	암호화 방식	Plain-text
cancel_process.jsp	Signature	Hex(Sha-256)	TID + MID + CancelAmt + 상점키(merchantKey)

[표 10-8] 승인 취소 응답 시 Signature 암호화 처리방법

```
String tid = "nicepay00m01012006221311045107";
String mid = "nicepay00m";
String cancelAmt = "1004";
String merchantKey = "EYzu8jGGMfqADEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLualFyoRNC6FkrzVjceg==";

//PlainText
nicepay00m01012006221311045107nicepay00m1004EYzu8jGGMfqADEp76gSckuvnaHHu+bC4opsSN6IHv3b2lurNYkVXrZ7Z1AoqQnXI3eLualFyoRNC6FkrzVjceg==

<암호화값>
9439b21e792ee41d1411d7b5e7e34a2062f0f42f19fa3f875f35c388954d4efd
```

[예시 10-8] 승인 취소 응답 시 Signature 암호화 구성 예시

새마을금고연합회	045
신협중앙회	048
상호저축은행	050
기타 외국계은행	051
모건스탠리은행	052
HSBC은행	054
도이치은행	055
알비에스피엘씨은행	056
제이피모간체이스은행	057

동양종합금융증권	209
현대증권	218
미래에셋증권	230
대우증권	238
삼성증권	240
한국투자증권	243
NH투자증권	247
교보증권	261
하이투자증권	262


[표 12-2] 은행, 증권사 코드 (가상계좌, 계좌이체)

## 13.기타

### 13.1 인증 응답코드 확인 방법

#### End-point 정보

PC	<a href="https://web.nicepay.co.kr/v3/webstd/js/nicepay-3.0.js">https://web.nicepay.co.kr/v3/webstd/js/nicepay-3.0.js</a>
Mobile	<a href="https://web.nicepay.co.kr/v3/v3Payment.jsp">https://web.nicepay.co.kr/v3/v3Payment.jsp</a>

[표 13-1] End-point 정보

#### 13.1.1 PC 결제창 응답코드 확인 방법 \*11.2 인증 응답코드 참고

##### Client-side 에서 인증 응답코드 확인 방법

- (1) 결제 창 종료 시 nicepaySubmit() 콜백
- (2) nicepaySubmit() 콜백 시 document.payForm.AuthResultCode.value 로 인증 응답코드 확인

##### Server-side 에서 인증 응답코드 확인 방법

- (1) 결제 창 종료 시 nicepaySubmit() 콜백
- (2) document.payForm action target 으로 submit
- (3) action target server-side 에서 form data(name=value)통해 AuthResultCode 획득

#### 13.1.2 Mobile 결제창 응답코드 확인 방법 \*11.2 인증 응답코드 참고

- (1) 결제 창 호출 시 returnUrl 에 설정된 end-point 로 form (name=value) type 으로 data 전달
- (2) server-side 에서 form data 통해 AuthResultCode 획득