



## **РУКОВОДСТВО ПО ИНТЕГРАЦИИ ДЛЯ ТОРГОВО-СЕРВИСНЫХ ПРЕДПРИЯТИЙ**

Версия: 1.18  
Выпуск: 20.03.2014

«CNP Processing GmbH»  
Республика Казахстан  
Г. Алматы, ул. Шевченко 118, офис 331  
Тел./Факс: +7 727 244 0477

---

---

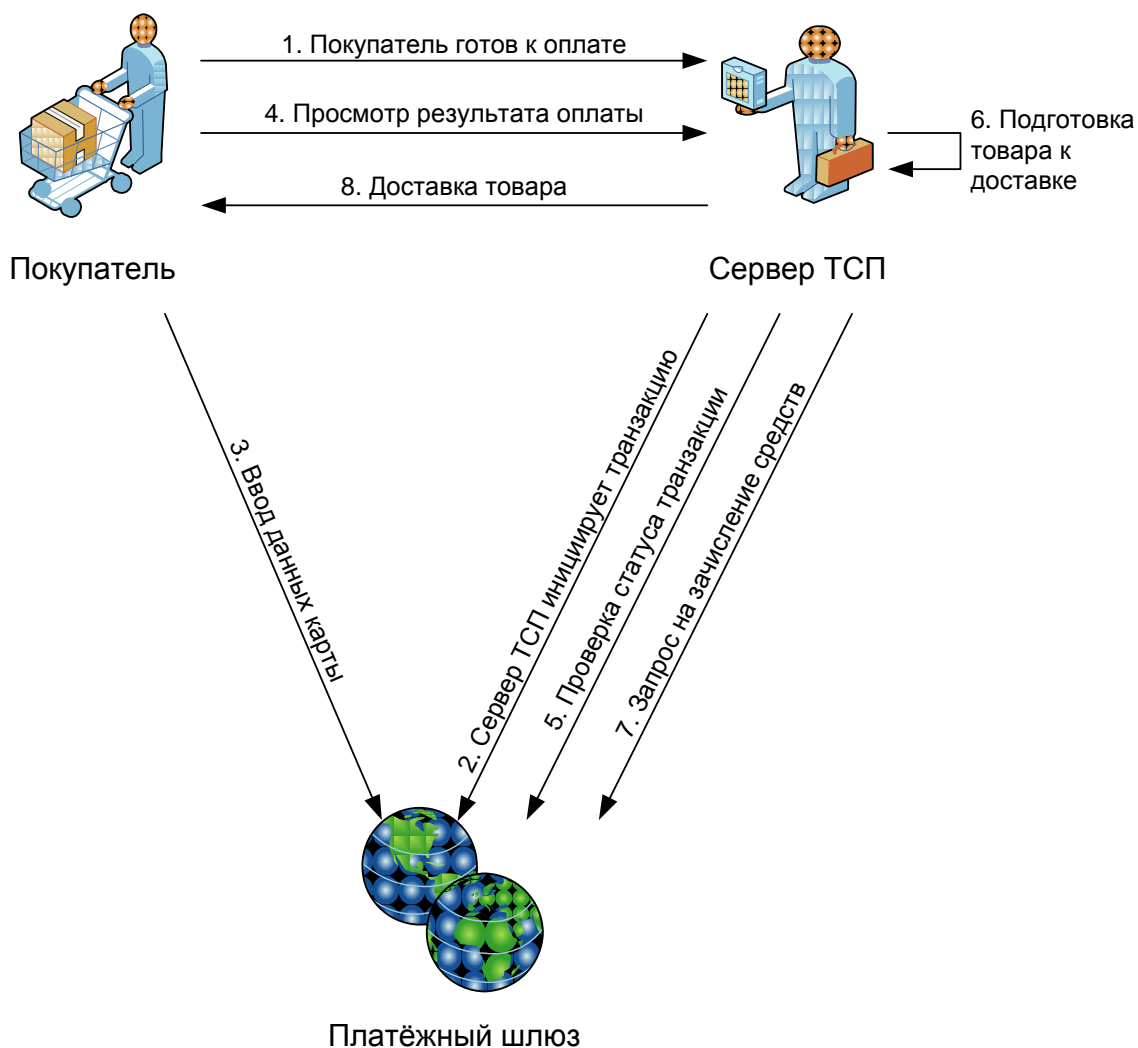
## Содержание

<b>СОДЕРЖАНИЕ .....</b>	<b>2</b>
<b>1 ВВЕДЕНИЕ.....</b>	<b>1</b>
<b>2 ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ (API) ДЛЯ ТОРГОВО-СЕРВИСНОГО ПРЕДПРИЯТИЯ (MERCHANT) .....</b>	<b>2</b>
2.1 STARTTRANSACTION (НАЧАТЬ ТРАНЗАКЦИЮ) .....	2
2.1.1 Запрос (Request) .....	2
2.1.2 Response (Ответ) .....	5
2.2 GETTRANSACTIONSTATUS (ПОЛУЧИТЬ СТАТУС ТРАНЗАКЦИИ) .....	5
2.2.1 Request (Запрос) .....	5
2.2.2 Response (Ответ) .....	6
2.3 COMPLETETRANSACTION (ЗАВЕРШИТЬ ТРАНЗАКЦИЮ) .....	7
2.3.1 Request (Запрос) .....	7
2.3.2 Response (Ответ) .....	8
2.4 REFUNDTRANSACTION (ВОЗВРАТ ПЛАТЕЖА ПО ТРАНЗАКЦИИ) .....	8
2.4.1 Request (Запрос) .....	8
2.4.2 Response (ответ) .....	9
2.5 GETEXTENDEDTRANSACTIONSTATUS (ПОЛУЧИТЬ ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ ТРАНЗАКЦИИ) .....	9
2.5.1 Request (Запрос) .....	9
2.5.2 Response (Ответ) .....	10
<b>3 REFERENCE IMPLEMENTATION (ПРИМЕРЫ РЕАЛИЗАЦИИ) .....</b>	<b>11</b>
3.1 CHECKOUT.ASPX.CS, CHECKOUT.PHP, CHECKOUT.JAVA.....	11
3.2 RESULT.ASPX.CS, RESULTS.PHP, RESULT.JAVA .....	11
3.2.1 .Net .....	11
3.2.2 Java .....	11
3.2.3 PHP.....	11
<b>4 WEBSITE LOOK AND FEEL (ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПЛАТЁЖНОЙ СТРАНИЦЫ).....</b>	<b>12</b>
<b>ПРИЛОЖЕНИЕ А MERCHANT WEB SERVICES WSDL.....</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ В TEST CONNECTIVITY (ТЕСТИРОВАНИЕ СОЕДИНЕНИЯ) .....</b>	<b>28</b>

---

## 1 Введение

Данный документ предназначен для того, чтобы помочь торгово-сервисным предприятиям успешно интегрировать свои Интернет магазины с платежным шлюзом CNP Processing GmbH. Мы предоставляем торгово-сервисным предприятиям стандартный интерфейс для интеграции, этот документ описывает непосредственно интерфейс программирования приложений (API), а так же процесс осуществления транзакции через платёжный шлюз. Общий вид процесса приведен ниже на схеме:



## 2 Интерфейс программирования приложений (API) для торгового-сервисного предприятия (Merchant)

Предоставляемый торгово-сервисным предприятиям интерфейс программирования приложений (API) реализован в виде веб-сервиса (смотрите WSDL описание в [Приложении А](#)). Мы предоставляем тестовый доступ к нашему тестовому серверу для возможности тестирования, для подробной информации смотрите [Приложение В](#). Ниже описывается сам процесс.

Когда покупатель (конечный клиент) завершает выбор продуктов на сайте Интернет магазина, и желает произвести оплату, ему обычно отображается корзина товаров и запрашивается способ оплаты товара. В случае, если покупатель выбирает способ оплаты картой, вам необходимо вызвать метод запуска транзакции (start transaction method) в API платёжного шлюза, используя при этом информацию с данными платежа (сумма, валюта и т.д.); в результате вы получите URL для переадресации покупателя. Как только покупатель осуществляет платеж, он переадресовывается обратно на ваш веб-сайт (Интернет магазин), на тот URL адрес, который вы указываете при инициации транзакции. Затем вы можете запросить статус транзакции и предоставить покупателю чек, если требуется. *Важно: Пожалуйста, обратите внимание, что на данном этапе еще не выполняется перевод денежных средств, средства только блокируются на карте покупателя.*

Следующим шагом вы подготавливаете к отправке покупателю его товары. Когда товары готовы к отгрузке, должен быть вызван метод завершения транзакции (complete transaction method). Тем самым запрашивается перевод денежных средств со счета покупателя на ваш счет. Товары отсылаются, как только был получен успешный ответ о завершении данной операции. Данный процесс должен выполняться даже в случае продаж цифровых лицензий и т.д., для обеспечения построения товаров в очередь доставки покупателю электронным способом или физическим.

### 2.1 startTransaction (начать транзакцию)

Этот раздел подробно описывает использование метода инициирования транзакции (start transaction). Данный метод должен вызываться для всех случаев, когда покупатель желает оплатить товар или услугу платёжной картой. Подробное описание параметров этого метода приводится в следующем разделе. При получении успешного ответа от web сервиса, вы получаете URL платёжной страницы, на которую должен быть переадресован покупатель. Если запрос завершается с ошибкой, необходимо предложить покупателю выбрать альтернативный метод оплаты или предложить повторить попытку позднее. Обязательно сохраните сообщение об ошибке в ваших логах и сообщите о нём нашим сотрудникам службы поддержки для дальнейшего анализа.

#### 2.1.1 Запрос (Request)

Запрос формируется с использованием следующих параметров:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantId</b> (идентификационный номер ТСП)	М	Уникальный идентификационный номер (ID) ТСП, предоставленный компанией CNP Processing GmbH. Длина поля 15 символов.
<b>terminalId</b> (идентификационный номер терминала)	С	Является обязательным, если CNP Processing GmbH присвоил ТСП terminal Id. Если terminal Id не присвоен для ТСП, не используйте это поле.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>customerReference</b> (уникальный идентификатор транзакции)	О	Вы можете снабдить каждую транзакцию уникальным номером для идентификации транзакции. Это поле должно составлять 12 символов в длину и соответствовать следующему регулярному выражению <code>^[0-9A-Za-z]{1,12}\$</code> . Сгенерированные номера должны оставаться уникальными, как минимум в течение последних шести месяцев. Рекомендуется заполнять поле по следующему правилу <code>AAxxxxxxx</code> , где AA - 2 кодовых буквы магазина, которые предоставляются CNP при подключении xxxxxxx - уникальный набор из 10 букв/цифр для идентификации транзакции. Если поле оставлено пустым, CNP сгенерирует уникальный номер (reference) и вернет его в ответном сообщении.
<b>orderId</b> (номер заказа)	М	Идентификатор номер заказа в магазине.
<b>totalAmount</b> (итоговая сумма)	М	Итоговая сумма транзакции; данная сумма будет списана с карты покупателя. Передаётся в виде целого числа без десятичной точки, фактически сумма заказа умноженная на 100. Например, для суммы 1000.00 нужно передавать 100000.
<b>currencyCode</b> (код валюты)	М	Код валюты, в которой обрабатывается итоговая сумма транзакции. В Казахстане валюта покупки – тенге, ее код 398. При необходимости использования другой валюты – сделайте запрос в <a href="mailto:integration-support@processing.kz">integration-support@processing.kz</a> .
<b>languageCode</b> (код языка)	М	Код языка, на котором отображается платёжная страница для покупателя. Если вы попытаетесь произвести транзакцию, используя неподдерживаемый код языка, вы получите сообщение о соответствующей ошибке, с указанием списка поддерживаемых языков. На данный момент платёжным шлюзом поддерживаются следующие коды языков: ru – русский / kz – казахский / en - английский
<b>Description</b> (описание)	О	Дополнительное описание покупки выполняемой покупателем на платёжной странице.
<b>billingAddress</b> (расчетный адрес)	О	Данное поле в странах СНГ не используется.
<b>merchantAdditionalInformationList</b> (список дополнительной информации ТСП)	О	Любая дополнительная информация относительно конкретной транзакции, которую вы захотите включить. Для более подробной информации смотреть раздел 2.1.1.1.
<b>goodsList</b> (список товаров)	М	Выборочный список товаров, которые составляют сумму транзакции. Общая сумма по списку не сверяется с итоговой суммой платежа, но этот список отображается покупателю на платёжной странице и в квитанции по оплате. Для более подробной информации смотреть раздел 2.1.1.2.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>returnURL</b> (URL адрес для возврата на сайт ТСП)	М	<p>URL адрес на который возвращается покупатель при завершении транзакции. Это поле должно предоставлять вам достаточно информации для завершения транзакции с покупателем, например, ID сессии покупателя</p> <p><a href="http://my.merchant.kz/finishedPayment?sessionID=A1B2V3">http://my.merchant.kz/finishedPayment?sessionID=A1B2V3</a></p> <p>Если вы используете автоматическую генерацию RRN и оставляете знак = (знак «равно») на конце URL адреса, в таком случае автоматически сгенерированный customerReference будет дополняться в конце URL адреса в вашем returnURL. Рекомендуем RRN брать из параметров, а не из сессии.</p> <p>Также, система автоматически вызывает указанный URL, сразу же после ответа банка-эмитента.</p> <p>Для магазина происходит аналогичное тому, что покупатель на платежной странице нажмет кнопку «Вернуться в магазин».</p> <p>В этом случае, у ТСП есть возможность своевременно узнать результат оплаты в случае, если покупатель не станет нажимать на кнопку «Вернуться в магазин» и закроет свой браузер после проведенной оплаты.</p>
<b>purchaserName</b>	О	ФИО покупателя. Максимальная длина 255 симв.
<b>purchaserEmail</b>	О	Email покупателя. Максимальная длина 255 симв.
<b>purchaserPhone</b>	О	Телефон покупателя. Максимальная длина 255 симв.
<b>merchantLocalDateTime</b>	М	Дата и время магазина. Формат dd.mm.yyyy hh24:mi:ss (01.02.2012 18:34:58)

#### 2.1.1.1 AdditionalInformation (Дополнительная информация)

Дополнительная информация – это любая информация, которую вы хотите предоставлять вместе с транзакцией в качестве административной информации.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>Key</b> (ключ)	О	Наименование дополнительной информации. Максимальная длина 40 символов.
<b>Value</b> (значение)	О	Значение, связываемое с ключом. Максимальная длина 4000 символов.

#### 2.1.1.2 GoodsItem (Список товаров)

Представляет единицу товара выбранного для покупки.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>nameOfGoods</b> (наименование товара)	М	Наименование единицы товара, которое будет отображено для покупателя. Максимальная длина 255 символов.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantsGoodsID</b> (идентификационный номер товара ТСП)	О	Идентификатор товара, например, внутренний идентификатор или артикул ТСП). Максимальная длина 256 символов.
<b>Amount</b> (сумма)	М	Стоимость единицы товара.
<b>currencyCode</b> (код валюты)	М	Код валюты для указанной суммы.

### 2.1.2 Response (Ответ)

Ответ формируется из объекта StartTransactionResult следующим образом:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>Success</b> (успешно)	М	Имеет значение «истина» (true), когда транзакция начата успешно, в противном случае значение – «ложь» (false). Причины сбоя при запуске транзакции включают в себя неактивное состояние ТСП (отключен или заблокирован), некорректные данные в запросе или технические проблемы в банковской системе.
<b>redirectURL</b> (URL переадресации)	М	Адрес URL для переадресации покупателя.
<b>errorDescription</b> (описание ошибки)	М	Когда значение поля success – «ложь» (false), данное поле содержит описание возникшей ошибки (на англ.яз).
<b>customerReference</b> (уникальный идентификатор транзакции)	М	Уникальный идентификатор (customerReference) присвоенный ТСП или платёжным шлюзом для данной транзакции. Указанный идентификатор должен быть указан во всех последующих запросах по данной транзакции.

## 2.2 getTransactionStatus (получить статус транзакции)

Когда покупатель вернулся на сайт ТСП по URL, указанному в параметре returnUrl, например, после того, как авторизация его карты завершена (или ранее, например, по причине истечения срока ожидания сессии и т.д.), метод getTransactionStatus используется для определения статуса транзакции и для продолжения обработки заказа должным образом, согласно состоянию транзакции. Кроме того, запрос getTransactionStatus может использоваться в службе технической поддержки ТСП для определения состояния транзакций, проведённых ранее.

### 2.2.1 Request (Запрос)

Запрос выполняется со следующими обязательными полями:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantId</b>	М	Уникальный идентификационный номер (ID) ТСП, предоставляемый компанией CNP Processing GmbH
<b>customerReference</b>	М	Уникальный идентификатор (customerReference) присвоенный ТСП или платёжным шлюзом для данной транзакции.

## 2.2.2 Response (Ответ)

Ответ формируется в виде объекта `StoredTransactionStatus`, причём опциональные поля не отправляются обратно в сторону ТСП, если транзакция еще не была отправлена на авторизацию. Объект `StoredTransactionStatus` содержит следующие поля:

Имя поля	Обязательное (М)/ Оptionальное (О)	Описание
<b>transactionStatus</b> (статус транзакции)	M	Текущий статус транзакции, все возможные значения указаны в разделе 2.2.2.1.
<b>transactionCurrencyCode</b> (код валюты транзакции)	O	Код валюты, в которой была авторизована карта.
<b>amountRequested</b> (запрашиваемая сумма)	O	Сумма, запрашиваемая в начальном запросе, в коде валюты, указанном выше.
<b>amountAuthorised</b> (авторизованная сумма)	O	Сумма фактической авторизации (0 – если отменена) в коде валюты, указанном выше.
<b>amountSettled</b> (выбранная сумма)	O	Сумма, запрошенная для завершения оплаты (0 – если еще не оплачена) в коде валюты.
<b>amountAuthorised</b> (авторизованная сумма)	O	Сумма авторизации (0 – если отменена) в коде валюты, указанном выше.
<b>amountRefunded</b> (сумма возврата)	O	Итоговая сумма всех возвратов по данной транзакции (0 – при отсутствии возвратов)
<b>goods</b> (товары)	O	Список товаров, переданный в методе <code>startTransaction</code> . Для подробной информации смотрите раздел 2.1.1.2.
<b>additionalInformation</b> (дополнительная информация)	O	Список дополнительной информации, переданный в методе <code>startTransaction</code> . Для подробной информации см. раздел 2.1.1.1
<b>authCode</b> (код авторизации)	O	В случае успешной авторизации транзакции, данное поле содержит код авторизации, который генерируется банком-эмитентом карты.
<b>purchaserName</b> (имя владельца карты)	O	Имя покупателя на карте, введенное покупателем на платежной странице.
<b>purchaserEmail</b> (эл. адрес покупателя)	O	Электронный адрес покупателя, введенный покупателем на платежной странице. На него будет отправлено уведомление по оплате от CNP Processing GmbH.
<b>purchaserPhone</b> (номер телефона)	O	Номер телефона покупателя, введенный покупателем на Платежной странице.
<b>merchantLocalDateTime</b>	O	Дата и время магазина, переданное магазином в методе <code>startTransaction</code> .
<b>merchantOnlineAddress</b>	O	Адрес веб-сайта для формирования торгового чека

### 2.2.2.1 Transaction Status (Статус транзакции)

Ниже приведен список всех возможных статусов транзакции.

Значение	Описание
<b>NO_SUCH_TRANSACTION</b> (данная транзакция не существует)	Транзакция с указанным <code>customer reference</code> не найдена.
<b>PENDING_CUSTOMER_INPUT</b> (ожидание ввода данных карты покупателем на платёжной странице)	Ожидание ввода данных карты покупателем на платёжной странице. Т.е. данные еще не введены и не отправлены в банк для авторизации.
<b>PENDING_AUTH_RESULT</b> (ожидание результата авторизации)	Данные отправлены в банк, ожидание результата по транзакции от банка.



Значение	Описание
<b>AUTHORISED</b> (авторизована)	Банк авторизовал запрос.
<b>DECLINED</b> (отклонена)	Банк отклонил запрос.
<b>REVERSED</b> (отменена)	Авторизация была отменена.
<b>PAID</b> (оплачена)	Транзакция отправлена для расчета (settlement).
<b>REFUNDED</b> (возвращена)	Сумма транзакции была частично или полностью возвращена магазином;
<b>INVALID_MID</b> (неверный ID ТСП)	Указанный идентификатор ТСП неверный, пожалуйста, свяжитесь со службой поддержки.
<b>MID_DISABLED</b> (ID ТСП заблокирован)	Указанный идентификатор ТСП более не действителен (блокирован), пожалуйста, свяжитесь со службой поддержки.

### 2.3 completeTransaction (завершить транзакцию)

Когда товары покупателя готовы к отгрузке, ТСП необходимо вызвать метод completeTransaction с значением «true» в поле «transactionSuccess». Денежные средства не будут перечислены на счёт ТСП до тех пор, пока не будет вызван этот метод. Если товары не могут быть отгружены покупателю (например, по причине отсутствия товара), то в поле «transactionSuccess» должно быть установлено значение «ложь», это позволит отменить резервирование денежных средств, выполненное при авторизации.

Для предприятий предоставляющих электронные услуги/товары (билеты, сертификаты и т.п.) предусмотрена возможность использования автоматического списания денежных средств с карт-счета без вызова completeTransaction. Списание производится автоматически по результату успешной авторизации после вызова со стороны магазина метода startTransaction. Детали по работе функционала можно получить по запросу в [integration-support@processing.kz](mailto:integration-support@processing.kz).  
Возможность использования “Auto Settlement” для ТСП рассматривает CNP Processing GmbH.

#### 2.3.1 Request (Запрос)

Запрос формируется при помощи следующих полей:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantId</b>	М	Уникальный идентификационный номер (ID) ТСП, предоставляемый компанией CNP Processing GmbH
<b>customerReference</b>	М	Уникальный идентификатор (customerReference) присвоенный ТСП или платёжным шлюзом для данной транзакции.
<b>transactionSuccess</b>	М	При значении «истина» (true), транзакция будет отправлена для осуществления платежа (settlement) и будет выполнен запрос на перевод заблокированных средств со счета покупателя на банковский счет ТСП. При значении «ложь» (false), будет осуществлена попытка отменить резервирование денежных средств, выполненное при онлайн авторизации.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>overrideAmount</b> (изменение исходной суммы транзакции)	О	ТСП может изменить сумму платежа (settlement), установив в этом поле значение, отличное от суммы авторизации. Данная сумма должна быть меньше или равной авторизованной сумме. Если это поле не будет установлено, используется сумма, указанная при авторизации.
<b>goodsList</b> (список товаров)	О	Выборочный список товаров, которые составляют сумму транзакции. Общая сумма по списку не сверяется с итоговой суммой платежа. Передается в случае если перечень товаров отличается от указанного при авторизации (startTransaction). Для более подробной информации см. раздел 2.1.1.2.

### 2.3.2 Response (Ответ)

Ответ может иметь значение «истина» (true) или «ложь» (false). Если значение поля – «истина», значит, транзакция была успешно отправлена для выполнения расчета. Если значение – «ложь», значит, отправка транзакции для расчета не удалась. Операция может закончиться неудачей по ряду причин, например, прошло слишком много времени с момента первоначальной авторизации (более 30 дней) и запись о транзакции покупателя уже отсутствует, либо по причине технических проблем в банковской системе. Чтобы перепроверить результат запроса, вы можете вызвать `getTransactionStatus` (получить статус транзакции) (смотреть раздел 2.2), чтобы определить воздействие вызова (`completeTransaction`) на систему.

## 2.4 refundTransaction (возврат платежа по транзакции) (используется по согласованию с CNP и Банком-эквайером)

Данный метод могут использовать только после согласования необходимости его использования и бизнес-условий с CNP Processing GmbH, а также Банком-эквайером. В настоящее время он заблокирован (отключен) и всегда возвращает отрицательный результат.

Метод может быть использован в случае, если вам потребуется выполнить возврат полной или части суммы платежа на карту покупателя при полном или частичном возврате товара.

### 2.4.1 Request (Запрос)

Запрос формируется из следующих полей:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantId</b>	М	Уникальный идентификационный номер (ID) ТСП, предоставляемый компанией CNP Processing GmbH
<b>customerReference</b>	М	Уникальный идентификатор, присвоенный ТСП или платёжным шлюзом для данной транзакции.
<b>refundAmount</b> (сумма возврата)	М	Сумма, подлежащая возврату, должна быть указана в валюте исходной транзакции.
<b>Description</b> (описание)	М	Описание причины возврата платежа.
<b>goodsToRefund</b> (список товаров)	О	Список товаров, по которым проводится возврат суммы. Смотреть раздел 2.4.1.2.
<b>additionalInformation</b> (доп. информация)	О	Дополнительная информация по возврату платежа. Смотреть раздел 2.4.1.1.

#### 2.4.1.1 AdditionalInformation (Дополнительная информация)

Дополнительная информация – это любая информация, которую вы хотите добавить к транзакции возврата платежа. В дальнейшем вы сможете видеть эту информацию в Merchant Portal-e.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>key</b> (ключ)	М	Наименование дополнительной информации.
<b>value</b> (значение)	М	Значение, связываемое с ключом.

#### 2.4.1.2 GoodsItem (единица товара)

Товары, подлежащие возврату.

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>nameOfGoods</b> (наименование товара)	М	Наименование товаров, которые могут быть отображены покупателю.
<b>merchantsGoodsID</b> (ID товара ТСП)	О	Ссылка на товар (внутренний идентификатор ТСП).
<b>amount</b> (сумма)	М	Стоимость данной единицы товара.
<b>currencyCode</b> (код валюты)	М	Код валюты, применимый к сумме.

#### 2.4.2 Response (ответ)

Значение ответа может быть «истина» (true) или «ложь» (false). Когда транзакция успешно отправлена для возврата платежа, значение будет – «истина». В случае если значение «ложь», отправка транзакции для возврата платежа не удалась. Причинами сбоя могут быть: сумма транзакции возврата превышает исходную сумму транзакции, первоначальная транзакция не может быть найдена, а также технические проблемы в системе банка.

### 2.5 getExtendedTransactionStatus (получить дополнительные параметры транзакции)

Перед отправкой товара ТСП имеет возможность проанализировать данные карты и покупателя, если заказ показался подозрительным и подвергся сомнению в попытке мошенничества.

Данный сервис является дополнительным и используется ТСП по своему усмотрению и необходимости.

#### 2.5.1 Request (Запрос)

Запрос выполняется со следующими обязательными полями:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>merchantId</b>	М	Уникальный идентификационный номер (ID) ТСП, предоставляемый компанией CNP Processing GmbH
<b>customerReference</b>	М	Уникальный идентификатор, присвоенный ТСП или платёжным шлюзом для данной транзакции.

## 2.5.2 Response (Ответ)

Ответ формируется в виде объекта StoredTransactionStatus, который содержит следующие поля:

Имя поля	Обязательное (М)/ Оptionальное (О)	Описание
<b>CardIssuerCountry</b> (страна банка-эмитента)	М	Наименование страны банка, выпустившего карту.
<b>MaskedCardNumber</b> (часть номера карты)	М	Маскированный номер карты.
<b>Verified3D</b> (результат проверки 3D пароля)	М	Результат проверки 3D пароля: YES – проверка была осуществлена успешно NO – проверки не было.
<b>PurchaserIpAddress</b> (IP адрес покупателя)	М	Сетевой IP адрес компьютера покупателя.

### 3 Reference Implementation (примеры реализации)

Исходные коды с примерами реализации доступны для скачивания с сайта <http://www.processing.kz/>, это позволит вам увидеть, как можно запускать и завершать транзакцию. Приложение доступно на следующих языках:

- .Net 2.0, можно запустить, используя Visual Studio 2005 и выше. Если у вас нет доступа к Visual Studio, но на вашем компьютере установлен IIS, вы можете просто поместить проект в IIS, чтобы увидеть результат. Вы также можете скачать экспресс версию Visual Studio по данной ссылке <http://www.microsoft.com/express/Downloads/>, если вы хотите увидеть этот проект в работе. Примеры можно скачать по ссылке <http://www.processing.kz/samples/TestMerchantWebsiteNET.zip>
- PHP 5.3, может запускаться при помощи любого веб сервера, поддерживающего PHP. Примеры можно скачать по ссылке <http://www.processing.kz/samples/TestMerchantWebsitePHP.zip>
- Java - это готовый проект для Eclipse, самый простой способ для запуска и просмотра исходных кодов; папка WebContent должна быть развернута в вашей папке WebApps на веб сервере с поддержкой Java EE 5. Примеры можно скачать по ссылке <http://www.processing.kz/samples/TestMerchantWebsiteJava.zip>

Наиболее важные страницы описаны ниже.

#### 3.1 Checkout.aspx.cs, Checkout.php, Checkout.java

Если вы посмотрите исходный код этих страниц, то увидите вызов startTransaction (запустить транзакцию) с соответствующими параметрами, а также пример кода по сохранению идентификатора транзакции для дальнейшего использования.

#### 3.2 Result.aspx.cs, Results.php, Result.java

##### 3.2.1 .Net

В функции Page\_Load можно увидеть, как выполняется запрос для получения статуса предыдущей покупки. Обычно, здесь вы показываете подтверждение покупки для покупателя. При просмотре функции btn\_RequestPayment\_Click вы увидите код, который может быть выполнен после отгрузки товара (или перед отгрузкой). При просмотре функции btn\_RequestReversal\_Click можно увидеть, как правильно выполнить операцию отмены авторизации, что необходимо сделать, например, в случае неудавшейся отгрузки товара.

##### 3.2.2 Java

При просмотре страницы Result.jsp и класса Result.java вы увидите, как выполняются вызовы от страницы, через класс, к веб серверу.

##### 3.2.3 PHP

При просмотре Result.php, в самом верху страницы, вы увидите, как выполняются вызовы к веб-сервису, используя библиотеку CNPMerchantWebServiceClient.php.

#### 4 Website Look and Feel (графический пользовательский интерфейс платёжной страницы)

Графический пользовательский интерфейс платёжной страницы оформлен по умолчанию по предпочтениям CNP Processing GmbH (см. рис.).

Оплата заказа - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Оплата заказа

processing.kzhttps://test.processing.kz/CNPConsumerWebsite/CaptureCardholderDetails

Google

PROCESSING.kze-commerce and internet acquiring

Оплата заказа

Получатель:www.testmarket.processing.kz

№ заказа:239006

Сумма платежа:3 000 KZT

Введите данные платежной карты:

Пример заполнения

К оплате принимаются платежные карты международных платежных систем Visa и MasterCard. Внимание! Если ваша карта имеет 3-D SecureCode, то после ввода данных платежной карты, Вы будете перенаправлены на страницу вашего банка эмитента для ввода дополнительного пароля.

Лицевая сторона

Номер карты:

Укажите номер карты

Срок действия карты:

VALID THRU

--месяц--

----год----

Укажите срок действия карты

Держатель карты:

IVANOV IVAN

как указано на карте

VISA

MasterCard

Обратная сторона

Код CVV2/CVC2:

Укажите CVV2/CVC2

CVV2/CVC2

Email:purchaser@processing.kz

Телефон:+7 (727) 244 04 77

Реквизиты вашей платежной карты используются только в момент совершения операции и передаются в Visa/MasterCard по защищенному каналу. Данные вашей платежной карты хранятся в зашифрованном виде и не могут быть переданы третьему лицу.

Оплатить

Приобретаемые товары:

Cheese and Chive Pringles

1 000 KZT

Chocolatein bars Lindt

2 000 KZT

© CNP Processing GmbH 2009–2012

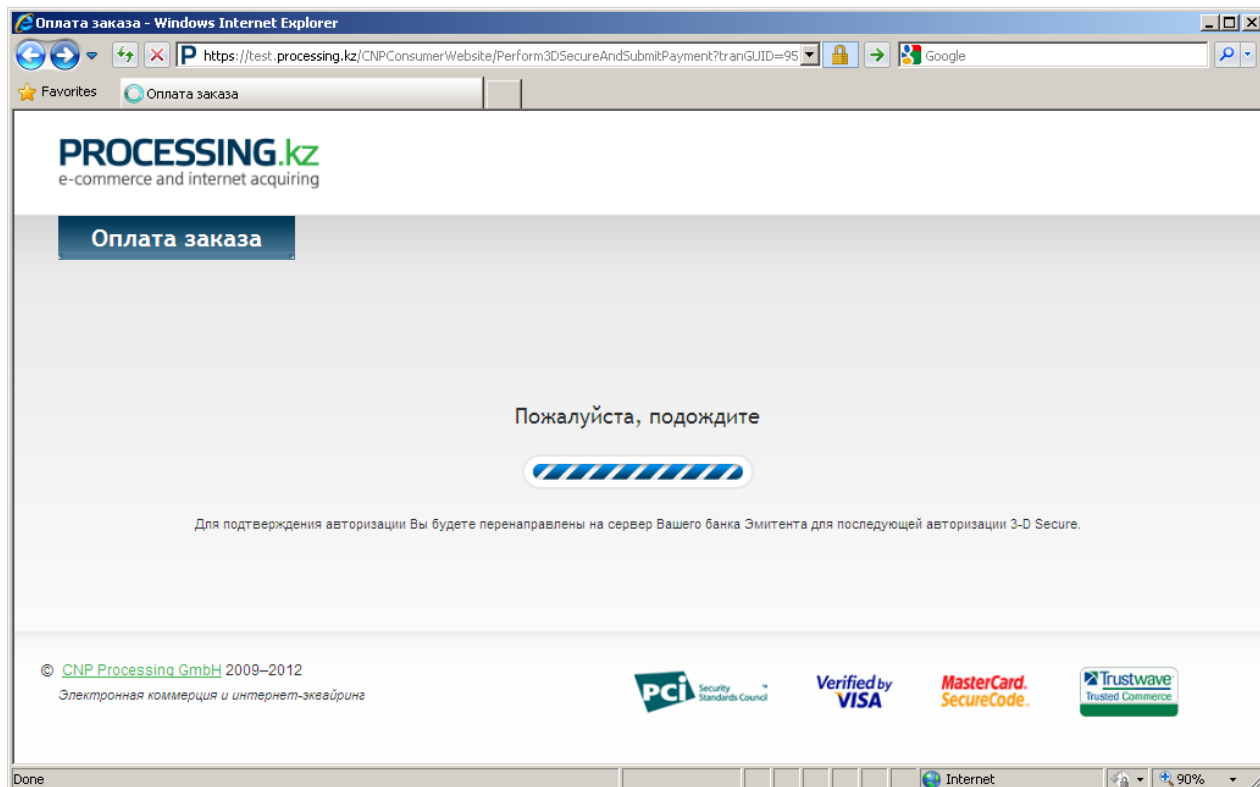
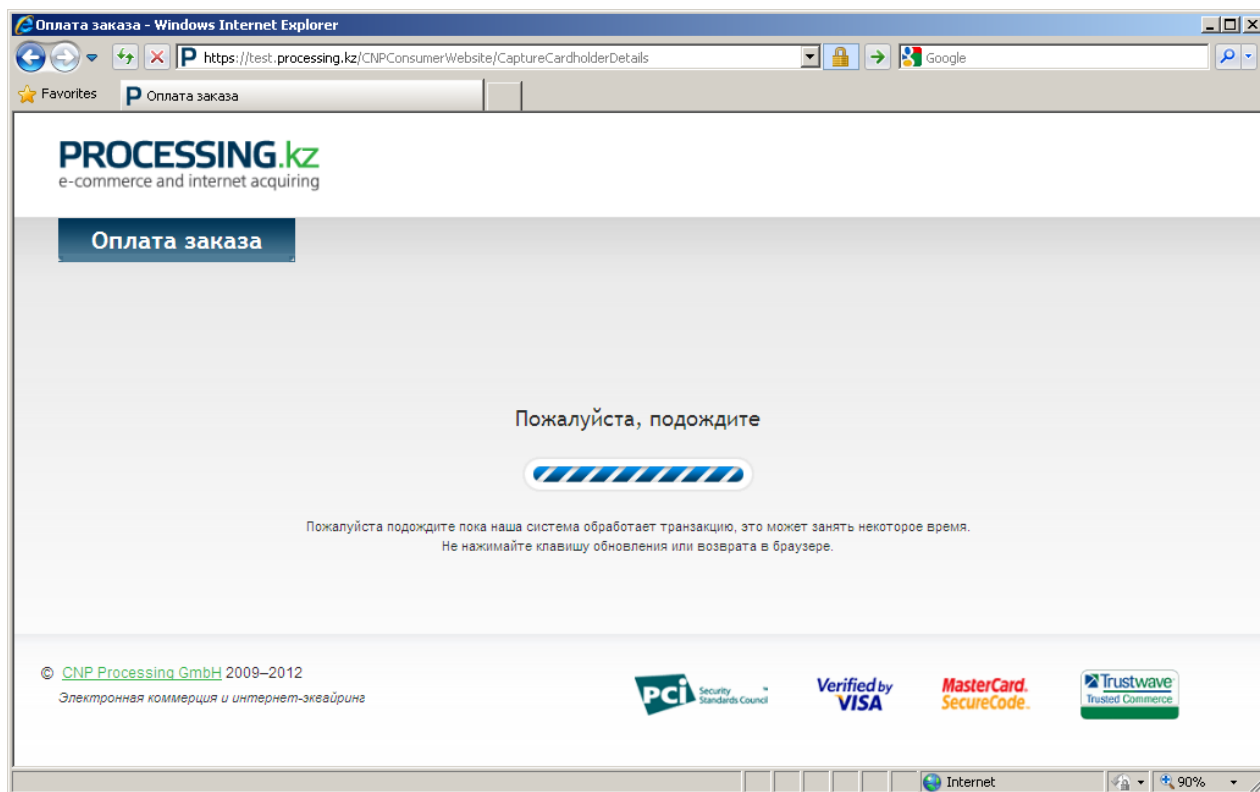
Электронная коммерция и интернет-эквайринг

pciSecurity Standards Council

Verified byVISA

MasterCardSecureCode.

TrustwaveTrusted Commerce



Оплата заказа - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Оплата заказа

processing.kz https://test.processing.kz/CNPConsumerWebsite/PaymentFinish?mid=0000000000000001&customerReference=133887239006&language=ru&returnUrl=https%3A%2F%2Ftest.processing.kz%2FTestMerchantWi

PROCESSING.kz  
e-commerce and internet acquiring

Оплата заказа

Оплата завершена успешно

Вернуться в магазин >

Печать чека

Наименование продавца	Processing Market
Адрес веб-сайта продавца	<a href="http://www.testmarket.processing.kz">www.testmarket.processing.kz</a>
№ заказа	239006
Дата транзакции	05.02.2012 10:59:50
Номер транзакции	133887239006
Имя покупателя	IVANOV IVAN
Код авторизации	A123B4
Тип операции	Покупка

**Политика возврата товара**

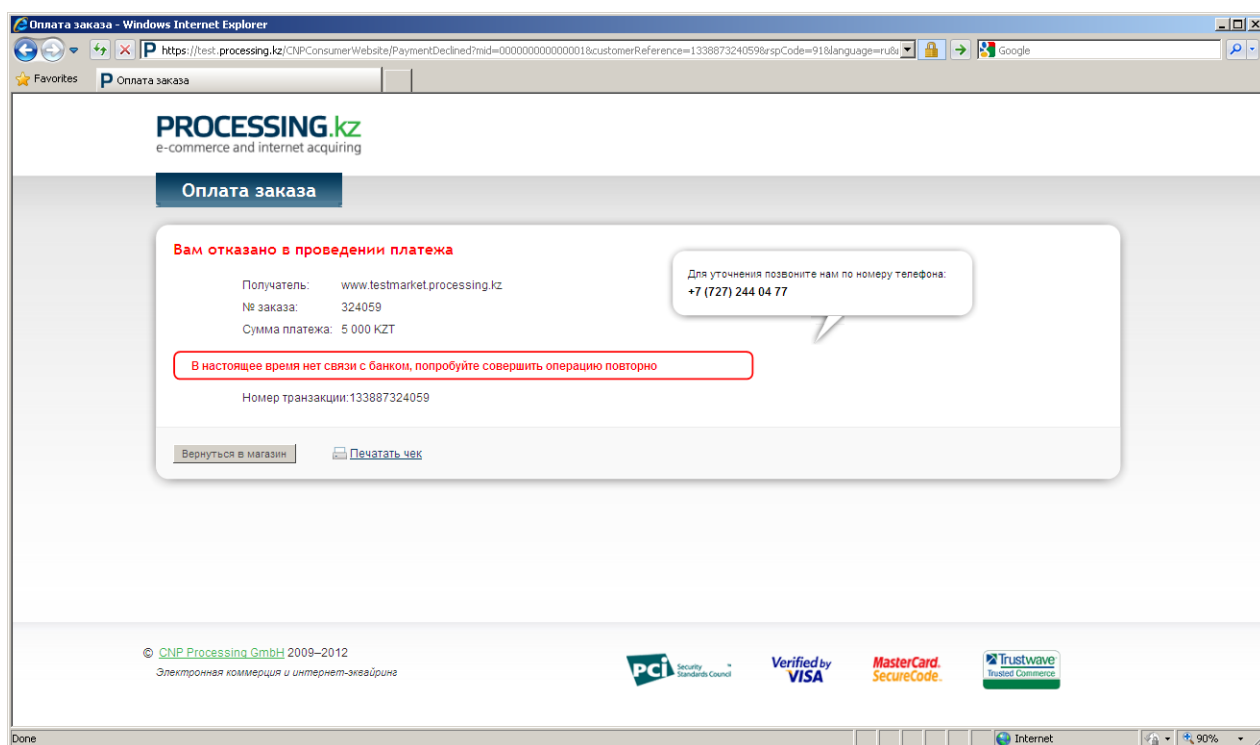
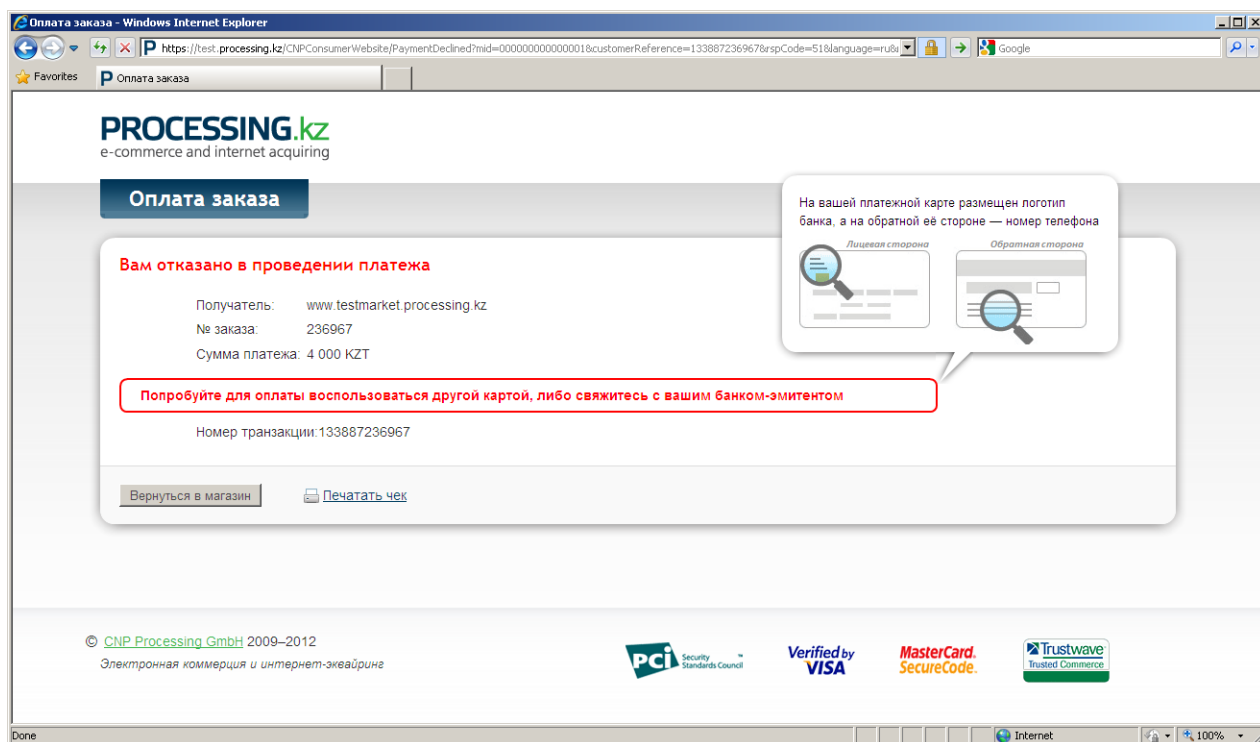
Вы можете отказаться от заказанного товара в любое время до его получения, а после получения - в течение 7 дней. Возврат товара надлежащего качества возможен в случае, если сохранены его товарный вид (упаковка, пломбы, ярлыки), потребительские свойства, а также документ, подтверждающий факт и условия покупки указанного товара (товарный или кассовый чек). Для возврата товара также необходим документ, удостоверяющий личность.

**Приобретаемые товары**

Cheese and Chive Pringles	1 000 KZT
Chocolatein bars Lindt	2 000 KZT
<b>ВСЕГО</b>	<b>3 000 KZT</b>

© CNP Processing GmbH 2009–2012  
Электронная коммерция и интернет-эквайринг





## Приложение А Merchant Web Services WSDL

```
<wsdl:definitions targetNamespace="http://kz.processing.cnp.merchant_ws/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:axis2="http://kz.processing.cnp.merchant_ws/" xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://kz.processing.cnp.merchant_ws/xsd"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:ax23="http://beans.merchant_web_services.cnp.processing.kz/xsd" xmlns:ax21="http://beans.common.cnp.processing.kz/xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/">
  <wsdl:documentation>CNPMerchantWebService</wsdl:documentation>
  <wsdl:types>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://beans.merchant_web_services.cnp.processing.kz/xsd">
      <xs:complexType name="StartTransactionResult">
        <xs:sequence>
          <xs:element minOccurs="0" name="customerReference" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="errorDescription" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="redirectURL" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="success" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="StartRegistrationResult">
        <xs:complexContent>
          <xs:extension base="ax23:StartTransactionResult">
            <xs:sequence/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:schema>
    <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://beans.common.cnp.processing.kz/xsd">
      <xs:complexType name="TransactionDetails">
        <xs:sequence>
          <xs:element minOccurs="0" name="billingAddress" nillable="true" type="ax21:Address"/>
          <xs:element minOccurs="0" name="cardId" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="currencyCode" type="xs:int"/>
          <xs:element minOccurs="0" name="customerReference" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="goodsList" nillable="true" type="ax21:GoodsItem"/>
          <xs:element minOccurs="0" name="LanguageCode" nillable="true" type="xs:string"/>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="merchantAdditionalInformationList" nillable="true" type="ax21:AdditionalInformation"/>
          <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="merchantLocalDateTime" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="orderId" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="paymentType" type="xs:int"/>
          <xs:element minOccurs="0" name="purchaserEmail" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="purchaserName" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="purchaserPhone" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="returnURL" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="terminalID" nillable="true" type="xs:string"/>
          <xs:element minOccurs="0" name="totalAmount" nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="Address">
```

```

<xs:sequence>
  <xs:element minOccurs="0" name="addressLine1" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="addressLine2" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="city" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="country" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="county" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="zip" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="GoodsItem">
  <xs:sequence>
    <xs:element minOccurs="0" name="amount" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="currencyCode" type="xs:int"/>
    <xs:element minOccurs="0" name="merchantsGoodsID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="nameOfGoods" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AdditionalInformation">
  <xs:sequence>
    <xs:element minOccurs="0" name="key" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="value" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RegistrationDetails">
  <xs:sequence>
    <xs:element minOccurs="0" name="currencyCode" type="xs:int"/>
    <xs:element minOccurs="0" name="customerReference" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="languageCode" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="merchantLocalDateTime" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="purchaserEmail" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="purchaserName" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="purchaserPhone" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="returnURL" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="terminalID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="totalAmount" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StoredTransactionStatus">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="additionalInformation" nillable="true" type="ax21:AdditionalInformation"/>
    <xs:element minOccurs="0" name="amountAuthorised" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="amountRefunded" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="amountRequested" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="amountSettled" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="authCode" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="cardHolder" nillable="true" type="xs:string"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="goods" nillable="true" type="ax21:GoodsItem"/>
    <xs:element minOccurs="0" name="merchantLocalDateTime" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="merchantOnlineAddress" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="orderId" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="purchaserEmail" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="purchaserName" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:element minOccurs="0" name="purchaserPhone" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="transactionCurrencyCode" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="transactionStatus" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="StoredRegistrationStatus">
    <xs:complexContent>
        <xs:extension base="ax21:StoredTransactionStatus">
            <xs:sequence>
                <xs:element minOccurs="0" name="cardId" nillable="true" type="xs:string"/>
                <xs:element minOccurs="0" name="maskedPan" nillable="true" type="xs:string"/>
                <xs:element minOccurs="0" name="registrationStatus" nillable="true" type="xs:string"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:schema>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://kz.processing.cnp.merchant_ws/xsd"
xmlns:ax24="http://beans.merchant_web_services.cnp.processing.kz/xsd" xmlns:ax22="http://beans.common.cnp.processing.kz/xsd">
    <xs:import namespace="http://beans.common.cnp.processing.kz/xsd"/>
    <xs:import namespace="http://beans.merchant_web_services.cnp.processing.kz/xsd"/>
    <xs:element name="startTransaction">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="transaction" nillable="true" type="ax22:TransactionDetails"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="startTransactionResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="return" nillable="true" type="ax24:StartTransactionResult"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="startRegistration">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="registration" nillable="true" type="ax22:RegistrationDetails"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="startRegistrationResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="return" nillable="true" type="ax24:StartRegistrationResult"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="refundTransaction">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>

```

```

        <xs:element minOccurs="0" name="password" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="refundAmount" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="goodsToRefund" nillable="true" type="ax22:GoodsItem"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="additionalInformation" nillable="true" type="ax22:AdditionalInformation"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="refundTransactionResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getVersionResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getTransactionStatus">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
            <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getTransactionStatusResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true" type="ax22:StoredTransactionStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getRegistrationStatus">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
            <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="getRegistrationStatusResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true" type="ax22:StoredRegistrationStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

<xs:element name="completeTransactionEx">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="transactionSuccess" type="xs:boolean"/>
      <xs:element minOccurs="0" name="overrideAmount" nillable="true" type="xs:string"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="goods" nillable="true" type="ax22:GoodsItem"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="completeTransactionExResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="completeTransaction">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="transactionSuccess" type="xs:boolean"/>
      <xs:element minOccurs="0" name="overrideAmount" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="completeTransactionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="completeRegistration">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="merchantId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="referenceNr" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="registrationSuccess" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="completeRegistrationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>

```

```
<wsdl:message name="completeRegistrationRequest">
  <wsdl:part name="parameters" element="ns:completeRegistration"/>
</wsdl:message>
<wsdl:message name="completeRegistrationResponse">
  <wsdl:part name="parameters" element="ns:completeRegistrationResponse"/>
</wsdl:message>
<wsdl:message name="startTransactionRequest">
  <wsdl:part name="parameters" element="ns:startTransaction"/>
</wsdl:message>
<wsdl:message name="startTransactionResponse">
  <wsdl:part name="parameters" element="ns:startTransactionResponse"/>
</wsdl:message>
<wsdl:message name="refundTransactionRequest">
  <wsdl:part name="parameters" element="ns:refundTransaction"/>
</wsdl:message>
<wsdl:message name="refundTransactionResponse">
  <wsdl:part name="parameters" element="ns:refundTransactionResponse"/>
</wsdl:message>
<wsdl:message name="getRegistrationStatusRequest">
  <wsdl:part name="parameters" element="ns:getRegistrationStatus"/>
</wsdl:message>
<wsdl:message name="getRegistrationStatusResponse">
  <wsdl:part name="parameters" element="ns:getRegistrationStatusResponse"/>
</wsdl:message>
<wsdl:message name="completeTransactionExRequest">
  <wsdl:part name="parameters" element="ns:completeTransactionEx"/>
</wsdl:message>
<wsdl:message name="completeTransactionExResponse">
  <wsdl:part name="parameters" element="ns:completeTransactionExResponse"/>
</wsdl:message>
<wsdl:message name="startRegistrationRequest">
  <wsdl:part name="parameters" element="ns:startRegistration"/>
</wsdl:message>
<wsdl:message name="startRegistrationResponse">
  <wsdl:part name="parameters" element="ns:startRegistrationResponse"/>
</wsdl:message>
<wsdl:message name="getVersionRequest"/>
<wsdl:message name="getVersionResponse">
  <wsdl:part name="parameters" element="ns:getVersionResponse"/>
</wsdl:message>
<wsdl:message name="getTransactionStatusRequest">
  <wsdl:part name="parameters" element="ns:getTransactionStatus"/>
</wsdl:message>
<wsdl:message name="getTransactionStatusResponse">
  <wsdl:part name="parameters" element="ns:getTransactionStatusResponse"/>
</wsdl:message>
<wsdl:message name="completeTransactionRequest">
  <wsdl:part name="parameters" element="ns:completeTransaction"/>
</wsdl:message>
<wsdl:message name="completeTransactionResponse">
  <wsdl:part name="parameters" element="ns:completeTransactionResponse"/>
</wsdl:message>
<wsdl:portType name="CNPMerchantWebServicePortType">
```



```

<wsdl:operation name="completeRegistration">
  <wsdl:input message="axis2:completeRegistrationRequest" wsaw:Action="urn:completeRegistration"/>
  <wsdl:output message="axis2:completeRegistrationResponse" wsaw:Action="urn:completeRegistrationResponse"/>
</wsdl:operation>
<wsdl:operation name="startTransaction">
  <wsdl:input message="axis2:startTransactionRequest" wsaw:Action="urn:startTransaction"/>
  <wsdl:output message="axis2:startTransactionResponse" wsaw:Action="urn:startTransactionResponse"/>
</wsdl:operation>
<wsdl:operation name="refundTransaction">
  <wsdl:input message="axis2:refundTransactionRequest" wsaw:Action="urn:refundTransaction"/>
  <wsdl:output message="axis2:refundTransactionResponse" wsaw:Action="urn:refundTransactionResponse"/>
</wsdl:operation>
<wsdl:operation name="getRegistrationStatus">
  <wsdl:input message="axis2:getRegistrationStatusRequest" wsaw:Action="urn:getRegistrationStatus"/>
  <wsdl:output message="axis2:getRegistrationStatusResponse" wsaw:Action="urn:getRegistrationStatusResponse"/>
</wsdl:operation>
<wsdl:operation name="completeTransactionEx">
  <wsdl:input message="axis2:completeTransactionExRequest" wsaw:Action="urn:completeTransactionEx"/>
  <wsdl:output message="axis2:completeTransactionExResponse" wsaw:Action="urn:completeTransactionExResponse"/>
</wsdl:operation>
<wsdl:operation name="startRegistration">
  <wsdl:input message="axis2:startRegistrationRequest" wsaw:Action="urn:startRegistration"/>
  <wsdl:output message="axis2:startRegistrationResponse" wsaw:Action="urn:startRegistrationResponse"/>
</wsdl:operation>
<wsdl:operation name="getVersion">
  <wsdl:input message="axis2:getVersionRequest" wsaw:Action="urn:getVersion"/>
  <wsdl:output message="axis2:getVersionResponse" wsaw:Action="urn:getVersionResponse"/>
</wsdl:operation>
<wsdl:operation name="getTransactionStatus">
  <wsdl:input message="axis2:getTransactionStatusRequest" wsaw:Action="urn:getTransactionStatus"/>
  <wsdl:output message="axis2:getTransactionStatusResponse" wsaw:Action="urn:getTransactionStatusResponse"/>
</wsdl:operation>
<wsdl:operation name="completeTransaction">
  <wsdl:input message="axis2:completeTransactionRequest" wsaw:Action="urn:completeTransaction"/>
  <wsdl:output message="axis2:completeTransactionResponse" wsaw:Action="urn:completeTransactionResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CNPMerchantWebServiceSoap11Binding" type="axis2:CNPMerchantWebServicePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <wsdl:operation name="startTransaction">
    <soap:operation soapAction="urn:startTransaction" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="completeRegistration">
    <soap:operation soapAction="urn:completeRegistration" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
  </wsdl:operation>

```



```
<wsdl:output>
  <soap:body use="Literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="refundTransaction">
  <soap:operation soapAction="urn:refundTransaction" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRegistrationStatus">
  <soap:operation soapAction="urn:getRegistrationStatus" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="completeTransactionEx">
  <soap:operation soapAction="urn:completeTransactionEx" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="startRegistration">
  <soap:operation soapAction="urn:startRegistration" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getVersion">
  <soap:operation soapAction="urn:getVersion" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getTransactionStatus">
  <soap:operation soapAction="urn:getTransactionStatus" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
```

```
</wsdl:input>
<wsdl:output>
  <soap:body use="Literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="completeTransaction">
  <soap:operation soapAction="urn:completeTransaction" style="document"/>
  <wsdl:input>
    <soap:body use="Literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="Literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="CNPMerchantWebServiceSoap12Binding" type="axis2:CNPMerchantWebServicePortType">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <wsdl:operation name="startTransaction">
    <soap12:operation soapAction="urn:startTransaction" style="document"/>
    <wsdl:input>
      <soap12:body use="Literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="Literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="completeRegistration">
    <soap12:operation soapAction="urn:completeRegistration" style="document"/>
    <wsdl:input>
      <soap12:body use="Literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="Literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="refundTransaction">
    <soap12:operation soapAction="urn:refundTransaction" style="document"/>
    <wsdl:input>
      <soap12:body use="Literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="Literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getRegistrationStatus">
    <soap12:operation soapAction="urn:getRegistrationStatus" style="document"/>
    <wsdl:input>
      <soap12:body use="Literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="Literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:operation>
```

```
<wsdl:operation name="completeTransactionEx">
  <soap12:operation soapAction="urn:completeTransactionEx" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="startRegistration">
  <soap12:operation soapAction="urn:startRegistration" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getVersion">
  <soap12:operation soapAction="urn:getVersion" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getTransactionStatus">
  <soap12:operation soapAction="urn:getTransactionStatus" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="completeTransaction">
  <soap12:operation soapAction="urn:completeTransaction" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="CNPMerchantWebServiceHttpBinding" type="axis2:CNPMerchantWebServicePortType">
  <http:binding verb="POST"/>
  <wsdl:operation name="startTransaction">
    <http:operation location="CNPMerchantWebService/startTransaction"/>
    <wsdl:input>
      <mime:content type="text/xml" part="startTransaction"/>
    </wsdl:input>
  </wsdl:operation>
</wsdl:binding>
```

```
<wsdl:output>
  <mime:content type="text/xml" part="startTransaction"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="completeRegistration">
  <http:operation location="CNPMerchantWebService/completeRegistration"/>
  <wsdl:input>
    <mime:content type="text/xml" part="completeRegistration"/>
  </wsdl:input>
  <wsdl:output>
    <mime:content type="text/xml" part="completeRegistration"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="refundTransaction">
  <http:operation location="CNPMerchantWebService/refundTransaction"/>
  <wsdl:input>
    <mime:content type="text/xml" part="refundTransaction"/>
  </wsdl:input>
  <wsdl:output>
    <mime:content type="text/xml" part="refundTransaction"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getRegistrationStatus">
  <http:operation location="CNPMerchantWebService/getRegistrationStatus"/>
  <wsdl:input>
    <mime:content type="text/xml" part="getRegistrationStatus"/>
  </wsdl:input>
  <wsdl:output>
    <mime:content type="text/xml" part="getRegistrationStatus"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="completeTransactionEx">
  <http:operation location="CNPMerchantWebService/completeTransactionEx"/>
  <wsdl:input>
    <mime:content type="text/xml" part="completeTransactionEx"/>
  </wsdl:input>
  <wsdl:output>
    <mime:content type="text/xml" part="completeTransactionEx"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="startRegistration">
  <http:operation location="CNPMerchantWebService/startRegistration"/>
  <wsdl:input>
    <mime:content type="text/xml" part="startRegistration"/>
  </wsdl:input>
  <wsdl:output>
    <mime:content type="text/xml" part="startRegistration"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getVersion">
  <http:operation location="CNPMerchantWebService/getVersion"/>
  <wsdl:input>
    <mime:content type="text/xml" part="getVersion"/>
```

```

    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml" part="getVersion"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getTransactionStatus">
    <http:operation location="CNPMerchantWebService/getTransactionStatus"/>
    <wsdl:input>
      <mime:content type="text/xml" part="getTransactionStatus"/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml" part="getTransactionStatus"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="completeTransaction">
    <http:operation location="CNPMerchantWebService/completeTransaction"/>
    <wsdl:input>
      <mime:content type="text/xml" part="completeTransaction"/>
    </wsdl:input>
    <wsdl:output>
      <mime:content type="text/xml" part="completeTransaction"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="CNPMerchantWebService">
  <wsdl:port name="CNPMerchantWebServiceHttpsSoap11Endpoint" binding="axis2:CNPMerchantWebServiceSoap11Binding">
    <soap:address
location="https://test.processing.kz:8443/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpsSoap11Endpoint/" />
  </wsdl:port>
  <wsdl:port name="CNPMerchantWebServiceHttpSoap11Endpoint" binding="axis2:CNPMerchantWebServiceSoap11Binding">
    <soap:address
location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpSoap11Endpoint/" />
  </wsdl:port>
  <wsdl:port name="CNPMerchantWebServiceHttpSoap12Endpoint" binding="axis2:CNPMerchantWebServiceSoap12Binding">
    <soap12:address
location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpSoap12Endpoint/" />
  </wsdl:port>
  <wsdl:port name="CNPMerchantWebServiceHttpsSoap12Endpoint" binding="axis2:CNPMerchantWebServiceSoap12Binding">
    <soap12:address
location="https://test.processing.kz:8443/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpsSoap12Endpoint/" />
  </wsdl:port>
  <wsdl:port name="CNPMerchantWebServiceHttpEndpoint" binding="axis2:CNPMerchantWebServiceHttpBinding">
    <http:address location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpEndpoint/" />
  </wsdl:port>
  <wsdl:port name="CNPMerchantWebServiceHttpsEndpoint" binding="axis2:CNPMerchantWebServiceHttpBinding">
    <http:address location="https://test.processing.kz:8443/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebServiceHttpsEndpoint/" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

## Приложение В Test Connectivity (Тестирование соединения)

Тестовые веб-сервисы доступны по ссылке:

<https://test.processing.kz/CNPMerchantWebServices/services/CNPMerchantWebService>

Пожалуйста, помните, что это тестовый сервис, используемый в публичной сети Интернет. Он постоянно используется несколькими клиентами, а техническая поддержка предоставляется лишь в рабочие часы (ALMT).

Чтобы получить ваш личный Merchant ID и номер тестовой карты для проведения тестовых платежей или дополнительную поддержку, пожалуйста, свяжитесь с командой тех. поддержки по интеграции компании CNP Processing GmbH, написав на электронный адрес: [integration-support@processing.kz](mailto:integration-support@processing.kz).