

**РЕФЕРАТ**

Тема выпускной квалификационной работы: «Разработка интерфейса для взаимодействия с Системой Быстрых Платежей (СБП) через банк-агент для платежей по QR кодам на подсистеме TranzAxis ServiceBus».

Ключевые слова: ПЛАТЁЖНАЯ СИСТЕМА, ИНТЕРФЕЙС,

QR-КОД, ТРАНЗАКЦИЯ, БАНК, ИНВОЙС

Пояснительная записка содержит 65 страниц без приложения, 27 иллюстраций, 10 информационных источников.

Объектом исследования в данной выпускной квалификационной работе являются платежные системы. Предмет исследования – интерфейсы платёжных систем.

Цель данной работы – разработка интерфейса класса ServiceBus для обработки платежей, инициированных при помощи Системы Быстрых Платежей Национальной Системы Платежных Карт России.

В пояснительной записке изучены особенности работы платёжных систем и интерфейсов. Рассмотрены проектные решения по разработке интерфейса класса ServiceBus. Проведена оценка результатов опытной эксплуатации интерфейса на системе клиента, что описано в отзыве менеджера проекта. Программное решение реализовано с использованием языка Java. Все инструменты для взаимодействия с пользователем предоставила система TranzAxis компании ООО «Компас Плюс». В результате создан интерфейс класса ServiceBus, который сотрудники и клиенты компании ООО «Компас Плюс» могут использовать в своей работе.

**ОТЗЫВВ**

на выпускную квалификационную работу, выполненную

обучающимся группы АВп-18-1

института энергетики и автоматизированных систем

Кошелевым Александром Евгеньевичем

на тему «*Разработка интерфейса для взаимодействия с Системой Быстрых Платежей (СБП) через банк-агент для платежей по QR кодам на подсистеме TranzAxis ServiceBus*»

В настоящее время в экономике стран большую роль играют финансовые посредники, которые предоставляют большой спектр различных сервисов, с другой стороны расширяется сеть частных и коммерческих банков, обладающих собственным программным обеспечением. Такое разнообразие затрудняет организацию межбанковского взаимодействия и требует разработки специализированного программного обеспечения, выполняющего роль интеграционной системы. Одной из таких систем является TranzAxis разработки компании CompassPlus. Платформа TranzAxis поддерживает работу с различными платежными системами, которые поддерживают различные виды программных продуктов, в том числе, систему быстрых платежей (СБП). Для расширения функциональных возможностей платформы TranzAxis возникла необходимость разработки программного приложения для обработки платежных транзакций через банк-агент инициированных QR кодами системы быстрых платежей национальной системы платежных карт по условиям, описанным в TSD банка-клиента.

Выпускная квалификационная работа выполнена в соответствии с заданием и имеет логическое изложение текстового материала, что показывает владение автором навыками подготовки технического документа.

В выпускной квалификационной Кошелева А.Е. рассмотрены структура различных платежных систем и описаны возможности платформы TranzAxis. Выполнана проектная разработка интерфейса для взаимодействия с Системой Быстрых Платежей подсистемы TranzAxis ServiceBus. Выполнена разработка интерфейса и проведена опытно-промышленная эксплуатация. Приведено подробное описание процесса работы с программным модулем.

В настоящее время, разработка Кошелева А.Е. принята для внедрения у клиентов компании CompassPlus.

В процессе работы над выпускной квалификационной работой Кошелев А.Е. показал умение применять полученные в процессе обучения знания на практике. Выпускная квалификационная работа выполнена в соответствии с требованиями СМК-О-СМГТУ-36-20.

Выпускная квалификационная работа, выполненная Кошелевым Александром Евгеньевичем, заслуживает оценки «отлично», а сам выпускник заслуживает присвоения квалификации «Бакалавр» по направлению Направление 09.03.01 Информатика и вычислительная техника профиль Автоматизированные системы обработки информации и управления

Содержание

[Введение 8](#_Toc106724348)

[1 Характеристика интерфейса для взаимодействия с Системой Быстрых на подсистеме TranzAxis ServiceBus 10](#_Toc106724349)

[1.1 Общая характеристика платежных систем 10](#_Toc106724350)

[1.1.1 Структура платёжных систем 10](#_Toc106724351)

[1.2 Национальная Система Платежных Карт 15](#_Toc106724352)

[1.3 Система Быстрых Платежей 15](#_Toc106724353)

[1.4 Платформа TranzAxis 17](#_Toc106724354)

[2 Проектные решения Разработки интерфейса для взаимодействия с Системой Быстрых Платежей на подсистеме TranzAxis ServiceBus 25](#_Toc106724355)

[2.1 Общие сведения об интерфейсе на подсистеме TranzAxis Servicebus 25](#_Toc106724356)

[2.1.1 Общие требования к системе 25](#_Toc106724357)

[2.1.2 Подсистема Servicebus 25](#_Toc106724358)

[2.2 Сценарии работы 27](#_Toc106724359)

[2.2.1 Сценарий формирования QR-кода 28](#_Toc106724360)

[2.2.2 Сценарий запроса информации по QR-коду 29](#_Toc106724361)

[2.2.3 Сценарий создания платежного инвойса 30](#_Toc106724362)

[2.2.4 Сценарий оплаты инвойса 31](#_Toc106724363)

[2.2.5 Сценарий запроса на возврат 32](#_Toc106724364)

[2.2.6 Сценарий подтверждения возврата 33](#_Toc106724365)

[2.3 Структура входной и выходной информации 34](#_Toc106724366)

[2.4 Схема работы интерфейса 35](#_Toc106724367)

[2.5 Основные функции 36](#_Toc106724368)

[2.6 Выводы по второй главе 37](#_Toc106724369)

[3 Результаты опытной эксплуатции интерфейса для взаимодействия с Системой Быстрых Платежей на подсистеме TranzAxis ServiceBus 38](#_Toc106724370)

[3.1 Создание нового конвейера ServiceBus в системе TranzAxis 38](#_Toc106724371)

[3.2 Общие настройки конвейера 40](#_Toc106724372)

[3.3 Коннекторы узлов 42](#_Toc106724373)

[3.4 Создание диаграммы конвейера 44](#_Toc106724374)

[3.5 Обработка исходящих запросов 47](#_Toc106724375)

[3.5.1 Узел RTP Pipeline Online Outgoing Interface 47](#_Toc106724376)

[3.5.2 Узел Transaction Request Processor 50](#_Toc106724377)

[3.5.3 Узел Network Channel – Client 51](#_Toc106724378)

[3.6 Обработка входящих запросов 54](#_Toc106724379)

[3.6.1 Узел Network Channel – Server 54](#_Toc106724380)

[3.6.2 Узел JML Processor 57](#_Toc106724381)

[3.6.3 Узел RTP Pipeline Online Incoming Interface 58](#_Toc106724382)

[3.7 Выводы по третьей главе 62](#_Toc106724383)

[Заключение 63](#_Toc106724384)

[Список использованных источников 65](#_Toc106724385)

[Приложение А 66](#_Toc106724386)

# Введение

Процесс аккумулирования и размещения финансовых ресурсов непосредственно связан с функционированием рынков капитала и деятельностью финансовых посредников. Сформировавшаяся как отдельное научное направление в середине XX в. теория финансового посредничества продолжает развиваться и обогащаться. На каждом этапе своего развития усложняется роль финансовых посредников в экономике, расширяются их функции, появляются новые типы финансовых посредников. На финансовых рынках все более заметную роль начинают играть коммерческие и частные банки, которые не имеют возможности самостоятельно разрабатывать программное обеспечение для своих производств.

CompassPlus является одной из компаний предоставляющих услуги проектирования, разработки и внедрения банковского программного обеспечения. На данный момент самой прогрессивной и актуальной системой, на базе которой строится система под конкретные нужды заказчика, является TranzAxis.

Для обработки уникальных транзакционных операций, не входящих в общую поставку TranzAxis, в системе существует отдельный класс объектов, называемый User Defined Transaction Type (UDTT). Данный тип операций реализуют программисты-кастомизаторы, руководствуясь технической документацией системы клиента, называемой Technical Structure Document (TSD). Но не редки случаи, когда несколько запрашиваемых на разработку клиентом типов транзакционных операций можно объединить в одну обработку. Помехой к упрощению и уменьшению количества разрабатываемых операций является маршрутизация транзакции внутри системы TranzAxis, а также протоколы, в соответствии с которыми из системы клиента приходят транзакционные запросы и возвращаются ответы после обработки.

Чаще всего запросы из внешних систем в TranzAxis приходят в разных форматах, в зависимости от используемого клиентом протокола передачи сообщений. Ранее приходилось писать дополнительные функции для перевода пришедшего сообщения в транзакционный запрос TranzAxis и перевода транзакционного ответа TranzAxis обратно в сообщение, воспринимаемое без ошибок протоколом клиента. Для оптимизации данных работ в TranzAxis был разработан новый тип объектов, предоставляющий инструментарий для упрощения и автоматизации межсистемного взаимодействия, названный ServiceBus. Внутри системы строится граф маршрутизации транзакции от принятия запроса из внешней системы, его перевода в транзакционный запрос TranzAxis, обработки, перевода ответа TranzAxis обратно в протокол, используемый внешней системой и отправки ответа во внешнюю систему в верном формате.

TranzAxis поддерживает работу с множеством платежных систем. На данный момент основной платежной системой на территории Российской Федерации является платежная система МИР, которая поддерживает множество платежных продуктов, одним из которых является система быстрыРазработка модуля системы CSIA для мониторинга описания работ по разработке ПО в системах контроля версий и управления проектамих платежей. Система быстрых платежей (СБП) — сервис, с помощью которого можно совершать межбанковские переводы по номеру мобильного телефона круглосуточно, без праздников и выходных. Комиссии за такие переводы низки или вовсе отсутствуют. К системе уже подключены более 200 банков, включая крупнейшие. СБП также позволяет оплачивать покупки в том числе по Quick Response (QR) коду и получать выплаты от организаций. СБП разработана Банком России и АО «Национальная система платежных карт». Банк России — оператор и расчетный центр Системы, а НСПК — операционный платежный и клиринговый центр.

Рассматриваемый в выпускной работе программный продукт является реализацией объекта ServiceBus в системе TranzAxis, для обработки платежных транзакций через банк-агент инициированных QR кодами системы быстрых платежей национальной системы платежных карт по условиям описанным в TSD банка-клиента.

# Характеристика интерфейса для взаимодействия с Системой Быстрых на подсистеме TranzAxis ServiceBus

## Общая характеристика платежных систем

### Структура платёжных систем

Платёжная система представляет собой совокупность финансовых учреждений, а также набор процедур и инструментов, которые используются для перевода безналичных денежных средств между субъектами (физическими и юридическими лицами) [2]. С платёжной системой тесно связаны такие понятия, как:

1) Транзакция – это банковская операция, которая влечёт за собой изменение состояния клиентского счёта (например, перевод денежных средств с одного счёта на другой, снятие наличных в банкомате, оплата товара или услуги и др.) [1];

2) Платёж – это погашение обязательств одного участника перед другим путём исполнения этим участником необходимого требования;

3) Расчёт – это окончание операции платежа и поступление денег на счёт получателя;

4) Окончательный расчёт – это погашение всех взаимных обязательств между участниками платёжной системы.

Платёжная система реализует три основных процесса, на основе которых строится работа всей системы [3]:

1) Процесс инициирования платежа, осуществляемый посредством платёжных инструментов (пластиковых карточек, электронных кошельков, мобильных приложений и т.д.). Этот процесс заключается в поручении обслуживающему банку перевода денег со счёта одного участника на счёт другого участника;

2) Обмен и передача платёжных инструментов между участниками;

3) Расчёт между участниками.

Любая платёжная система обладает целостной упорядоченной структурой, которая представляется в виде, как показано на рисунке 1.

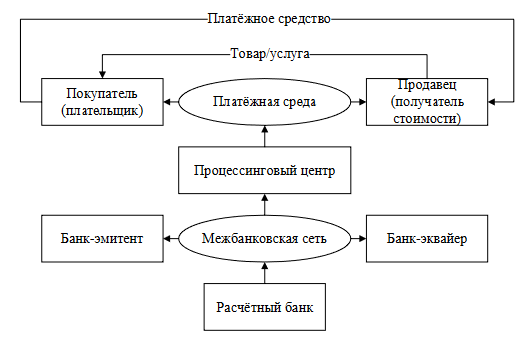


Рисунок 1 – Структура платёжной системы с участием кредитной организации

Платёжная система состоит из десяти элементов [2]:

1) Платёжного средства – им может являться банковская карта, электронный кошелёк;

2) Товара или услуги;

3) Покупателя – покупателем является физическое или юридическое лицо, которое осуществляет оплату при помощи платёжного средства с целью приобретения товара или услуги;

4) Продавца – продавцом является физическое или юридическое лицо, которое предоставляет покупателю товар или услугу за определённую плату;

5) Платёжной среды – важное условие для обеспечения существования электронной коммерции. Платёжная среда позволяет обрабатывать сделки в реальном времени в стране и за её пределами, на международных рынках. Примером платежных сред могут служить различные интернет магазины;

6) Процессингового центра – он играет роль посредника между субъектами расчётов. Процессинговый центр отвечает за безопасность транзакций (так как является третьим лицом, которое не заинтересовано в извлечении выгоды нелегальным путём) и быстроту проведения расчётных операций;

7) Банка-эмитента (или эмиссионного банка) – это банк, который выпустил и обслуживает карту клиента;

8) Банка-эквайера – это банк, через который осуществляется проведение операций по карте. Он предоставляет клиентам специальные терминалы для использования полного функционала карт. В некоторых ситуациях банк-эквайер может выступать и в качестве банка-эмитента;

9) Межбанковской сети – это такая система, которая объединяет банкоматы разных банков: к примеру, имея карту, выпущенную одним банком, клиент может воспользоваться банкоматом другого банка, хотя перечень доступных операций с картой в таком случае будет ограничен;

10) Расчётного банка – это банк, который осуществляет взаиморасчёты между участниками платёжной системы.

Так как кредитная организация (банк) не является обязательной составляющей платёжной системы [5], то структура представляется в общем виде, изображённом на рисунке 2.

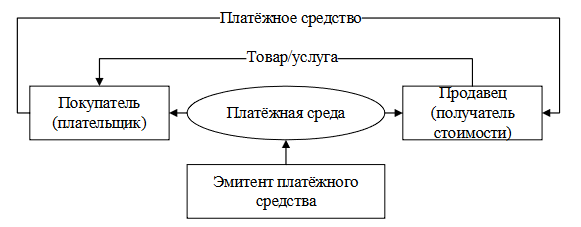


Рисунок 2 – Структура платёжной системы в общем виде

Количество элементов структуры сокращается до шести [8]. Вместо межбанковской сети и её составляющих выступает эмитент платёжного средства – юридическое лицо, гарантирующие исполнение обязательств, которые связаны с использованием выпущенного им платёжного средства (например, пластиковой карты).

Взаимодействие между участниками платёжной системы осуществляется следующим образом [4]:

1) Покупатель и продавец становятся участниками одной и той же платёжной системы или же взаимосвязанных систем, чтобы осуществить перевод денежных средств;

2) Покупатель и продавец открывают счета в платёжной системе, в которой они являются участниками;

3) Покупатель вносит деньги/их эквивалент на свой счёт;

4) По распоряжению покупателя или другим способом, оговоренным между покупателем и продавцом, платёжная система производит платёж.

Без определённых условий взаимодействие участников платёжной системы не представляется возможным. Поэтому при построении систем должны учитываться семь принципов:

1) Обеспечение функционирования платёжной системы в рамках права (деятельность платёжной системы должна осуществляться в рамках закона и иметь ясную и чёткую юридическую базу, подкреплённую действующим правом);

2) Управляемость (платёжная система должна иметь ясную и чёткую систему управления, благодаря которой обеспечивается эффективность, стабильность и безопасность системы);

3) Комплексное управление рисками (для предотвращения убытков платёжная система должна иметь систему управления рисками различных видов: инвестиционными, кредитными, правовыми, операционными, коммерческими и др.);

4) Обеспечение качества расчётов (означает завершённость расчётов, проведение расчётов с помощью средств коммерческих банков или центрального банка, проведение неденежных расчётов);

5) Равный доступ к участию в системе (обеспечивает иерархию в платёжной системе путём разбиения участников на группы с определёнными обязанностями и правами);

6) Прозрачность (информация о деятельности платёжной системы должна быть полной и общедоступной; это позволяет выявлять проблемы в деятельности, пагубно влияющие на развитие экономики страны);

7) Эффективность (деятельность платёжной системы должна давать положительный эффект, который может выражаться не только в денежных средствах, но и в высокой скорости расчётов и обслуживания клиентов, высоком качестве осуществления платежей, полной удовлетворённости разнообразных потребностей клиентов и т.д.).

В осуществлении эффективной деятельности платёжной системы и соответствии вышеуказанным принципам помогают различные элементы, которые представлены на рисунке 3.



Рисунок 3 – Виды элементов платёжной системы

Как видно, платёжная система состоит из институциональных и инфраструктурных элементов. К институциональным элементам относятся кредитные институты, а также совокупность разного рода правил, законов, стандартов, образующих основу для деятельности платёжной системы. Инфраструктурные же элементы включают комплекс систем, платежных инструментов и процедур, позволяющих осуществлять платёжные операции [6].

Таким образом, платёжная система – это большая система со сложной структурой и тесно взаимосвязанными составляющими, каждая из которых выполняет свои определённые функции.

## Национальная Система Платежных Карт

23 июля 2014 года было создано акционерное общество «Национальная система платёжных карт» (АО НСПК), перед которым были поставлены две задачи:

1. Создание операционного и клирингового центра для обработки внутрироссийских операций по картам международных платёжных систем;
2. Выпуск и продвижение национальной платёжной карты.

Генеральным директором АО НСПК был назначен Владимир Комлев, ранее возглавлявший АО «Компания объединённых кредитных карточек» (UCS). С апреля 2015 года все внутрироссийские транзакции MasterCard полностью переведены на процессинг НСПК [9].

НСПК подписала несколько соглашений о совместном выпуске карт с международными платёжными системами:

1. 19 июня 2015 года с MasterCard по картам системы Maestro;
2. 7 июля 2015 года с японской системой JCB.
3. Июль 2015 с американской системой American Express.
4. 15 декабря 2015 года НСПК объявила о выпуске первых карт «Мир» семью российскими банками.

Согласно финансовому отчёту, чистая прибыль АО НСПК за 2015 год составила 1,2 млрд рублей, оборот — 2,9 млрд рублей.

В 2017 году Национальная система платёжных карт вошла в состав Совета по стандартам безопасности данных индустрии платёжных карт (PCI SSC).

## Система Быстрых Платежей

Система регистрации участников Системы Быстрых Платежей [5] позволяет однозначно определить участника по соответствующему идентификатору и обеспечивает взаимодействие с расчетным центром для проведения расчетов по счетам участников Системы.

Центральный репозиторий является централизованной базой хранения информации, необходимой для адресации запросов отправителей и получателей

платежей. Данные репозитория не передаются за рамки Системы Быстрых Платежей.

Система дополнительных сервисов обеспечивает возможность подключения к Системе Быстрых Платежей дополнительных инновационных сервисов участников.

Система инвойсинга поддерживает возможность направления запроса платежа, выставления счетов и их оплату.

Система взаимодействия с Расчетным центром обеспечивает расчеты по счетам Участников в Платежной Системе Банка России.

Система фрод-мониторинга обеспечивает дополнительный уровень защиты участников Системы Быстрых Платежей и потребителей ее услуг [7].

Основные задачи, которые ставятся перед системой быстрых платежей:

1. В условиях активного роста электронной коммерции и развития цифрового взаимодействия между участниками рынка удобные и быстрые платежи становятся необходимостью как для населения, так и для организаций;
2. Системы, позволяющие физическим и юридическим лицам проводить платежи в режиме реального времени, уже работают более чем в 10 странах, еще около 20 стран находятся на стадии построения или подготовки к внедрению таких систем;
3. На российском рынке клиентам банков доступны удобные и, как правило, бесплатные платежные сервисы внутри одного банка;
4. В России назрела необходимость создания инфраструктуры, которая позволит проводить платежи в онлайн-режиме, в том числе, в случае, если счета отправителя и получателя находятся в разных банках.

Создание инфраструктуры быстрых платежей — один из ключевых проектов по развитию розничного платежного пространства на российском рынке.

Сервисы Системы Быстрых платежей ориентированы на физических лиц – пользователей мобильных устройств, на компании в сфере электронной и мобильной коммерции (e-commerce и m-commerce), на компании, предоставляющие розничные продукты и услуги.

Основные задачи Системы Быстрых Платежей:

1. Простота и удобство инициации платежа по номеру мобильного телефона, а в дальнейшем – по e-mail и другим упрощенным идентификаторам;
2. Доступность сервиса быстрых платежей в режиме 365/7/24;
3. Мгновенное зачисление денежных средств получателю. Информация об исполнении платежа в онлайн режиме.

## Платформа TranzAxis

TranzAxis – это специализированный инструмент для разработки решений, помогающих автоматизировать процессы розничного банковского бизнеса. Платформа включает в себя такие необходимые и базовые модули, как [10]:

1. Поддержка процессингового центра,
2. Электронные платежи,
3. Учет и анализ деятельности,
4. Составление отчетов,
5. Индикаторы подозрительной (мошеннической) деятельности

TranzAxis предоставляет банку готовые решения, ориентированные на классические банковские продукты и услуги - кредитные и депозитные (вкладные) продукты, услуги, связанные с осуществлением безналичных расчетов, с приемом и выдачей наличных денег, банковские карточные продукты, консультационные и информационные услуги, в том числе по электронным каналам и каналам самообслуживания.

TranzAxis предоставляет пользователям широкий набор инструментальных

средств разработки приложений (конструкторов, дизайнеров, редакторов, отладчиков и т.п.), а также набор специализированных компонент, необходимых при реализации приложений (универсальных классов и прикладных служб).

Ниже на рисунке 4 в виде диаграммы представлен общий механизм и возможности функционала платформы TranzAxis.



Рисунок 4 – Платформа TranzAxis

Механизм TranzAxis предоставляет следующие возможности:

1. Проектирование, адаптация и разработка самостоятельных приложений.
2. Написание функций для расширения приложений.
3. Внедрение базовых возможностей во внешние системы, таких как:
   1. Передача сообщений из внешней системы в ядро клиентской банковской системы;
   2. Формирование атомарных транзакционных операций;
   3. Сохранение данных в базе данных клиентской банковской системы и хранение их в течение установленного стандартами промежутка времени для формирования возвратов и предоставления данных об операциях правоохранительным органам при возникновении необходимости;
   4. Создание и удаление объектов финансовой системы клиента (контракты, карты, корпоративные клиенты, индивидуальные клиенты и т.п.).
4. Гибкая настройка и изменение стандартно-базовых приложений под любые стандарты бизнес-требований клиента.

С помощью TranzAxis возможно максимально точно задать и описать организационные структуры финансовых институтов начиная от ее геолокационных составляющих (филиалы, кассы операционного обслуживания, отделения), заканчивая департаментами, подразделениями и локальными точками. Балансовые филиалы банков описываются в системе как отдельные финансовые институты, небалансовые - как отделения финансового института (банка). Финансовые институты, деятельность которых необходимо автоматизировать, описываются в системе как локальные финансовые институты, такие институты будут обладать собственными подобъектами: транзакциями, счетами, терминалами и т.п. Программный продукт TranzAxis помогает реализовать связи с внешними системами для внешних финансовых институтов. Гибкость системы помогает создавать решения под различные ситуации, такие как:

1. Автоматизация отдельной кредитно-финансовой организации, которая входит в состав розничного банковского обслуживания, имеющая под собой балансовые филиалы и общую клиентскую базу. Также, каждый филиал банка может иметь непосредственно древовидную иерархическую структуру небалансовых отделений, которые наследуют свойства, определенные для филиала или для системы в целом. Это можно сравнить с наследованием, где общая логика описывает деятельность банка, а финансовые институты – характерные для них особенности, характерные для деятельности отделений данного банка.
2. Автоматизация деятельности нескольких независимых кредитнофинансовых организаций, в каждой из которых отсутствуют балансовые филиалы. При этом, каждый из финансовых институтов также может включать в себя собственное множество небалансовых отделений. В данном случае каждый из институтов будет настраиваться в соответствии с собственными правилами. Доступ пользователей к информации, связанной с финансовым институтом, ограничивается через механизм ролей, реализованный в системе.

Основные объекты системы следующие:

1. Физические и юридические лица - это реальные или потенциальные клиенты финансовых институтов, а также их финансовые партнеры. Средства системы позволяют в любой момент времени получить полную информацию по любому клиенту, историю его операций, список транзакций и т.п. Клиентская база может вестись не только по лицам, которые являются клиентами финансовой организации, но и по любым субъектам, оказавшимся в поле ее деятельности;
2. Контракты. Взаимодействие финансового института и клиента описывается в системе через механизм контрактов. Контракт в TranzAxis не обязательно подразумевает формально заключаемый контракт между клиентом и финансовым институтом - как правило, это системный объект, характеризующий определенную сторону взаимоотношений между клиентом и институтом. В связи с этим выделяют два класса контрактов - финансовые и сервисные. Сервисные контракты используются для описания технических аспектов взаимоотношений банка с клиентом. Финансовые контракты регулируют финансовые аспекты взаимоотношений с клиентом;
3. Финансовые контракты - финансовые отношения, а именно: правила проведения финансовых операций, принципы выбора процентных ставок, начисление и капитализация процентов, определение сроков погашения кредитов, общие принципы взаиморасчетов и т.п., - описываются в системе через финансовые контракты. Правила отражения средств на счетах, то есть бухгалтерская логика, регулируется схемой бухгалтерского учета. Схемы бухгалтерского учета могут настраиваться самостоятельно пользователями TranzAxis. Также могут использоваться стандартные схемы бухгалтерского учета, поставляемые с соответствующими классами финансовых контрактов;
4. Сервисные контракты - технические аспекты обслуживания клиентов - издание и обслуживание карт, техническое обслуживание устройств, схемы оповещения клиента по финансовым операциям и т.п., - определяются с помощью сервисных контрактов;
5. Транзакции - транзакции в системе TranzAxis могут приниматься, обрабатываться, генерироваться, инициироваться, перенаправляться, модифицироваться и редактироваться;
6. Проводки - в TranzAxis для отражения движения средств используется смешанная схема. Каждая проводка существует в рамках прикладной транзакции и состоит из нескольких частей. Основные части проводки описывают изменения баланса дебетуемого и кредитуемого счета, определяют счет и суммы в валюте проводки, в расчетной валюте и в валюте счета. В конверсионных проводках может быть еще одна специфическая часть - курсовая разница;
7. Счета - Счета в системе используются для организации финансового учета и контроля. Счета могут принадлежать либо финансовому институту, либо финансовому контракту, либо терминалу. Бухгалтерская классификация счетов, а также их технические атрибуты определяются в системе через план счетов. План счетов может быть общим (распространяться на все институты и контракты) или составляться индивидуально по каждому финансовому институту;
8. Ограничения и лимиты - ограничение и контроль допустимости исполнения транзакций производится с помощью аппарата ограничений и лимитов. Ограничение - это запрет на исполнение транзакций определенного вида при определенных условиях (например, запрет на проведение транзакций по картам определенных типов). Лимит - это подкласс ограничения, который используется для снижения финансовых рисков при проведении финансовых операций. Лимит устанавливает верхний допустимый предел на сумму или на количество транзакций определенного вида для участника.
9. Тарифы - TranzAxis имеет единый универсальный аппарат тарификации, который используется для расчета: комиссий на транзакции, периодических плат, процентных ставок, эквайринговых доплат. Условия взимания и параметры для вычисления конкретной комиссии, платы или процентной ставки определяются через специальный объект системы - тариф. Для тарифов поддерживается версионность - каждая версия тарифа действует в пределах определенного диапазона операционных дней и периода времени. Параметры версии тарифа определяют правила для применения и расчета тарифа (валюту, группу курсов обмена, знак тарифа, условие применимости и т.п.). Изменение параметров тарифа возможно только в рамках новой версии тарифа, для которой должны быть запланированы день и время начала ее использования. Протестировать использование тарифов можно с помощью встроенных средств тестирования. Для комплексного описания правил тарификации используются тарифные планы. Тарифные планы могут связываться отношениями наследования, образуя древовидную структуру, по которой осуществляется полное или выборочное наследование тарифов. При наследовании параметры тарифа из родительского тарифного плана можно перекрывать и изменять на заданную величину или процент;
10. Терминалы - под терминалом в системе понимается любое аппаратное или программное устройство, при помощи которого инициируются транзакции. TranzAxis предоставляет средства для гибкой настройки технических и операционных параметров терминалов, а также возможности мониторинга активности терминала, просмотра транзакций и событий по терминалу. В системе терминалы могут принадлежать финансовому институту - банковские терминалы, или торговцу - торговые терминалы. Обслуживание торговых терминалов выполняется на основе сервисного контракта торговца; данный контракт осуществляет тарификацию транзакций и взимает периодические платы за обслуживание терминала. Для расчетов с торговцем используется финансовый контракт взаиморасчетов, связанный с сервисным.   
    Особенностью банковских терминалов является наличие собственных счетов, которые описываются с помощью схем кассовых счетов. Схема кассовых счетов содержит параметры, регулирующие механизм учета счетов приема и выдачи (раздельно или совместно), а также позволяющие при необходимости динамически создавать недостающие счета.   
    Интерактивные операции терминала позволяют пользователю в диалоговом режиме выполнять действия над терминалом или от имени терминала. Система мониторинга терминальной сети позволяет оператору наблюдать за состоянием терминалов, а также управлять ими. Параметры мониторинга терминалов настраиваются администратором системы;
11. Токены - это средства/объекты идентификации и аутентификации клиента. В качестве токена может использоваться пластиковая карта, регистрационное имя (login), цифровой сертификат и т.д. Каждый тип токена поддерживает определенный набор способов аутентификации. Например, EMV-карта поддерживает аутентификацию при помощи PIN, CVV, CVV2, ARQC и т.д., а login – только аутентификацию по паролю. Для аутентификации клиента в рамках одной транзакции могут использоваться несколько токенов и несколько способов аутентификации. Политика аутентификации настраивается отдельно для каждого подмножества типов транзакций. Текущее состояние токена характеризуется его статусом, который влияет на процесс аутентификации и регулирует доступные для данного токена операции. В системе реализована подсистема логистики токенов, позволяющая управлять их жизненным циклом с момента заказа и до момента выдачи клиенту или уничтожения. Этапы жизненного цикла токена (изготовление, генерация криптографических величин, обработка, доставка в отделения выдачи и т.д) описываются в системе посредством фаз жизни токена;
12. Интерфейсы – инструменты для организации передачи данных между системами. Интерфейсы используются для:
    1. Передачи транзакций между финансовыми институтами и системами (online, clearing).
    2. Загрузки транзакций из другой системы в пределах одного финансового института (например, с терминального хоста).
    3. Выгрузки транзакций во внешние системы (extract).
    4. Выгрузки и загрузки различных объектов во внешние системы (refresh).

Интерфейсы различаются направлением (incoming/outgoing), типом (online/offline) и протоколом:

1. Outgoing Online интерфейсы непосредственно участвуют в обработке транзакции. Транзакция может передаваться по двухфазной (online и offline) или однофазной схеме (либо online, либо offline).
2. Incoming Online интерфейсы создают и исполняют транзакции в системе.
3. Outgoing Offline интерфейсы создают пакеты служебных файловых транзакций, в результате выгрузки, которых создаются клиринговые, extract, refresh файлы.
4. Incoming Offline интерфейсы загружают файлы и создают пакеты транзакций.

Интерфейсы подключаются к транзакции сервисными Interchange контрактами. Эти контракты выбирают подходящие интерфейсы, контролируют ограничения и лимиты. Основным недостатком TranzAxis является отсутствие интерфейсов для некоторых финансовых механизмов. Одним из таких является механизм поддержки системы быстрых платежей.

# Проектные решения Разработки интерфейса для взаимодействия с Системой Быстрых Платежей на подсистеме TranzAxis ServiceBus

## Общие сведения об интерфейсе на подсистеме TranzAxis Servicebus

### Общие требования к системе

В системе TranzAxis компании ООО «Компас Плюс» предусмотрены такие объекты, как интерфейсы. Интерфейс – это объект системы, предназначенный для выгрузки, приёма и передачи транзакций между финансовыми институтами и системами (например, международными системами Visa, MasterCard, JCB, различными коммерческими банками и т. д.) по определённым критериям и правилам. Клиент компании (к примеру, банк), через который проходит множество транзакций, устанавливает систему TranzAxis на свои рабочие станции и пользуется интерфейсом для получения необходимой по транзакциям информации, задавая собственные настройки.

Для установки TranzAxis и работы с интерфейсом пользователь должен иметь на рабочей станции одну из следующих операционных систем: Windows (7-10), Linux (Linux Mint, Ubuntu, Debian, Fedora) или Mac OS.

Также на рабочей станции необходимо установить Java версии 8 или выше. Все модули, классы объектов (в т. ч. классы интерфейсов) и прочие элементы системы TranzAxis реализованы на языке программирования Java, что обеспечивает кроссплатформенность системы, а также отсутствие необходимости разрабатывать с нуля виртуальную машину.

### Подсистема Servicebus

Подсистема Servicebus предназначена для обработки и преобразования онлайн-сообщений. При помощи данной подсистемы администратор может разрабатывать конвейеры, решающие следующие задачи:

1. Отправка запросов во внешнюю систему и получение ответов;
2. Прием запросов от внешних систем, их обработка и отправка ответов;
3. Прием сообщений от внешних систем по одному протоколу, их преобразование и отправка в другую внешнюю систему по другому протоколу;
4. И т. д.

Каждый узел может иметь несколько входных и несколько выходных коннекторов, к которым подключаются другие узлы. Коннектор характеризуется именем и типом данных. Могут использоваться следующие типы данных:

1. Bin. Двоичные данные;
2. Structure. Произвольные данные, формат которых определяется MSDL-схемой;
3. XML. Данные в формате XML со структурой, соответствующей XSD-схеме;
4. Tran. Транзакционный запрос или ответ;

Кроме данных, по конвейеру могут передаваться уведомления о событиях и ошибках.

Один и тот же конвейер может использоваться для обработки как запросов, так и ответов на них. Каждый ответ ассоциируется с соответствующим запросом и передается по тем же узлам конвейера в обратном направлении.

Узел может преобразовывать сообщение, выполнять какие-либо действия по его обработке и передавать сообщение следующему узлу.

В конвейере используются узлы следующих видов:

1. Стартовые узлы. Получают запросы от других подсистем TranzAxis или внешних систем, и передают их другим узлам.
2. Промежуточные узлы. Преобразуют и обрабатывают сообщения.
3. Узлы маршрутизации. Разновидность промежуточных узлов. Передают сообщения одному из нескольких узлов, в зависимости от определенных условий.
4. Конечные узлы. Получают ответы на запросы от других узлов и передают их другим подсистемам TranzAxis или внешним системам.

Некоторые стартовые и конечные узлы реализуются специальными системными модулями. При создании конвейера в списке модулей инстанции появляются системные модули, соответствующие стартовым и конечным узлам конвейера. Такие модули не могут быть созданы из контекста инстанции.

Настройка узлов конвейера выполняется в основном при помощи пользовательских функций, XSLT-преобразований и XPath-выражений.

## Сценарии работы

Общая схема взаимодействия представлена на рисунке 5

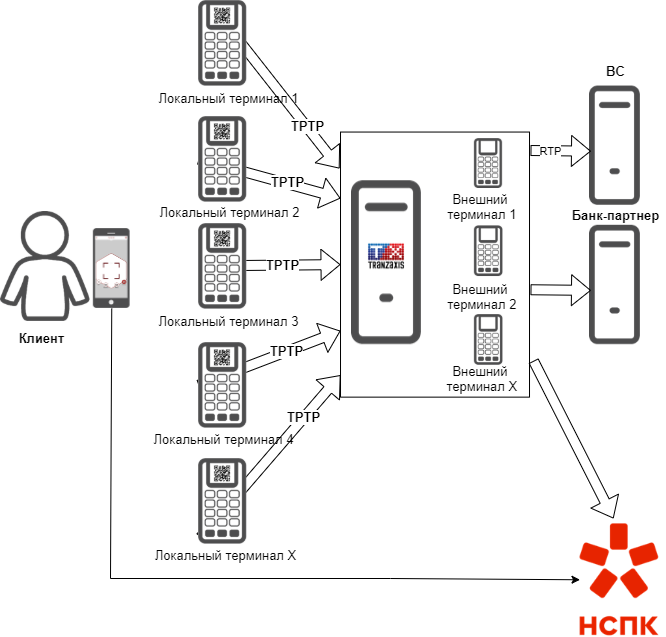


Рисунок 5 – Общая схема взаимодействия

Схема работы интерфейса представлена на рисунке 6

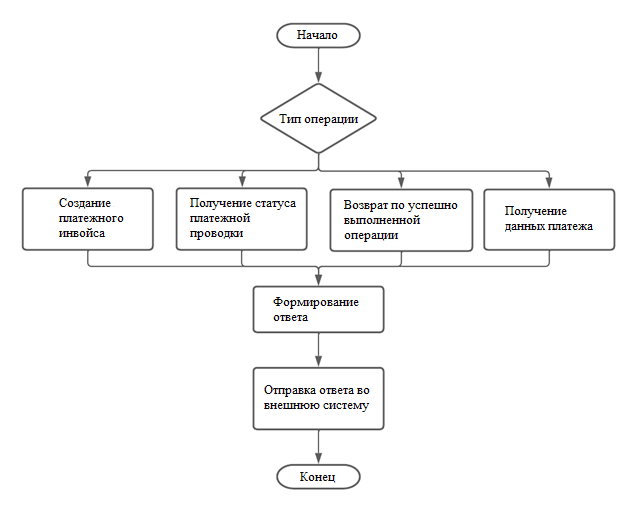


Рисунок 6 – Схема работы интерфейса

Интерфейс предусматривает шесть сценариев работы:

1. Формирование QR-кода;
2. Запрос информации по QR-коду;
3. Создание платежного инвойса;
4. Оплата инвойса;
5. Запрос на возврат;
6. Подтверждение возврата.

### Сценарий формирования QR-кода

Структура объектов и потоки обработки сценария представлена на рисунке 7. Взаимодействие начинается с формирования QR-кода [10]. Он может быть статическим, т.е. формируется для терминала один раз и прикрепляется в виде картинки у продавца, и динамическим, т.е. формируется для конкретной покупки.



Рисунок 7 – Запрос на формирование QR-кода

Алгоритм сценария включает в себя следующие действия:

1. На терминале проводится операция формирования QR-кода;
2. TranzAxis маршрутизирует транзакцию в необходимый финансовый институт;
3. TranzAxis производит поиск внешнего терминала для дочерней транзакции и формирует запрос транзакции по внешнему терминалу;
4. После того, как QR-код будет принят в НСПК, ответ от внешней системы будет содержать ссылку на QR-код, номер инвойса и статус регистрации – Created. Эти данные из ответа внешней транзакции передаются во внутреннюю и далее – на терминал. В функции корректировки ответа терминалу данные о зарегистрированном QR-коде записываются в специальный справочник “QrDictionary”;

### Сценарий запроса информации по QR-коду

Структура объектов и потоки обработки сценария представлена на рисунке 8. Данный сценарий является вспомогательным для остальных сценариев.



Рисунок 8 – Запрос информации по QR-коду

Алгоритм сценария включает в себя следующие действия:

1. Другой сценарий инициирует запрос информации по платежному QR-коду;
2. В специальном справочнике QR-кодов “QrDictionary” производится поиск необходимого кода;
3. Формируется ответ, содержащий значения параметров, перечисленных в запросе.

### Сценарий создания платежного инвойса

Структура объектов и потоки обработки сценария представлена на рисунке 9. Сценарий необходим для создания и регистрации платежной транзакции в системе клиента.



Рисунок 9 – Создание платежного инвойса

Алгоритм сценария включает в себя следующие действия:

1. QR-код отображается на терминале/размещается на картинке у торговца. Оплату инициирует Клиент, сканируя QR-код через специальное приложение Банка издателя;
2. После того, как НСПК получает запрос на оплату от Банка издателя, через BC/Банк-партнер/НСПК запрос на создание инвойса получателя для торговца попадает в TX. Создается инвойс типа C2BRecv в статусе Ready (для внешнего терминала). Также формируется транзакция Payment Single PreprocessOnly для проверки возможности проведения платежа, на которую функцией формируется такая же дочерняя транзакция для финансового контракта локального торговца (поиск локального торговца через справочник “QrDictionary”)). Институт для указания source contract (т.е. контракт, по которому будет проверяться и впоследствии выполняться дебетовая часть платежа) будет определяться, исходя из интерфейса (институт-партнер) –> Interchange контракт, где этот институт указан в качестве партнерского –> расчетный контракт для этого Interchange-контракта. Добавление source contract id реализуется в Функции корректировки статуса инвойса, назначенную на тип инвойса C2BRecv. Функция должна проверять переход статуса инвойса с null на Ready.
3. В системе клиента записывается инвойс, в состоянии ожидания оплаты.

### Сценарий оплаты инвойса

Структура объектов и потоки обработки сценария представлена на рисунке 10. Сценарий необходим для проведения оплаты и получения статуса платежа: оплачено или отклонено.



Рисунок 10 – Оплата инвойса

Алгоритм сценария включает в себя следующие действия:

1. После инициализации оплаты клиентом с помощью словаря “QRDictionary” выполняется поиск созданного ранее платежного инвойса;
2. В случае успешной проверки возможности проведения транзакции клиент проверяет данные о торговце и платеже в специальном приложении и инициирует оплату;
3. После того, как НСПК получает подтверждение оплаты от Банка издателя, через BC/Банк-партнер/НСПК запрос на изменение статуса попадает в систему клиента. Статус инвойса меняется на Paid, при этом формируется транзакция, которая инициирует проводку со счета контракта взаиморасчетов BC/Банк-партнер/НСПК на счет внешнего торговца;
4. Для транзакции по внешнему терминалу формируется дочерняя транзакция, которая копирует запрос и подменяет данные о терминале и финансовом институте-инициатора. Проводка формируется со счета контракта взаиморасчетов на счет локального торговца;
5. В случае неуспешной оплаты TX получает из BC/Банк-партнер/НСПК транзакцию, которая меняет статус инвойса на Declined.

### Сценарий запроса на возврат

Структура объектов и потоки обработки сценария представлена на рисунке 11. Возврат инициируется на локальном терминале.

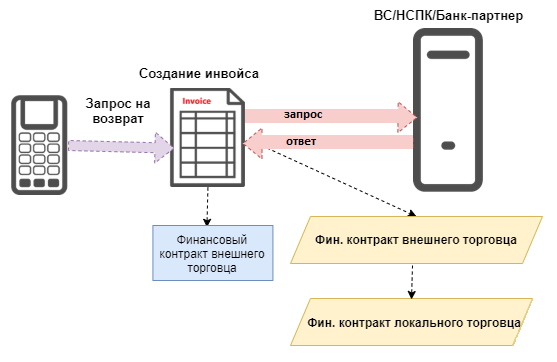


Рисунок 11 – Запрос на возврат

Алгоритм сценария включает в себя следующие действия:

1. На терминале проводится операция возврата. Запрос от терминала помимо стандартных полей должен включать в себя следующую информацию: сумма возврата, номер инвойса, номер получателя возврата, идентификатор банка, куда сделать возврат;
2. Система клиента формирует транзакцию, создается инвойс типа B2CSend в статусе Ready. При создании инвойса система клиента корректирует данные о терминале на внешний терминал. Формируется транзакция, которая передается в BC/Банк-партнер/НСПК.

### Сценарий подтверждения возврата

Структура объектов и потоки обработки сценария представлена на рисунке 12.

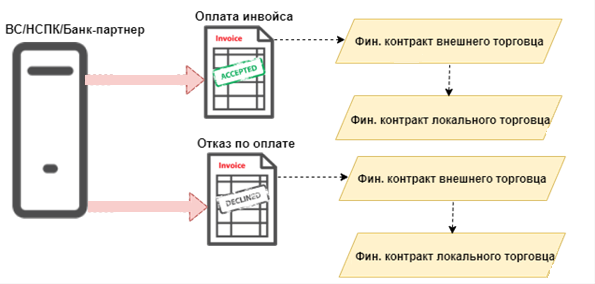


Рисунок 12 – Подтверждение возврата

Алгоритм сценария включает в себя следующие действия:

1. Терминал формирует финансовый запрос. Запрос от терминала помимо стандартных полей должен включать в себя номер предыдущей транзакции в системе клиента. Статус инвойса меняется на PaymentAuth при этом формируется транзакция, которая передается в BC/Банк-партнер/НСПК;
2. При успешной обработке транзакции из BC/Банк-партнер/НСПК приходит запрос, меняющий статус инвойса на Paid. При этом формируется транзакция, которая списывает деньги со счета внешнего торговца на счет BC/Банк-партнер/НСПК. Также формируется дочерняя транзакция для списания средств со счета локального торговца. В случае неуспешной обработки статус инвойса меняется на Declined и формируется реверс на Payback Auth для финансового контракта внешнего торговца, а затем - для локального торговца;
3. Во время проведения возврата, терминал формирует административный запрос. Формируется транзакция, в ответе на которую терминалу передается статус возврата.

## Структура входной и выходной информации

Используемая в процессе работы входная и выходная информация существует двух видов: условно-постоянная и переменная. К условно-постоянной информации относится та, которую не нужно менять при каждом обращении к программному решению или не нужно менять часто.

Переменной входной информацией считается та, которая меняется при каждом обращении к программному решению.

К условно-постоянной входной информации относится транзакционный запрос, приходящий из внешней системы в TranzAxis.

Переменной входной информацией является тип транзакционной операции, варианты которых приведены на рисунке.

Условно-постоянной выходной информацией в программном продукте является информация об одобрении проведения платежа.

Переменной выходной информацией является текст ответа.

Ментальная карта входной и выходной информации приведена на рисунке 13.

Диаграмма основных функций приведена на рисунке 14.

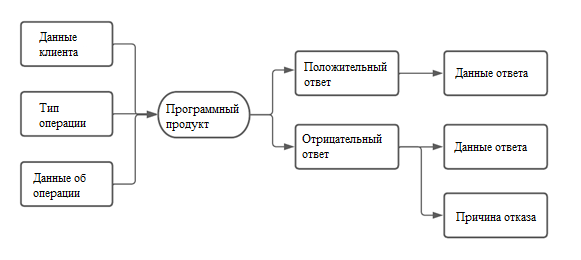


Рисунок 13 – Ментальная карта входной и выходной информации

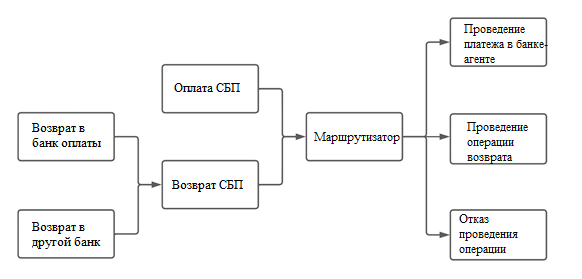


Рисунок 14 – Диаграмма основных функций

## Схема работы интерфейса

Работа интерфейса разбивается на три этапа:

1. Получение запроса от внешней системы и выбор ветки обработки;
2. Обработка запроса:
   1. Обработка входящего запроса. Регистрация запроса в базе данных, создание объектов в системе клиента, формирование ответа;
   2. Обработка исходящего запроса. Получение перечисленных в запросе данных по платежу из системы клиента, формирование ответа;
3. Работа с ответом:
   1. Регистрация транзакционного ответа в системе клиента и отправка статуса операции во внешнюю систему;
   2. Отправка ответа в формате Servicebus на обработку ядром, регистрация ответа в системе клиента и отправка запрошенных данных во внешнюю систему.

Общая блок-схема алгоритма представлена на рисунке 15.

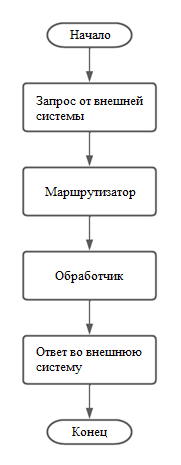


Рисунок 15 – Общая блок-схема алгоритма

## Основные функции

Базовым классом в TranzAxis является Entity. Он является основным для всех объектов, хранящихся в базе данных. От него инстанцируется объект класса Pipeline, который и является объектом класса Servicebus.

Основными функциями интерфейса являются следующие:

1. TranXsd:Response process (Processor.Jml processor, SbXsd:PipelineMessageRq rq, TranXsd:Request tranRq). Функция отвечает за обработку входящих запросов. К параметрам функции относятся следующие:
   1. Processor.Jml processor. Хранит в себе информацию об узле, в котором происходит обработка;
   2. SbXsd:PipelineMessageRq rq. Запрос в формате объекта ServiceBus;
   3. TranXsd:Request tranRq. Транзакционный запрос, обрабатываемый конвейером.

Функция формирует транзакционный ответ, записываемый в базу данных

1. SbXsd:PipelineMessageRs process (Processor.Jml processor, SbXsd:PipelineMessageRq rq). Функция отвечает за обработку исходящих запросов. К параметрам функции относятся следующие:
   1. Processor.Jml processor. Хранит в себе информацию об узле, в котором происходит обработка;
   2. SbXsd:PipelineMessageRq rq. Запрос в формате объекта ServiceBus.

Функция возвращает ответ в формате Servicebus для последующей обработки конвейером и отправки во внешнюю систему.

## Выводы по второй главе

Во второй главе проведен анализ поставленной сформулированной в задании на ВКР.

Рассмотрены общие функциональные возможности системы, на которой будет работать ПО и рассмотрен основной объект реализации поставленной задачи.

Разобрана структура входной и выходной информации, выделены особенности, с помощью которых составлена ментальная карта разработки и диаграмма основных функций.

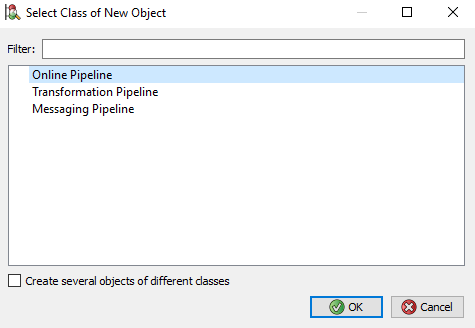
Выделены и разобраны сценарии работы, разработана общая схема работы конвейера и описаны основные функции и методы интерфейса.

# Результаты опытной эксплуатции интерфейса для взаимодействия с Системой Быстрых Платежей на подсистеме TranzAxis ServiceBus

## Создание нового конвейера ServiceBus в системе TranzAxis

При разработке конвейера ServiceBus используется графический интерфейс, упрощающий создание и редактирование элементов. Основными элементами интерфейса являются рабочая область построения диаграммы и кнопки действий (создать, редактировать, удалить и т.д.). Для добавления нового конвейера Servicebus в систему необходимо перейти в Entire System / Financial Institutions / Financial Institution / Interchange / Servicebus Pipelines и нажать на кнопку Create object, после чего перед пользователем появится окно создания нового конвейера, представленное на рисунке 16.

Рисунок 16 – Окно добавления нового конвейера



Пользователь может выбрать тип конвейера при создании:

1. Конвейер Online. Конвейер Servicebus класса Конвейер Online подразумевает, что вызывающая сторона ожидает ответ, и может использоваться для следующих случаев:
   1. Отправка запросов во внешнюю систему и получение ответов;
   2. Прием запросов от внешних систем, их обработка и отправка ответов.

Для конвейера данного класса доступны следующие узлы:

1. Сетевой канал – клиент;
2. Сетевой канал – сервер;
3. Сервисный канал – клиент;
4. Сервисный канал – сервер;
5. Сетевой канал – концентратор;
6. Получатель очереди сообщений;
7. Канал персональных коммуникаций;
8. Отправитель очереди сообщений;
9. RTP Pipeline Online Incoming Interface;
10. RTP Pipeline Online Outgoing Interface;
11. RTP Pipeline Online Outgoing CBA Interface;
12. Исполнитель транзакций;
13. Криптопроцессор;
14. Обработчик MAC;
15. Концентратор транзакций;
16. Разборщик сообщений;
17. Сборщик сообщений;
18. JML процессор;
19. Трансформатор;
20. JML маршрутизатор;
21. XPath маршрутизатор;
22. Вызов через API;
23. Старт;
24. Эхо.
25. Конвейер преобразований - Конвейер Servicebus класса Конвейер преобразований выполняет следующие функции:
    1. Прием запросов от внешних систем и их обработка;
    2. Прием сообщений от внешних систем по одному протоколу и их преобразование по другому протоколу.

Для конвейера данного класса доступны следующие узлы:

1. Исполнитель транзакций;
2. Криптопроцессор;
3. Обработчик MAC;
4. Концентратор транзакций;
5. Разборщик сообщений;
6. Сборщик сообщений;
7. JML процессор;
8. Трансформатор;
9. JML маршрутизатор;
10. XPath маршрутизатор.

Узлы Старт и Эхо присутствуют на диаграмме по умолчанию.

1. Конвейер сообщений - Конвейер Servicebus класса Конвейер сообщений необходим для приема запросов от внешних систем, их обработки и отправки ответов. Для конвейера данного класса доступны узлы, аналогичные конвейеру класса Конвейер Online, кроме узла Вызов через API.

## Общие настройки конвейера

Вкладка общее, представленная на рисунке 17 содержит следующие параметры:

1. Ид. Идентификатор конвейера. Генерируется автоматически. Редактирование запрещено.
2. Класс конвейера. Класс конвейера. Заполняется автоматически. Редактирование запрещено.
3. Название. Название конвейера.
4. Профиль трассировки. Профиль трассировки, используемый для событий, порожденных данным конвейером. Редактируется при помощи специального редактора
5. Примечания. Описание конвейера.
6. Ссылочный ID лицензии. Используется в конвейерах Service Bus, поставляемых в дистрибутиве TranzAxis (в каталоге uds\ServiceBus\etc\), для их отражения в лицензионном отчете.
7. Время последней модификации. Дата и время, когда конвейер редактировался в последний раз. Значение присваивается автоматически. Редактирование запрещено.
8. Автор последней модификации. Имя пользователя, который последним редактировал конвейер. Значение присваивается автоматически. Редактирование запрещено.
9. Журнал изменений. При помощи редактора журнала изменений пользователь, создавший / изменивший текущий конвейер, может заполнить информацию о внесенных ревизиях.
10. Пропустить проверку для узлов. Список узлов конвейера, которые не будут проверяться при выполнении проверки диаграммы конвейера на наличие ошибок (редактор конвейера, вкладка Диаграмма, команда Проверить). Список формируется в диалоге редактирования параметра путем установки флагов напротив узлов, которые необходимо пропускать при проверке.

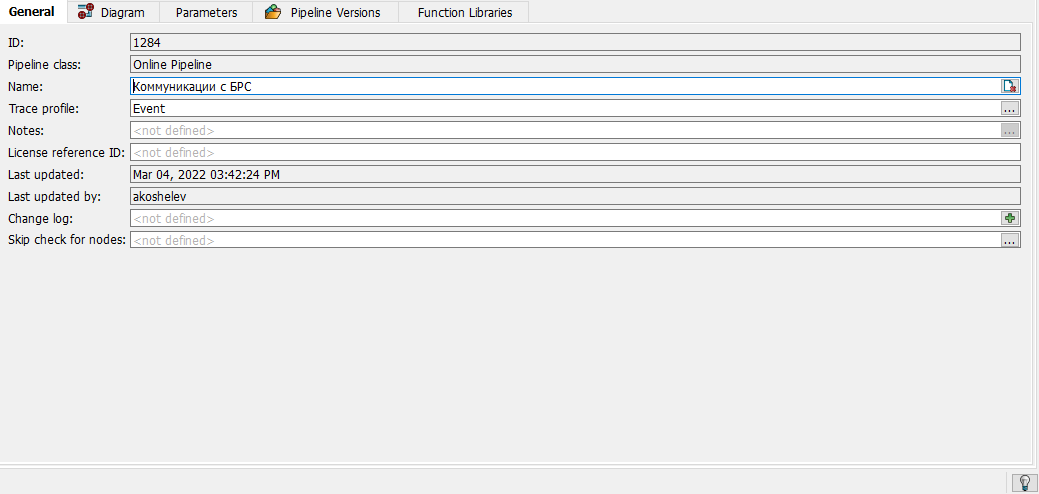


Рисунок 17 – Общие настройки конвейера

Вкладка параметры представлена на рисунке 18 На вкладке определяется набор параметров, используемых узлами конвейера. Редактор параметра содержит следующие атрибуты:

1. Имя. Имя параметра. Обязательный атрибут;
2. Значение. Значение параметра;
3. Локальный. Если флаг установлен, то параметры конвейеров являются локальными и их значения не заменяются при импорте.

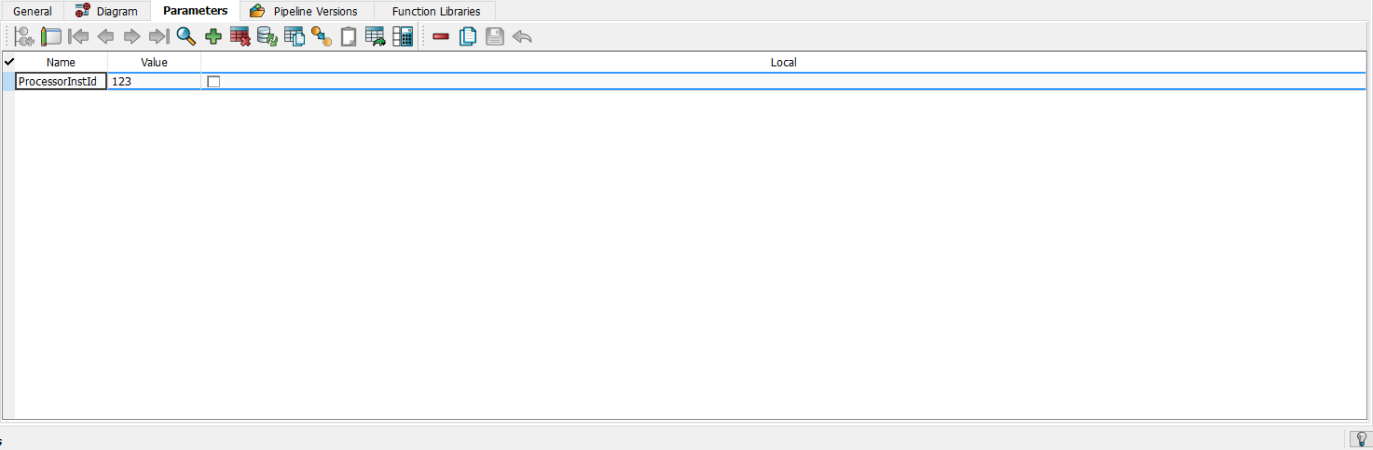


Рисунок 18 – Параметры конвейера

## Коннекторы узлов

Каждый узел может иметь несколько входных и несколько выходных коннекторов, к которым подключаются другие узлы. Коннектор характеризуется именем и типом данных. Могут использоваться следующие типы данных:

1. Bin. Двоичные данные;
2. Structure. Произвольные данные, формат которых определяется MSDL-схемой;
3. XML. Данные в формате XML со структурой, соответствующей XSD-схеме;
4. Tran. Транзакционный запрос или ответ.

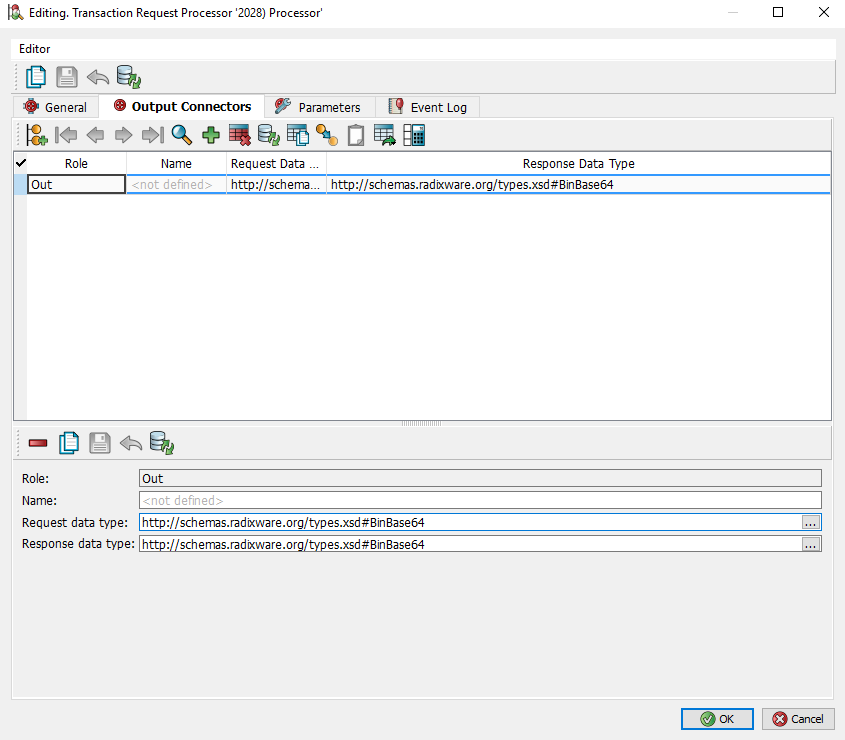


Рисунок 19 – Вкладка Output Connectors узла

При создании нового коннектора заполняются следующие параметры:

1. Роль – название узла, используемое в коде;
2. Имя – название узла, отображаемое в графическом интерфейсе;
3. Тип данных запроса – формат данных, приходящих в запросе;
4. Тип данных ответа – формат данных, уходящих в ответе.

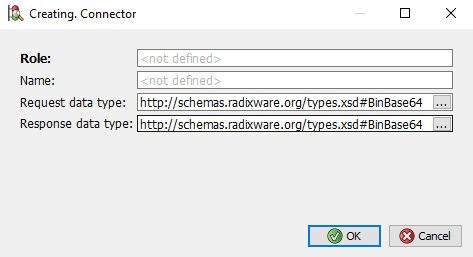


Рисунок 20 – Создание нового коннектора

## Создание диаграммы конвейера

На вкладке расположен визуальный редактор диаграммы. Кнопки панели инструментов визуального редактора следующие:

1. Удалить - удалить выбранный объект (узел или сегмент) с диаграммы;
2. Сменить направление - изменить направление выбранного узла на противоположное;
3. Создать копию узла - создать копию выбранного узла;
4. Выбрать узел - кнопка-переключатель. Переключить визуальный редактор в режим выбора и перемещения объектов;
5. Создать связь - кнопка-переключатель. Переключить визуальный редактор в режим создания связей;
6. Проверить - проверить диаграмму на наличие ошибок (неподключенных узлов и т.п.). Внешний вид кнопки зависит от текущего состояния диаграммы (не проверено / проверено);
7. Комментарий - добавить текстовый комментарий на диаграмму.

Добавление узлов на диаграмму осуществляется посредствам следующих кнопок:

1. Сетевой канал - клиент - добавить на диаграмму узел типа Сетевой канал – клиент;
2. Сетевой канал – сервер - добавить на диаграмму узел типа Сетевой канал – сервер;
3. Сервисный канал - клиент - добавить на диаграмму узел типа Сервисный канал – клиент;
4. Сервисный канал - сервер - добавить на диаграмму узел типа Сервисный канал - сервер ;
5. Сетевой канал - концентратор - добавить на диаграмму узел типа Сетевой канал – концентратор;
6. Получатель очереди сообщений - добавить на диаграмму узел типа Получатель очереди сообщений;
7. Канал персональных коммуникаций - добавить на диаграмму узел типа Канал персональных коммуникаций;
8. Отправитель очереди сообщений - добавить на диаграмму узел типа Отправитель очереди сообщений;
9. RTP Pipeline Online Incoming Interface - добавить на диаграмму узел типа RTP Pipeline Online Incoming Interface;
10. RTP Pipeline Online Outgoing Interface - в кнопке с выпадающим списком сгруппированы команды добавления на диаграмму узлов следующих типов:
    1. RTP Pipeline Online Outgoing Interface;
    2. RTP Pipeline Online Outgoing CBA Interface;
11. Исполнитель транзакций - добавить на диаграмму узел типа Исполнитель транзакций;
12. Криптопроцессор - добавить на диаграмму узел типа Криптопроцессор;
13. Обработчик MAC - добавить на диаграмму узел типа Обработчик MAC;
14. Концентратор транзакций - в кнопке с выпадающим списком сгруппированы команды добавления на диаграмму узлов следующих типов:
    1. Концентратор транзакций;
    2. Концентратор транзакций (CBA);
15. Разборщик сообщений - добавить на диаграмму узел типа Разборщик сообщений;
16. Сборщик сообщений - добавить на диаграмму узел типа Сборщик сообщений;
17. JML процессор - добавить на диаграмму узел типа JML процессор;
18. Трансформатор - добавить на диаграмму узел типа Трансформатор;
19. JML маршрутизатор - добавить на диаграмму узел типа JML маршрутизатор;
20. XPath маршрутизатор - добавить на диаграмму узел типа XPath маршрутизатор;
21. Вызов через API - добавить на диаграмму узел типа Вызов через API;
22. Старт - добавить на диаграмму узел типа Старт;
23. Эхо - добавить на диаграмму узел типа Эхо.

Для решения поставленной задачи необходимо создать диаграмму, представленную на рисунке 21. Диаграмма содержит две независимые ветки:

1. Ветка обработки исходящих запросов. Необходима для запроса дополнительной информации клиентом или уточнения данных о платеже. Состоит из трех узлов:
   1. RTP Pipeline Online Outgoing Interface;
   2. Transaction Request Processor;
   3. Network Channel – Client.
2. Ветка обработки входящих запросов. Необходима для регистрации данных в системе клиента и проведения финансовых транзакций. Состоит из трех узлов:
   1. Network Channel – Server;
   2. JML Processor;
   3. RTP Pipeline Online Incoming Interface.

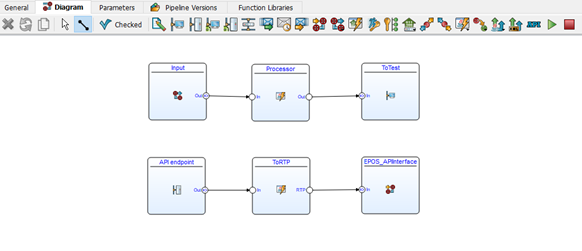


Рисунок 21 – Диаграмма конвейера

## Обработка исходящих запросов

### Узел RTP Pipeline Online Outgoing Interface

Первым узлом в ветке исходящих запросов является интерфейс класса RTP Pipeline Outgoing. Настройка узла продемонстрирована на рисунке 22. Интерфейс RTP Online Outgoing является сервис-ориентированным интерфейсом. Данный интерфейс подготавливает SOAP запрос и вызывает соответствующий сервис. При этом запрос содержит ссылочный идентификатор инициатора, от имени которого происходит взаимодействие (необходимо для указания принимающей стороне, какой входящий интерфейс должен обрабатывать данный запрос).

Основными параметрами интерфейса являются:

1. Ид - идентификатор интерфейса. Заполняется автоматически. Редактирование запрещено;
2. Класс - класс интерфейса. Задается при создании. Значение выбирается из дерева классов в диалоговом окне. Редактирование запрещено;
3. Название - название интерфейса;
4. Институт - владелец интерфейса - название финансового института-владельца интерфейса. Параметр автоматически заполняется при создании значением или названием финансового института, в контексте которого задается интерфейс. Обязательный параметр для офлайн-интерфейсов. Редактирование запрещено;
5. Профиль трассировки - параметр указывает, какие события интерфейса должны быть записаны в журнал событий;
6. Время последней модификации - время создания или последней модификации любого из параметров интерфейса. Значение присваивается автоматически. Редактирование запрещено;
7. Автор последней модификации - имя пользователя, который последним редактировал параметры интерфейса. Значение присваивается автоматически. Редактирование запрещено;
8. Журнал изменений - при помощи редактора журнала изменений пользователь, создавший / изменивший параметры интерфейса, может заполнить информацию о внесенных ревизиях.

Дополнительные параметры интерфейса следующие:

1. PIN working key - Ключ перекодировки PIN-блока. Если параметр не задан, то перекодировка не производится;
2. Ожидание ответа на исходящий запрос (с.) – максимальное время ожидания ответа на запрос, по истечению которого возникает исключение;
3. Ожидание установки соединения (с.) - Интервал времени (в секундах), в течение которого ожидается установка соединения с внешней системой. Если в течение этого времени соединение не было установлено, то транзакция может быть обработана в режиме STIP. Если значение параметра не задано, то в качестве тайм-аута для установки соединения используется значение параметра Ожидание ответа на исходящий запрос (с). Минимальное значение - 1 секунда;
4. Период повторов передачи (с.) – время, по истечению которого интерфейс будет генерировать повтор запроса при отсутствии ответа;
5. Количество повторов передачи – максимальное количество повторов передачи запроса;
6. Коэффициент увеличения периода повторов – коэффициент, влияющий на увеличение времени между повторами;
7. Максимальный период повторов передачи (с) – максимальное время между повторами;
8. Требуется передача подтверждения DCC. Флаг, определяющий поведение интерфейса при DCC. Если флаг установлен, то интерфейс используется не только для динамической конвертации, но и для передачи во внешнюю систему результата выполнения конвертации (после подтверждения или отказа от DCC клиентом). По умолчанию значение флага не задано - признак того, что значение не определялось пользователем; при этом поведение соответствует неустановленному флагу;
9. Маскировать чувствительные данные в запросе. Если флаг установлен, то в передаваемом сообщении будут маскироваться чувствительные данные (PAN, PIN-блок и др. данные). Данный флаг имеет смысл устанавливать для интерфейсов, работающих для выполнения динамической конвертации валюты (DCC) или для аутентификации во внешней системе, в случаях если внешняя система не должна получать чувствительные данные. По умолчанию флаг не установлен;
10. Корректировка исходящего сообщения – функция, код которой применяется при формировании исходящего сообщения;
11. Предобработка входящего сообщения - функция, код которой применяется до ядровой обработки запроса;
12. Корректировка ответа транзакции - функция, код которой применяется при формировании ответа транзакции;
13. Корректировка внешних сообщений - функция, код которой применяется при формировании внешнего сообщения;
14. Необходимость подготовки повторов транзакции – функция, проверяющая условия для генерации повтора.

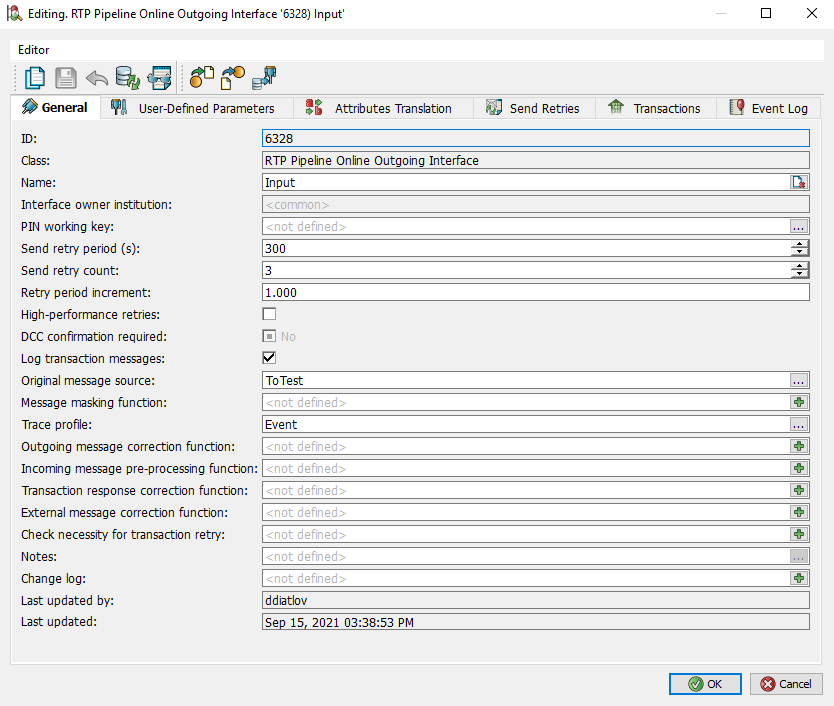


Рисунок 22 – Настройка узла RTP Pipeline Outgoing Interface

### Узел Transaction Request Processor

Вторым узлом является исполнитель транзакций. Узел принимает транзакционные запросы, выполняет транзакции в ядре TranzAxis и возвращает транзакционные ответы. Настройка узла продемонстрирована на рисунке 23. Узел имеет один входной коннектор, принимающий запросы в типе данных http:// schemas.tranzaxis.com/ tran.xsd#Request и возвращающий ответ в типе данных http:// schemas.tranzaxis.com/ tran.xsd#Response.

На вкладке настройки общее мы можем настроить следующие параметры:

1. Ид - идентификатор исполнителя транзакций. Заполняется автоматически. Редактирование запрещено;
2. Название – название исполнителя транзакций;
3. Время последней модификации - время создания или последней модификации любого из параметров исполнителя транзакций. Значение присваивается автоматически. Редактирование запрещено;
4. Автор последней модификации - имя пользователя, который последним редактировал исполнителя транзакций. Значение присваивается автоматически. Редактирование запрещено.

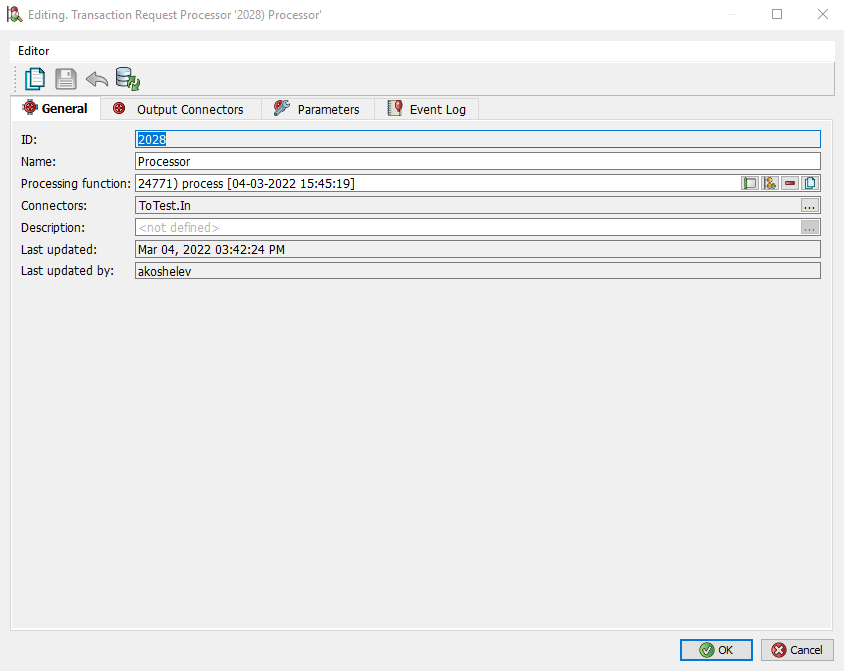


Рисунок 23 – Настройка узла Transaction Request Processor

### Узел Network Channel – Client

Узел выполняет следующие функции:

1. Открытие сетевого соединения;
2. Передача запроса и получение ответа;
3. Разрыв или удержание соединения, в зависимости от содержания запроса и ответа.

Узел имеет один входной коннектор, передающий исходящие запросы в бинарном формате.

Вкладка Общее содержит следующие параметры:

1. Ид - идентификатор сетевого узла. Заполняется автоматически. Редактирование запрещено;
2. Ссылочный ид - предназначенный для использования во внешних системах. Значение параметра должно быть уникальным в пределах модуля, которому принадлежит сетевой канал. Проверка уникальности выполняется при создании и редактировании канала, а также при импорте конвейера, которому принадлежит канал. При этом в процессе импорта значение, которое не является уникальным, очищается. По умолчанию значение параметра на задано;
3. Класс – класс узла;
4. Название – название узла;
5. Модуль системы - системный модуль, при помощи которого узел должен взаимодействовать с внешней системой. Значение параметра выбирается при помощи селектора. Обязательный параметр;
6. Адрес - сетевой адрес, с которым должен соединяться узел. Обязательный параметр;
7. Максимальное число сессий - определяет максимально возможное количество одновременно активных сетевых подключений;
8. Синхронный режим - если флаг установлен, то канал будет работать в синхронном режиме, в котором каждый запрос на отправку данных во внешнюю систему выполняется в отдельном подключении в параллельном режиме;
9. Записывать количество занятых соединений в базу данных - флаг, определяющий необходимость ведения счетчика занятых соединений в параметре Количество занятых соединений. Флаг отображается, если установлен флаг Синхронный режим. По умолчанию флаг не установлен;
10. Протокол - протокол, используемый каналом для взаимодействия с внешними системами. Возможные значения:
    1. Не определен (по умолчанию);
    2. HTTP.
11. Формат исходящего пакета - формат фрейма, используемый для формирования запросов;
12. Формат входящего пакета - формат фрейма, используемый для разбора ответов внешней системы;
13. Тайм-аут соединения - интервал времени, в течение которого узел должен ожидать соединения с внешней системой, в секундах;
14. Тайм-аут приема - интервал времени, в течение которого узел должен ожидать входящее сообщение, в секундах;
15. Тайм-аут передачи - интервал времени, в течение которого узел должен ожидать отправки исходящего сообщения, в секундах;
16. Время удержания соединения - интервал времени, в течение которого узел должен удерживать соединение после получения ответа, в секундах. Если значение параметра не задано, то узел будет использовать постоянное соединение;
17. Длительность удержания соединения - пользовательская функция, определяющая время удержания. Если значение параметра не задано, то будет использоваться значение параметра Время удержания соединения;
18. Протокол безопасности;
19. Приватные ключи канала;
20. Доверенные сертификаты удаленной стороны;
21. Проверка сертификата удаленной стороны;
22. Использовать TCP keepalive - если флаг установлен, то для проверки соединения с внешними системами будет использоваться опция протокола TCP KeepAlive. Настройки данной опции (тайм-аут, количество попыток проверки активности соединения) выполняются на уровне ОС. По умолчанию флаг установлен;
23. Количество открытых соединений - количество сетевых подключений, обслуживаемых каналом в данный момент;
24. Количество занятых соединений - счетчик количества занятых соединений канала на текущий момент. Для клиентского канала соединение считается занятым, если по нему отправлен исходящий запрос во внешнюю систему, но не получен ответ. Параметр отображается, если установлен флаг Синхронный режим. Значение обновляется при каждой отправке запроса и получении ответа, если установлен флаг записывать количество занятых соединений в базу данных.

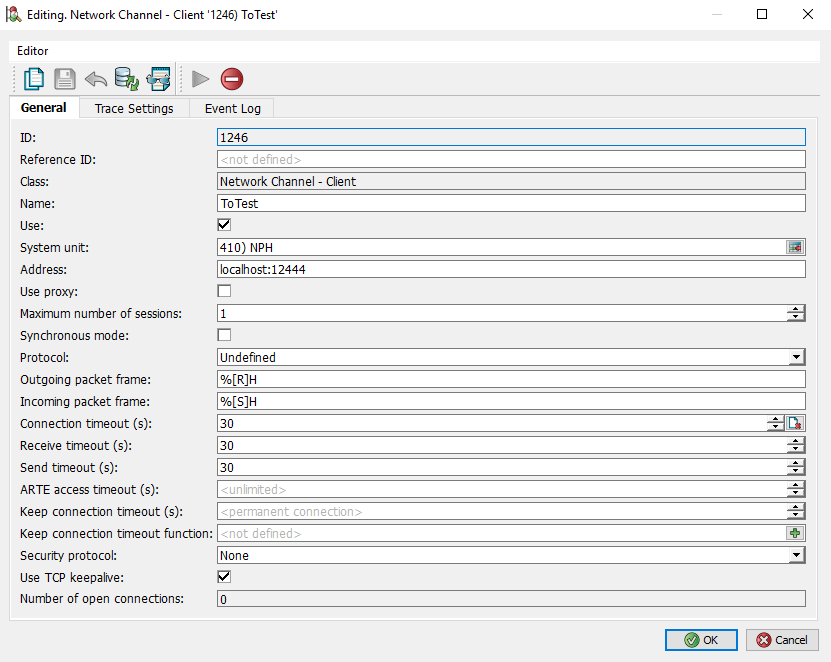


Рисунок 24 – Настройки узла Network Channel – Client

## Обработка входящих запросов

### Узел Network Channel – Server

Узел выполняет следующие функции:

1. Прием сетевых соединений;
2. Прием запросов и передача ответов;
3. Разрыв или удержание соединения, в зависимости от содержания запроса и ответа;

Узел имеет один выходной коннектор, передающий данные в бинарном формате. Вкладка общее содержит следующие параметры:

1. Ид – уникальный идентификатор узла. Создается автоматически. Редактирование запрещено;
2. Ссылочный ид - предназначенный для использования во внешних системах. Значение параметра должно быть уникальным в пределах модуля, которому принадлежит сетевой канал. Проверка уникальности выполняется при создании и редактировании канала, а также при импорте конвейера, которому принадлежит канал. При этом в процессе импорта значение, которое не является уникальным, очищается. По умолчанию значение параметра на задано;
3. Ссылочный идентификатор сетевого канала (RID) - предназначенный для использования во внешних системах. Значение параметра должно быть уникальным в пределах модуля, которому принадлежит сетевой канал. Проверка уникальности выполняется при создании и редактировании канала, а также при импорте конвейера, которому принадлежит канал. При этом в процессе импорта значение, которое не является уникальным, очищается. По умолчанию значение параметра на задано;
4. Класс – класс узла;
5. Название – название узла. Задается пользователем;
6. Модуль системы. Системный модуль, при помощи которого узел должен взаимодействовать с внешней системой. Значение параметра выбирается при помощи селектора, содержащего все определенные в системе модули типа Обслуживание сетевых портов. Обязательный параметр;
7. Коннектор. Название узла, подключенного к исходящему коннектору данного узла. Заполняется автоматически. Редактирование запрещено;
8. Адрес. Сетевой адрес, на котором узел должен ожидать соединений;
9. Синхронный режим. Серверные сетевые каналы фактически всегда работают в синхронном режиме, поэтому для них флаг Синхронный режим влияет только на возможность создавать по данному каналу метрики синхронных сетевых каналов. По умолчанию флаг не установлен;
10. Записывать количество занятых соединений в базу данных. Флаг, определяющий необходимость ведения счетчика занятых соединений в параметре Количество занятых соединений. Флаг отображается, если установлен флаг Синхронный режим. По умолчанию флаг не установлен;
11. Протокол. Протокол, используемый каналом для взаимодействия с внешними системами. Возможные значения:
    1. Не определен (по умолчанию).
    2. HTTP.

Подробнее об особенностях настройки сетевых каналов при работе по протоколу HTTP см. в разделе Настройки сетевых каналов.

1. Формат входящего пакета. Формат фрейма, используемый для разбора запросов;
2. Формат исходящего пакета. Формат фрейма, используемый для формирования ответов;
3. Максимальное количество сессий. Максимальное количество сетевых сессий, которое должен поддерживать узел. При достижении максимального количества сессий новые соединения будут отклоняться;
4. Количество активных сессий. Количество сетевых сессий, активных в настоящий момент. Редактирование запрещено;
5. Тайм-аут приема. Интервал времени, в течение которого узел должен ожидать входящее сообщение, в секундах;
6. Тайм-аут передачи. Интервал времени, в течение которого узел должен ожидать отправки исходящего сообщения, в секундах;
7. Время удержания соединения. Интервал времени, в течение которого узел должен удерживать соединение после отправки ответа, в секундах. Если значение параметра не задано, то узел будет использовать постоянное соединение;
8. Длительность удержания соединения. Пользовательская функция, определяющая время удержания. Если значение параметра не задано, то будет использоваться значение параметра Время удержания соединения;
9. Протокол безопасности;
10. Приватные ключи канала;
11. Доверенные сертификаты удаленной стороны;
12. Проверка сертификата удаленной стороны;
13. Использовать TCP keepalive - если флаг установлен, то для проверки соединения с внешними системами будет использоваться опция протокола TCP KeepAlive. Настройки данной опции (тайм-аут, количество попыток проверки активности соединения) выполняются на уровне ОС. По умолчанию флаг установлен;
14. Количество открытых соединений - количество сетевых подключений, обслуживаемых каналом в данный момент;
15. Количество занятых соединений - счетчик количества занятых соединений канала на текущий момент. Для серверного канала соединение считается занятым на время вызова обработчика очередного полученного сетевого пакета. Параметр отображается, если установлен флаг Синхронный режим. Значение обновляется раз в 5 секунд, если установлен флаг записывать количество занятых соединений в базу данных.

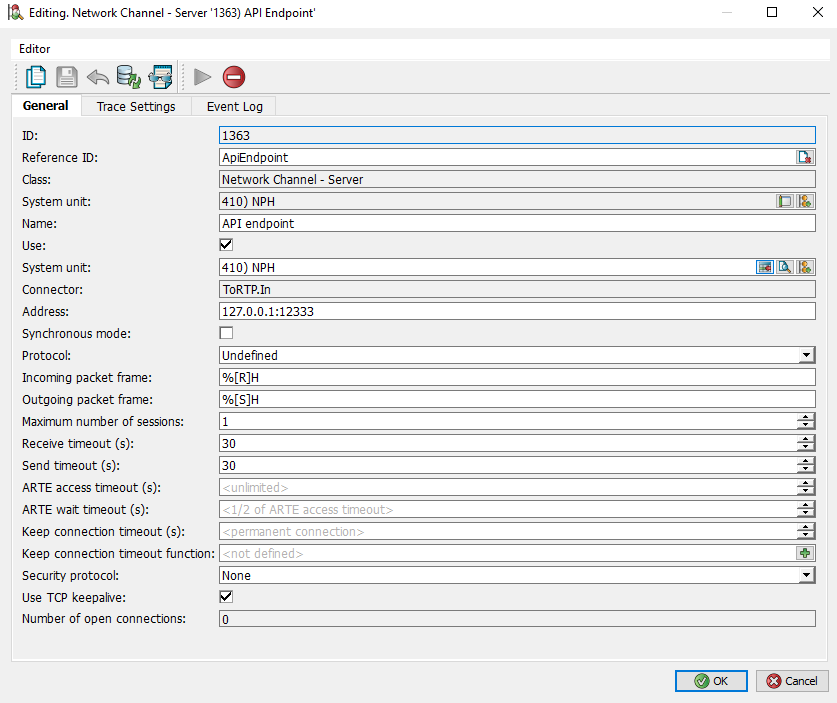


Рисунок 25 - Настройки узла Network Channel – Server

### Узел JML Processor

Узел выполняет обработку сообщения, определяемую пользовательской функцией, и возвращает результат. Узел может передавать сообщение для обработки одному из узлов, подключенных к выходным коннекторам данного узла. Поддерживаются следующие виды JML процессоров:

1. JML процессор. Обрабатывает сообщения в формате XML;
2. JML процессор (Bin). Обрабатывает бинарные сообщения;
3. Процессор транзакционных запросов. Обрабатывает транзакционные запросы;
4. Процессор транзакционных запросов (CBA). Обрабатывает транзакционные запросы, принятые из АБС.

На вкладке общее можно настроить следующие параметры:

1. Ид.
2. Название.
3. Тип данных запроса. Выбирается при помощи селектора, содержащего XSD-схемы. Отображается только для узлов типа JML процессор.
4. Тип данных ответа. Выбирается при помощи селектора, содержащего XSD-схемы. Отображается только для узлов типа JML процессор.
5. Функция обработки. Пользовательская функция, обрабатывающая сообщение и возвращающая ответ. Может передавать сообщение для обработки одному из узлов, подключенных к выходным коннекторам данного узла
6. Коннекторы. Список узлов, которые подключены к выходным коннекторам данного узла. Заполняется автоматически. Редактирование запрещено.
7. Описание.
8. Время последней модификации.
9. Автор последней модификации

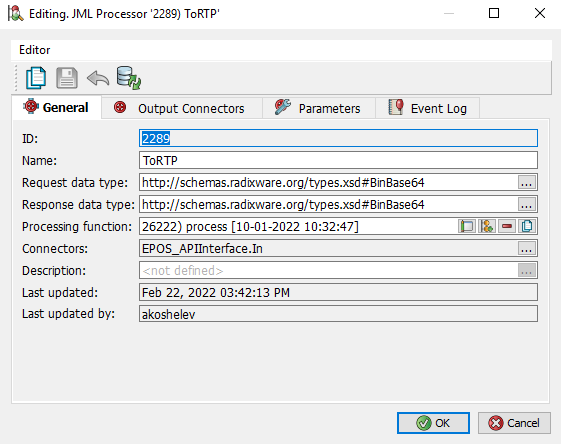


Рисунок 26 - Настройки узла JML Processor

### Узел RTP Pipeline Online Incoming Interface

Интерфейс RTP Online Incoming является сервис-ориентированным интерфейсом. При получении SOAP-сообщения для сервиса ARTE обращается к зарегистрированному сервисному объекту для обработки запроса. Сервисный объект извлекает из запроса ссылочный идентификатор инициатора и ищет нужный входящий интерфейс, которому передает полученный запрос.

На вкладке общее, кроме общих параметров настройки, интерфейс имеет специфичные параметры:

1. Поддерживаемая версия. Параметр предназначен для выявления несовместимости протоколов RTP между разными версиями TranzAxis при приеме/ передаче сообщений с помощью интерфейса. Список возможных значений содержит версии, с которыми обнаружена несовместимость в более поздних версиях. При получении входящего сообщения, в котором указана версия TranzAxis, выполняется проверка совместимости настроек, и при несовпадении в трассу выводится предупреждение;
2. Институт-партнер;
3. Финансовая сеть;
4. Ссылочный ид. инициатора. Должен совпадать со ссылочным идентификатором исходящего интерфейса. Обязательный параметр;
5. Сервисные модули. Параметр предназначен для привязки входящего интерфейса к нескольким ARTE-модулям системы. Список модулей формируется пользователем в типовом редакторе массива значений. Значения элементов массива необходимо выбрать из селектора модулей, определенных в системе;
6. Формат PIN-блока. Формат, используемый для передачи PIN-блока внешним системам. Значение выбирается из предопределенного списка. Возможные значения:
   1. ANSI X9.8;
   2. Docutel;
   3. Diebold;
   4. PLUS;
   5. ISO Format 1;
   6. ISO Format 2;
   7. По умолчанию - ANSI X9.8;
7. Ключ перекодировки PIN-блока. Если параметр не задан, то перекодировка не производится;
8. Количество потоков обработки извещений;
9. Не связывать транзакции. Если флаг не установлен, то для редактирования доступны следующие параметры:
   1. Глубина поиска связи для первой попытки;
   2. Не связывать транзакции при второй попытке;
   3. Глубина поиска связи для второй попытки. Параметр доступен для редактирования, если не установлен флаг Не связывать транзакции при второй попытке;
   4. Не связывать транзакции при второй попытке для длительных связей;
   5. Глубина поиска связи для второй попытки для длительных связей. Параметр доступен для редактирования, если не установлен флаг Не связывать транзакции при второй попытке для длительных связей;
10. Проверять запросы. Если флаг установлен, то выполняется проверка входящих запросов на соответствие XSD-схеме данных. В случае несоответствия запрос отклоняется интерфейсом с соответствующим кодом ошибки. Данный флаг предназначен для тестирования интерфейса;
11. Сохранять транзакционные сообщения;
12. Предобработка входящего сообщения;
13. Корректировка запроса транзакции;
14. Корректировка исходящего сообщения;
15. Корректировка внешних сообщений.

Кроме того, редактор узла на вкладке Общее содержит следующие специфические параметры:

1. Обрабатывать Advice асинхронно. Если флаг установлен, то обработка транзакций в режиме Advicе выполняется асинхронно;
2. Источник оригинальных сообщений. Узел, передающий исходные сообщения, которые должны сохраняться в данном узле. Значение выбирается из списка узлов того же конвейера, имеющих выходные коннекторы;
3. Маскирование сообщений. Пользовательская функция, выполняющая маскирование чувствительных данных в сохраняемых исходных сообщениях;

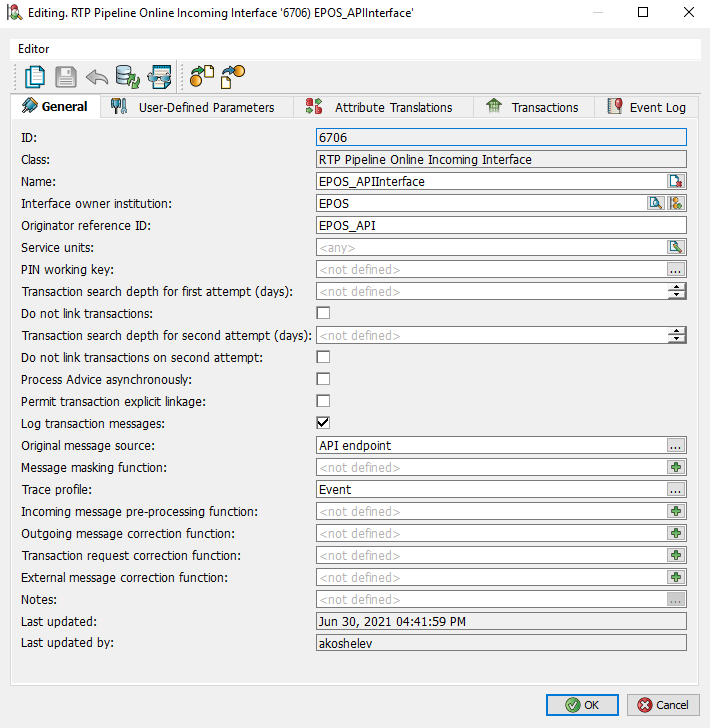


Рисунок 27 – Узел RTP Pipeline Online Online Incoming Interface

## Выводы по третьей главе

В третьей главе была рассмотрена реализация конвейера ServiceBus для обработки платежей инициированных посредством Системы Быстрых Платежей.

Были описаны инструменты для работы с конвейером и продемонстрирована работа по созданию диаграммы конвейера.

Были описаны все используемые в конвейере узлы, их роль в обработке платежей и все настройки данных узлов.

В дальнейшем планируется сопровождение данного конвейера после его загрузки на среду клиента инженером.

# Заключение

В первой главе выпускной квалификационной работы осуществлено решение задачи, связанной с изучением теоретических вопросов по платёжным системам и разрабатываемым для них интерфейсам. В частности, изучены основные понятия в сфере платёжных систем, их структура, функции и виды. Выяснено, что платёжная система – это большая система со сложной структурой и тесно взаимосвязанными составляющими, каждая из которых выполняет свои определённые функции. Рассмотрены характеристики и история развития Национальной Системы Платежных Карт, и проекта Система Быстрых Платежей, для которой разрабатывается исходящий интерфейс.

Во второй главе выпускной квалификационной работы осуществлено решение задачи, связанной с проектированием модели интерфейса. В частности, определены структура получаемых на входе и выходе данных, сценарии и схемы работы интерфейса, основные методы и классы, связанные с интерфейсом. Построены блок-схемы основных модулей программного решения и общая блок-схема работы программы, и ментальная карта исходящего интерфейса на основе структуры входной и выходной информации. Также приведены общие сведения об объекте класса ServiceBus. Использованные проектные решения разработки интерфейса позволили чётко представить структуру программного решения.

В третьей главе выпускной квалификационной работы осуществлено решение задачи, связанной с непосредственно разработкой интерфейса для обработки платежей СБП на основе исследованной теории и результатов проектирования. Определена технология обработки входящих и исходящих запросов. Также проведена опытная эксплуатация интерфейса. Интерфейс успешно обработал тестовый пример, данные, полученные из внешней системы, прошли все необходимые этапы маршрутизации, обработки и записи в базу данных для дальнейшего использования, что подтверждает пригодность интерфейса к использованию в работе сотрудниками и клиентами компании ООО «Компас Плюс», что так же подтверждает отзыв менеджера проекта.

Таким образом, посредством решения вышеизложенных задач достигнута цель данной выпускной квалификационной работы. Разработан объект типа ServiceBus с помощью инструментов системы TranzAxis компании ООО «Компас Плюс», для обработки платежных транзакций через банк-агент инициированных QR кодами системы быстрых платежей национальной системы платежных карт по условиям описанным в TSD банка-клиента. Интерфейс будут использовать в работе сотрудники и клиенты компании ООО «Компас Плюс». Решена поставленная проблема отсутствия интерфейса для обработки транзакций, инициированных при помощи Системы Быстрых Платежей, вызванная тем, что проект Национальной Системы Платежных Карт России получили достаточное большое распространение лишь в последние несколько лет.

# Список использованных источников

1. Словарь банковских терминов. Транзакция [Электронный ресурс] / Элек трон. текстовые дан. – Режим доступа:

https://myfin.by/wiki/term/tranzakciya, свободный.

1. Усоскин, В. М. Платежные системы и организация расчетов в коммерче ском банке [Текст]: учеб. пособие / В. М Усоскин, В. Ю. Белоусова. – М.:  Высшая школа экономики, 2012. – 193 с
2. Платежные системы / С. Е. Дубова, А. С. Обаева, А. А. Валинурова [и  др.]. – М.: МЦНИП, 2014. – 500 с.
3. Международные платежные системы [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа:  
   http://www.iccwbo.ru/blog/2016/mezhdunarodnye-platezhnye-sistemy/, сво бодный.
4. Сайт Банка России [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа:  
   <https://www.cbr.ru/PSystem/sfp/>
5. Экономическая энциклопедия [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа:  
   <https://vocable.ru/termin/finansovyi-posrednik.html>
6. Международный журнал прикладных и фундаментальных исследований [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа:  
   <https://applied-research.ru/en/article/view?id=9888>
7. Деньги, кредит, банки: Учебник для бакалавров / под ред. В.Ю. Катасо нова, В.П. Биткова. - М.: Юрайт, 2014. – 575 с. - ISBN 978-5-9916-3277-5.  – (Бакалавр. Базовый курс).
8. Руэксперт История Национальной Системы Платежных Карт [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа:  
   <http://ruxpert.ru/nspk>
9. Сайт платежной системы Мир. НСПК [Электронный ресурс] / Электрон.  текстовые дан. – Режим доступа: <https://mironline.ru/nspk/>

# Приложение А

Рецензия аналитика проекта:

В рамках задачи PRJDEV-8010 был разработан Service Bus для интеграции с Банком-партнером по проекту Клиента E-POS “СБП C2B/B2C”. Разработчиком было поддержано четыре исходящих запроса (getQrCode, getQrdStatus, qrRefundBk, getRefundStatus) и один входящий (qrdStatusResult).

Совместно с Клиентом были полностью протестированы запросы, относящиеся к части C2B проекта: getQrCode, getQrdStatus. При этом пока не был протестирован входящий запрос по C2B - qrdStatusResult, т.к. Банк-партнер утверждает, что такие запросы они пока не имеют возможности отправить в TX. Поэтому разработка была несколько усложнена для того, чтобы два запроса: getQrdStatus и qrdStatusResult, работали по единой логике, а именно: создавали инвойс в нужном статусе либо при получении ответа на getQrdStatus, либо при получении входящего запроса qrdStatusResult (что из этого будет обработано раньше) и не дублировали действие при обработке более позднего запроса/ответа.

Части, относящиеся к B2C (qrRefundBk, getRefundStatus), были протестированы частично ввиду изменения политической и экономической ситуации и сдвига планов по запуску проекта у Клиента.

Насколько мне известно, C2B часть перенесена в продакшен, и в течение двух месяцев Клиент не обращался с ошибками или замечаниям по функционалу. Планы по запуску части B2C (qrRefundBk, getRefundStatus) на данный момент пока не известны.

Полищук Ксения Сергеевна

Старший аналитик решений Управления аналитики

26.05.2022 г.

