

**Integrovaný agent obchodníka elektronického obchodu
(IMA)
Verze 2.10.0
Příručka správce**

2. Příprava na provoz IMA

2.1. Systémové požadavky

2.1.1. Požadovaný software třetí strany

<u>Komponenta</u>	<u>Verze</u>	<u>Poznámky</u>
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	Požadována CURL knihovna

2.2. Instalace systému

2.2.1. Instalace produktu IMA

Instalace IMA je omezena na rozbalení archivu „EcommMerchant-2.10.0.zip“ do kořenového adresáře. Tím se vytvoří podadresář „EcommMerchant-2.10.0“, 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

- Bezpečnostní certifikáty:
 - 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**.
 - 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 mají platnost 1 rok a před ukončením jejich platnosti Obchodník obdrží z Banky nové produkční certifikáty.
- Keystore soubor se používá k vytvoření SSL spojení s ECOMM serverem a také k identifikaci Obchodníka vůči ECOMM systému.
- Možnosti – JAVA a PHP řešení

- a. V případě použití JAVA, modifikujte soubor „merchant.properties“ (v adresáři „java“):
- Bank.server.url** – adresa pro přístup k ECOMM serveru (od Bank/Procesora).
 - https.proxy.host** – adresa proxy serveru HTTPS (volitelná). **https.proxy.port** – port proxy serveru 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 získaný z banky.
 - keystore.type** – typ souboru certifikátu. Navrhovaná hodnota je „JKS“.
 - 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 přesměrovaný 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 přesměrování klienta zpět k obchodníkovi. Nezapomeňte, že přesměrování se provádí podle metody HTTP POST, což znamená, že adresa URL nemůže obsahovat žádné parametry.
- returnOkUrl** – klient bude přesměrovaný na tuto adresu po ověření 3D Secure a ukončení transakce (bez ohledu na její výsledek)
 - returnFailUrl** – klient bude přesměrovaný na tuto adresu v případě technické chyby v činnosti systému ECOMM.

3. Integrace IMA do řešení obchodníka

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 účtě držitele karty jsou rezervovány (zablokovány).

DMS transakce jsou odsouhlaseny příkazem `-t`. V případě PHP řešení to je funkce `makeDMSTrans()`. Když je provedena SMS transakce, prostředky jsou okamžitě 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 přesměrován 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 přesměrován zpět k obchodníkovi (s uvedením identifikátoru transakce).
6. Obchodník dostane informace z ECOMM o výsledku transakce (uskutečněná nebo neuskutečněná) 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"
```

Příklad s daty:

```
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_Testovacie prostredie IBIS_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: bAt6JLX52DUbibbzD9gDF15Ppr4=
```

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	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_Testovacie prostredie IBIS_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 startDMSTrans(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: bAt6JLX52DUbibbzD9gDF15Ppr4=
```

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 přenášený 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 account)
```

Parametry http post:

```
command=t&trans_id=<auth_id>&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&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 getTransResult(String trans_id, String ip)
```

Volání metody: PHP

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

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
	TIMEOUT časový limit transakce vypršel
	PENDING neukončená transakce – neúspěšná transakce
result_code	kód výsledku transakce tak, jak je vrácen z autorizačního systému (3 číslice)
3dsecure	3D Secure status
	DECLINED – 3D Secure autorizace byla neúspěšná
	AUTHENTICATED – 3D Secure autorizace byla úspěšná
	NOTPARTICIPATED – Neúčast v 3D schématu
	NO_RANGE – not enrolled transakce
	ATTEMPTED – platný pokus o autentifikaci
	UNAVAILABLE – autentifikace je nedostupná

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 skryta). 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 přesměrován 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 First Data zamítne pokus o reverzování autorizace s response code 914.
- 2) SMS a DMS transakce může 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 i částečnou částku, v závislosti na schopnosti acquirera/procesora. K objasnění této záležitosti kontaktujte prosím svého acquirera/procesora.

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 = $merchant -> reverse(urlencode($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 uzavěrci 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 požadavek na storno transakce

Volání metody: JAVA

```
public String closeDay()
```

Volání metody: PHP

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

Výsledek

```
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 výsledek uzávěrky obchodního dne:
 OK obchodní den je uzavřený

result_code kód výsledku uzávěrky obchodního 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:“

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. Provedení přesměrování klienta

Přesměrování 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 přesměrování 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 PHO environment `urlencode()`). V průběhu přesměrování mohou být navíc posílány další proměnné. Tyto parametry se vrátí zpět obchodníkovi při přesměrování 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 takto:

```
<!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.

3.3.8. 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 přesměrováni na stránku, na které zadají údaje z karty (cardinfo.html); jedná se o dynamicky generovanou HTML stránku.

3.3.9. 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 registrování 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
```

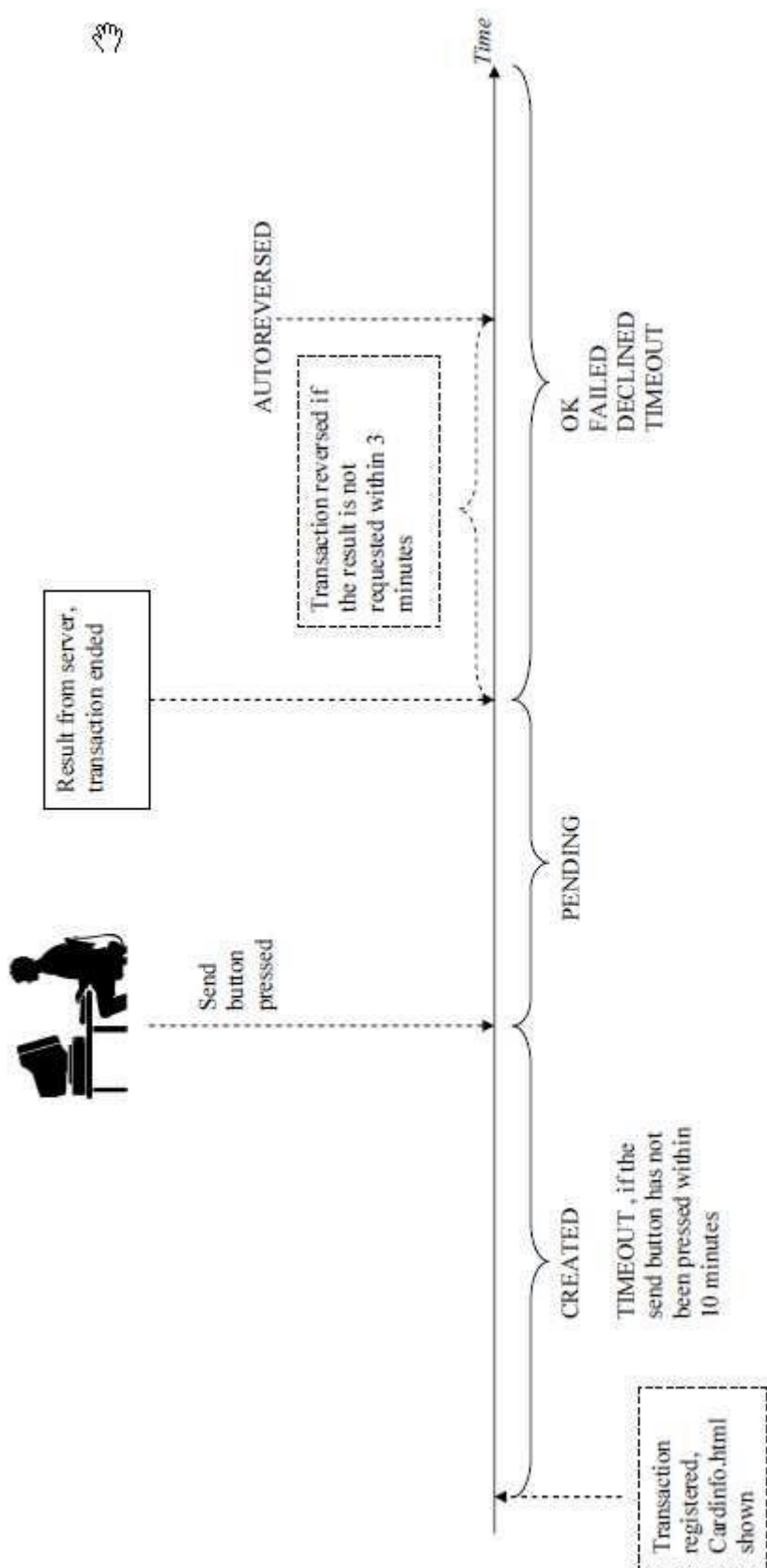


Picture 1: Tvorba jazykových adresářů

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 **secureshop.firstdata.lv**, 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 secureshop.firstdata.lv.
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.**