



---

# Virtual Currency

## API 가이드

Xsolla

2012 년 3 월

---

## 저작권 정보

© 2012 Xsolla. 모든 저작권은 저작권자에게 있습니다..

본 매뉴얼과 소프트웨어의 모든 저작권은 저작권자에 있다. 미합중국 저작권법과 국제 저작권법에 의거하여 본 매뉴얼과 소프트웨어를 어떠한 형태로든 특정부분 또는 전체를 복제하거나 재생산할 수 없으며 저장할 수 없다. 또한 복제 시스템, 전기적 또는 기계적 형태로, 서면동의 없이 복제할 수 없다. 단 정상적인 소프트웨어 사용을 위한 백업용도의 복제만 허가한다.

## 상표

Xsolla 상표와 상품은 미합중국과 그 외의 국가에서 상표권 등록을 마친 상표입니다. Xsolla 의 서면 동의 없이 본 상표를 게시하거나 사용할 수 없습니다.

### Xsolla USA Inc

15301 Ventura Blvd

Suite 100 Bldg E

Sherman Oaks, CA 91403

전화: 1 (818) 435-6613

미국 내 무료 안내 번호: 1 (877)

797-6552 [www.xsolla.com](http://www.xsolla.com)

# 내용

내용.....	3
API 특징.....	4
독자.....	4
Xsolla 트랜잭션 해부.....	4
요구사항.....	5
Request 파라미터.....	5
Response 포맷.....	5
사용자 ID 존재여부 Request .....	7
체크(Check) .....	7
예제 코드.....	10
잔고 증가 Request.....	11
지불(Pay)....	11
예제 코드.....	14
취소 Request.....	15
취소(Cancel).....	15
예제 코드.....	17
예제 코드.....	18
IP 주소 확인.....	18
MD5 시그니처(Signature) 체크 .....	18

# API 특징

Virtual Currency API 는 개발자가 virtual currency 를 설정한 통화 환전율에 따라서 사용제한 없이 판매할 수 있도록 해준다. API 를 사용하여 결제금액을 virtual currency 로 환전한다. Advanced Virtual Currency API 를 사용하여 실제 화폐를 virtual currency 로 개발자가 규정한 공식에 따라서 환전할 수 있습니다.

Virtual Currency API 는 특히 사전에 가격이 정해져 있는 인게임 virtual currency 로 변환하도록 프로젝트를 위해 쉽고 용이하게 구성할 수 있다. 사용자는 게임 내의 계정에 충전 명령을 사용하여 사전에 설계된 virtual currency 금액을 구입할 수 있다. 플레이어는 e-wallet, 결제용 키오스크, 온라인 banking 등의 방식을 사용하여 결제할 수 있다.

Virtual Currency API 는 다음과 같은 장점을 제공한다:

전체 결제 기능 API 는 결제액을 virtual currency 로 환전하여 준다. 게임 개발사의 의도에 따라 플레이어의 최소/최대 결제 한도를 조정할 수 있다.

자동 결제 플레이어가 Xsollar 일련번호를 부여 받게 되면 시간 장소에 상관없이 Xsolla 웹사이트 또는 게임 내 폼을 작성하여 게임 계정을 충전할 수 있다. 플레이어는 결제 시스템 페이지에서 Xsolla 버튼을 누르고 Xsolla 일련번호, email, ID 와 같은 일부 정보를 입력해야 한다.

결제 시스템 연동, Virtual Currency API 를 사용하면 다른 결제 시스템 웹사이트에서 "기타 서비스 사이트" 섹션에서 Xsolla 의 게임 카탈로그를 확인할 수 있습니다. 개발사의 프로젝트 역시 카탈로그에 추가 될 수 있다. 따라서 사용자들은 Xsolla 웹사이트 또는 결제 시스템에서 직접 결제를 할 수 있는 것이다.

## 독자

본 문서는 Xsollar Currency API 를 이용하여 인게임 결제 시스템을 통합하고자 하는 개발자를 위한 학습서이다. 본 학습서는 학습자가 html 기본 명령어, HTTP Request 명령과 소프트웨어 아키텍처 [REST](#) 스타일에 대한 지식을 숙지하고 있다고 가정하고 있다.

## Xsolla 트랜잭션 해부

Xsolla 시스템은 2 개의 필수 단계를 수행한다: 사용자 상태 체크 과정과 트랜잭션 처리과정, 그리고 부가기능으로 트랜잭션을 취소 기능이 있다. 명령어(command) 파라미터 Request 유형에는 체크, 결제, 취소가 있습니다. 사용자의 상태 정보를 체크하는 동안 정확한 ID 정보가 데이터베이스에 존재하는 가를 확인해야 합니다. 트랜잭션 처리 단계 동안 결제 ID 와 고객의 잔고 변경에 대한 내부적 확인 작업을 수행해야 합니다.

데이터 베이스 내에 동일 ID 를 사용한 결제가 2 개 이어서는 안 된다. 만약 시스템에 트랜잭션 요청 아이디가 반복되어 들어 온다면 시스템은 이전에 요청으로

# API 특징

수행된 결과 값을 출력해야 한다.

## 요구사항

응용프로그램:

다음 IP 로부터 HTTP, HTTPS 의 request 명령을 허가해야 한다:  
94.103.26.178, 94.103.26.181

CP1251 XML 인코딩으로 된 GET 명령의 파라미터를 처리해야 한다

CP1251 XML 포맷으로 Response 를 만들어야 한다(만약 Response 가 국가 코드를 포함한 경우)

Request 와 Response 모드로 데이터를 교환해야 한다. Response 는 최대 60 초 이내에 실행되어야 한다. 60 초를 초과하면 타임아웃에 의해 접속이 종료된다.

## Request 포맷

Xsolla 시스템이 HTTP 또는 HTTPS URL 을 통해 Request 를 보낸다. 예제:

```
https://test.project.com
project=133&command=pay&id=14332453&v1=BiIMURka&v2=&v3=&sum=902.481&date=2012-03-26+08%3A14%3A43&md5=92f93db6770adccf54d8063612858693
```

Xsollar 시스템의 각 결제는 사용자 id 파라미터를 키로하여 개발자에게 고유한 식별코드를 제공한다. 이 식별코드로 트랜잭션의 문제를 해결하거나 결과의 일치성을 확보하는 데 사용한다.

결제 Request 에서 Xsolla 시스템은 결제 일자 파라미터를 다음 포맷 형태로 사용한다: YYYYMMDDHHMMSS. 이 날짜 정보로 트랜잭션의 문제를 해결하거나 결과의 일치성을 확보할 수 있다.

금액은 소수점 두 자리까지 사용한다. "." (소수점)은 구분자이다. 고객이 구매한 virtual currency 금액을 나타낸다.

## Response 포맷

CP1251 XML 포맷으로 Response 를 작성하여 시스템에 리턴한다( Response 가 국가 코드를 포함한 경우), 예제:

```
<?xml version="1.0" encoding="windows-1251"?>
<response>
  <id>1234567</id>
  <id_shop>9876543</id_shop>
  <result>0</result>
  <comment>Success</comment>
</response>
```

## 결과 값

코드	설명	치명적인가요?
0	예	아니오

1	일시적인 오류, 차후에 다시 시도하세요	아니오
---	-----------------------	-----

코드	설	치명적인가요?
2	유효하지 않은 ID	예
3	유효하지 않은 MD5 시그니처	예
4	유효하지 않은 request 포맷 (금액 오류, 파라미터 부족)	예
5	기타 오류(이하 오류 정보 작성)	예
7	기술 문제로 인한 고객의 결제 처리 미완료/거부	예



# 사용자 ID 존재여부 Request

## 확인

Xsolla 서버는 개발사의 프로젝트를 Xsolla Merchant 계정에 등록된 결제 스크립트 URL 을 사용하여 고객의 ID 존재여부 확인을 시도한다. 트랜잭션 내역은 GET 방식으로 사용하여 CP1251 XML 형태로 전달되며 다음과 같은 파라미터를 사용한다.

### Request 파라미터

필드 명	타입	설명	필수 항목입니까?	예제
명령어	문자열	고객의 ID 존재여부를 확인하려면 반드시 체크해야 합니다. 고객의 ID 가 확인되었음을 뜻합니다	예	
v1	문자열	고객 ID 는 플랫폼상에 고유한 정보여야 합니다(예, 닉네임, email). 문자, 숫자, 특수문자를 사용할 수 있습니다. 문자열의 길이는 최대 255	예	데모
v2	문자열	추가 ID 파라미터(예제 게임 서버용) 문자열의 길이는 최대 200	아니오	0
v3	String	추가 ID 파라미터(예제 게임 서버용) 문자열의 길이는 최대 100	아니오	0
md5	문자열	불법 사용 방지 문자열	예	bdfa807b47c58c4 3e3d6dcaaa3a1301 d

## MD5 시그니처(Signature) 생성

시그니처가 있어야 결제 처리 보안을 통과할 수 있습니다. 시그니처는 MD5 해쉬 알고리즘으로 만들어지며 다음 수록된 파라미터와 합쳐진 문자를 암호화한다.

```
md5(<command><v1><secret_key>)
```

*secret\_key* 는 Merchant 계정에서 선택한 비밀 워드입니다.

## 예제

시그니처 계산 문자열에는 다음의 파라미터가 포함된다:

```
command =  
check v1 =  
demo  
secret_key = password
```

MD5 암호화를 사용하기 전에 값:

```
checkdemopassword
```

MD5 암호화가 적용된 문자열은 다음과 같다:

```
bdfa807b47c58c43e3d6dcaaa3a1301d
```

본 Request 에 Response 로서 결제 프로그램은 다음과 같은 XML Response 를 생성해야 합니다.

개발 시스템에 고객 ID 가 존재하면 다음과 같은 잔고 입금이 처리된다:

```
<response>  
  <result>0</result>  
</response>
```

개발 시스템에 고객 ID 가 존재하지 않으면 잔고에 입금이 처리되지 않는다:

```
<response>  
  <result>7</result>  
  <comment>고객의 계정을 사용할 수 없거나 존재하지 않습니다</comment>  
</response>
```

## 결과 필드 값

결과	설명
0	Request 에 표시된 고객 ID 의 잔고를 증액할 수 있습니다. 성공적인 고객 정보를 확인 후에 잔고 증액 Request 를 전송합니다.
7	이 고객 ID 의 트랜잭션은 처리할 수 없습니다.

## 추가 사기 방지 보호 정책

추가 파라미터를 전달하여 고객에 대한 신상정보를 제공할 수 있습니다. 이 추가 대응 기법으로 고객의 신뢰도 확보하고 보안 필터를 제작할 수 있습니다. XML Response 의 포맷은 다음과 같습니다:

```
<response>

  <result>...</result>

  <comment>...</comment>

<specification>

  <s1>...</s1>

  <s2>...</s2>

  ...

</specification>

</response>
```

모든 파라미터는 프로토콜 문서에 규정된 방식으로 사용되어야 한다. 누계액은 실수형태로 전송되어야 하며 "."을 구분자로 사용해야 합니다. ('.' 뒤에 0,1,2 문자) 날짜는 문자열 형태로 전송되어야 합니다.

YYYYMMDDHHMMSS. 예제에서 s1 는 승인에 대한 파라미터입니다(예를 들어 승인이 100%인 고객은 신뢰성을 의미한다). s2 는 등록 일자이며, s3 은 그동안 사용한 virtual currency 의 누계액이며, s4 는 생년월일이다. 계정 매니저나 인티그레이션 매니저에게 이 파라미터들을 설정하도록 요청하고 각각의 파라미터의 뜻을 추가로 기록한다.

## 예제 코드

---

본 예제 코드는 고객의 존재여부를 체크하는 명령을 수행하는 예제입니다. [첨부를 확인하세요](#)  
MD5 시그니처와 IP 주소를 확인하는 예제이다.

```
<?php

$v1 = $_GET['v1'];

// 닉네임 체크
// check_nickname - 데이터베이스에 닉네임이 존재하는 지를 체크하는 함수
$check = check_nickname($v1);

// 만약 닉네임이 존재하면 if
($check) {
    $code = '0';
    $comment = 'success';
} // 아니면
else {
    $code = '7';
    $comment = 'fail';
}

echo '<?xml version="1.0" encoding="windows-1251"?>
    <response>
        <code>' . $code . '</code>
        <comment>' . $comment . '</comment>
    </response>';

// 체크 함수
function check_nickname()
{
    // 원하는 코드를 입력하세요

    return true;
}
```

# 잔고 증가 Request

## 지불

트랜잭션이 성공하면 Xsolla 서버는 개발자의 결제 스크립트 URL 을 실행하여 결과를 알린다. Xsolla Merchant 계정에 개발자 프로젝트 상세정보에 기록되어 있다. 트랜잭션 내역은 GET 방식으로 사용하여 CP1251 XML 형태로 전달되며 다음과 같은 파라미터를 사용한다.

## Request 파라미터

필드 명	타입	설명	필수 항목입니까?	예제
명령어	문자열	잔고 증가 Request 를 보내려면 금액이 지불완료 되어야 합니다	예	
id	문자열	Xsolla 시스템 Order ID	예	7555545
v1	문자열	고객 ID 는 프로젝트에서 고유해야 합니다. 문자열의 길이는 최대 255	예	데모
v2	문자열	추가 ID 파라미터 문자열의 길이는 최대 200	아니오	0
v3	문자열	추가 ID 파라미터 문자열의 길이는 최대 100	아니오	0
합계	실수	virtual currency 결제 금액 구분자 '.' ('.' 뒤에 0,1,2 문자)	예	100
날짜	문자열	사용 날짜 포맷 YYYYDDMMHHMMSS	예	20060425180622
보너스	문자열	프로모션 캠페인이란 *계정 매니저가 제공하는 secret word 를 말합니다	아니오	bonussum
md5	문자열	불법 사용 방지 문자열	예	9286b1ff8c5226b 666a20ddb4cc03 c2b

\*Xsolla 계정 매니저에게 프로모션 캠페인에 대한 정보를 얻으실 수 있다

.

## MD5 시그니처 생성

시그니처가 있어야 결제 처리 보안을 통과할 수 있다. 시그니처는 MD5 해쉬 알고리즘으로 만들어지며 다음 수록된 파라미터와 합쳐진 문자를 암호화한다.

```
md5(<command><v1><id><secret_key>)
```

*secret\_key 는 Merchant 계정에서 선택한 비밀 워드이다.*

## 예제

시그니처 계산 문자열에는 다음의 파라미터가 포함됩니다:

```
secret key = "password"

command = pay

id = 7555545 v1 = demo

date=20060425180622
```

MD5 암호화를 사용하기 전에 값:

```
paydemo7555545password
```

MD5 암호화가 적용된 문자열은 다음과 같다:

```
9286b1ff8c5226b666a20ddb4cc03c2b
```

본 Request 에 Response 로서 결제 프로그램은 다음과 같은 XML Response 를 생성해야 한다:

잔고 증액 트랜잭션을 성공 처리:

```
<response>

  <id>7555545</id>

  <id_shop>1234</id_shop>

  <sum>10</sum>

  <result>0</result>

</response>
```

잔고 증액 트랜잭션을 실패 처리:

```
<response>

  <id>7555545</id>

  <id_shop>1234</id_shop>

  <sum>10</sum>

  <result>1</result>

  <comment>Temporary database error</comment>

</response>
```

## 결과 필드 값

코드	설명	치명적인가요?
0	예	아니오
1	일시적인 오류, 차후에 다시 시도하세요	아니오
2	유효하지 않은 ID	예



코드	설명	치명적인가요?
3	유효하지 않은 MD5 시그니처	예
4	유효하지 않은 request 포맷 (금액 오류, 파라미터 부족)	예
5	기타 오류(이하 오류 정보 작성)	예
7	기술 문제로 인한 고객의 결제 처리 미완료/거부	예

잔고 증액 Request 가 중복된 경우 잔고를 증액하지 않고 정확하게 금액을 처리해야 한다.

## 예제 코드

다음은 코드는 잔고 증액하는 예제입니다. [부록 A](#)의 MD5와 IP 주소를 체크하는 예제를 참조하세요.

```
<?php

$v1 = $_GET['v1'];
$sum = $_GET['sum'];
$id = $_GET['id'];

// 결제를 하고 ID를 리턴합니다
$paymentId = pay($v1, $sum, $id);

// 답
// 결제 성공 시 if ($paymentId)
    echo '<?xml version="1.0" encoding="windows-1251"?>

    <response>
    <id>' . $id . '</id>
    <id_shop>' . $paymentId . '</id_shop>
    <sum>' . $sum . '</sum>
    <result>0</result>
    <comment>Success</comment>
    </response>';
else
    echo '<?xml version="1.0" encoding="windows-1251"?>

    <response>
    <id>' . $id . '</id>
    <id_shop>' . $paymentId . '</id_shop>
    <sum>' . $sum . '</sum>
    <result>1</result>
    <comment>Temporarily database error</comment>
    </response>';

// 결제 함수 function pay($v1, $sum,
$id)
{
    // 원하는 코드를 입력하세요

    return $paymentId;
}
```

# 취소 Request

## 취소 기능(Cancel)

이 함수는 옵션입니다. Xsolla 계정 매니저가 개발자에게 결제 취소 기능을 만들 필요가 있는지 상담을 제공한다. Xsolla 서버는 결제 취소 시 Xsolla Merchant 계정에 입력된 결제 스크립트 URL 로 결제 정보를 전송한다. 결제 내역은 GET 방식으로 사용하여 CP1251 XML 형태로 전달되며 다음과 같은 파라미터를 사용한다.

## Request 파라미터

필드 명	타입	설명	필수 항목입니까?	예제
명령어	문자열	결제 취소 Request 를 보내려면 금액이 취소되어야 합니다	예	
id	문자열	Xsolla 시스템 Order ID 는 결제 Request 의 ID 파라미터로 전송됩니다	예	7555545
md5	문자열	불법 사용 방지 문자열	예	e9b9777e9c0a4595ad009eca90ba9977

## MD5 시그니처 생성

시그니처가 있어야 결제 처리 보안을 통과할 수 있다. 시그니처는 MD5 해쉬 알고리즘으로 만들어지며 다음 수록된 파라미터와 합쳐진 문자를 암호화한다.

```
md5(<command><id><secret_key>)
```

*secret\_key 는 Merchant 계정에서 선택한 비밀 워드이다.*

## 예제

시그니처 계산 문자열에는 다음의 파라미터가 포함된다:

```
secret key = "password" id
= "7555545"
command = "cancel"
```

MD5 암호화를 사용하기 전에 값:

```
cancel7555545password
```

MD5 해쉬 문자열은 다음과 같다.

```
e9b9777e9c0a4595ad009eca90ba9977
```

본 Request 에 Response 로서 결제 프로그램은 다음과 같은 XML Response 를 생성해야 한다:

결제 취소 트랜잭션을 성공 처리:

```
<response>  
  <result>0</result>  
</response>
```

결제 취소 트랜잭션을 실패 처리:

```
<response>  
  <result>2</result>  
  <comment> 결제 ID 가 존재하지 않습니다</comment>  
</response>
```

## 결과 필드 값

결과	설명
0	결제 취소 Request 성공적으로 실행됐습니다
2	요청하신 결제 취소 트랜잭션을 찾을 수 없습니다
7	요청하신 결제 취소 Request 를 완료할 수 없습니다

## 예제 코드

---

다음 코드는 잔고 증액 취소 예제입니다. [부록 A](#)의 MD5와 IP 주소를 체크하는 예제를 참조하세요.

```
<?php

$id = $_GET['id'];

// 결제 취소하기
$cancelResult = cancel($id);
// 결제 취소 성공 시 if
($cancelResult)
    echo '<?xml version="1.0" encoding="windows-1251"?>
        <response>
            <result>0</result>
        </response>';
else
    echo '<?xml version="1.0" encoding="windows-1251"?>
        <response>
            <result>2</result>
            <comment>Payment with given ID does not exist</comment>
        </response>';

// 취소 함수 function cancel($id)
{
    // 원하는 코드를 입력하세요

    return true;
}
```

# 예제 코드

부록에서 IP 체크와 MD5 시그니처 체크 예제를 참조하세요.

## IP 주소 체크

```
<?php
/**
 * IP 체크 함수
 */

$isAllowed = checkIP($_SERVER['REMOTE_ADDR']);

/**
 * 시그니처 체크
 *
 * @return Boolean
 */
function checkIP($ip)
{
    return in_array($ip, array("94.103.26.178", "94.103.26.181"));
}
```

## MD5 시그니처 체크

```
<?php
/**
 * 시그니처 체크 함수
 */

$command = $_GET["command"];
$v1 = $_GET["v1"];
$id = $_GET["id"];
$md5 = $_GET["md5"];
$key = 'secretKey';

$check = checkSignature($command, $v1, $id, $key, $md5);

/**
 * 시그니처 체크
 *
 * @return Boolean
 */
function checkSignature($command, $v1, $id, $key, $md5)
{
    return md5($command . $v1 . $id . $key) === $md5;
}
```