

E-commerce solution

RSB ECOMM Integrated Merchant Agent (IMA)

Инструкция интеграции



Содержание

1 ВВЕДЕНИЕ	3
1.1 Назначение документа	3
1.2 Аудитория	3
1.3 Использование документа	3
2 ПОДГОТОВКА К РАБОТЕ	4
2.1 Системные требования	4
2.1.1 Необходимое программное обеспечение для IMA	4
2.2 Установка системы	4
2.2.1 Установка продукта IMA	4
2.2.2 Дополнительные файлы класса Java для работы с IMA	4
2.2.3 Конфигурация продукта IMA	4
3 ИНТЕГРАЦИЯ РЕШЕНИЯ	6
3.1 Общая функциональная схема	6
3.2 Интеграция	7
3.2.1 Регистрация транзакций	8
3.2.2 Регистрация DMS авторизации	9
3.2.3 Выполнение DMS транзакции	10
3.2.4 Результат транзакции	11
3.2.5 Отмена транзакции	13
3.2.6 Возврат суммы транзакции	14
3.2.7 Завершение бизнес - дня	15
3.2.8 Регулярные платежи	16
3.2.8.1 Регистрация	16
3.2.9 Выполнение	17
3.2.10 Применение параметра Properties	18
3.2.11 Переадресация клиента	19
3.2.12 Создание шаблона для платежной страницы (cardinfo.html)	20
3.2.13 Создание шаблона - confirm_fee.html	20
3.2.14 Функциональность поддержки многих языков	21
3.2.14.1 Перевод сообщений об ошибках javascript в окне ввода карт	21
3.2.15 Функциональность регулярных платежей	22
3.2.15.1 Регистрация и первый платеж	22
3.2.15.2 Регулярные платежи	23
3.3 Дополнительная информация	23
3.3.1 Помощь в отслеживании транзакций	23
3.3.2 Примечания относительно браузера Платежного сервера (Payment Server)	23
3.3.3 Примечания относительно информации о маршруте полета (Airline itinerary)	23



1 Введение

1.1 Назначение документа

Документ содержит описание этапов инсталляции и запуска системы Integrated Merchant Agent (IMA).

Система IMA предназначена для интеграции торгово-сервисных предприятий (ТСП) к системе ECOMM 3D-Secure для обработки транзакций, источником которой является Интернет.

1.2 Аудитория

Документ предназначен для технических специалистов ТСП.

1.3 Использование документа

Настоящий документ соответствует следующим модулям программного обеспечения:

- Tieto Ecomm
- ECOMM Integrated Merchant Agent (IMA).



2 Подготовка к работе

2.1 Системные требования

2.1.1 Необходимое программное обеспечение для IMA

КОМПОНЕНТ	ВЕРСИЯ	ПРИМЕЧАНИЯ
Sun Java Runtime Environment (JRE)	1.4. или выше	Устанавливается согласно http://www.oracle.com/technetwork/java/index.html

Если технологически невозможно использовать IMA в работе, необходимо использовать HTTPS POST запросы.

2.2 Установка системы

2.2.1 Установка продукта IMA

Для того чтобы установить IMA, распакуйте архив «Example.zip» в рабочий каталог. В результате этого будет создан подкаталог «Example» и следующие каталоги:

- «examples» – различные примеры, которые могут пригодиться при интеграции IMA;
- «jars» – JAR файл для работы IMA (IMA находится в архиве «ecommm_merchant.jar»).

2.2.2 Дополнительные файлы класса Java для работы с IMA

Необходимы следующие файлы библиотек Java:

- jcert.jar;
- jnet.jar;
- jsse.jar.

Настоящие файлы обычно является частью стандартной инсталляции Java. Их необходимо поместить в один каталог с файлом «ecommm_merchant.jar».

2.2.3 Конфигурация продукта IMA

- 1 ТСП должно получить сгенерированный банком SSL сертификат (keystore) и пароль для доступа к SSL сертификату. Данный сертификат применяется для установки подключения к серверу ECOMM и идентификации торговца в системе ECOMM при запросе TRANSACTION_ID.

Для этого необходимо подготовить запрос на SSL сертификат в формате *.csr, где CN="Merchant ID". Merchant ID, как правило присваивается после регистрации договора.

- 2 Необходимо изменить файл «Merchant.properties» (файл находится в каталоге «examples»):

`bank.server.url` – адрес для доступа к серверу ECOMM (необходимо получить от Банка)



`https.proxy.host` – адрес прокси-сервера HTTPS (необязателен)

`https.proxy.port` – порт прокси-сервера HTTPS (необязателен)

`https.handler` – библиотека для поддержки протокола HTTPS (необязательна) при работе IMA.

`https.cipher` – алгоритм шифрования подключения HTTPS (необязателен). Обычно применяется „SSL_RSA_WITH_RC4_128_MD5”.

`keystore.file` – полученный от Банка файл SSL сертификата (keystore)

`keystore.type` – тип файла SSL сертификата. Мы рекомендуем всегда использовать “JKS”.

`keystore.password` – пароль, полученный от Банка.

`ecomm.server.version` – если применяется версия сервера ECOMM 2. Если необходим возврат дополнительных деталей, характерных для сервера ECOMM 2, должно быть указано „2.0”. Если параметр не указан или равен „1.0”, дополнительные детали не возвращаются.

`connection.timeout` – время в секундах для подключения к серверу ECOMM. Заметьте, что параметр позволяет только уменьшить таймаут по сравнению со значением по умолчанию, зависящим от конкретной платформы (используется значение по умолчанию на стороне Банка).

- 3 Создать и отправить банку шаблон платежной страницы в формате HTML (`cardinfo.html`) и вспомогательные файлы для ввода информации о картах или используйте шаблон по умолчанию предоставленный Банком. Более подробное описание приводится в главе 3.2.12 *Создание шаблона для платежной страницы* (`cardinfo.html`) или в документе «Бизнес-функциональные требования на разработку платежных страниц для решения Tieto Ecomm».
- 4 Банк должен получить информацию относительно URL, применяемых для переадресации клиента обратно на сайт ТСП. Важно помнить, что для переадресации используется метод HTTPS POST. Это значит, что URL не может содержать параметров.

`returnOkUrl` – клиент будет переадресован на данный адрес после аутентификации 3D Secure и выполнения транзакции (независимо от результата).

`returnFailUrl` – клиент будет переадресован на данный адрес в случае технической неполадки в системе ECOMM.



3 Интеграция решения

3.1 Общая функциональная схема

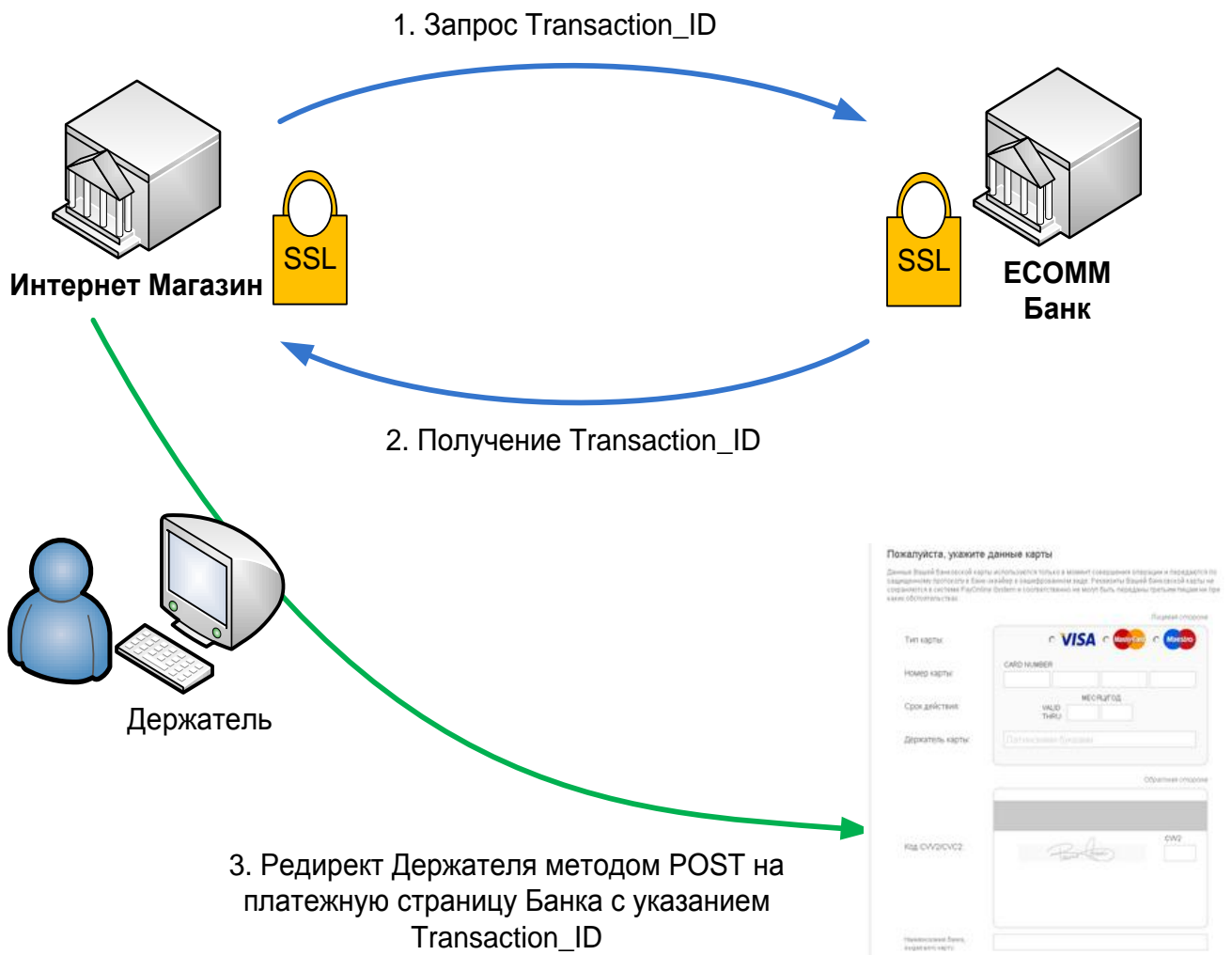
- 1 Клиент выбирает продукт и готов заплатить за покупку. При нажатии кнопки/ссылки «Checkout»/«Оплатить», контроль передается ТСП.
- 2 ТСП регистрирует транзакцию в системе ECOMM (указывая сумму, валюту, IP адрес клиента и краткое описание транзакции (необязательно)), в ответе получая идентификатор транзакции.
 - ТСП имеет возможность определить и отправить детали платежа. Описание данной возможности дано в документации к решению.
- 3 Клиент (с указанным идентификатором транзакции) переадресуется платежному серверу ECOMM для ввода карточных данных. Данные вводятся, используя предоставленный ТСП шаблон.
- 4 В некоторых случаях (когда применяется комиссия эквайер банка) дополнительный экран с информацией о комиссии показывается клиенту для подтверждения. Экран подтверждения использует предоставленный ТСП шаблон.
- 5 После ввода карточных данных, в зависимости от наличия поддержки 3D Secure, выполняются следующие операции:
 - ТСП поддерживает 3D Secure. После ввода данных карты, аутентификация клиента выполняется в рамках 3D Secure. Результат аутентификации передается системе ECOMM.
 - ТСП не поддерживает 3D Secure. В таком случае выполняется транзакция.
- 6 Транзакция выполняется в случае успешной аутентификации.
- 7 Клиент переадресуется на сайт ТСП (при указанном идентификаторе транзакции).
- 8 ТСП, при наличии идентификатора транзакции, получает информацию о результате транзакции от ECOMM (транзакция завершена или нет).
- 9 В случае DMS (Авторизация/Расчет/Authorisation/Captuer) транзакции, для платежа клиента необходима дополнительная транзакция – Авторизации, за которой следует Расчет. Такая транзакция типично выполняется по факту проверки наличия товара на складе, после доставки продукта клиенту, после проведения дополнительных проверок т.д. Следует отметить, что Расчет является основанием для перечисления денег со счета клиента (финансовая операция), в то время как Авторизация подразумевает лишь блокировку средств на счете клиента.
- В случае SMS (Покупка/Purchase) транзакции – единая транзакция, которая позволяет перевести деньги со счета держателя на счет ТСП. Это означает, что авторизация платежа и ее расчет осуществляются в рамках одной транзакции.
- 10 При необходимости ТСП может запросить, чтобы платежный сервер ECOMM отменил транзакцию (reversal). По завершению бизнес – дня, отмену произвести не удастся.
- 11 При необходимости можно вернуть сумму транзакции (refund) только в случае, если статус платежа FINISHED.
- 12 ТСП регулярно (раз в день) отправляет серверу ECOMM запрос на завершение бизнес-дня.
- 13 Платеж может одновременно использоваться и для регистрации регулярного платежа.



14 ТСП может выполнять регулярные платежи (см. пункт 3.2.14).

15 ТСП может просматривать свои транзакции через браузер Merchant Portal. Инструкция по настройке доступа к Payment Server Workplace.

Рисунок 3-1 – Функциональная схема



3.2 Интеграция

Возможен вызов платежного сервера ECOMM посредством IMA (находится в файле «jars» в архиве «ecommm_merchant.jar») различными способами.

- 1 Вызовом архива Java «ecommm_merchant.jar» из командной строки. Примеры показаны в каталоге «examples».
- 2 Непосредственным вызовом сервисных методов класса `lv.tietoenator.cs.ecomm.merchant.Merchant`. Классу `Merchant` необходимо присвоить название файла конфигурации при создании данного класса. Название файла позволяет инициировать IMA, а в случае возникновения ошибки возвращается `ConfigurationException`.



Пример:

```

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);

```

Дополнительные примеры можно найти в каталоге 'example'.

Вызов сервера ECOMM возможен и без применения Integrated Merchant Agent – посредством прямых вызовов http (используя SSL). Для передачи запроса со всеми необходимыми параметрами применяется метод POST.

3.2.1 Регистрация транзакций**Параметры командной строки:**

-v	идентифицирует запрос на регистрацию транзакции
amount	сумма транзакции в дробных единицах. Параметр обязателен (до 12 цифр)
currency	код валюты транзакции (ISO 4217), обязателен (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции, необязателен (до 125 знаков)
language	идентификатор языка авторизации, необязателен (до 32 знаков)

Вызов метода:**Параметры IMA запроса:**

```

public String
    startSMSTrans(String amount, String currency, String ip, String desc, String
language, Properties properties)

    // old methods for backward compatibility
    public String
        startSMSTrans(String amount, String currency, String ip, String desc, String
language)

    public String
        sendTransData(String amount, String currency, String ip, String desc, String
language)

```

Параметры Https post:


```
command=v&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&msg_type=SMS (&<property_name>=<property_value>)*
```

Результат:

```
TRANSACTION_ID: <trans_id>
```

trans_id идентификатор транзакции (28 знаков в кодировке base64)

В случае ошибки возвращаемая строка символов начинается с 'error:'.

Пример результата:

```
TRANSACTION_ID: bAt6JLX52DUbibbzD9gDF15Ppr4=
```

3.2.2 Регистрация DMS авторизации**Параметры командной строки:**

-a	идентифицирует запрос на регистрацию транзакции
amount	сумма транзакции в дробных единицах. Параметр обязателен (до 12 цифр)
currency	код валюты транзакции (ISO 4217), обязателен (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции, необязателен (до 125 знаков)
language	идентификатор языка авторизации, необязателен (до 32 знаков)

Вызов метода:**Параметры IMA запроса:**

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

    // old methods for backward compatibility
public String
    startDMSAuth(String amount, String currency, String ip, String desc, String
language)
    public String
    startDMSAuth(String amount, String currency, String ip, String desc, String
language)
```

Параметры Https post:

```
command=a&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&msg_type=DMS (&<property_name>=<property_value>)*
```

Результат:

```
TRANSACTION_ID: <trans_id>
```

trans_id идентификатор транзакции (28 знаков в кодировке base64)

В случае ошибки возвращаемая строка символов начинается с 'error:'

Пример результата:

```
TRANSACTION_ID: bAt6JLX52DUbibbzD9gDF15Ppr4=
```



3.2.3 Выполнение DMS транзакции

Параметры командной строки:

-t	идентифицирует запрос на регистрацию транзакции
auth_id	идентифицирует авторизацию финансовой транзакции
amount	сумма транзакции в дробных единицах. Параметр обязателен (до 12 цифр)
currency	код валюты транзакции (ISO 4217), обязателен (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции, параметр необязателен (до 125 знаков)

Вызов метода:

Параметры IMA запроса:

```
public String
    makeDMSTrans(String auth_id, String amount, String currency, String ip, String
desc, String language, Properties properties)

    // old methods for backward compatibility
    public String
        makeDMSTrans(String auth_id, String amount, String currency, String ip, String
desc, String language)

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

Параметры Https post:

```
command=t&trans_id=<auth_id>&amount=<amount>&currency=<currency>&client_ip_addr=<i
p>&description=<desc>&language=<language>&msg_type=DMS (&<property_name>=<property_
value>)*
```

Результат:

```
RESULT: <result>
RESULT_CODE: <result_code>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER <pan>
```

result – результаты транзакции: OK – транзакция выполнена успешно, FAILED – выполнение транзакции не состоялось

result_code – код результата транзакции, возвращенный от Card Suite Processing RTPS (3 цифры)

rrn – номер поисковой ссылки, возвращенный от Card Suite Processing RTPS (12 знаков)

app_code – код подтверждения, возвращенный от Card Suite Processing RTPS (максимум 6 знаков)

pan – замаскированный номер карты



Поля RESULT_CODE несут только информационный характер. Информационные поля RRN и APPROVAL_CODE отображаются только в случае успешных транзакций и применяются для более легкого отслеживания транзакций в системе Card Suite Processing RTPS. Основой для принятия решения об успешном или неудачном выполнении транзакции должно быть только значение поля RESULT.

В случае ошибки, возвращенная строка символов начинается с 'error:'.

В случае предупреждения, возвращенная строка символов начинается с 'warning:' (зарезервировано для использования в будущем).

Пример результата:

```
RESULT: OK
RESULT_CODE: 000
RRN: 123456789012
APPROVAL_CODE: 123456
CARD_NUMBER: 9*****9999
```

3.2.4 Результат транзакции

Параметры командной строки:

-c	идентифицирует запрос результата транзакции
trans_id	идентификатор транзакции, обязателен (28 знака)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)

Вызов метода:

Параметры IMA запроса:

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

Параметры Https post:

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

Результат:

```
RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER: <pan>
AAV: <aav>
RECC_PMNT_ID: <rcc_pmnt_id>
RECC_PMNT_EXPIRY: <rcc_pmnt_ex>
MRCH_TRANSACTION_ID: <mrch_tx_id>
```

result – статус результата транзакции:

OK – транзакция завершена успешно,



FAILED – транзакция не состоялась,

CREATED – транзакция зарегистрирована в системе,

PENDING – выполнение транзакции продолжается,

DECLINED – транзакция отклонена системой ECOMM, так как ECI находится в списке заблокированных ECI (конфигурация серверной части ECOMM),

REVERSED – транзакция возвращена,

AUTOREVERSED – транзакция возвращена автореверсалом,

TIMEOUT – транзакция просрочена

result_ps – результат транзакции в интерпретации Платежного сервера (Payment Server) (показывается только если сконфигурирован возврат специфичных данных ECOMM2 – см. параметр `ecomm.server.version` в главе 2.2.3):

FINISHED – платеж завершен успешно,

CANCELLED – платеж отменен,

RETURNED – платеж возвращен,

ACTIVE – платеж зарегистрирован, но не завершен.

result_code – код результата транзакции, возвращенный системой Card Suite Processing RTPS (3 цифры)

3dsecure – статус 3D Secure:

AUTHENTICATED – успешная авторизация 3D Secure

DECLINED – неуспешная авторизация 3D Secure

NOTPARTICIPATED – держатель карты не является участником схемы 3D Secure

NO_RANGE – карта не входит в интервал 3D secure карт указанных на стороне эмитента

ATTEMPTED – авторизация 3D secure через ACS attempts сервер

UNAVAILABLE – авторизация 3D secure не доступна

ERROR – сообщение об ошибке возвращено из ACS сервера

SYSERROR – 3D secure авторизация закончилась системной ошибкой

UNKNOWNSCHEME – попытка выполнить 3D secure авторизацию для неизвестного типа карты (Dinners club, American Express)

rrn – номер поисковой ссылки, возвращенный системой Card Suite Processing RTPS

app_code – код подтверждения, возвращенный системой Card Suite Processing RTPS (максимум 6 знаков)

rap – замаскированный номер карты

aav – результат проверки значения хеш-функции в AAV merchant name (если не успешно)

FAILED – значение хеш-функции не совпадает

rss_pmnt_id – идентификатор регулярного платежа (при наличии такового) в системе Payment Server.

rss_pmnt_ex – дата завершения действия регулярного платежа (при наличии такового) в Payment Server в формате ГГММ



mrch_tx_id – идентификатор платежа Merchant Transaction Identifier (если доступен) – предоставляется, если отправлен как дополнительный параметр под названием "mrch_transaction_id" при регистрации платежа.

Поля RESULT_CODE и 3DSECURE носят чисто информативный характер и могут не отображаться. Поля RRN и APPROVAL_CODE отображаются только для успешных транзакций, в целях информации, и облегчают отслеживание транзакций в системе Card Suite Processing RTPS. Основой для принятия решения об успешном или неудачном выполнении транзакции должно быть только значение поля RESULT.

В случае ошибки возвращаемая строка символов начинается с 'error:'.

В случае предупреждения возвращаемая строка символов начинается с 'warning:' (зарезервировано для использования в будущем).

Пример результата:

```
RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: ATTEMPTED
RRN: 123456789012
APPROVAL_CODE: 123456
CARD_NUMBER: 9*****9999
RECC_PMNT_ID: 1258
RECC_PMNT_EXPIRY: 1108
```

3.2.5 Отмена транзакции

Параметры командной строки:

- r идентифицирует запрос отката транзакции
- trans_id идентификатор транзакции, обязателен (28 знака)
- amount необязательный параметр – сумма отката в дробных единицах (до 12 знаков). Для DMS авторизации возможен откат только полной суммы авторизации, т.е. суммы авторизации и отката должны быть одинаковыми. В других случаях возможен также частичный откат.
- suspected_fraud необязательный параметр – флаг, указывающий, что откат делается из-за подозрения в мошенничестве. В таких случаях значение этого параметра должно быть установлено в "да". Если этот параметр используется, возможен откат только полной суммы.

Вызов метода:

Параметры IMA запроса:

```
public String
reverse(String trans_id)

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



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

Параметры Https post:

```
command=r&trans_id=<trans_id>&amount=<amount>
```

```
command=r&trans_id=<trans_id>&amount=<amount>&suspected_fraud=yes
```

Результат:

```
RESULT: <result>
```

```
RESULT_CODE: <result_code>
```

result – результат отката:

OK – успешная транзакция отката

REVERSED – транзакция прежде уже была возвращена

FAILED – неудачная транзакция отката (статус транзакции остается, как было)

result_code – код результата отката, возвращенный системой Card Suite Processing RTPS (3 цифр)

В случае ошибки возвращаемая строка символов начинается с 'error:'.

В случае предупреждения возвращаемая строка символов начинается с 'warning:' (зарезервировано для использования в будущем).

Пример результата:

```
RESULT: OK
```

```
RESULT_CODE: 400
```

3.2.6 Возврат суммы транзакции**Параметры командной строки:**

-k идентифицирует запрос возврата суммы транзакции

trans_id идентификатор транзакции, обязателен (28 знаков), всегда возвращается полная сумма оригинальной транзакции

Вызов метода:**Параметры IMA запроса:**

```
public String
refund(String trans_id)
```

Параметры Https post:

```
command=k&trans_id=<trans_id>
```

Результат:

```
RESULT: <result>
```

```
RESULT_CODE: <result_code>
```

result – результат возврата суммы:



OK	успешная транзакция возврата суммы
FAILED	неудачная транзакция возврата суммы

result_code – код результата возврата суммы, возвращенный системой Card Suite Processing RTPS (3 цифры)

refund_transaction_id – идентификатор транзакции возврата суммы – применим для получения деталей платежа возврата суммы или для запроса отката платежа возврата суммы.

В случае ошибки возвращаемая строка символов начинается с 'error:'.

В случае предупреждения возвращаемая строка символов начинается с 'warning:' (зарезервировано для использования в будущем).

Статус транзакции в платежном сервере после возврата суммы не меняется.

3.2.7 Завершение бизнес - дня

Бизнес-день завершается при проведении сверки итогов, затем формируется электронный журнал для указанного ТСП.

Параметры командной строки:

-b идентифицирует запрос завершения бизнес-дня

Вызов метода:

Параметры IMA запроса:

```
public String
closeDay()
```

Параметры Https post:

```
command=b
```

Результат:

```
RESULT: <result>
RESULT_CODE: <result_code>
FLD_075: <fld_075>
FLD_076: <fld_076>
FLD_087: <fld_087>
FLD_088: <fld_088>
```

result – результат завершения бизнес-дня:

OK	бизнес-день завершен успешно
FAILED	бизнес-день завершен неудачно

result_code – код завершения бизнес-дня, возвращенный системой Card Suite Processing RTPS (3 цифры)

fld_075 – количество кредитных реверсалов (до 10 цифр), отображается только если result_code начинается с 5

fld_076 – количество дебетовых реверсалов (до 10 цифр), отображается только если result_code начинается с 5



fld_087 – общая сумма кредитных реверсалов (до 16 цифр), отображается только если result_code начинается с 5

fld_088 – общая сумма дебетовых реверсалов (до 16 цифр), отображается только если result_code начинается с 5

В случае ошибки возвращаемая строка символов начинается с 'error:'

Пример результата:

```
RESULT: OK
RESULT_CODE: 500
FLD_075: 12
FLD_076: 31
FLD_087: 3201
FLD_088: 10099
```

3.2.8 Регулярные платежи

Регулярные платежи регистрируются, или одновременно с авторизацией на определённую сумму но без расчета транзакции, или при первом платеже, в обоих режимах - SMS и DMS. Описание функций обычных платежей дано в секции 3.2.14. Параметр *rec_pmnt_id* может отсутствовать в вызове JAVA. В таком случае TRANSACTION_ID становится идентификатором обычного платежа.

3.2.8.1 Регистрация

Параметры командной строки:

-z или -d	запрос SMS транзакции/регистрации DMS авторизации
amount	сумма транзакции в дробных единицах, обязателен (до 12 цифр)
currency	код валюты транзакции (ISO 4217), обязателен, (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции (до 125 знаков)
rec_pmnt_id	выбранный торговцем идентификатор регулярного платежа
expiry	предпочитаемый предельный срок действия регулярного платежа в формате ММГГ

Параметры командной строки для регистрации регулярного платежа с авторизацией но без первого платежа:

-p	запрос на регистрацию платежа
currency	код валюты транзакции (ISO 4217), обязателен, (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции (до 125 знаков)
rec_pmnt_id	выбранный торговцем идентификатор регулярного платежа
expiry	предпочитаемый предельный срок действия регулярного платежа в формате ММГГ

Вызов метода:




```

Параметры IMA запроса:public String startSMSTransRP(String amount, String
currency,
String ip, String desc, String language, String recc_pmnt_id, String expiry,
Properties properties)

public String startDMSAuthRP(String amount, String currency,
String ip, String desc, String language, String recc_pmnt_id, String expiry,
Properties properties)

public String registerRP(String currency, String ip, String desc, String language,
String recc_pmnt_id, String expiry, Properties properties)

```

Параметры Https post:

```

command=z&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<des
c>&language=<language>&msg_type=SMS&biller_client_id=<recc_pmnt_id>&perspayee_expi
ry=<expiry>&perspayee_gen=1(&<property_name>=<property_value>)*

command=d&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<des
c>&language=<language>&msg_type=DMS&biller_client_id=<recc_pmnt_id>&perspayee_expi
ry=<expiry>&perspayee_gen=1(&<property_name>=<property_value>)*

command=p&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<lan
guage>&msg_type=AUTH&biller_client_id=<recc_pmnt_id>&perspayee_expiry=<expiry>&per
spayee_gen=1(&<property_name>=<property_value>)*

```

Ответ на запрос регистрации аналогичен ответу в случае вызова *startSMSTrans*, *startDMSAuthRP* или *registerRP* за исключением следующих дополнительных полей:

RECC_PMNT_ID: rec_pmnt_id, если указан. Иначе – TRANSACTION_ID

RECC_PMNT_EXP min (дата завершения действия карты, параметры *expiry*).

Для DMS транзакции генерируются посредством того же вызова *makeDMSTrans*.

Перезапись существующих регулярных платежей:

Если в рамках шаблона уже указан регулярный платеж для данного клиента (карты), то этот платеж необходимо перезаписать. Перезапись регулярных платежей возможна с указанием дополнительного параметра *perspayee_overwrite=1*. В этом случае будут удалены все регулярные платежи для шаблона, установленного для данного клиента (карты).

В командной строке, после обязательных параметров, можно указать дополнительный параметр. При этом используется синтаксис:

```
--parameter_name=parameter_value,
```

который можно указать в вызове метода, в переменной свойств (*properties*).

В параметрах Https post этот параметр можно добавить как обычный параметр (как показано в ранее приведенных примерах).

3.2.9 Выполнение

Параметры командной строки:

-e запрос регистрации SMS транзакции



amount	сумма транзакции в дробных единицах, параметр обязателен (до 12 цифр)
currency	код валюты транзакции (ISO 4217), обязателен, (3 цифры)
client_ip_addr	IP адрес клиента, обязателен (15 знаков)
description	детали транзакции (до 125 знаков)
recc_pmnt_id	выбранный торговцем идентификатор регулярного платежа

Вызов метода:**Параметры IMA запроса:**

```
public String makeRP(String recc_pmnt_id, String amount, String currency,
String ip, String desc, Properties properties)
```

Параметры Https post:

```
command=e&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&biller_client_id=<recc_pmnt_id>(&<property_name>=<property_value>)*
```

Результат:

```
TRANSACTION_ID: <trans_id>
RESULT: <result>
RESULT_CODE:<result_code>
RRN:<rrn>
APPROVAL_CODE:<appr_code>
```

trans_id – идентификатор транзакции (28 знака в кодировке base64)

result – результаты транзакции: OK – транзакция выполнена успешно, FAILED – выполнение транзакции не состоялось

В случае регулярного платежа коды ответа (result_code) имеют дополнительное значение:

108 – Торговцу необходимо связаться с клиентом;

114 – Торговец может повторить транзакцию позднее;

180 – Клиент отказался от дальнейшего сотрудничества. Регулярный платеж был удален;

2xx – Регулярный платеж был удален.

В случае ошибки, возвращенная строка символов начинается с 'error:'.

3.2.10 Применение параметра Properties

Параметр Properties используется для указания дополнительных деталей для Payment Server. Применение параметра зависит от решения и объясняется в соответствующей документации. Одним из возможных применений является указание деталей маршрута авиационного билета.

Параметры *Properties* можно указать в командной строке:

```
<Ecomm command line call> --<property_name1>=<property_value1> --
<property_name2=<property_value2> ...
```



3.2.11 Переадресация клиента

Возможна переадресация клиента (для ввода данных карты) на указанный URL (необходимо получить от банка), посредством метода GET или POST запроса. Важно, чтобы во время переадресации передавалась переменная `trans_id`. Эта переменная содержит идентификатор подлежащей к оплате транзакции (необходимо обратить внимание на то, что `trans_id` содержит знаки '+', '=' и '/', которые перед отправлением идентификатора необходимо заменить на кодировку URL (например, заменить '=' на группу '%3D'). Если Вы работаете с IMA, это возможно посредством метода `URLEncoder.encode`). Во время переадресации возможна передача дополнительных параметров, которые будут возвращены ТСП в момент переадресации клиента на страницу сайта ТСП.

Кодировка URL:

!	"	#	%	&	'	*	,	:	;	<	=	>	?	[]	^	`	{		}	<пробел>
%21	%22	%23	%25	%26	%27	%2a	%2c	%3a	%3b	%3c	%3d	%3e	%3f	%5b	%5d	%5e	%60	%7b	%7c	%7d	%20

Пример применения метода POST посредством JavaScript приводится в каталоге «examples» → «client_to_ecomm.html»:

```

<!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>

```

Строку символов `%%post_url%` в примере необходимо заменить на предоставленный банком URL сервера ECOMM, а `%%trans_id%` – на идентификатор транзакции.



3.2.12 Создание шаблона для платежной страницы (cardinfo.html)

Карточные данные вводятся в динамично сгенерированный шаблон HTML, дизайн которого можно изменить согласно нуждам ТСП. Пример шаблона приводится в «examples» → «cardinfo.html».

ECOMM сервер распознает следующие строки символов в данном шаблоне:

%%javascript%%	Заменяется на JavaScript для проверки полей ввода. Если ТСП сам желает использовать данную функциональную возможность, эта строка может отсутствовать.
%%formdef%%	Заменяется на <form id="cardentry" action=<url> method="post"> <input type="hidden" name="trans_id" value="<trans_id>" readonly> <input type="hidden" name="count" value="X" readonly> где X является номером попытки отправления формы
%%cardname%%	<input type="text" name="cardname" size="19" maxlength="100">
%%cardnr%%	<input type="text" name="cardnr" size="19" maxlength="19">
%%expmonth%%	<input type="text" name="validMONTH" size="2" maxlength="2">
%%expmonth_select%%	<select name="validMONTH"><option value="01">01</option>...</select>
%%expyear%%	<input type="text" name="validYEAR" size="2" maxlength="2">
%%expyear_select%%	<select name="validYEAR"><option value="08">08</option>...</select>
%%cvc2%%	<input type="text" name="cvc2" size="4" maxlength="4">
%%cvc2_password%%	<input type="password" name="cvc2" size="4" maxlength="4">
%%amount%%	сумма транзакции
%%ccyalpha%%	валюта транзакции
%%description%%	детали транзакции, переданные торговцем на ECOMM сервер

3.2.13 Создание шаблона - confirm_fee.html

В некоторых случаях эквайер банка может брать комиссию - в таких случаях держатель карты должен быть информирован о комиссии. Это делается, показывая экран подтверждения комиссии после ввода данных карты. Экран подтверждения комиссии содержит всю необходимую информацию о комиссии и обеспечивает владельца карты с возможностью продолжить платёж или отменить его.

Экран подтверждения комиссии необходим (и будет показан держателю карты) только если будет применена комиссия.

Экран подтверждения комиссии реализуется с помощью динамически генерируемого HTML-шаблона, дизайн которого можно изменить согласно нуждам ТСП. Пример шаблона приводится в «exarmples» → «confirm_fee.html».

ECOMM сервер распознает следующие строки символов в данном шаблоне:

%%formdef%%	Заменяется на <form id="cardentry" action=<url> method="post"> <input type="hidden" name="trans_id" value="<trans_id>" readonly> <input type="hidden" name="commission_confirmed" value="1" readonly>
%%cancel_link%%	Заменяется на URL для отмены оплаты
%%cardname%%	Имя владельца карты (ранее введенный в экран для ввода данных карты)
%%cardnr%%	Последние 4 цифры номера карты (ранее введенный в экран для ввода данных карты)
%%amount%%	сумма транзакции
%%amount_fee%%	сумма комиссии
%%amount_total%%	общая сумма (сумма транзакции + сумма комиссии)
%%ccyalpha%%	валюта транзакции
%%description%%	детали транзакции, переданные ТСП на ECOMM сервер



3.2.14 Функциональность поддержки многих языков

ECOMM поддерживает многоязыковой пользовательский интерфейс. Создано множество шаблонов. Язык каждого набора соответствует выбору пользователя.

Для выбора необходимого языка в пользовательском интерфейсе, для клиента необходимо создать упомянутые HTML шаблоны и отправить их персоналу обслуживания ECOMM сервера, указав идентификатор языка для каждого набора шаблонов. На сервере каждому языку может соответствовать один идентификатор, максимальная длина которого равна 32 ASCII-символам. Идентификатор языка может содержать строчные буквы, цифры и знак подчеркивания (_) в кодировке ASCII.

Если ECOMM сервер не содержит языковые установки для выбранного ТСП (не установлены языковые шаблоны), для пользовательского интерфейса применяются указанные по умолчанию шаблоны.

Если IMA вызывается из командной строки, то кроме параметра языка необходимо указать параметры описания. Параметры описания и идентификатор языка являются необязательными параметрами транзакции. Если нет нужды в описании, вместо описания используется строка пробелов ("").

Функциональность различных языков внедрена только для интерфейса конечного пользователя. Для операций завершения бизнес-дня и реверсалов нет языковых установок.

Локализация отчета об ошибках осуществляется посредством файлов locale.properties. Файл под таким названием должен присутствовать в каждом каталоге шаблонов, отчеты, об ошибках которого требуется локализовать. По умолчанию применяются следующие отчеты об ошибках серверной части:

```
invalid_cardnr=Invalid card number
invalid_cardnr_length=Card number length should be in range 13 - 19
invalid_cardname=Card name length should be in range 1 - 100
invalid_exp_year=Expiration year should be in range 01 - 99
invalid_exp_month=Expiration month should be in range 01 - 12
invalid_cvc2=Invalid CVC2 value
```

3.2.14.1 Перевод сообщений об ошибках javascript в окне ввода карт

В файле cardinfo.html, перед тегом %<script language="JavaScript">
strLocale = new Object();
strLocale_loc.enter_cvc = 'Enter CVC2/CVV2 or CID code!';
strLocale_loc.enter_cid = 'Enter Amex CID code!';
strLocale_loc.enter_cvc2 = 'Enter CVC2/CVV2 code!';
strLocale_loc.enter_name = 'Enter name, surname!';
strLocale_loc.enter_cardnr = 'Enter card number!';
strLocale_loc.enter_expiry = 'Enter card expiry date!';
</script>



Если strLocale не указан, то сообщения будут на английском языке, как и в данном примере.

Для локализации сообщений об ошибках JavaScript можно использовать и файл локализации под названием locale.properties, содержащий описание необходимых переводов. По умолчанию применяются следующие сообщения об ошибках клиентской части:

```
enter_cardnr=Enter card number
enter_expiry=Enter card expiry date
enter_cardname=Enter name, surname
enter_cvc=Enter CVC2/CVV2 or CID code
enter_cid=Enter Amex CID code
enter_cvc2=Enter CVC2/CVV2 code
```

3.2.15 Функциональность регулярных платежей

3.2.15.1 Регистрация и первый платеж

- 1 Держатель карты подключается к серверу ТСП для выполнения регулярной услуги:
 - Следуя требованиям международных платежных систем, держатель карты указывает свои контакты карты, регулярность платежа и дату завершения действия договора;
 - Регистрация совершается при первом платеже, поэтому надо обязательно указать сумму платежа (> 0).
- 2 ТСП инициирует регистрацию регулярного платежа (Regular Payment (RP)) посредством модуля E-comm agent и выполняет переадресацию браузера держателя карты на страницу платежей ECOMM.
- 3 Если необходимо, выполняется аутентификация держателя карты. Для этого применяется инфраструктура 3D Secure.
- 4 При положительном ответе ECOMM вызывает Платежный сервер (Payment Server (PS)) и регистрирует регулярный платеж (RP).
- 5 ТСП присваивает RP определенный идентификатор (например, номер договора, уникальный в записях самого ТСП) для незамедлительной идентификации определенного шаблона для регулярных платежей. Если параметр (recc_pmnt_id) не установлен, то применяется значение trans_id.
- 6 Если необходимо, ТСП может указать детали платежа (заполнить поля параметров *Properties*, т.е. mrch_transaction_id).
- 7 Атрибуты карты хранятся на сервере Платежного сервера вместе с описанием RP – для создания будущих платежей.
- 8 Результаты успешных и неудачных вызовов передаются в ECOMM и ТСП.
- 9 PS отправляет авторизацию (возможны оба режима – SMS и DMS) в платежную сеть, на сумму первого платежа.
- 10 Для каждой карты можно зарегистрировать неограниченное количество договоров RP (шаблонов). Если создан новый шаблон RP и присвоен существующий recc_pmnt_id, то такой шаблон регистрируется вместо ранее созданного шаблона.



3.2.15.2 Регулярные платежи

- 1 ТПС (ECOMM Merchant Agent) инициирует каждую последующую транзакцию RP:
 - ТСП применяет присвоенные идентификатор договора RP (*гесс_pmnt_id*) и сумму транзакции;
 - ECOMM идентифицирует транзакцию по идентификатору договора RP и отправляет транзакцию в PS;
 - PS использует полученный *гесс_pmnt_id*, находит атрибуты карты и выполняет процесс авторизации (это влияет на регулярные платежи только в режиме SMS) как для транзакции MoTo.
 - Ответ ТСП отсылается посредством ECOMM.
- 2 Регулярные платежи имеют место до наступления одного из нижеуказанных событий (после чего шаблон автоматически удаляется):
 - Завершился срок действия шаблона RP;
 - Полученный код ответа (*response_code*) регулярного платежа - 180 или 2xx.

3.3 Дополнительная информация

3.3.1 Помощь в отслеживании транзакций

После заполнения любого поля параметра *properties* можно осуществлять поиск (в браузере Платежного сервера) и следить за статусом транзакции.

Детали платежа - *payee* и *biller* – передаются системе обработки карт, в полях ISO FLD_098 и FLD_104.

3.3.2 Примечания относительно браузера Платежного сервера (Payment Server)

Для каждого платежа Payment Server может выполнить команды *Finish payment*, *Cancel payment* и *Return payment*. Однако решение *есomm* имеет следующие ограничения:

- Активные платежи DMS (*status = Active*) можно отменить посредством Payment Server, после выбора опции *Cancel payment*. Сообщение реверсала генерируется как при выборе *Reverse*.
- Активные платежи DMS можно завершить выбором опции *Finish Payment*.
- Суммы завершенных платежей можно вернуть выбором опции *Return Payment*.

Причина отказа платежа (Action code) содержится в *auth_action_code*.

3.3.3 Примечания относительно информации о маршруте полета (Airline itinerary)

При покупке авиабилета, после заполнении деталей параметра *Properties*, Платежный сервер создает соответствующее сообщение *Airline itinerary*.

НАЗВАНИЕ ПОЛЯ	ФОРМАТ
<i>passenger_name</i>	ans 20
<i>departure_date</i>	n 8
<i>origination_city_airport</i>	an 3



НАЗВАНИЕ ПОЛЯ	ФОРМАТ
carrier_code1	an 2
service_class1	an 1
stop_over_code1	an 1
destination_city_airport1	an 3
carrier_code2	an 2
service_class2	an 1
stop_over_code2	an 1
destination_city_airport2	an 3
carrier_code3	an 2
service_class3	an 1
stop_over_code3	an 1
destination_city_airport3	an 3
carrier_code4	an 2
service_class4	an 1
stop_over_code4	an 1
destination_city_airport4	an 3
travel_agency_code	ans 8
travel_agency_name	ans 25
ticket_number	n 14
restriction_sign	a 1

Сокращения:

- n – номер,
- an – буквенно-цифровая строка,
- ans – буквенно-цифровая строка, дополненная пробелами до необходимой длины.

