

Инструкция за изграждане на връзка към системата за онлайн картови плащания на Първа инвестиционна банка

Версия 1.2

Съдържание:

1 ВЪВЕДЕНИЕ	3
1.1 ЦЕЛ.....	3
1.2 ПРЕДНАЗНАЧЕНИЕ.....	3
1.3 ПРИЛОЖЕНИЯ КЪМ НАСТОЯЩИЯ ДОКУМЕНТ.....	3
2 КОМУНИКАЦИЯ МЕЖДУ ТЪРГОВЕЦ И ЕСОММ	4
2.1 ФУНКЦИОНАЛНА СХЕМА НА КОМУНИКАЦИЯ С ЕСОММ ПРИ КАРТОВО ПЛАЩАНЕ	4
2.2 АДРЕСИ ЗА КОМУНИКАЦИЯ С ЕСОММ	5
2.3 СРЕДСТВА ЗА КОМУНИКАЦИЯ НА ТЪРГОВЕЦ С ЕСОММ	6
2.4 ПОДГОТОВКА НА ТЪРГОВСКАТА СИСТЕМА ЗА КОМУНИКАЦИЯ С ЕСОММ.....	7
2.5 ПОДГОТОВКА НА CARD_INFO ФАЙЛ ОТ СТРАНА НА ТЪРГОВЕЦ.....	9
2.6 НАСОЧВАНЕ НА ПОТРЕБИТЕЛЯ КЪМ ПЛАТЕЖНАТА СТРАНИЦА В БАНКОВАТА СИСТЕМА	9
3 ИЗПРАЩАНЕ НА ЗАЯВКИ КЪМ ЕСОММ.....	11
3.1 ЗАЯВКА ЗА РЕГИСТРАЦИЯ НА ТРАНЗАКЦИЯ	11
3.2 ЗАЯВКА ЗА ПОЛУЧАВАНЕ НА РЕЗУЛТАТ ОТ ТРАНЗАКЦИЯ	13
3.3 ВРЪЩАНЕ НА СУМА ОТ КАРТОВО ПЛАЩАНЕ (REFUND)	17
3.4 АНУЛИРАНЕ НА КАРТОВА ТРАНЗАКЦИЯ (REVERSAL).....	20
4 ВГРАЖДАНЕ НА ЕСОММ КЛИЕНТ В ТЪРГОВСКИЯ САЙТ	22
4.1 ПРИМЕР ЗА ВГРАЖДАНЕ НА ЕСОММ КЛИЕНТ В ТЪРГОВСКИЯ САЙТ.....	22
5 ИЗГРАЖДАНЕ НА ВРЪЗКА С БАНКОВАТА СИСТЕМА БЕЗ ИЗПОЛЗВАНЕ НА ЕСОММ КЛИЕНТИТЕ	28
5.1 ПРИМЕР ЗА КОМУНИКАЦИЯ С ЕСОММ ОТ РНР СРЕДА.....	28
6 ПРОВЕЖДАНЕ НА ТЕСТОВЕ.....	30
6.1 ТЕСТОВИ КАРТИ	30
6.2 ПОСЛЕДВАЩИ ТЕСТОВЕ	30

Списък с фигури

1	Фигура	4.1.1
.....	17
2	Фигура	4.1.2
.....	17

1 Въведение

1.1 Цел

Този документ представлява инструкция за изграждане на връзка между системата на онлайн Търговец и системата на FIBANK за сигурни разплащания с банкови карти през интернет (ECOMM).

В настоящата инструкция се описва интерфейса между системата на онлайн Търговец и ECOMM системата на Банката, като се специфицират: изискванията към търговската платформа; средствата, чрез които се осъществява сигурна връзка през интернет с банковата система; форматът и съдържанието на съобщенията, които се обменят между двете системи и логиката, по която протича този обмен.

1.2 Предназначение

Документът е предназначен за разработчици на търговски уеб сайтове, които трябва да имплементират функционалност за картови плащания през интернет чрез системата на FIBANK в съответната търговска платформа.

1.3 Приложения към настоящия документ

Настоящата инструкция трябва да бъде използвана заедно със следните документи и пакети:

- 1 Instruction 3D_2010_ECOMM.pdf
- 2 Fibank_integration_package.zip пакет за интеграция
- 3 merchant.properties.txt
- 4 ResponseCodes.txt
- 5 ECOMM_addresses_for_communication.txt

Изброените документи се намират в директорията "Doc" на пакета за интеграция.

2 КОМУНИКАЦИЯ МЕЖДУ ТЪРГОВЕЦ И ECOMM

Комуникацията между Търговец и ECOMM се осъществява според следните принципи:

- комуникационната сесия винаги се инициира от търговеца (ECOMM само отговаря на получени заявки);
- търговецът се идентифицира пред ECOMM при всяка своя заявка;
- комуникацията се осъществява по сигурен канал през Интернет;
- за идентифицирането на търговец в ECOMM и за договаряне на TLS сесия през Интернет се използва Java Keystore файл със сертификати, който се генерира от Банката и се предоставя на търговеца след сключване на договор за разплащания. Генерираният Keystore файл е защитен с парола и е индивидуален за всеки един Търговец.

2.1 Функционална схема на комуникация с ECOMM при картово плащане

- 1 Потребителят е избрал продукт в търговския уеб сайт и е готов да направи картово плащане за този продукт.
- 2 Потребителят натиска съответният бутон за потвърждение на своята покупка в търговския уеб сайт (Checkout button).
- 3 Търговският сървър инициира TLS сесия с ECOMM, идентифицира се чрез своя сертификат в Keystore файла и прави заявка за регистриране на картова транзакция. Индикира стойността на следните параметри: сума на транзакцията; валута на транзакцията; IP адрес на потребителя; описание на транзакцията.
- 4 ECOMM обработва заявката и регистрира тази транзакция в платежния сървър на Банката.
- 5 За регистрираната транзакция в платежния сървър на Банката се генерира уникален идентификатор (TRANSACTION_ID).
- 6 ECOMM връща отговор на Търговеца, в който изпраща TRANSACTION_ID на регистрираната транзакция.
- 7 Търговецът получава регистрирания TRANSACTION_ID параметър, след което насочва потребителския браузър към платежната страница на ECOMM (чрез POST или GET, като при тази заявка трансферира и TRANSACTION_ID параметъра).

- 8 В платежната страница на ECOMM потребителят въвежда данните на своята банкова карта и потвърждава картовото плащане.
- 9 Транзакцията се изпълнява в банковата система.
- 10 След приключване на картовата транзакция, ECOMM връща потребителя обратно към сайта на Търговеца чрез POST заявка (към предварително спесифициран URL адрес), като подава и TRANSACTION_ID параметъра на съответната транзакция.
- 11 Търговецът инициира нова TLS сесия с ECOMM, идентифицира се чрез своя сертификат в Keystore файла и изпраща заявка за получаване на резултат от приключилото картово плащане.
- 12 ECOMM изпраща отговор за резултата от картовото плащане (успешно плащане / неуспешно плащане).

2.2 Адреси за комуникация с ECOMM

Банката разполага с реална и с тестова система за онлайн картови плащания. Процесът по изграждане на връзка между търговската платформа и банковата система, задължително се осъществява - първо в тестовата среда на Банката.

След като, изградената връзка с тестовата среда на Банката бъде проверена и се извършат съответните тестове, на Търговеца се осигурява достъп до реалната система на Банката. Адресите за комуникация на Търговец с ECOMM системата на Банката се намират във файла „ECOMM_addresses_for_communication.txt“ в пакета за интеграция “Fibank_integration_package.zip”:
Fibank_integration_package.zip \cs.bc.ecomm_merchant-1.1.59\doc

2.3 Средства за комуникация на Търговец с ECOMM

ECOMM модулът за комуникация с Търговец е Java базиран. Комуникацията на Търговец с този модул се осъществява чрез ECOMM клиент (IMA / Integrated Merchant Agent). Този клиент представлява JAR архив (ecommm_merchant.jar), който съдържа готови класове и методи за инициране на връзка с ECOMM и изпращане на заявки за транзакции. Търговецът получава ECOMM клиента от Банката чрез пакета "Fibank_integration_package.zip". Този клиент трябва да бъде вграден в системата на Търговеца. В пакета "Fibank_integration_package.zip" има версия на ECOMM клиента, която може да работи в .NET среда (ecommm_merchant.dll). Тази версия на клиента се намира в папка "dot_NET" от пакета за интеграция.

Връзката с ECOMM може да бъде инициализирана чрез ECOMM клиента (IMA) по два начина:

- 1 Извикване на Java архива "ecommm_merchant.jar" от команден ред (примери могат да бъдат намерени в "example" директорията на пакета "Fibank_integration_package.zip").
- 2 Директно извикване на методите от lv.tietoenator.cs.ecommm_merchant.Merchant класа, който се намира в Java архива "ecommm_merchant.jar". За Merchant класа, който се създава - трябва да бъде зададено име на конфигурационен файл. В този файл се задават параметри, свързани с комуникационната сесия с ECOMM (bank.server.url; keystore.file; keystore.type; keystore.password). Пример за конфигурационен файл се предоставя, като приложение към настоящата инструкция (merchant.properties.txt). Задаването на конфигурационен файл инициализира ECOMM клиента.

Пример:

```
Merchant merchant;
try
{
    merchant = new Merchant(propFile);
} catch (ConfigurationException e)
{
    System.err.println("error: " + e.getMessage());
    return;
}

String result = merchant.startSMSTrans(amount, currency, client_ip, description);
```

Възможно е Търговец да комуникира с ECOMM без да бъде използван IMA клиента. Подобна комуникация може да се осъществи чрез директни http заявки по TLS сесия, като параметрите на транзакцията се изпращат към ECOMM чрез POST метод. Търговецът трябва да използва получените от Банката сертификати в Keystore файла, за да осъществи криптирана TLS сесия с ECOMM и да изпрати съответната заявка.

2.4 Подготовка на търговската система за комуникация с ECOMM

В системата на Търговеца трябва да има среда, която е съвместима с ECOMM (IMA) клиентите, предоставени от Банката (JRE или .NET). ECOMM клиентът “ecommerce_merchant.jar” изисква Java Runtime Environment (JRE), версия 1.8 или по-висока в системата на Търговеца.

Клиентът “ecommerce_merchant.dll” изисква .NET среда в системата на Търговеца. За да бъде инициализиран е необходима Java виртуална машина с имплементация за .NET. За тази цел се използва продуктът IKVM.NET (<http://www.ikvm.net>). Файловете на този продукт се намират в директория “Fibank_integration_package.zip\dot_NET\ecommerce_client\dot_net_java virtual machine” на пакета за интеграция.

Подготовката на търговската система за комуникация с ECOMM включва, следните стъпки:

- 1 Търговецът разархивира “Fibank_integration_package.zip” пакета в своята работна директория.
- 2 В директорията, в която се намира файла “ecommerce_merchant.jar” се поставят, следните Java библиотеки: jcert.jar, jnet.jar, jsse.jar (тези файлове обикновено са включени в стандартна JRE инсталация).

В случай, че Търговецът използва .NET версията на ECOMM клиента - в директорията, в която се намира файла “ecommerce_merchant.dll”, се поставят файловете на IKVM.NET (IKVM.OpenJDK.Core.dll, IKVM.OpenJDK.Security.dll, IKVM.OpenJDK.Text.dll, IKVM.OpenJDK.Util.dll, IKVM.Runtime.dll).

- 3 Търговецът получава Keystore файл със сертификати и парола от Банката. Keystore файлът се използва за идентифициране на Търговец в банковата система и за установяване на сигурна връзка с ECOMM.

- 4 Търговецът получава примерен конфигурационен файл от Банката (merchant.properties.txt). В него се задават параметри, свързани с комуникационната сесия с ECOMM. В получения файл, Търговецът трябва да зададе параметри за:

bank.server.url – адрес за комуникация на Търговеца с банковата система (виж точка 2.2 от настоящата инструкция);

keystore.file – трябва да бъде посочен пътя до Keystore файла в директорията на Търговеца ;

keystore.type – тип на Keystore файла (JKS);

keystore.password – паролата за Keystore файла, получена от Банката.

- 5 Банката трябва да получи информация от Търговеца за IP адреса на търговския сървър, от който ще бъде иницирана комуникацията с ECOMM, за да бъде осигурен съответния мрежов достъп в банковата система.
- 6 Банката трябва да получи информация за URL адресите в търговската система, към които да препраща потребителя, след приключване на картовото плащане (OKUrl и FailUrl). Препращането на потребителя, обратно към търговския сайт се изпълнява чрез HTTP POST метод. В този процес OKUrl адресът се използва при успешно обработена и приключила картова транзакция (без значение от резултатът от картовото плащане: успешно; отказано, поради недостатъчна наличност и др.). FailUrl адресът се използва при неуспешно обработена транзакция, поради комуникационен или технически проблем в ECOMM. OKUrl и FailUrl адресите могат да бъдат идентични.
- 7 Търговецът трябва да получи примерен card_info файл от банката. Този файл е HTML форма за попълване на картовите данни от страна на потребителя. Card_info файлът се зарежда в платежната страница на банковата система и чрез него потребителят въвежда и изпраща картовите параметри към платежния сървър на Банката. Търговецът трябва да модифицира този файл според своите изисквания, като спазва указанията за процеса, описани в точка 2.5 от настоящата инструкция.

2.5 Подготовка на card_info файл от страна на Търговец

Пакетът за интеграция съдържа папка “merchant_cardinfo” (Fibank_integration_package\cs.bc.ecomm_merchant-1.1.59\merchant_cardinfo). В тази папка се намира базовата форма за въвеждане на картовите данни от страна на потребителя. Формата представлява HTML темплейт. Темплейтът се поставя в банковата система и служи за въвеждане и изпращане на картовите параметри от страна на потребителя към банковия платежен сървър. Търговецът може да модифицира дизайна на темплейта според своите изисквания. Не трябва да бъде изтриван или променян “JavaScript” кода в темплейта, който служи за проверка на въведените параметри от потребителя и за коректното изпращане на параметрите към платежния сървър на Банката.

Формата за картовите данни се състои от HTML темплейт и файлове с изображения в растерен формат (лога на приеманите банкови карти, лого на търговец, цвят на фон и др.). Файловете с изображенията се поставят в папка с името на търговеца, като се запази структурата на базовия темплейт (пример: ако търговецът се казва Merchant ABC, изображенията трябва да се поставят в папка merchant_abc. Папката трябва да бъде поддиректория на директорията images.). Адресацията на изображенията в HTML темплейта трябва да бъде относителна и да е съобразена с имената на съответните папки (пример: ако файлът merchant_logo.jpg се намира в папката merchant_abc, адресацията на този файл в темплейта трябва да бъде: images/merchant_abc/merchant_logo.jpg). Търговецът може да съхранява файловете с изображения на своя сървър. В този случай, адресацията в темплейта трябва да бъде абсолютна (пример: https://merchantabc.bg/shop/images/merchant_logo.jpg), като изображението трябва да се зарежда в клиентския браузър чрез HTTPS комуникация.

Търговецът не може да добавя свои скриптове в HTML темплейта.

2.6 Насочване на потребителя към платежната страница в банковата система

След като, Търговецът регистрира транзакция в ECOMM и получи TRANSACTION_ID от банковата система, потребителят се насочва към платежната страница на Банката. На тази страница потребителят трябва да въведе данните на своята карта в card_info формата и да направи картово плащане. Насочването на потребителя към платежната страница трябва да бъде изпълнено чрез GET или POST метод. По време на този трансфер Търговецът трябва да изпрати към Банката и TRANSACTION_ID параметъра.

```
<!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="-ECOMM-communication-address" 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>
```

TRANSACTION_ID параметърът може да съдържа някои от следните символи ('+', '=' и '/'). Тези символи са неприемливи в WEB среда и трябва да бъдат заменени с приемливи символи при трансфериране на TRANSACTION_ID параметъра към платежната страница (например: '=' трябва да бъде заменен с '%3D'). В Java среда за тази цел би могъл да се използва URLEncoder.encode методът. След приключване на картовото плащане, ECOMM връща потребителя обратно в търговския сайт (според спесифицираните OKUrl/FailUrl), като трансферира TRANSACTION_ID параметъра на картовата транзакция. Търговецът не трябва да използва i-frame при насочването на потребителя към платежната страница на Банката (за потребителя трябва да бъде ясно, че той е прехвърлен към банкова платежна страница, за да въведе данните на своята карта).

3 Изпращане на заявки към ECOMM

Процесът на комуникация на Търговец с ECOMM включва изпращане на четири типа заявки от страна на Търговеца към банковата система:

- заявка за регистрация на картова транзакция в банковата система;
- заявка за получаване на резултат от картова транзакция.;
- заявка за връщане на сума от картова транзакция;
- заявка за анулиране на картова транзакция.

Търговецът трябва да инициира TLS сесия с ECOMM и да се идентифицира чрез сертификата от Keystore файла, при всяка своя заявка от четирите типа.

3.1 Заявка за регистрация на транзакция

При избор на съответния продукт и натискане на бутона за потвърждение (checkout button) от страна на потребител в търговския уеб сайт, Търговецът трябва да изпрати заявка за регистриране на транзакция в ECOMM в следният формат:

-v	командата, идентифицираща заявка за регистрация на транзакция
amount	сума на транзакцията (до 12 символа)
currency	код на валута на транзакцията (3 символа)
client_ip_addr	ip адрес на потребителя (до 15 символа)
description	описание на транзакцията (до 125 символа)

1 Сума на транзакцията (amount): сумата на картовата транзакция трябва да бъде изпращана към ECOMM – без десетичен знак (пример: 125,67 лв. трябва да бъдат изпратени, като 12567).

2 Валута на транзакцията (currency): валутата на транзакцията се заявява чрез ISO 4217 код. Поддържаната валута на транзакция от ECOMM е: български лев (ISO 4217 код: 975);

3 Описание на транзакцията (description): описание на транзакцията (номер на поръчка, уникална референция на заявката от търговския сайт).

Изпълнение:

Изпълнение на заявката от команден ред чрез извикване на “ecommm_merchant.jar” архива:

```
java -jar ecomm_merchant.jar merchant.properties -v 100 975 192.168.100.1 test transaction
```

Изпълнение на заявката чрез извикване на съответният метод от lv.tietoenator.cs.ecomm.merchant.Merchant класа:

```
public String  
    startSMSTrans(String amount, String currency, String ip, String description)
```

При изпълнение на заявката чрез startSMSTrans метода, не трябва да се подава (-v) командата.

Изпълнение на заявката чрез HTTP POST

```
command=v&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<description>&msg_type=SMS
```

Резултат:

При изпълнение на заявката за регистрация на транзакция от страна на Търговец, ECOMM връща отговор с TRANSACTION_ID на транзакцията, регистрирана в банковата система. При наличие на проблем, ECOMM отговаря със стринг от символи, започващ с „error”. TRANSACTION_ID параметърът е стринг с дължина от 28 символа. Този стринг е base64 кодиран.

Пример за резултат от заявка за регистрация на транзакция:

```
TRANSACTION_ID: bAt6JLX52DUbibbzD9gDF15Ppr4=
```

Всяка регистрирана транзакция в банковата система е активна за плащане в продължение на 10 минути.

3.2 Заявка за получаване на резултат от транзакция

След приключване на картовото плащане в банковата система, ECOMM връща потребителския браузър към OK/Fail Url адреса в търговския уеб сайт, като подава TRANSACTION_ID параметъра на картовата транзакция и описанието на транзакцията, което Търговецът е изпратил в заявката за регистрация на това плащане.

Параметри, изпращани от ECOMM при връщане на потребителя към търговския сайт:

Пример:

```
[trans_id] = bAt6JLX52DUbibbzD9gDF15Ppr4=  
[Ucaf_Cardholder_Confirm] = 0
```

В случай, че заявката за транзакция на Търговеца не е била обработена коректно и не е приключила успешно (поради технически или комуникационен проблем в ECOMM), банковата система пренасочва потребителя към FailUrl адреса в търговския сайт, като стрингът от параметри, които Банката изпраща започва с „error”.

С препращането на потребителя към OKUrl адреса, ECOMM показва на Търговеца, че това картово плащане е било обработено и е приключило. На този етап Търговецът все още не знае – какъв е конкретният резултат от картовата транзакция. За да, получи този резултат, търговската система трябва да изпрати заявка за получаване на резултат от картово плащане към ECOMM. Заявката има следния формат:

-c	командата, идентифицираща заявка за резултат от транзакция
trans_id	идентификаторът на транзакцията (задължително поле; 28 символа)
client_ip_addr	ip адрес на потребителя (задължително поле; до 15 символа)

Изпълнение:

Изпълнение на заявката от команден ред чрез извикване на “ecommm_merchant.jar” архива:

```
java -jar ecomm_merchant.jar merchant.properties -c bAt6JLX52DUbibbzD9gDF15Ppr4=  
192.168.100.1
```

Изпълнение на заявката чрез извикване на съответният метод от
lv.tietoanator.cs.ecomm.merchant.Merchant класа:

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

При изпълнение на заявката чрез getTransResult метода, не трябва да се подава (-с) командата.

Изпълнение на заявката чрез HTTP POST

```
command=c&trans_id=<trans_id>&client_ip_addr=<ip>
```

Резултат:

При изпълнение на тази заявка, ECOMM връща резултат от картовото плащане под формата на следните параметри:

```
RESULT: <резултат от картовото плащане>  
RESULT_PS: <статус на плащането в платежния сървър>  
RESULT_CODE: <код за резултата от картовото плащане>  
3DSECURE: <резултат от 3D Secure автентикацията на потребителя>  
RRN: <референтен номер на картовото плащане в платежния сървър>  
APPROVAL_CODE: <авторизационен код на картовото плащане>  
CARD_NUMBER: <маска на банковата карта във формат X*****XXXX>
```

Примерен отговор от ECOMM с резултат от картово плащане:

```
RESULT: OK  
RESULT_PS: FINISHED  
RESULT_CODE: 000  
3DSECURE: OK  
RRN: 123456789012  
APPROVAL_CODE: 123456  
CARD_NUMBER: 4*****6789
```

Параметърът, който показва – дали картовото плащане е успешно е “RESULT”.
Търговецът трябва да вземе решение за резултата от плащането, според стойността на този
параметър. Останалите

параметри от отговора на ECOMM имат информативен характер.

Параметърът

“RESULT” може да има следните стойности:

OK – успешно плащане

FAILED – неуспешно плащане

CREATED – плащане, регистрирано в банковата система (незавършено)

PENDING – плащане, чакащо да бъде платено (незавършено)

DECLINED – отказано плащане от ECOMM

REVERSED – анулирано (отменено) плащане

AUTOREVERSED – плащане, отменено чрез автоматична анулация

TIMEOUT – времето за плащане е изтекло (незавършено)

За успешно плащане може да се смята, само това, за което ECOMM е върнал “RESULT” параметър със стойност “OK”.

За всяка приключила транзакция Търговецът трябва да съхранява следните параметри, получени от ECOMM:

TRANSACTION_ID

RESULT

RESULT_PS

RESULT CODE

3DSECURE

RRN

APPROVAL_CODE

CARD_NUMBER

При необходимост, Банката може да изиска информация от Търговеца за конкретен потребител, който е направил картово плащане за поръчка от търговската система. Банката ще специфицира конкретната транзакция чрез изброените по-горе параметри.

Параметърът “RESULT_CODE” показва код на резултата от картовото плащане. Търговецът получава файла “ResponseCodes.txt” като приложение към настоящата инструкция. Файлът съдържа описание на всички кодове за резултат от картово плащане, които Търговецът може да получи от ECOMM.

Поради характеристиките на комуникацията през Интернет е възможно – възникването на ситуация, при която потребителят приключва своето плащане в банковата система, но пренасочването му обратно към търговския уеб сайт не е успешно (потребителят затваря своят браузър, преди да бъде препратен към търговския сайт; временен срив в Интернет комуникацията и др.). При възникване на подобна ситуация Търговецът не би разбрал, че картовото плащане е приключило. Търговецът може да изпрати заявка за получаване на резултат от плащането, без потребителят да е бил върнат успешно в търговския уеб-сайт.

Възможно е Търговецът да насочи потребител към страницата за плащане в банковата система, но този потребител да не завърши съответната картова транзакция, поради някаква причина (отказал се е от плащането; забавил се е повече от 10 минути и др.). Потребител, който не е финализиран своето плащане в страницата на Банката - не се пренасочва обратно към търговския уеб-сайт. При този случай, Търговецът също може да изпрати заявка за получаване на резултат от картово плащане към ECOMM без потребителят да е върнат в търговската система. ECOMM ще отговори с резултат "PENDING", ако плащането не е завършено, но все още е активно за плащане. В случай, че са изминали повече от 10 минути, след като транзакцията е била регистрирана в Банковата система и тази транзакция не е била платена, ECOMM деактивира транзакцията и при заявка за резултат от страна на Търговеца, банковата система отговаря с резултат "TIMEOUT". Деактивирана транзакция не може да бъде активирана повторно (Търговецът трябва да направи нова заявка за регистрация на транзакция и да получи нов TRANSACTION_ID параметър).

Препоръчително е Търговецът да изгради механизъм в своята система, чрез който да проверява всички заявки за транзакции, за които няма получен отговор от ECOMM.

3.3 Връщане на сума от картово плащане (Refund)

Интерфейсът за комуникация между Търговец и ECOMM осигурява възможност за изпълнение на заявка за връщане на сума от приключило плащане. Тази заявка може да бъде изпълнена само за успешно приключило картово плащане (RESULT: OK). Заявката за връщане може да бъде изпълнена за цялата сума на приключилото плащане или за част от нея. Могат да бъдат изпълнени няколко заявки за връщане на сума за дадено плащане при следното ограничение: общата сума на заявките за връщане не трябва да надвишава сумата на приключилото плащане.

Заявката има следният формат:

-k командата, идентифицираща заявка за връщане (Refund)
trans_id идентификаторът на транзакцията, за която се изпълнява връщане (задължително поле; 28 символа)
amount сума за връщане (без десетичен знак; до 12 символа дължина) – опционално поле. При заявка за връщане без изпращане на "amount" параметър се изпълнява връщане на цялата сума на приключилото плащане.

Изпълнение:

Изпълнение на заявката от команден ред чрез извикване на "ecommerce_merchant.jar" архива:

```
java -jar ecommerce_merchant.jar merchant.properties -k bAt6JLX52DUbibbzD9gDF15Ppr4=13658
```

Изпълнение на заявката чрез извикване на съответният метод от `lv.tietoenator.cs.ecomm.merchant.Merchant` класа:

```
public String  
refund(String trans_id, Properties properties)  
  
public String  
refund(String trans_id, String amount, Properties properties)
```

При изпълнение на заявката чрез `refund` метода, не трябва да се подава (-k) командата.

Изпълнение на заявката чрез HTTP POST

```
command=k&trans_id=<trans_id>&amount=<amount>(&<property_name>=<property_value>)
```

Препоръчва се при изпълнение на заявка за връщане да се изпраща допълнителен description параметър, в който още веднъж да се подава trans_id на плащането, за което се прави връщане.

```
command=k&trans_id=<trans_id>&amount=<amount>(&<description>=<Refund for trans_id>)
```

```
command=k&trans_id=bAt6JLX52DUbibbzD9gDF15Ppr4&amount=5697(&description=Refund for  
bAt6JLX52DUbibbzD9gDF15Ppr4)
```

Резултат:

При изпълнение на тази заявка, ECOMM връща отговор с резултат от това изпълнение:

```
RESULT: <резултат от картовото плащане>  
RESULT_CODE: <код за резултата от картовото плащане>  
REFUND_TRANS_ID: <ID на транзакцията за връщане>
```

Примерен отговор от ECOMM с резултат от заявка за връщане:

```
RESULT: OK  
RESULT_CODE: 000  
REFUND_TRANS_ID: ecXzh5Bi3oL1ZUHegwZ0/adVw1U=
```

Параметърът “RESULT” може да има следните стойности:

OK – успешно връщане

FAILED – неуспешно връщане

При неуспешна обработка на заявката за връщане ECOMM изпраща отговор със стринг, който започва с “error”.

ECOMM ще изпрати подобен отговор при заявка за връщане на сума, която надвишава сумата на приключилото плащане.

Пример за подобен отговор:

```
error: refund amount exceeds original amount
```

Всяка транзакция за връщане се регистрира в ECOMM системата с уникален ID параметър (REFUND_TRANS_ID).

Този REFUND_TRANS_ID параметър на изпълнената заявка трябва да се съхранява, заедно с останалите параметри от отговора, получен от ECOMM.

Търговецът може да проверява резултатът от транзакция за връщане на сума чрез (-с) заявката за получаване на резултат, която се използва за проверка на транзакциите за плащане.

3.4 Анулиране на картова транзакция (reversal)

Интерфейсът за комуникация между Търговец и ECOMM осигурява възможност за изпълнение на заявка за анулиране на приключило плащане. Тази заявка може да бъде изпълнена само за успешно приключило картово плащане (RESULT: OK). Заявката за анулиране може да бъде изпълнена за цялата сума на приключилото плащане или за част от нея. Тази заявка може да бъде изпълнявана при следните ограничения: за дадено успешно плащане може да бъде изпълнена само една заявка за анулиране (пълно или частично); заявката за анулиране може да бъде изпълнена до 23:59:59ч. в същият ден, в който е била изпълнена оригиналната транзакция.

Заявката има следният формат:

-r командата, идентифицираща заявка за анулиране (Reversal)
trans_id идентификаторът на транзакцията, за която се изпълнява анулиране (задължително поле; 28 символа)
amount сума за анулиране (без десетичен знак; до 12 символа дължина) – опционално поле. При заявка за анулиране без изпращане на "amount" параметър се изпълнява анулиране на цялата сума на приключилото плащане.

Изпълнение:

Изпълнение на заявката от команден ред чрез извикване на "ecommerce_merchant.jar" архива:

```
java -jar ecommerce_merchant.jar merchant.properties -r bAt6JLX52DUbibbzD9gDF15Ppr4=13658
```

Изпълнение на заявката чрез извикване на съответният метод от `lv.tietoenator.cs.ecomm.merchant.Merchant` класа:

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

При изпълнение на заявката чрез reverse метода, не трябва да се подава (-r) командата.

Изпълнение на заявката чрез HTTP POST

```
command=r&trans_id=<trans_id>&amount=<amount>(&<property_name>=<property_value>)
```

Препоръчва се при изпълнение на заявка за анулиране да се изпраща допълнителен description параметър, в който още веднъж да се подава trans_id на плащането, за което се прави анулиране.

```
command=r&trans_id=<trans_id>&amount=<amount>(&<description>=<Reversal for trans_id>)
```

```
command=r&trans_id=bAt6JLX52DUbibbzD9gDF15Ppr4&amount=5697(&description=Reversal for  
bAt6JLX52DUbibbzD9gDF15Ppr4)
```

Резултат:

При изпълнение на тази заявка, ECOMM връща отговор с резултат от това изпълнение:

```
RESULT: <резултат от картовото плащане>
```

```
RESULT_CODE: <код за резултата от картовото плащане>
```

Примерен отговор от ECOMM с резултат от заявка за анулиране:

```
RESULT: OK
```

```
RESULT_CODE: 400
```

Параметърът "RESULT" може да има следните стойности:

OK – успешно анулиране

FAILED – неуспешно анулиране

REVERSED – за тази транзакция, вече е била успешно изпълнена заявка за анулиране

При неуспешна обработка на заявката за връщане ECOMM изпраща отговор със string, който започва с "error".

4 ВГРАЖДАНЕ НА ECOMM КЛИЕНТ В ТЪРГОВСКИЯ САЙТ

В пакета “Fibank_integration_package.zip” се намират двете версии на ECOMM клиента (ecommm_merchant.jar и ecomm_merchant.dll).

- Клиентът ecomm_merchant.jar се намира в директорията: “Fibank_integration_package\cs.bc.ecomm_merchant-1.1.59\jars” на пакета за интеграция.
- Клиентът ecomm_merchant.dll се намира в директорията “Fibank_integration_package\cs.bc.ecomm_merchant-1.1.59\dot_NET\ecommm_client” на пакета за интеграция.

Пакетът за интеграция съдържа примери за вграждане на двете версии на ECOMM клиентите (в съответните директории на двата клиента).

4.1 Пример за вграждане на ECOMM клиент в търговския сайт

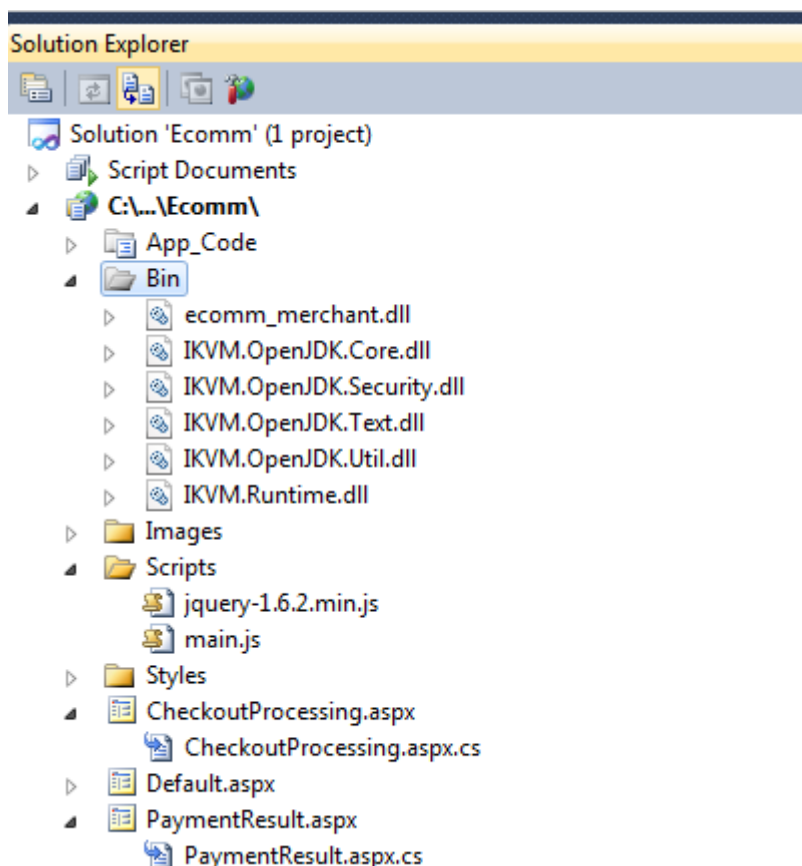
Приложеният пример показва вграждане на ecomm_merchant.dll клиента в търговски уеб сайт. Файловете от този пример могат да бъдат намерени в “Fibank_integration_package\cs.bc.ecomm_merchant-1.1.59\dot_NET\example_code” директорията на пакета за интеграция. В този пример, търговският уеб-сайт се състои от две страници: страница за потвърждение на направената поръчка (CheckoutProcessing.aspx) и страницата, към която ECOMM връща потребителя след приключване на картовото плащане (PaymentResult.aspx). Кодът, който се изпълнява от търговския сървър се съдържа във файловете CheckoutProcessing.aspx.cs и PaymentResult.aspx.cs.

За извикването на CheckoutProcessing.aspx.cs и за пренасочване на потребителя към платежната страница в ECOMM се използва JavaScript (javascript_ecomm project.txt). Файлът

BullionParams.cs служи за подаване на пътя до конфигурационният merchant.properties файл.


Конфигурационният merchant.properties файл съдържа параметри, свързани с комуникационната сесия с ECOMM: bank.server.url; keystore file; keystore type; keystore password. Тези файлове

са поставени в съответната работна директория на Търговеца.




Фиг. 4.1.1 Работна директория на Търговеца

Страницата CheckoutProcessing.aspx съдържа потребителска поръчка, готова за плащане.



ВАШАТА ПОРЪЧКА

	ПРОДУКТ	ЦЕНА	КОЛИЧЕСТВО	ОБЩО
	Сребърно ключче 1oz	76.00 BGN	1 Бройки	76.00 BGN
			Цена на продуктите:	76.00 BGN
			Цена за доставка:	5.15 BGN
			Обща сума:	81.15 BGN

[Продължи](#)

Фиг. 4.1.2 Файл CheckoutProcessing.aspx, зареден в клиентския браузър

При натискане на бутон „Продължи”, чрез JavaScript се извиква файла CheckoutProcessing.aspx.cs. Този файл съдържа логика за: изпращане на заявка за регистрация на транзакция към ECOMM; посрещане на отговора от ECOMM, съдържащ TRANSACTION_ID параметъра; конвертиране на TRANSACTION_ID параметъра чрез UriEncode и пренасочване на потребителя към платежната страница на банковата система (параметрите на заявката са със статични стойности: 1 лев, ip 10.14.112.20, описание - Purchase and delivery of 3 item from Fibank Gold Shop).

CheckoutProcessing.aspx.cs:

```
using System;
using System.IO;
using System.Web;
using System.Diagnostics;
using Bullions;
using lv.tietoenator.cs.ecomm.merchant;

public partial class CheckoutProcessing : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string res = "";
        string trans_id_urlEncode = "";
        string trans_id_base64 = "";

        lv.tietoenator.cs.ecomm.merchant.Merchant merchant;

        try
        {
            merchant = new Merchant(BullionParams.GetPaymentMerchPropFilePath());

            //decimal totalPricePurchaseString =
            decimal.Parse(PurchasesBLController.GetTotalPricePurchase());

            //Params ( amount, curency code, client IP Address, text message)
            String result = merchant.startSMSTrans("100", "975", "10.14.112.20", "Purchase and
            delivery of 3 item from Fibank Gold Shop.");
            if (result.Length >= 28 && result.Substring(0, 14) == "TRANSACTION_ID")
            {
                trans_id_base64 = result.Substring(16, 28);
                trans_id_urlEncode = HttpUtility.UrlEncode(trans_id_base64);
            }
            else
            {
                throw new Exception(result);
            }
        }
    }
}
```



```
catch (Exception ex)
{
    res = "{\"response_status\": \"Transaction ID get error - \" + ex.Message + \"\"}";
    Response.Write(res);
    Response.End();
}

if (trans_id_urlEncode != "")
{
    try
    {
        //purchase.payment_transaction_id = trans_id_base64;
        string url = BullionParams.GetPaymentClientHendlerUrl() + trans_id_urlEncode;
        res = "{\"response_status\": \"REDIRECT\", \"url\": \"\" + url + \"\"}";
    }
    catch (Exception exc)
    {
        res = "{\"response_status\": \"Service error(DB) - \" + exc.Message + \"\"}";
    }
    Response.Write(res);
    Response.End();
}
else
{
    res = "{\"response_status\": \"Service error(PAY)\"}";
    Response.Write(res);
    Response.End();
}
}
```

След като, потребителят завърши своето картово плащане в банковата система, ECOMM го пренасочва към PaymentResult.aspx страницата в търговския веб-сайт. Файлът PaymentResult.aspx.cs съдържа логика за: посрещане на заявката от ECOMM, с която потребителят се пренасочва обратно към търговската система; изпращане на заявка към ECOMM за получаване на резултат от картовото плащане; посрещане на отговора от ECOMM за резултат от картовото плащане.

PaymentResult.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Diagnostics;
using System.Xml;
using System.Xml.Xsl;
using lv.tietoenator.cs.ecomm.merchant;
using Bullions;

public partial class PaymentResult : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string trans_id = "";
        string res = "";

        string payment_result = "";
        string payment_result_code = "";
        string payment_retrieval_ref_code = "";
        string payment_approval_code = "";
        string payment_card_number = "";

        if (Request.Params.Get("trans_id") != null)
            trans_id = Request.Params.Get("trans_id").ToString();

        if (trans_id != "")
        {
            lv.tietoenator.cs.ecomm.merchant.Merchant merchant;

            try
            {
                merchant = new Merchant(BullionParams.GetPaymentMerchPropFilePath());
                string result = merchant.getTransResult(trans_id,
BullionParams.GetPaymentIpAddress());
                result = result.Replace('\n', ' ');

                string[] resultArray = result.Split(' ');
                Dictionary<string, string> resultDict = new Dictionary<string, string>();
                for (int i = 0; i < resultArray.Length; i += 2)
                {
                    if (resultArray[i].Trim() != "" && resultArray[i + 1].Trim() != "")
                        resultDict.Add(resultArray[i].Replace(":", ""), resultArray[i + 1]);
                }
            }
        }
    }
}
```

```
        string temp = "";
        resultDict.TryGetValue("RESULT", out temp);
        payment_result = temp;

        temp = "";
        resultDict.TryGetValue("RESULT_CODE", out temp);
        payment_result_code = temp;

        temp = "";
        resultDict.TryGetValue("RRN", out temp);
        payment_retrieval_ref_code = temp;

        temp = "";
        resultDict.TryGetValue("APPROVAL_CODE", out temp);
        payment_approval_code = temp;

        temp = "";
        resultDict.TryGetValue("CARD_NUMBER", out temp);
        payment_card_number = temp;
    }
    catch (Exception exc)
    {
        res = "{\"response_status\": \"Purchase error(" + exc.Message + ")\"}";
    }
}
else //Payment get info error.
{
    res = "{\"response_status\": \"Missing transaction ID\"}";
}

Response.Write(res);
}
}
```

5 Изграждане на връзка с банковата система без използване на ECOMM клиентите

Търговецът може да осъществява комуникация с ECOMM системата и без да бъдат използвани ECOMM клиентите от пакета за интеграция. Подобна комуникация може да бъде осъществена чрез изпращане на директни HTTP POST заявки по TLS сесия към ECOMM. За договаряне на тази TLS сесия и за идентификацията на Търговеца се използват сертификатите от keystore файла, предоставен от Банката. При такава комуникация с ECOMM, Търговецът трябва да използва в своята програмна среда - приложение (модул), което може да работи със сертификати и може да изпраща POST заявки по TLS сесия. Параметрите, които Търговецът трябва да изпраща към ECOMM чрез директна HTTP POST заявка са описани в точка 3. на настоящата инструкция.

Сертификатът за комуникация на Търговец с банковата система се предоставя от Банката, като Java keystore файл. В случай, че Търговецът не използва ECOMM клиентите, за да комуникира с банковата система, а използва директни HTTP POST заявки чрез съответно приложение (модул), Java keystore файлът трябва да бъде конвертиран в подходящ формат за съответното приложение.

5.1 Пример за комуникация с ECOMM от PHP среда

Приложеният пример показва възможно решение за изпращане на директни HTTP POST заявки към ECOMM от PHP сайт, без да бъдат използвани ECOMM клиентите. Заявката се изпълнява чрез PHP libcurl модул. Файловете с примерната реализация се намират в папка „php_example” (Fibank_integration_package\cs.bc.ecomm_merchant-1.1.59\PHP\php_example). Примерът е реализиран в PHP 5.6.14. Параметрите, които трябва да бъдат изпращани към ECOMM при директна HTTP POST заявка са посочени в точки 3.1 – 3.4 от настоящата инструкция. Подготовката за реализация на това решение включва конвертиране на keystore файла със сертификати в .pem файл. За изпълнението на това конвертиране са използвани софтуерните приложения keytool и openssl при следният подход:

1. Конвертиране на keystore файла от JKS формат в PKCS12 формат чрез keytool.

```
keytool -importkeystore -srckeystore imakstore -destkeystore imakstore.p12 -srcstoretype jks -deststoretype pkcs12 -srcstorepass XXXXXX -deststorepass XXXXXX -srcalias ima -destalias ima
```

2. Конвертиране на p.12 файла в .pem файл чрез openssl

```
openssl pkcs12 -in imakstore.p12 -out imakstore.pem -passin pass:XXXXXX -passout pass:XXXXXX
```

При изпращане на POST заявки към системата на Банката, libcurl модулът трябва да бъде настроен да не проверява сертификатът на банковият уеб сървър.

```
<?php
$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $prm_data);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($ch, CURLOPT_SSLCERT, dirname(__FILE__) . '/imakstore.pem');
curl_setopt($ch, CURLOPT_SSLCERTPASSWD, 'IMAKSTORE PASSWORD');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
// error debugging
/*$fileHandle = fopen(dirname(__FILE__) . "/error.txt", "w");
curl_setopt($ch, CURLOPT_VERBOSE, true);
curl_setopt($ch, CURLOPT_STDERR, $fileHandle);*/

// grab URL and pass it to the browser
$response = curl_exec($ch);
// close cURL resource, and free up system resources
curl_close($ch);
return $response;
}
?>
```

6 Провеждане на тестове

Търговецът може да осъществи необходимата проверка на изградената връзка с ECOMM чрез извършване на тестови транзакции. След приключване на тестовете от страна на Търговеца, Банката проверява изградената връзка от своя страна чрез съответните тестови процедури. Тестовите процедури обхващат, както проверка на изградената връзка с ECOMM, така и проверка на търговския уеб-сайт за спазване на всички изисквания на Банката към него (изискванията са описани в документа “ Instruction 3D_2010_ECOMM.pdf”). Всички тестови процедури трябва да приключат успешно, за да бъде разрешено на Търговеца да изгради връзка с реалната ECOMM система на Банката.

6.1 Тестови карти

За провеждане на тестове от страна на Търговеца, Банката осигурява съответните тестови карти. Карти се предоставят на Търговеца, след като той изгради връзка с ECOMM (изпълни успешна заявка за регистрация на транзакция и получи TRANSACTION_ID за нея от банковата система), според следната процедура:

- 1 След изпълнение на успешна заявка за регистрация на транзакция и отговор от ECOMM с TRANSACTION_ID, Търговецът уведомява Банката за изградената връзка.
- 2 Банката прави проверка на съответната заявка и изпраща на Търговеца – тестови карти и инструкции за симулиране на успешни и неуспешни транзакции с тях.

6.2 Последващи тестове

При необходимост от последващи тестове, след като Търговецът вече е започнал да работи с реалната ECOMM система (надграждане на търговската система, отстраняване на грешки в работата на търговската система, нов проект и др.), Банката може да предостави нов достъп до тестовата ECOMM система.