



**Integrovaný agent obchodníka elektronického obchodu
(IMA)**
Verzia 2.11.0
Príručka správcu

SIA Virtual POS

Informace v tomto materiálu mohou být změněny bez oznámení a **jsou považované za důvěrné**. Pro účely technické podpory při zavádění a správě systému ECOMM v rámci akceptace e-commerce transakcí kontaktujte prosím společnost SIA Central Europe, a.s.:

SIA Central Europe, a.s.
 Digital Park II, Einsteinova 25
 851 01 Bratislava
 Email: siask_ecommerce@sia.eu

Přehled

Version	Date	Author	Changes
1.0	2009-01-12	Ilze Saucuka, Atis Rusins	Initial version
1.1	2009-01-12	Atis Rusins	Added information about <i>frameset</i> usage restriction
1.2	2009-03-03	Atis Rusins	Minor changes – installation paths
1.4	2009-08-13	Atis Rusins	Added information about DMS transaction reversal and business day closure
1.5	2010-06-14	Atis Rusins	Added information about transaction status AUTOREVERSED
1.6	2010-06-29	Atis Rusins	Added SMS transaction reversal description
1.7.	2011-10-28	Jevgenijs Milovanovs, Ilze Saucuka	Added information about transaction result requesting and partial reversal of transaction amount. Minor format/style changes of the document
2.09.8	2012-10-02	Janis Kulins	DMS1 transaction reversal time
2.09.9	2013-05-08	Elina Ringa	“Back to order” functionality realization description
2.10.0	2014-06-09	Elina Ringa	Changes related to the new version of Ecomm IMA. Transaction result request description update. ECOMM IMA purpose and installation description update. Minor text and format changes. Description of additional response fields to business day closure request added.
2.11.0	2020-04-20	Petra Polomská	Changes related to the new version of Ecomm IMA – URL, branding and formal changes.

Obsah

1. ÚVOD	4
1.1. O TÉTO PŘÍRUČCE	4
1.2. PŘÍJEMCI	4
1.3. POUŽITÍ.....	4
2. PŘÍPRAVA NA PROVOZ IMA	5
2.1. SYSTÉMOVÉ POŽADAVKY	5
2.1.1. <i>Požadovaný software třetí strany</i>	5
2.2. INSTALACE SYSTÉMU	5
2.2.1. <i>Instalace produktu IMA</i>	5
2.2.2. <i>Technická konfigurace produktu IMA</i>	5
3. INTEGRACE IMA DO ŘEŠENÍ OBCHODNÍKA	7
3.1.1. <i>SMS</i>	7
3.1.2. <i>DMS</i>	7
3.2. VŠEOBECNÉ PROVOZNÍ SCHÉMA	7
3.3. INTEGRACE	8
3.3.1. <i>Provedení transakce SMS</i>	9
3.3.2. <i>Registrace autorizace DMS</i>	9
3.3.3. <i>Provedení transakce DMS</i>	10
3.3.4. <i>Výsledek transakce</i>	11
3.3.5. <i>Zrušení transakce</i>	13
3.3.6. <i>Uzávěrka obchodního dne</i>	13
3.3.7. <i>Další možnosti nastavení</i>	15
3.3.7.1 <i>Použití dalších atributů v dotaze</i>	15
3.3.7.2 <i>Další 3D Secure atributy</i>	15
3.3.8. <i>Provedení předadresování klienta</i>	16
3.3.9. <i>Vytvoření cardinfo.html</i>	16
3.3.10. <i>Funkce rozlišování jazyků</i>	17
3.4. ČASOVÁ OSA VÝSLEDKU TRANSAKCE.....	19
3.5. POŽADAVKY NA BEZPEČNOST	20

1. Úvod

1.1. O této příručce

Tento dokument byl vytvořen s cílem popsat kroky, které jsou zapotřebí při instalaci a ovládní integrovaného obchodního agenta (IMA). Předpokládá se, že systém ITA má zabezpečit spojení mezi obchodním systémem a systémem ECOMM pro provádění transakcí 3D-Secure a MasterCard SPA/UCAF v prostředí www.

1.2. Příjemci

Tento dokument je určený pro správce systému.

1.3. Použití

Tento dokument odpovídá softwarovým modulům:

- Systém elektronického obchodu ECOMM, verze 1.2.243
- ECOMM Integrated Merchant Agent (IMA – integrovaný agent obchodníka) verze 2.11.0

2. Příprava na provoz IMA

2.1. Systémové požadavky

2.1.1. Požadovaný software třetí strany

Komponenta	Verze	Poznámky
Sun Java Runtime Environment (JRE)	1.5 nebo vyšší	Instalace musí být provedena podle pokynů uvedených v http://docs.oracle.com/cd/E51849_01/gg-winx/GDRAD/java.htm
PHP	PHP 4.0.2. nebo vyšší	Požadovaná CURL knihovna
TLS	TLS 1.1, 1.2 nebo vyšší	Váš server musí podporovat a používat následující TLS protokol verzia: - nejmeně TLS 1.2 pro připojení na Test server - nejmeně TLS 1.1 pro připojení na produkční server

2.2. Instalace systému

2.2.1. Instalace produktu IMA

Instalace IMA je omezena na rozbalení archivu „EcommMerchant-2.11.0_SIA.zip“ do kořenového adresáře. Tím se vytvoří podadresář „EcommMerchant-2.11.0“, který obsahuje adresář „php“ – příklad PHP kódu, který může být užitečný pro integraci IMA, adresář „java“ – JAR soubory potřebné pro fungování IMA (IMA je v archivu ecomm_merchant.jar), adresář „c#“ – soubory pro použití na .net serveru a adresář „doc“, který obsahuje tyto soubory.

2.2.2. Technická konfigurace produktu IMA

1. Bezpečnostní certifikáty:
 - a. Obchodník obdrží **test certifikáty** v instalačním balíku služby z Banky, které je zapotřebí implementovat do systému obchodníka a které jsou určeny **pro přístup do testovacího prostředí služby**.
 - b. Po překlopení do produkčního provozu Obchodník obdrží z Banky produkční certifikáty, které je třeba implementovat do systému Obchodníka a které jsou určeny **pro přístup do produkčního prostředí služby**. Produkční certifikáty použité pro komunikaci mezi vaší aplikací a platobní bránou mají obmezenou platnost a před ukončením jejich platnosti Obchodník obdrží z Banky **nové produkční certifikáty**.
2. Keystore soubor se používá k vytvoření SSL spojení s ECOMM serverem a také k identifikaci Obchodníka vůči ECOMM systému.
3. Možnosti – JAVA a PHP řešení
 - a. V případě použití JAVA, modifikujte soubor „**merchant.properties**“ (v adresáři „java“):
Bank.server.url - adresa pro přístup k serveru ECOMM:
<https://vpos.te.sia.eu:8443/ecommerce/MerchantHandler> – **testovací provoz**
<https://vpos.sia.eu:8443/ecommerce/MerchantHandler> – **produkční provoz**

https.proxy.host – adresa proxy servera HTTPS (volitelná)

https.proxy.port – port proxy servera HTTPS (volitelný)

https.handler – knihovna podporující protokol HTTPS (volitelná). Když běží IMA na platformě Java 1.3, s použitím SUN JSSE, potom by se hodnota parametru měla nastavit na „com.sun.net.ssl.internal.www.protocol“.

https.cipher – algoritmus šifrování spojení HTTPS (volitelný). Obvykle se používá „SSL_RSA_WITH_RC4_128_MD5“.

keystore.file – soubor keystore, který je získáný z banky.

keystore.type – typ souboru certifikátu. Navrhovaná hodnota je „JKS“.

keystore.password – heslo použitý při vytvoření certifikátu

connection.timeout – čas v sekundách na spojení s ecomm serverem. Nezapomeňte, že tento parametr lze použít pouze ke zkrácení předvoleného času, který je dán konkrétní platformou

- b. V případě použití PHP, musí být modifikován soubor „config.php“ (nachází se adresáři php / v podadresáři):

\$ecomm_server_url – URL pro obchodníka pro komunikaci s ECOMM serverem

\$ecomm_client_url – URL, na kterou přeměrovaný klient a kde může zadat údaje o kartě

\$cert_url – cesta k certifikátu na routery obchodníka

\$cert_pass – heslo k certifikátu

\$currency – transakční měna ve formátu ISO 4217

\$db_user – MySQL databázové uživatelské jméno

\$db_pass – MySQL databázové heslo

\$db_host – MySQL databázový host

\$db_database – MySQL název databáze

\$db_table_transaction – MySQL název tabulky pro transakce

\$db_table_batch-MySQL název tabulky pro denní uzávěrky

\$db_table_err – MySQL název tabulky pro chyby

- c. Šablona formuláře HTML (cardinfo.html) a požadované pomocné soubory pro zadávání údajů karty klienta se vytvoří automaticky. Lze měnit html kód cardinfo,htm – nahrazení kódem z balíku i doplnění vlastního banneru obchodníka. Podrobný popis je uveden v kapitole **Vytvoření cardinfo.html**, resp. v dokumentu **README FIRST.docx v balíku**.

4. Banka musí dostat informace o adrese URL, která se bude používat k přeadresování klienta zpět k obchodníkovi. Nezapomeňte, že přeadresování se provádí podle metody HTTP POST, což znamená, že adresa URL nemůže obsahovat žádné parametry.

returnOkUrl – klient bude přeadresovaný na tuto adresu po ověření 3D Secure a ukončení transakce (bez ohledu na její výsledek)

returnFailUrl – klient bude přeadresovaný na tuto adresu v případě technické chyby v činnosti systému ECOMM.

3. Integrace IMA do řešení obchodníka

Obchodník má na výběr z dvou způsobů zúčtování plateb. Má možnost výběru mezi:

- **SMS Platba** – znamená automatické online zpracování transakce. t.j. po zaplacení zákazníkem je platba automaticky zařazená na zúčtování.
- **DMS Platba** – tento typ platby vyžaduje dva kroky a je plně v kompetenci prevádzkovatele e-shopu tedy obchodníka. První krok je registrace platby DMS (t.j. vykoná se rezervace prostředků na účte zákazníka), druhý krok obchodník je povinen odsúhlasit transakciu až po odeslání tovaru zákazníkovi (t.j. potvrdí zúčtování DMS)- blokáce transakce je v prospěch obchodníka až po zúčtování (zúčtování by mělo být realizováno max do 30 dnů).

3.1.1. SMS

SMS transakce jsou prováděny příkazem `-v`. V případě PHP řešení to je funkce `startSMSTrans()`. Když je provedena SMS transakce, prostředky jsou okamžitě odepsány z účtu držitele karty

3.1.2. DMS

DMS transakce jsou autorizované příkazem `-a`. V případě PHP řešení to je funkce `startDMSAuth()`. Když je proveden tento autorizační požadavek, finanční prostředky na účte držitele karty jsou rezervovány (zablokovány).

DMS transakce jsou odsouhlaseny příkazem `-t`. V případě PHP řešení to je funkce `makeDMSTrans()`. Když je tento příkaz proveden, rezervovány (zablokovány) prostředky jsou odepsány z účtu držitele karty.

Merchant musí provést DMS transakci:

- Pokud bylo zboží odesláno klientovi, ale ne později než 30 dní od příkazu `-a/startDMSAuth()` provedení
- Pokud zboží nebylo odesláno klientovi (v případě, kdy jsou služby klientovi doručeny elektronicky), ale ne později než 30 dní od příkazu `-a/startDMSAuth()` provedení

3.2. Všeobecné provozní schéma

1. Klient si vybral produkt a je připravený zaplatit za nákup. Po stisknutí tlačítka/spojky „pokladna /checkout“, je řízení převedeno na řešení obchodníka.
2. Obchodník zaregistruje transakci v systému ECOMM (s uvedením sumy, měny, IP adresy klienta a stručného popisu transakce (volitelné)), a jako odpověď dostane identifikátor transakce.
3. Klient (uvádějící identifikátor transakce) je preadresovaný na server plateb ECOMM pro zadání údajů karty. Zadání je provedeno s použitím šablony formuláře (`cardinfo.html`).
4. Po zadání údajů karty jsou tyto ověřeny a je vygenerován výsledek transakce.
5. Klient je preadresován zpět k obchodníkovi (s uvedením identifikátoru transakce).
6. Obchodník dostane informace z ECOMM o výsledku transakce (uskutečněna nebo neuskutečněna) s použitím přijatého identifikátoru transakce (uskutečněna nebo ne).
7. V případě transakce DMS se musí provést dodatečná transakce, aby se od klienta získali peníze (příkaz `-t`, v PHP implementaci je to funkce `makeDMSTrans()`) – toto se obvykle provádí, když je zboží doručeno klientovi.

8. V případě potřeby může obchodník požadovat od serveru plateb ECOMM stornování transakce.
9. Obchodník by měl posílat pravidelné požadavky na uzavření obchodního dne do ECOMM serveru.

3.3. Integrace

Volání platebního serveru ECOMM využívající IMA (nacházející se v archivu ecomm_obchodník.jar) lze provést několika způsoby.

- 1) Zavolání JAVA archivu ecomm_merchant.jar z příkazového řádku. Příklady jsou uvedeny v adresáři „java“..
- 2) Volání služebních metod třídy lv.konts.ecomm.merchant.Merchant. Název souboru konfigurace musí být přiřazený třídě Merchant, když je tato třída vytvářena. IT umožní inicializovat IMA a poskytne ConfigurationException v případě chyby.

Příklad: JAVA

```
Merchant merchant;
try
{
    merchant = new Merchant(propFile);
} catch (ConfigurationException e)
{
    System.err.println("error: " + e.getMessage());
    return;
}
String result = merchant.sendTransData(amount, currency, client_ip, description);
```

Posílání VS pro transakci:

```
java -jar ecomm_merchant.jar merchant.properties -v amount currency client_ip_addr description
language --additional_parameter1="value1" --additional_parameter2="value2"
```

Sample with data:

```
java -jar ecomm_merchant.jar merchant.properties -v 10 978 192.168.1.1 "description" "sk" --
additional_parameter1="value1" -- additional_parameter2="value2"
```

And example with account:

```
java -jar ecomm_merchant.jar merchant.properties -v 10 978 192.168.1.1 "description" "sk" --
account="1234567890"
```

Příklad: PHP

```
$merchant = new Merchant($ecomm_url,$cert_url, $cert_pass, 1);

$resp=$merchant->startDMSAuth(
    $amount,
    $currency,
    $client_ip_addr,
    $description,
    $language
    $account

);

Echo "$resp\n";
```


3.3.1. Provedení transakce SMS

Parametry příkazového řádku:

-v	identifikuje požadavek registrace transakce
amount	suma transakce v malých jednotkách, povinné (do 12 číslic)
currency	kód měny transakce, povinné (ISO 4217) (3 číslice)
client_ip_addr	IP adresa klienta, povinná (15 znaků)
description	stručný popis transakce, volitelný (mělo by být urlencoded – do 125 znaků)
language	identifikátor jazyka autorizace, volitelný (do 32 znaků) – alternativně lze toto pole použít i pro zobrazení cardinfo.html v responzivním designu (mobil, tablet) – pro detailní info viz – dokument 014_Testovací_prostredi_manual.pdf, str.3
account	identifikátor variabilního symbolu obchodníka (do 28 znaků) – je přenášený do HTML, XML výpisu pro obchodníka, pro identifikaci jednotlivých klientských transakcí

Volání metody:JAVA

```
public String  
startSMSTrans(String amount, String currency, String ip, String desc, String language, String  
account)
```

Parametry http post:

```
command=v&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&desc=<desc>&language=<language>&account=<account>&msg_type=SMS
```

Volání metody:PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);  
$resp = $merchant -> startSMSTrans($amount, $currency, $ip, $description, $language, $account);
```

Výsledek:

TRANSACTION_ID: <trans_id>

trans_id identifikátor transakce (28 znaků v kódování na základě 64)

Pokud je přítomna chyba, vrácený řetězec začíná „error:“

Příklad výsledku:

```
TRANSACTION_ID: bAt6JLX52DUbibbzD9gDFI5Ppr4=
```

3.3.2. Registrace autorizace DMS

Parametry příkazového řádku:

-a	identifikuje požadavek registrace transakce
amount	suma transakce v malých jednotkách, povinné (do 12 číslic)
currency	kód měny transakce, povinné (ISO 4217) (3 číslice)
client_ip_addr	IP adresa klienta, povinná (15 znaků)
description	kód měny transakce, povinné (mělo by být urlencoded – do 125 znaků)
language	identifikátor jazyka autorizace, volitelný (do 32 znaků) – alternativně lze toto pole použít i pro zobrazení cardinfo.html v responzivním designu (mobil, tablet) – pro detailní info viz – dokument 014_Testovací_prostredi_manual.pdf, str.3
account	identifikátor variabilního symbolu obchodníka (do 28 znaků) – je přenášený do HTML, XML výpisu pro obchodníka, pro identifikaci jednotlivých klientských transakcí

Volání metody:JAVA

```
public String  
startDMSAuth(String amount, String currency, String ip, String desc, String language, String account)
```

Parametry http post:

```
command=a&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&desc=<desc>&language=<language>&account=<account>&msg_type=DMS
```

Volání metody:PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);  
$resp = $merchant -> startDMSAuth($amount, $currency, $ip, $description, $language, $account);
```

Výsledek:

TRANSACTION_ID: <trans_id>

trans_id identifikátor transakce (28 znaků v kódování na základě 64)

Pokud je přítomna chyba, vrácený řetězec začíná „error:“

Příklad výsledku:

```
TRANSACTION_ID: bAt6JLX52DUbibbzD9gDFI5Pr4=
```

3.3.3. Provedení transakce DMS

Parametry příkazového řádku:

-t	identifikuje požadavek registrace transakce
auth_id	id předchozí úspěšné provedené autorizace
amount	suma transakce v malých jednotkách, povinné (do 12 číslic)
currency	kód měny transakce, povinné (ISO 4217) (3 číslice)
client_ip_addr	IP adresa klienta, povinná (15 znaků)
account	identifikátor variabilního symbolu obchodníka (do 28 znaků) – je přenášeno do HTML, XML výpisu pro obchodníka, pro identifikaci jednotlivých klientských transakcí

Volání metody:JAVA

```
public String  
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc, String  
account)
```

Parametry http post:

```
command=t&trans_id=<auth_id>&amount=<amount>&currency=<currency>&client_ip_addr=<ip>  
&desc=<desc>&language=<language>&account=<account>&msg_type=DMS
```

Volání metody:PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);  
$resp = $merchant -> makeDMSTrans($auth_id, $amount, $currency, $ip, $account);
```

Výsledek:

RESULT: <result>
RESULT_CODE: <result_code>
RRN: <rrn>
APPROVAL_CODE: <app_code>

result výsledek transakce:
 OK úspěšná transakce
 FAILED neúspěšná transakce

result_code kód výsledku transakce tak, jak je vrácen z autorizačního systému (3 číslice)

rrn číslo odkazu hledání, vrácené z autorizačního systému (12 znaků)

app_code kód schválení vrácený z autorizačního systému (maximálně 6 znaků)

Pole RESULT_CODE je pouze informativní. Pole RRN a APPROVAL_CODE se zobrazují pouze u úspěšných transakcí a jejich účelem je pomoci sledování transakce v autorizačním systému.

Rozhodnutí o úspěšnosti nebo selhání transakce musí být učiněno pouze na základě hodnoty v poli RESULT.

Pokud je přítomna chyba, vrácený řetězec začíná „error:“

Pokud je přítomné varování, vrácený řetězec začíná „warning:“

Příklad výsledku:

```
RESULT: OK
RESULT_CODE: 000
RRN: 123456789012
APPROVAL_CODE: 123456
```

3.3.4. Výsledek transakce

Výsledek SMS transakce musí být Obchodníkem vyžádán do tří minut od doby, kdy se klient vrátí na stránku obchodníka po provedení platby. Mějte prosím na paměti, že pokud výsledek transakce neproběhne do tří minut, transakce bude automaticky reverzována.

V případě DMS1 transakce musíte **požádat o výsledek** transakce **do tří minut**. Pokud však po DMS1 do tří minut provedete DMS2, pak nemusíte dodatečně požádat o výsledek DMS1 transakce. V případě DMS2 transakce bude výsledek vrácen automaticky.

Parametry příkazového řádku:

-c identifikuje požadavek na výsledek transakce
trans_id identifikátor transakce, povinný (28 znaků)
client_ip_addr IP adresa klienta, povinná (15 znaků)

Volání metody: JAVA

```
public String
getResult(String trans_id, String ip)
```

Volání metody: PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);
$resp = $merchant->getResult(urlencode($trans_id), $client_ip_addr);
```

Výsledek:

RESULT: <result>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
AAV: <aav>
RRN: <rrn>
APPROVAL_CODE: <app_code>

Výsledek RESULT: <result> RESULT_CODE: <result_code> 3DSECURE: <3dsecure> AAV:
<aav> RRN: <rrn> APPROVAL_CODE: <app_code>

result	výsledek transakce: OK úspěšná transakce FAILED neúspěšná transakce (nesprávná data) DECLINED neúspěšná transakce (neexistující data) REVERSED transakce je stornovaná AUTOREVERSED – transakce automaticky stornovaná ak nebylo požádáno o výsledek transakce v specifikovaném čase (do 3 minut) TIMEOUT časový limit transakce vypršel
result_code	kód výsledku transakce tak, jak je vrácen z autorizačního systému (3 číslice) 3dsecure 3D Secure status: AUTHENTICATED – 3D Secure autentifikace byla úspěšná NOT_AUTHENTICATED – Držitel karty nebyl autentifikován UNAVAILABLE – 3D Secure autentifikace není k dispozici ATTEMPTED – platný pokus o autentifikaci CHALLENGE – odpověď na autentifikační požiadavku nebyla ještě doručena REJECTED – vydávateľ platební karty odmítl autentifikaci SKIPPED – 3D Secure autentifikace byla vynechána na základe dynamických pravidel 3D Secure NOTPARTICIPATED – Neúčast v 3D schématu ERROR – 3 Secure chyby SYSERROR – systémová chyba UNKNOWNSCHEME – neznámé kartové schéma FAILED – stav po timeout
rrn	číslo odkazu hledání, vrácení autorizačního systému (12 znaků)
app_code	kód schválení vrácený z autorizačního systému (maximálně 6 znaků)
card_number	plně nebo částečně maskované číslo karty

Pole RESULT_CODE a 3DSECURE jsou pouze informativní (mohou být skrytá). Pole RRN a APPROVAL_CODE se zobrazují pouze u úspěšných transakcí a jejich účelem je pomoci sledování transakce v autorizačním systému. Rozhodnutí o úspěšnosti nebo selhání transakce musí být učiněno pouze na základě hodnoty v poli RESULT.

Pozn.: Výsledek transakce by neměl být vyžádán, pokud klient není přesměrovaný na adresu obchodníka: returnOKUrl / returnFailUrl. Pokud se klient nevrátí na returnOKUrl/returnFailUrl, pak výsledek transakce může být vyžádán za 13 minut. Detaily naleznete ve schématu **v kapitole 3.4**

Pokud je přítomna chyba, vrácený řetězec začíná „error:“
Pokud je přítomné varování, vrácený řetězec začíná „warning:“

Příklad výsledku:

```
RESULT: OK
RESULT_CODE: 000
3DSECURE: ATTEMPTED
RRN: 123456789012
APPROVAL_CODE: 123456
CARD_NUMBER: 5*****0014
```

3.3.5. Zrušení transakce

- 1) DMS autorizace (DMS1) může být zrušena pouze během počátečních 72 hodin od registrace autorizace (Step1). Po 72 hodinách SIA Central Europe zamítne pokus o reverzování autorizace s response code 914.
- 2) SMS a DMS transakce může být zrušena nezávisle, pokud není obchodní den uzavřený, nebo ne.
- 3) Zrušení DMS a SMS transakce může být provedeno do 90 dnů od data transakce. Po 90 dnech systém zrušení odmítne. Pro jednu transakci může být zrušení posláno pouze jednou.

Parametry příkazového řádku:

-r identifikuje požadavek na storno transakce
trans_id identifikátor transakce, povinný (28 znaků)
amount stornovaná suma v malých jednotkách, povinné (do 12 číslic) Stornovat lze celou částku.

Volání metody: JAVA

```
public String
reverse(String trans_id, String amount)
```

Volání metody: PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);
$resp = $merchant->reverse($trans_id, $amount);
```

Výsledek:

```
RESULT: <result>
RESULT_CODE: <result_code>
```

result výsledek storna:
 OK transakce je stornovaná
 FAILED transakce není je stornovaná
 REVERSED transakce je stornovaná

result_code kód výsledku storna tak, jak je vrácen z autorizačního systému (3 číslice)

Pokud je přítomna chyba, vrácený řetězec začíná „error:“
Pokud je přítomné varování, vrácený řetězec začíná „warning:“

Příklad výsledku:

```
RESULT: OK
RESULT_CODE: 400
```

3.3.6. Uzávěrka obchodního dne

Obchodní den je uzavřený při uzavěrce poslední otevřené dávky pro obchodníka. Uzavření obchodního dne se musí provádět každý den. Další dávka se otevře pouze s další úspěšnou transakcí.

Parametry příkazového řádku:

-b identifikuje požiadavku na uzávierku obchodného dňa

Voláni metody: JAVA

```
public String  
closeDay()
```

Volanie metódy: PHP

```
$merchant = new Merchant(($ecomm_server_url,$cert_url, $cert_pass, 1);  
$resp = $merchant->closeDay();
```

Výsledok:

```
RESULT: <result>  
RESULT_CODE: <result_code>  
FLD_074: <fld_074>  
FLD_075: <fld_075>  
FLD_076: <fld_076>  
FLD_077: <fld_077>  
FLD_086: <fld_086>  
FLD_087: <fld_087>  
FLD_088: <fld_088>  
FLD_089: <fld_089>
```

result	výsledok uzávierky obchodného dne:
	OK obchodní den je uzavřený
	FAILED obchodní den není uzavřený

result_code	kód výsledku uzávierky obchodného dne, jak byl přijatý z autorizačního systému (3 číslice)
FLD_074	počet kreditních transakcí (do 10 číslic), pouze když result_code začíná na 5
FLD_075	počet stornovaných kreditů (do 10 číslic), pouze když result_code začíná na 5
FLD_076	počet debetních transakcí (do 10 čísel), pouze když result_code začíná na 5
FLD_077	počet stornovaných debetů (do 10 číslic), pouze když result_code začíná na 5
FLD_086	celkový počet kreditních transakcí (do 16 číslic), pouze když result_code začíná na 5
FLD_087	celková hodnota stornovaných kreditů (do 16 číslic), pouze když result_code začíná na 5
FLD_088	celková hodnota debetních transakcí (do 16 číslic), pouze když result_code začíná na 5
FLD_089	celková hodnota stornovaných debetů (do 16 číslic), pouze když result_code začíná na 5

Pokud je přítomna chyba, vrácený řetězec začíná „error:“

Příklad výsledku:

```
RESULT: OK  
RESULT_CODE: 500  
FLD_074: 0  
FLD_075: 8  
FLD_076: 464  
FLD_077: 0  
FLD_086: 0  
FLD_087: 151100  
FLD_088: 24461939  
FLD_089: 0
```

3.3.7. Další možnosti nastavení

3.3.7.1 Použití dalších atribútů v dotaze

Další atribúty mohou být použitý pro poskytnutí vícero detailů Platobnímu serveru. Pro volání příkazového řádku je možný přidat volby/vlastnosti pomocí následující skladby:

```
<Ecomm command line call>
--<additional_property_name1>=<value1>
--<additional_property_name2>=<value2> ...
```

Pro volání metody JAVA, mohou být vlastnosti vloženy do argumentu metody vlastností. Pro HTTP POST volání mohou být vlastnosti vloženy jako standardní parameter:

```
command=<command>&<regular
parameters>&<additional_property_name1>=<value1>&
<additional_property_name2>=<value2> ...
```

3.3.7.2 Další 3D Secure atribúty

Některé další informace o držiteli karty mohou být posláni do služby 3D Secure 2.x verzie

Parametr	Popis
cardholder_email	E-mailová adresa držitele karty (max. 254 znaků)
Cardholder_phone_cc	Předvolba krajiny mobilního telefonního čísla držitele karty (1-3 číslice)
Cardholder_phone	Mobilní telefonní číslo držitele karty (max. 15 znaků)
Cardholder_home_phone_cc	Předvolba krajiny domácího telefonního čísla držitele karty (1-3 číslice)
Cardholder_home_phone	Domácí telefonní číslo držitele karty (max. 15 znaků)
Address_match	Označení či se adresa doručení a fakturační adresa shodují (Y=shoduje; N=neshoduje)
Bill_city	Město fakturace (max. 50 znaků)
Bill_country	Kód krajiny fakturace (3 písměna jako v ISO 3166-1 alpha-3 standard)
Bill_street	Fakturační adresa (max. 50 znaků)
Bill_street2	Fakturační adresa – druhá část ak je potřebná (max. 50 znaků)
Bill_street3	Fakturační adresa – třetí část ak je potřebná (max. 50 znaků)
Bill_postcode	Poštovní směrové číslo pro fakturaci (max. 16 znaků)
Bill_state	Kód krajiny pro fakturaci (max 3 znaky jako v ISO 3166-2)
Shipto_city	Město doručení (max. 50 znaků)
Shipto_country	Kód krajiny pro doručení (3 písměna jako v ISO 3166-1 alpha-3 standard)
Shipto_street	Adresa pro doručení (max.50 znaků)
Shipto_street2	Adresa pro doručení – druhá část ak je potřebná (max. 50 znaků)
Shipto_street3	Adresa pro doručení – třetí část ak je potřebná (max. 50 znaků)
Shipto_postcode	Poštovní směrové číslo pro doručení (max. 16 znaků)
Shipto_state	Kód krajiny pro doručení (max. 3 znaky jako v ISO 3166-2)

Poznámka: Atribúty platby by neměli být odesílaný ak nejsou přítomné v systéme obchodníka nebo jsou jinak nedostupný. Ak jsou atribúty odesílané musej mít platnou hodnotu.

3.3.8. Provedení přeadresování klienta

Přeadresování klienta (kvůli zadání údajů z platební karty) na adresu URL, která je definována bankou, může být provedeno podle metody GET a také POST. Je důležité zajistit, aby v průběhu přeadresování byla přenesena změněná trans_id. Tato proměnná obsahuje identifikátor transakce, která musí být uhrazena. (Pamatujte na to, že trans_id může obsahovat symboly jako „+“, „=“ a „/“, které musí být nahrazeny řetězci přijatelnými pro internetové prostředí (například, „=“ nahradit „%3D“) ještě před odesláním. V prostředí Java to lze provést pomocí metody nazývané URLEncoder.encode, v PHP environment urlencode()). V průběhu přeadresování mohou být navíc posílány další proměnné. Tyto parametry se vrátí zpět obchodníkovi při přeadresování klienta zpět na stránku obchodníka, parametry mohou být přijaty metodou POST.

Příklad metody POST, která se používá s využitím JavaScript je uveden v adresáři „example/client_to_ecomm.html“ a vypadá takhle:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Merchant example post template to ECOMM</title>
<script type="text/javascript" language="javascript">
function redirect() {
    document.returnform.submit();
}
</script>
</head>
<body onLoad="javascript:redirect()">
<form name="returnform" action="%%post_url%%" method="POST">
    <input type="hidden" name="trans_id" value="%%trans_id%%">

    <!-- To support javascript unaware/disabled browsers -->
<noscript>
    <center>Please click the submit button below.<br>
    <input type="submit" name="submit" value="Submit"></center>
</noscript>
</form>

</body>
</html>
```

Řetězec příkladů v této ukázce, %%post_url%%, musí být nahrazen za adresu URL serveru ECOMM, který poskytne banka a řetězec %%trans_id%% – bude nahrazen identifikátorem transakce.

3.3.9. Vytvoření cardinfo.html

Údaje karty klienta se zadají s použitím šablony cardinfo.html předdefinované Bankou s možností úpravy ze strany obchodníka – podrobné informace naleznete v dokumentu **README FIRST.docx ve svém balíku**. K tomu se používají speciální templates. ECOMM server rozeznává následující řetězce symbolů této šablony:

%%javascript%%	je nahrazený JavaScript k ověření vstupních polí. Pokud chce samotný obchodník použít tuto funkci, lze tento řetězec vypustit.
%%formdef%%	je nahrazený za <form action=<url> method="post" onSubmit="return FormValidator(this)"> <input type="hidden" name="trans_id" value="<trans_id"> readonly>
%%cardname%%	<input type="text" name="cardname" size="19" maxlength="50">
%%cardnr%%	<input type="text" name="cardnr" size="19" maxlength="19">
%%expmonth%%	<input type="text" name="validMONTH" size="2" maxlength="2">
%%expyear%%	<input type="text" name="validYEAR" size="2" maxlength="2">
%%cvc2%%	<input type="text" name="cvc2" size="4" maxlength="3">
%%amount%%	částka transakce pro tisk
%%ccalpha%%	měna transakce
%%description%%	popis transakce zadaný obchodníkem do ECOMM

V cardinfo.html je povinně obsažen validátor JavaScriptu:

```
<script src="/template/javascripts/formvalidator_en.js"></script>
```

Kde „en“ je jazyk rozhraní.

Není dovoleno integrovat platební stránku do IFRAME/FRAMESET.

POZN.: Není dovoleno používat vlastní Java script, atributy pro správu events (např. „onclick“, „onmouseover“, ...), ani externí linky – všechny tyto prvky budou automaticky odstraněny při uploadu souboru na ECOMM server.

Po výběru a potvrzení prodeje v doméně Obchodníka budou klienti přeměrováni na stránku, na které zadají údaje z karty (cardinfo.html); jedná se o dynamicky generovanou HTML stránku.

3.3.10. Funkce rozlišování jazyků

Počínaje verzí 2.08 programového vybavení ECOMM IMA, server podporuje funkci rozlišování jazyků. Tato funkce byla vyvinuta proto, aby serveru ECOMM mohl obsluhovat uživatelské rozhraní v různých jazycích.

Kromě implicitních šablon (cardinfo.html) by měl obchodník vytvořit další skupinu šablon pro každý požadovaný jazyk. Každá skupina šablon musí být spojena s kódem jazyku, na který se bude spoléhat personál serveru ECOMM. Kód jazyku musí mít maximální délku 32 znaků v ASCII kódu a měl by obsahovat pouze malá písmena, čísla a znak „_“ v ASCII kódu. Doporučené hodnoty kódů jazyka s obvykle používanými identifikátory jazyka tvoří dvě písmena (us, uk, fr, lv, ru, lt atd.). Parametr jazyka transakce musí být stejný jako název adresáře, ve kterém je příslušná card_info.html.

Jazyk použitý pro každou transakci lze specifikovat během registrace transakce v serveru ECOMM. Pokud jazyk transakce není uveden nebo není známý (server nenalezl šablony pro uvedený kód jazyka), používají se implicitní šablony dodané obchodníkem.

Příklad vytvoření stromové struktury s více jazyky pro cardinfo.html:

```
/[parametr jazyka]/cardinfo.html  
template/[ID_terminalu]/en  
                          logo.gif (images files)  
                          logo_en.gif  
                          logo_cz.gif
```

Download Template files for customization | Validation xml

▶ /cz	Apr 03 08:41	▶ /
▶ /en	Apr 03 08:41	<input type="radio"/> Create
▶ /fd	Apr 03 08:41	<input type="radio"/> Upload
▶ /kt	Apr 03 08:41	<input type="radio"/> Edit
📄 /business_error.html	1605 bytes Apr 03 08:42	<input type="radio"/> Delete
📄 /cardinfo.html	7124 bytes Apr 03 08:42	<input type="radio"/> File Preview
📄 /fail_to_merchant.html	2112 bytes Apr 03 08:42	<input type="radio"/> Reset file to Default
🎨 /locale.properties	521 bytes Apr 03 08:42	<input type="radio"/> Upload all files to 'Production' server
📄 /ok_to_merchant.html	2208 bytes Apr 03 08:42	
📄 /post_to_mpi.html	2795 bytes Apr 03 08:42	
▶ /template/IO999999	Nov 10 11:57	

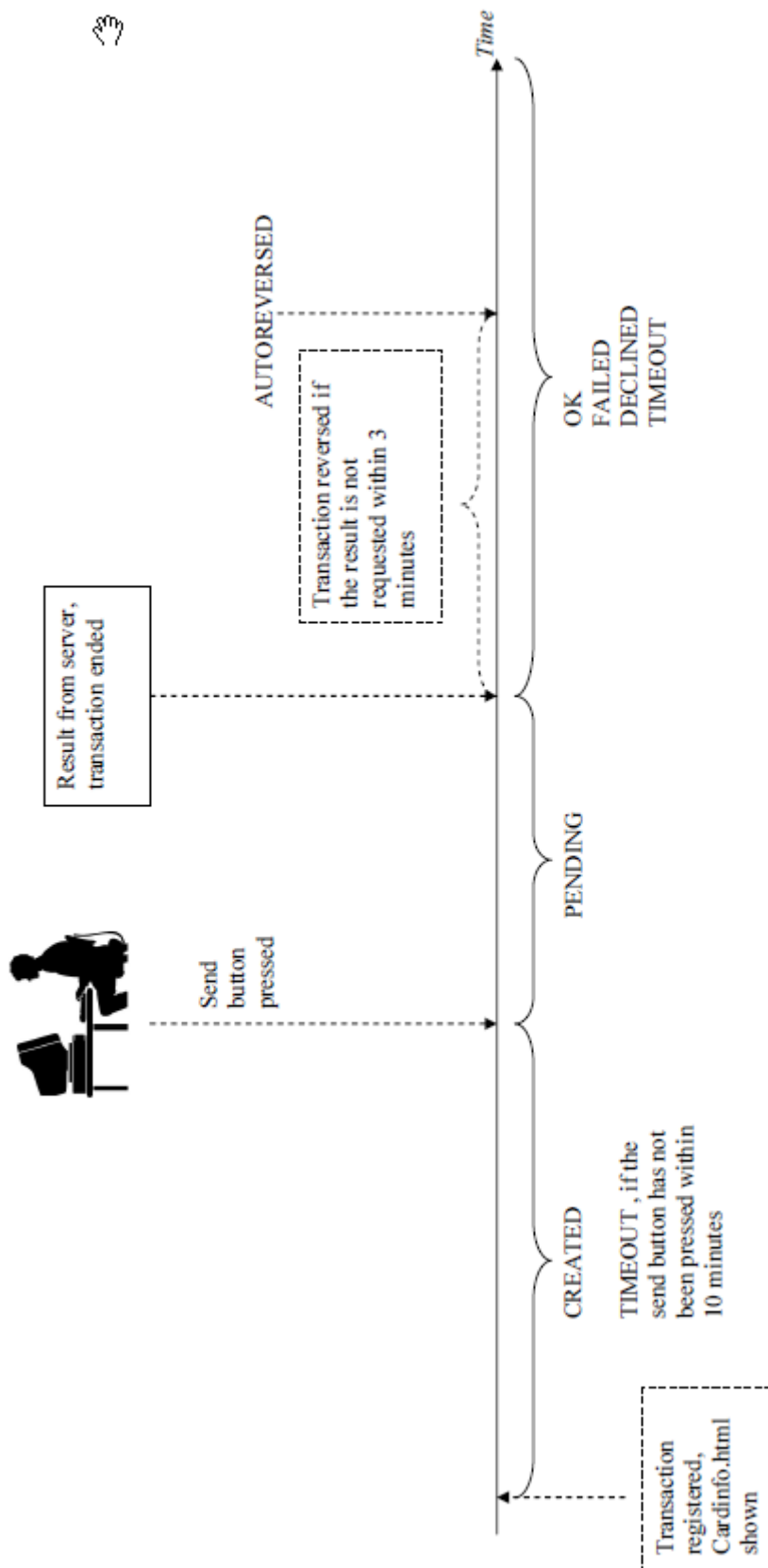
Reset Send

Picture 1: Tvorba jazykových adresářů

Pokud je IMA volané z příkazového řádku, měly by být nastaven i popisný parametr, jak je nastaven parametr jazyka. Je to kvůli tomu, že identifikátory popisu a jazyka nejsou povinné transakční parametry. Pokud je vyžadován popis, ale není vyžadován jazyk, pak by popis měl být specifikovaný prázdným řádkem ("").

Funkce rozlišování jazyků je přístupná pouze v uživatelském rozhraní. Pro uzavření obchodního dne a reverzály (storna) nelze zvolit jazyk.

3.4. Časová osa výsledku transakce



3.5. Požadavky na bezpečnost

1. Soubor cardinfo.html může obsahovat pouze popisné textové prvky a grafické podklady. Nejsou povoleny žádné aktivní prvky, jako například scripting v libovolném programovacím jazyku, inframes a ActiveX objekty.
2. Není povoleno, aby v souboru cardinfo.html byly přítomny url odkazy směřující mimo doménu **vpos.sia.eu**, s **výjimkou** povinných odkazů definovaných v grafickém manuálu. Pokud v souboru cardinfo.html používáte odkaz Zpět na objednávku, pak tento odkaz musí mít jeden z následujících kódů:
Back to order
Back to order
Back to order
3. Linky všeobecně nesmí obsahovat žádnou informaci z formulářových dat. Pokud chcete pro klienta zobrazit jiné údaje (například informaci, kde se na platební kartě nacházejí potřebné údaje k provedení transakce), směřujte odkazy na takovéto stránky na doménu **vpos.sia.eu**.
4. Do šablon (například cardinfo.html) nesmí být přidávána žádná dodatečná pole.
5. Form submit akce a adresy nesmí být nikde jakýmkoliv způsobem změněny. Formulář musí mít pouze jeden prvek s funkcí Submit.
6. Znaková sada pro šablonu je UTF-8. Není povolena její změna.
7. **Při přechodu do produkčního provozu je každá stránka kontrolována z hlediska dodržení těchto pravidel.**
8. **Při přechodu do produkce je každý obchodník automaticky zařazen do systému on-line realtime monitoringu transakcí.**