



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

Версия: 1.27  
Выпуск: 24.09.2019

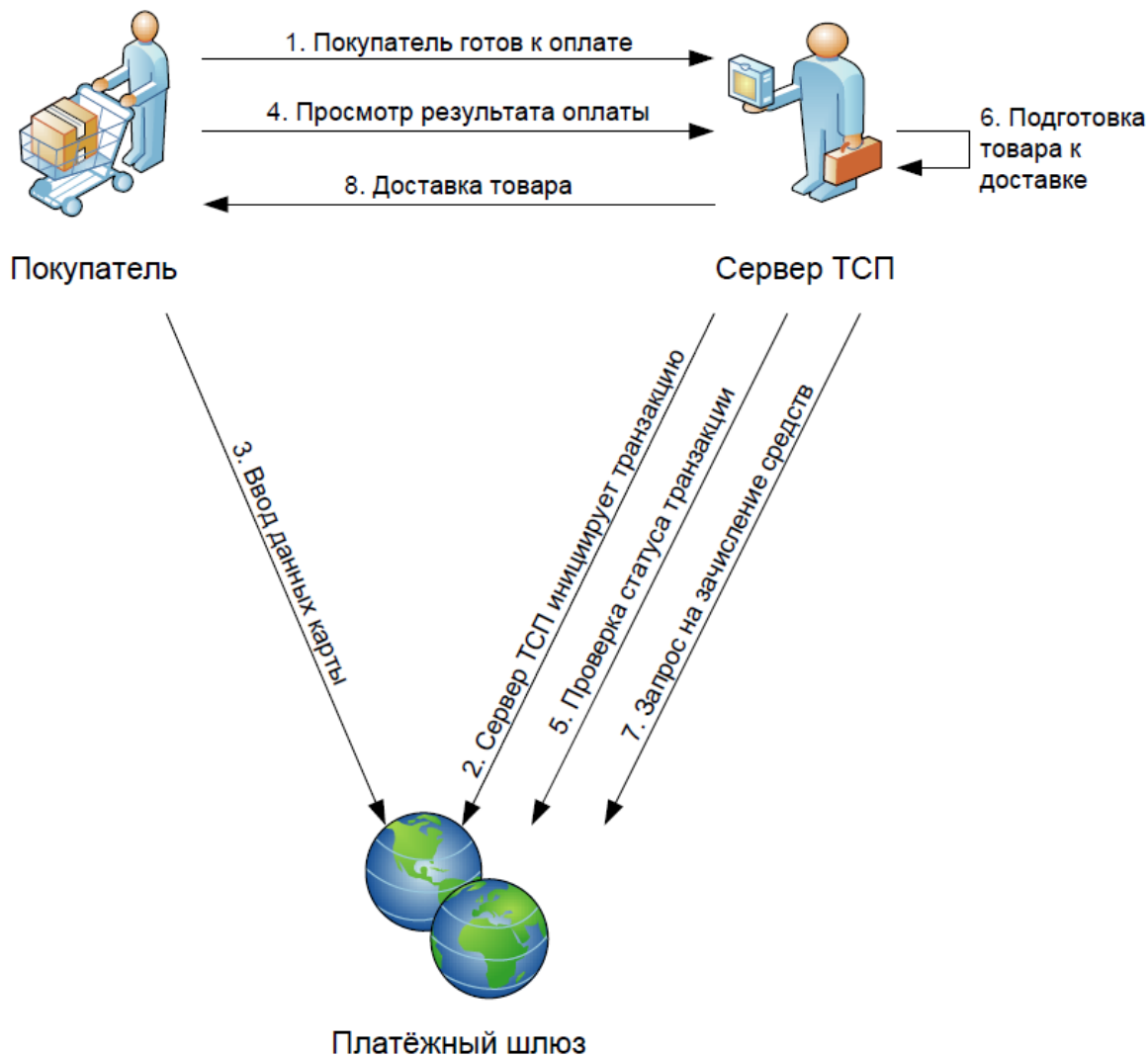
«CNP Processing GmbH»  
Республика Казахстан  
Г. Алматы, ул. Шевченко 118, офис 331  
Тел./Факс: +7 727 244 0477  
[www.processing.kz](http://www.processing.kz)

## Содержание

<b>СОДЕРЖАНИЕ .....</b>	<b>2</b>
<b>1 ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>2 ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ (API) ДЛЯ ТОРГОВО-СЕРВИСНОГО ПРЕДПРИЯТИЯ (MERCHANT) .....</b>	<b>4</b>
2.1 STARTTRANSACTION (НАЧАТЬ ТРАНЗАКЦИЮ) .....	4
2.1.1 Регистрация карты после оплаты.....	4
2.1.2 Механика работы с методом StartTransaction.....	5
2.1.3 Запрос (Request).....	5
2.1.4 Response (Ответ).....	7
2.2 GETTRANSACTIONSTATUSCODE (ПОЛУЧИТЬ СТАТУС ТРАНЗАКЦИИ) .....	8
2.2.1 Request (Запрос).....	8
2.2.2 Response (Ответ).....	8
2.3 COMPLETETRANSACTION (ЗАВЕРШИТЬ ТРАНЗАКЦИЮ) .....	10
2.3.1 Request (Запрос).....	10
2.3.2 Response (Ответ).....	11
2.4 GETEXTENDEDTRANSACTIONSTATUS (ПОЛУЧИТЬ ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ ТРАНЗАКЦИИ) .....	11
2.4.1 Request (Запрос).....	11
2.4.2 Response (Ответ).....	11
<b>3 СПЛИТ ПЛАТЕЖИ ИНТЕГРАЦИЯ С GALILEO/AMADEUS .....</b>	<b>13</b>
3.1 ИНТЕГРАЦИЯ С GALILEO.....	13
3.2 ИНТЕГРАЦИЯ С AMADEUS .....	13
<b>4 WEBSITE LOOK AND FEEL (ГРАФИЧЕСКИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПЛАТЁЖНОЙ СТРАНИЦЫ).....</b>	<b>15</b>
ПРИЛОЖЕНИЕ А MERCHANT WEB SERVICES WSDL .....	19
ПРИЛОЖЕНИЕ В TEST CONNECTIVITY (ТЕСТИРОВАНИЕ СОЕДИНЕНИЯ) .....	30

## 1 Введение

Данный документ предназначен для того, чтобы помочь торгово-сервисным предприятиям<sup>1</sup> 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 также процесс осуществления транзакции через платёжный шлюз. Общий вид процесса приведен ниже на схеме:



<sup>1</sup>Торгового-Сервисным Предприятием являются электронные магазины, финансовые институты, оказывающие услуги держателям платежных карт VISA, MasterCard посредством интернет коммуникаций.

## 2 ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ (API) ДЛЯ ТОРГОВО-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 Регистрация карты после оплаты

В случае если требуется сохранять карту после успешной оплаты необходимо сообщить о данном требовании в CNP.

После успешной транзакции CNP вызывает returnUrl, который будет содержать userId и cardId.

Если статус карты будет REGISTERED. Данная карта сохранена и можно использовать для последующих транзакций без повторного ввода карточных данных, более подробно необходимо см. спецификацию CNP Merchant Integration Guide\_OneClick метод Pay. Для проверки статуса карты необходимо вызвать метод getCardStatus, более подробно необходимо см. спецификацию CNP Merchant Integration Guide\_OneClick. В случае если статус карты будет в UNCOMPLETED для завершения регистрации необходимо вызвать метод completeCardRegistration, более подробно необходимо см. спецификацию CNP Merchant Integration Guide\_OneClick.

\*В случае если первый платеж отменяется (статус транзакции REVERSED), то карту запрещено использовать для последующих оплат.

### 2.1.2 Механика работы с методом StartTransaction

- 1) Merchant вызывает сервис StartTransaction. В ответ получает URL страницы ввода карточных данных. Merchant перекидывает клиента на URL – на этом этапе нет необходимости вызывать метод проверки статуса getTransactionStatusCode;
- 2) По завершению транзакции CNP вызывает returnUrl Merchant'a. Только после получения вызова Вашего returnUrl необходимо вызвать метод проверки статуса getTransactionStatusCode.

В случае если Вы не получили вызов Вашего returnUrl, то необходимо ждать 20 минут с момента получения URL страницы ввода карточных данных и после вызывать метод проверки статуса getTransactionStatusCode.

- 3) Есть статус транзакции один из ниже указанных, то повторно нет необходимости вызывать метод проверки статуса getTransactionStatusCode, + необходимо учитывать временной интервал, указанный в пункте 2:

NO\_SUCH\_TRANSACTION

DECLINED

REVERSED

PAID

### 2.1.3 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)	О	Данное поле не заполняется значением (передавать необходимо пустым).
<b>customerReference</b> (уникальный идентификатор транзакции)	М	Вы можете снабдить каждую транзакцию уникальным номером для идентификации транзакции. Это поле должно составлять 12 цифр в длину. Сгенерированные номера должны оставаться уникальными, как минимум в течение последних шести месяцев. Рекомендуется заполнять поле по следующему правилу: [1-9]xxxxxxxxxx, где [1-9] – первая цифра номера, которая находится в диапазоне от 1 до 9. xxxxxxxxxx - уникальный набор из 11 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)	M	Код валюты, в которой обрабатывается итоговая сумма транзакции. В Казахстане валюта покупки – тенге, ее код 398.
<b>languageCode</b> (код языка)	M	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)	O	Дополнительное описание покупки выполняемой покупателем на платёжной странице.
<b>billingAddress</b> (расчетный адрес)	O	Данное поле в странах СНГ не используется.
<b>merchantAdditionalInformationList</b> (список дополнительной информации ТСП)	O	Любая дополнительная информация относительно конкретной транзакции, которую вы захотите включить. Для более подробной информации смотреть раздел 2.1.2.1.
<b>goodsList</b> (список товаров)	M	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 2.1.2.2.
<b>returnURL</b> (URL адрес для возврата на сайт ТСП)	M	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 <a href="http://my.merchant.kz/finishedPayment?sessionID=A1B2V3">http://my.merchant.kz/finishedPayment?sessionID=A1B2V3</a> 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. Также, система автоматически вызывает указанный URL, сразу же после ответа банка-эмитента. Для магазина происходит аналогичное тому, что покупатель на платёжной странице нажмет кнопку «Вернуться в магазин». 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.
<b>purchaserName</b>	O	ФИО покупателя. Максимальная длина 255 симв.
<b>purchaserEmail</b>	O	Email покупателя. Максимальная длина 255 симв.
<b>purchaserPhone</b>	O	Телефон покупателя. Максимальная длина 255 симв.
<b>merchantLocalDateTime</b>	M	Дата и время магазина. Формат dd.mm.yyyy hh24:mi:ss (01.01.2019 18:34:58)

#### 2.1.3.1 AdditionalInformation (Дополнительная информация)

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

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

#### 2.1.3.2 GoodsItem (Список товаров)

Представляет единицу товара выбранного для покупки.

Имя поля	Обязательное (M)/ Опциональное (O)	Описание
<b>nameOfGoods</b> (наименование товара)	M	Наименование единицы товара, которое будет отображено для покупателя. Максимальная длина 255 символов.
<b>merchantsGoodsID</b> (идентификационный номер товара ТСП)	O	Идентификатор товара, например, внутренний идентификатор или артикул ТСП). Максимальная длина 256 символов.
<b>Amount</b> (сумма)	M	Стоимость единицы товара.
<b>currencyCode</b> (код валюты)	M	Код валюты для указанной суммы.

#### 2.1.4 Response (Ответ)

Ответ формируется из объекта StartTransactionResult следующим образом:

Имя поля	Обязательное (M)/ Опциональное (O)	Описание
<b>Success</b> (успешно)	M	Имеет значение «истина» (true), когда транзакция начата успешно, в противоположном 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>	M	Адрес URL для переадресации покупателя.

(URL переадресации)		
<b>errorDescription</b> (описание ошибки)	M	Когда значение поля success – «ложь» (false), данное поле содержит описание возникшей ошибки (на англ.яз).
<b>customerReference</b> (уникальный идентификатор транзакции)	M	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 getTransactionStatusCode (получить статус транзакции)

Для предыдущих версий используется метод getTransactionStatus. 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 getTransactionStatusCode используется для определения статуса транзакции и для продолжения обработки заказа должным образом, согласно состоянию транзакции. Кроме того, запрос getTransactionStatusCode 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 (M)/ Опциональное (O)	Описание
<b>merchantId</b>	M	Уникальный идентификационный номер (ID) ТСП, предоставленный компанией CNP Processing GmbH.
<b>referenceNr</b>	M	Уникальный идентификатор (customerReference) присвоенный ТСП или платёжным шлюзом для данной транзакции.

### 2.2.2 Response (Ответ)

Ответ формируется в виде объекта StoredTransactionStatusCode, 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 (M)/ Опциональное (O)	Описание
<b>transactionStatus</b> (статус транзакции)	M	Текущий статус транзакции, все возможные значения указаны в разделе «Статус транзакции».
<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>goods</b> (товары)	O	Список товаров, переданный в методе startTransaction. Для подробной информации смотрите раздел 2.1.2.2.
<b>orderId</b> (номер заказа)	M	Идентификатор номер заказа в магазине.



<b>additionalInformation</b> (дополнительная информация)	O	Список дополнительной информации, переданный в методе startTransaction. Для подробной информации см. раздел 2.1.2.1
<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 startTransaction.
<b>merchantOnlineAddress</b>	O	Адрес веб-сайта для формирования торгового чека
<b>RspCode</b> (код ответа)	O	Код ответа банка эмитента. Указывает на успешность прохождения транзакции. Если значение 00 – транзакция прошла успешно, если значение отлично от 00 – транзакция не успешна, подробную причину см. в RspCodeDesc.
<b>RspCodeDesc</b> (детальное описание кода ответа)	O	В случае не успешного прохождения транзакции, поле будет содержать детальное описание причины отказа. В случае успешного – поле не заполняется.
<b>IssuerBank</b>	O	Банк эмитент
<b>BankRRN</b>	O	Референс транзакции в Банке эмитенте
<b>SinkNode</b>	O	Канал передачи транзакции

### 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 customer reference не найдена.
<b>PENDING_CUSTOMER_INPUT</b> (ожидание ввода данных покупателем)	Ожидание ввода данных карты покупателем на платёжной странице. Т.е. данные еще не введены и не отправлены в банк для авторизации. GUID страницы ввода карточных данных действителен не более 20 минут.
<b>PENDING_AUTH_RESULT</b> (ожидание результата авторизации)	Данные отправлены в банк, ожидание результата по транзакции от банка.
<b>AUTHORISED</b> (авторизована)	Банк авторизовал запрос.
<b>DECLINED</b> (отклонена)	Банк отклонил запрос.
<b>REVERSED</b> (отменена)	Авторизация была отменена.
<b>PAID</b>	Транзакция отправлена для расчета (settlement).

(оплачена)	
<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.

Внимание! Списание платежа/отмену платежа необходимо произвести в течение 3 календарных дней с момента оплаты клиентом. В случае проведения поздних списаний (списание денежных средств по истечению 3 календарных дней) и последующего поступления претензий от Банков-эмитентов, суммы транзакций будут удерживаться со счетов коммерсанта вне зависимости от того, что была оказана услуга или предоставлен товар, т.к. отклонить подобные претензии по правилам Международных Платежных Систем не представляется возможным.

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, предоставленный компанией CNP Processing GmbH.
<b>referenceNr</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.
<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.

### 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 (по отмене более 72 часов, по подтверждению более 7 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), чтобы определить воздействие вызова (`completeTransaction`) на систему.

### 2.4 `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.

#### 2.4.1 Request (Запрос)

Запрос выполняется со следующими обязательными полями:

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

#### 2.4.2 Response (Ответ)

Ответ формируется в виде объекта `StoredTransactionStatusExtended`, который содержит следующие поля:

Имя поля	Обязательное (М)/ Опциональное (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 СПЛИТ ПЛАТЕЖИ ИНТЕГРАЦИЯ С GALILEO/AMADEUS

#### 3.1 Интеграция с Galileo

В поле merchantAdditionalInformationList (Дополнительная информация) необходимо передавать следующие параметры:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>Key</b> (ключ)	О	Наименование дополнительной информации. Максимальная длина 40 символов. 1. FEE_AMOUNT – сумма комиссии 2. GALILEO_WS_PROFILE – профайл тревел агентства в системе Galileo 3. GALILEO_WS_PASSWORD – пароль в системе Galileo 4. GALILEO_WS_USERNAME – пользователь для выдачи э.билета в системе Galileo 5. PNR - номер брони
<b>Value</b> (значение)	О	Значение, связываемое с ключом. Максимальная длина 4000 символов.

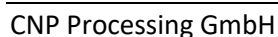
#### 3.2 Интеграция с Amadeus

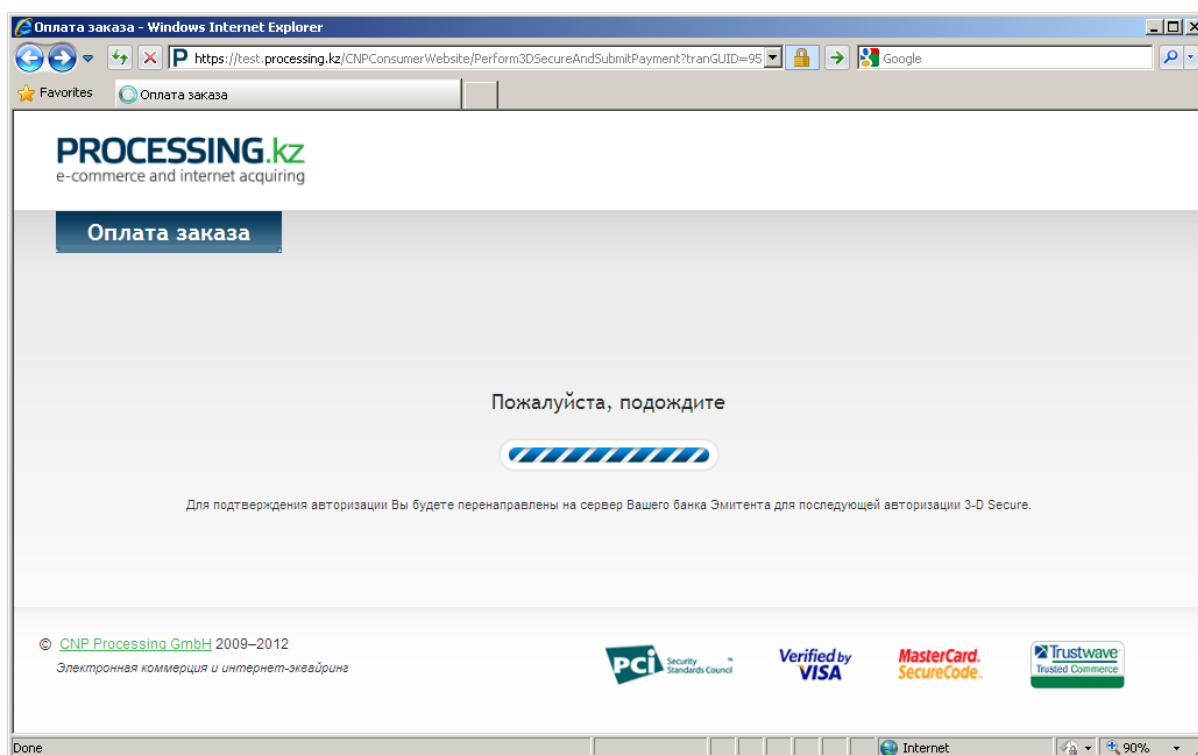
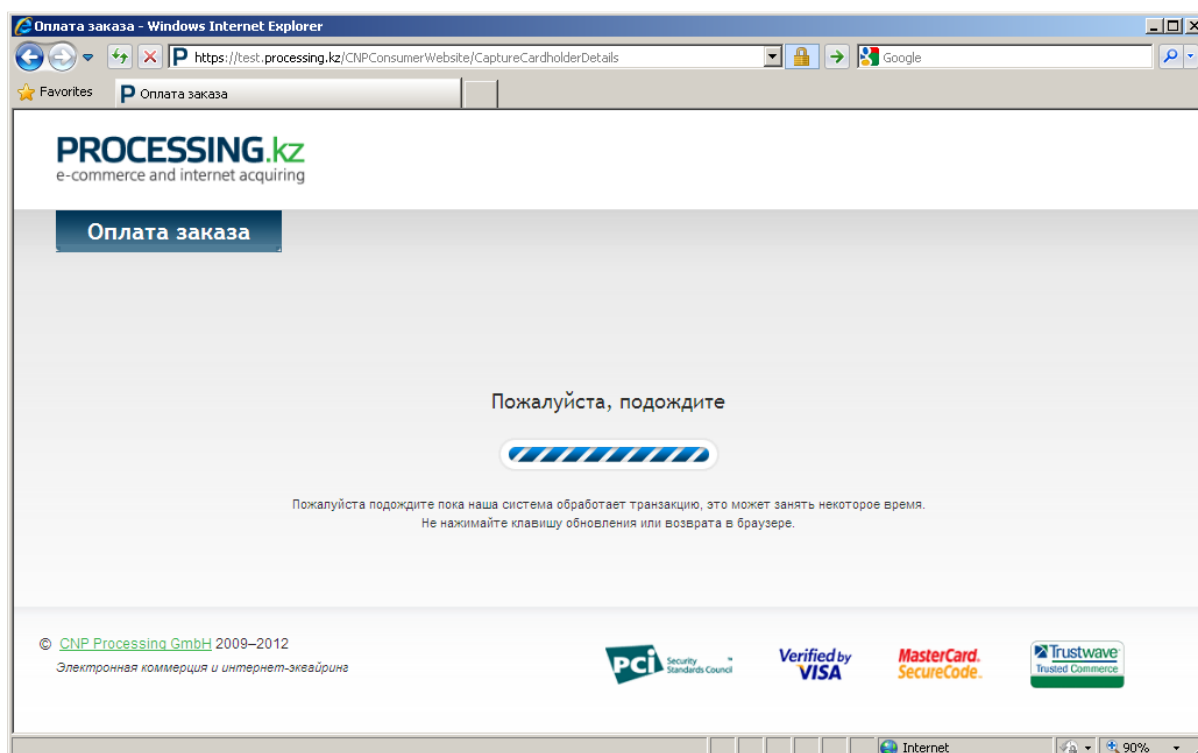
В merchantAdditionalInformationList необходимо передавать следующие параметры:

Имя поля	Обязательное (М)/ Опциональное (О)	Описание
<b>Key</b> (ключ)	О	Наименование дополнительной информации. Максимальная длина 40 символов. 1. PNR – номер брони 2. PERSON_SURNAME – имя пассажира 3. TICKET_AMOUNT – стоимость билета 4. FEE_AMOUNT - сумма комиссии 5. PAY_VIA_AMADEUS – признак оплаты через Amadeus (поддерживает ли авиакомпания оплату с карты через Амадеус, true - да, false - нет) 6. WEB_SERVICE_SECURITY_NUMBER – код агентства для работы с E-Power Amadeus WS 7. AMADEUS_WS_PORTAL_CODE – портал код в системе E-Power Amadeus WS 8. AMADEUS_WS_URL – url E-Power Amadeus WS куда надо отправлять запросы, сделано для теста. 9. AMADEUS_WS_USERNAME – пользователь для выдачи э. билета в системе E-Power Amadeus WS 10. AMADEUS_WS_PASSWORD – пароль в системе E-Power Amadeus WS 11. PNR – номер брони 12. PERSON_SURNAME – имя пассажира

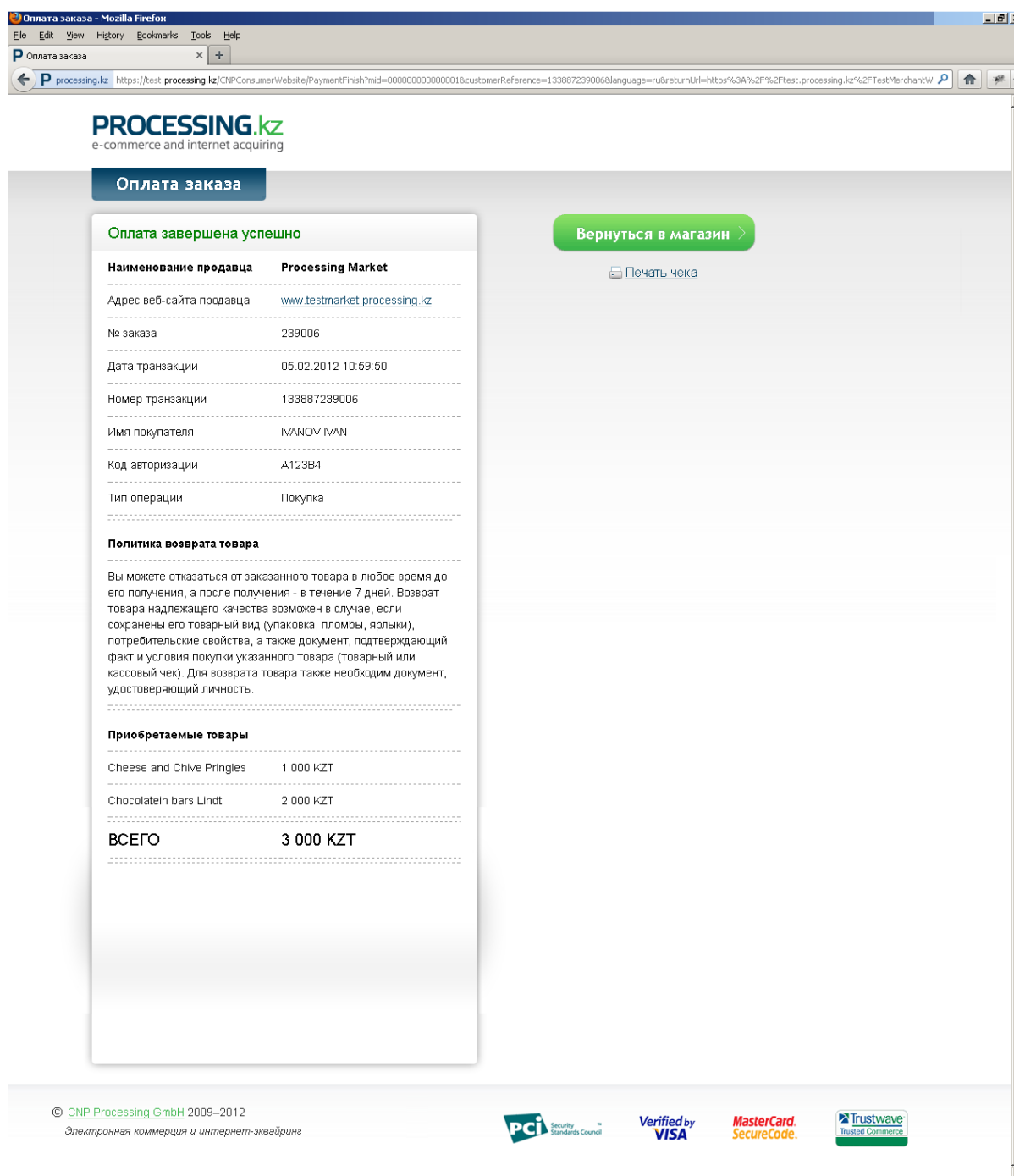
		<p>13. TICKET_AMOUNT – стоимость билета</p> <p>14. FEE_AMOUNT - сумма комиссии</p> <p>15. PAY_VIA_AMADEUS - признак оплаты через Amadeus (поддерживает ли авиакомпания оплату с карты через Амадеус, true - да, false - нет)</p> <p>16. WEB_SERVICE_SECURITY_NUMBER – код агентства для работы с E-Power Amadeus WS</p> <p>17. AMADEUS_WS_PORTAL_CODE – портал код в системе E-Power Amadeus WS</p> <p>18. AMADEUS_WS_URL – url E-Power Amadeus WS куда надо отправлять запросы, сделано для теста.</p> <p>19. AMADEUS_WS_USERNAME – пользователь для выписки э. билета в системе E-Power Amadeus WS</p> <p>20. AMADEUS_WS_PASSWORD – пароль в системе E-Power Amadeus WS</p>
<b>Value</b> (значение)	O	<p>Значение, связываемое с ключом.</p> <p>Максимальная длина 4000 символов.</p>

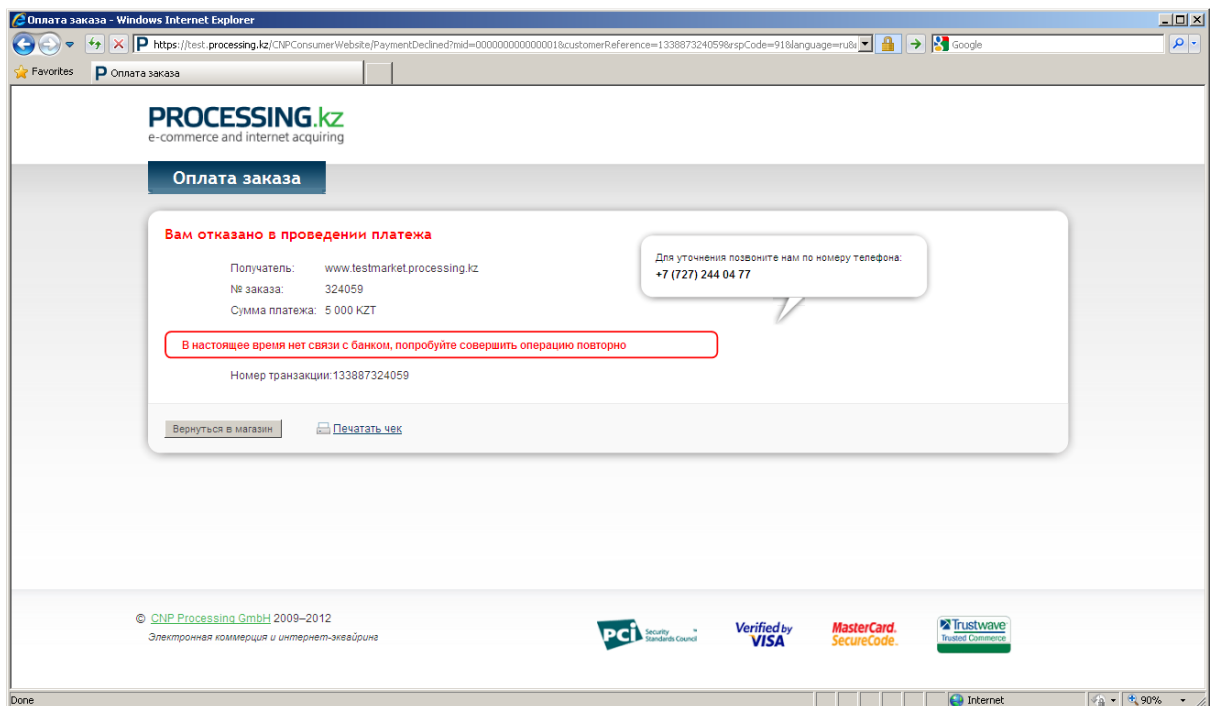
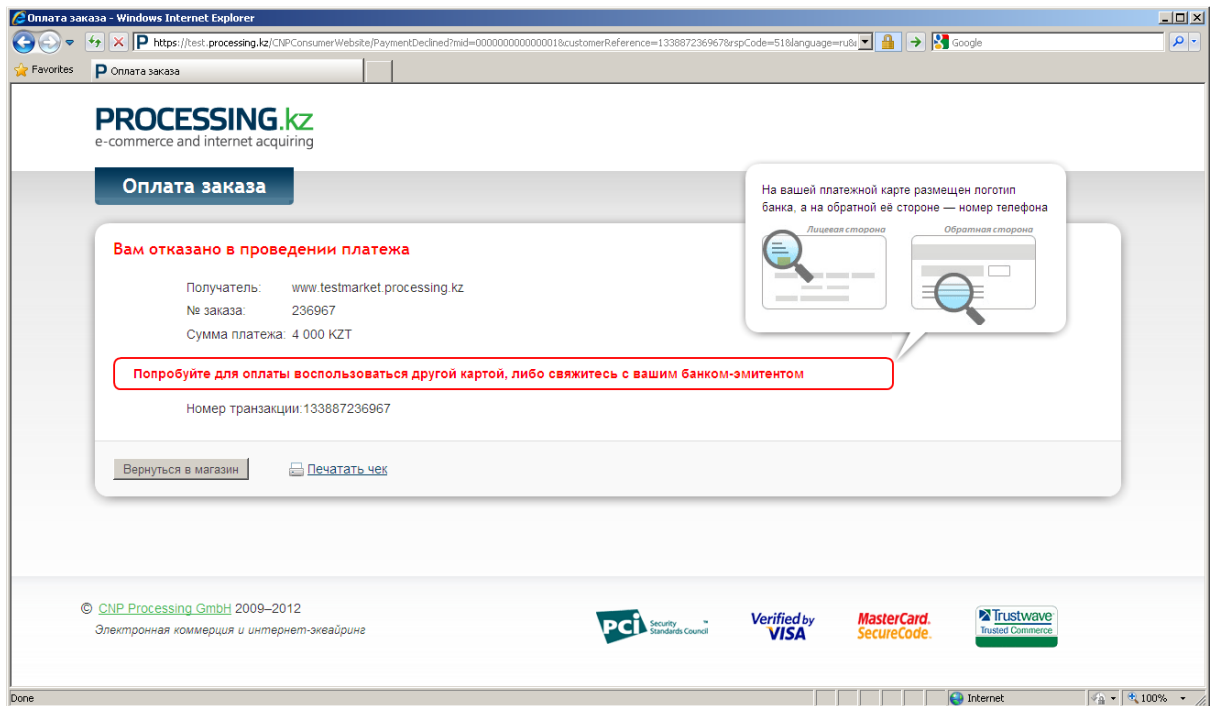
Графический пользовательский интерфейс платёжной страницы оформлен по умолчанию по предпочтениям CNP Processing GmbH (см. рис.).











## Приложение A 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: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: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 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: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.CNPMerchantWeb
ServiceHttpsSoap11Endpoint/">
</wsdl:port>
<wsdl:port name="CNPMerchantWebServiceHttpSoap11Endpoint"
binding="axis2:CNPMerchantWebServiceSoap11Binding">
<soap:address
location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebS
erviceHttpSoap11Endpoint/">
</wsdl:port>
<wsdl:port name="CNPMerchantWebServiceHttpSoap12Endpoint"
binding="axis2:CNPMerchantWebServiceSoap12Binding">
<soap12:address
location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebS
erviceHttpSoap12Endpoint/">
</wsdl:port>
<wsdl:port name="CNPMerchantWebServiceHttpsSoap12Endpoint"
binding="axis2:CNPMerchantWebServiceSoap12Binding">
<soap12:address
location="https://test.processing.kz:8443/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWeb
ServiceHttpsSoap12Endpoint/">
</wsdl:port>
<wsdl:port name="CNPMerchantWebServiceHttpEndpoint" binding="axis2:CNPMerchantWebServiceHttpBinding">
<http:address
location="http://test.processing.kz:8080/CNPMerchantWebServices/services/CNPMerchantWebService.CNPMerchantWebS
erviceHttpEndpoint/">
</wsdl:port>
<wsdl:port name="CNPMerchantWebServiceHttpsEndpoint" binding="axis2:CNPMerchantWebServiceHttpBinding">

```

```
<http:address
location="https://test.processing.kz:8443/CNPMerchantWebServices/services/CNPMerchantWebServiceHttpsEndpoint/" />
</wsdl:port>
</wsdl:service>
  </wsdl:definitions>
```

## Приложение В Test Connectivity (Тестирование соединения)

1. Тестовые веб-сервисы доступны по ссылке:

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

2. Тестовый Merchant Web Portal для управления транзакциями доступен по ссылке:

<https://testportal.processing.kz/CNPMerchantWebPortal>

Логин и пароль по запросу на [integration-support@processing.kz](mailto:integration-support@processing.kz)

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).