

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки / специальности)

Разработка программного обеспечения
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка интеграционного решения для синхронизации данных
электронной почты и системой управления проектами в отделе подключений
службы сопровождения ключевых клиентов в СКБ Контур»

Обучающийся

Н.В. Гончаренко

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Казаченок

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Настоящая работа посвящена разработке интеграционного решения синхронизации данных между системой управления проектами и сервисом электронной почты в отделе подключений службы сопровождения ключевых клиентов. Цель работы заключается в том, чтобы описать получившееся интеграционное решение, позволяющее обрабатывать все задачи в одной системе и автоматизировать сбор статистики. В ходе работы были решены следующие задачи: проведен анализ бизнес-процессов отдела подключений и выявлены проблемы, которые обуславливают необходимость разработки интеграции, разработаны требования и произведено проектирование системы с помощью диаграмм UML, описано получившееся решение и процесс его разработки.

Актуальность работы обусловлена тем, что в настоящее время многие сервисы интегрируются друг с другом, чтобы создать наибольшую пользу и удобство для клиента и для получения выгоды в виде коммерческой прибыли, или лояльности клиентов, или удобства внутренних сотрудников компании. В данной работе проанализирован опыт внедрения интеграции между системой управления проектами и сервисом электронной почты, но охарактеризована не только сама интеграция, но и автоматизация бизнес-процессов, способы повышения эффективности работы сотрудников. Представляется, что описанный опыт может быть полезен для дальнейших разработок в указанной области.

Работа состоит из трех глав: первая глава посвящена теоретическому описанию предметной области и разработке требований к интеграционному решению, вторая глава посвящена проектированию и описанию информационной системы, третья глава посвящена практической реализации и тестированию интеграционного решения.

Оглавление

Введение	5
Глава 1 Постановка задачи на разработку интеграционного решения для синхронизации данных	8
1.1 Анализ бизнес-процессов в отделе подключений службы сопровождения ключевых клиентов	8
1.2 Разработка требований к интеграционному решению по синхронизации системы управления проектами и сервисом электронной почты	12
Глава 2 Проектирование интеграционного решения для синхронизации данных между сервисом электронной почты и системы управления проектами	17
2.1 Описание интеграционного решения в отделе подключений службы сопровождения ключевых клиентов	17
2.2 Логическое моделирование интеграционного решения	22
Глава 3 Разработка и тестирование интеграционного решения для обмена данными между системой управления проектами и сервисом электронной почты	31
3.1 Реализация интеграционного решения для отдела подключений службы сопровождения ключевых клиентов	31
3.2 Тестирование интеграционного решения для синхронизации данных системы управления проектами и сервиса электронной почты	48
Заключение	52
Список используемой литературы	52
Приложение А Код правила для парсинга данных	57
Приложение Б Код правила для автоматического проставления значения поля «SP из справочника»	59
Приложение В Код правила для заполнения поля «Тип задачи»	61
Приложение Г Код правила для заполнения поля «SP учтено»	64

Приложение Д Код расширения для браузера, автоматизирующего отправку электронных заявок ключевым клиентам.....	70
---	----

Введение

Бакалаврская работа написана на основе деятельности IT-компании СКБ Контур; компания является крупнейшим поставщиком услуг в области электронного документооборота во многих сферах [22, 7], в том числе в области ритейла, и является EDI-провайдером и ЭДО-оператором. EDI-провайдер осуществляет обмен формализованными документами между торговыми сетями, дистрибьюторами и поставщиками [3, 4, 5]. В СКБ Контур работой над указанным направлением деятельности компании занимается управление внедрений. Работа посвящена анализу деятельности отдела подключений службы сопровождения ключевых клиентов, который входит в структуру управления внедрений. Служба сопровождения ключевых клиентов занимается сопровождением работы клиентов в системах Контур.EDI и Диадок. Вариантов работы в Контур.EDI может быть два: первый – это веб-интерфейс, второй – интеграционное решение для взаимодействия Контур.EDI и информационной системой, которую использует поставщик (например, 1С). Для обоих вариантов работы необходима настройка, чтобы поставщики и торговые сети могли осуществлять обмен формализованными сообщениями. Сотрудники отдела подключений работают в нескольких информационных системах и взаимодействуют с разными подразделениями – как с внутренними сотрудниками, так и внешними контрагентами. Для того, чтобы организовать эффективное взаимодействие и оперативное выполнение задач в отделе подключений, используется система управления проектами. Но из-за того, что сотрудники получают заявки на подключение в разных информационных системах – системе управления проектами и сервисе электронной почты – работа отдела представляется не совсем эффективной, кроме того, указанная причина не позволяет автоматизировать сбор статистики. Решение видится в разработке интеграционного решения по синхронизации данных системы управления проектами и сервиса электронной почты [20].

Работа посвящена разработке такого решения, чтобы вся работа сотрудников отдела подключений фиксировалась только в одном информационном сервисе. Это повысит эффективность сотрудников за счет того, что им не придется переключаться из одной системы в другую, а это исключит человеческий фактор (например, ситуации, когда сотрудник забыл проверить ту или иную систему), позволит автоматизировать рутинные действия, а также позволит автоматизировать сбор статистики для подсчета эффективности работы отдела.

Объект исследования – интеграция между сервисом электронной почты и системой управления проектами для оптимизации работы группы прямых подключений.

Предмет исследования – разработка интеграции между системой управления проектами и сервисом электронной почты для синхронизации данных.

Цель работы: разработать интеграционное решение, которое позволит собирать все задачи на подключение в одной системе и формировать отчет по выполненным задачам на каждого сотрудника.

Задачи:

- описать основные бизнес-процессы отдела подключений;
- разработать требования к интеграции систем;
- спроектировать решение на основе требований, составить диаграммы;
- описать получившееся решение;
- описать техническую реализацию полученного интеграционного решения для синхронизации данных.

Актуальность работы видится в том, что в настоящее время многие сервисы стремятся к созданию общей интегрированной среды либо с коммерческой точки зрения, либо для удобства сотрудников или клиентов; причины такого стремления понятны: данная среда помогает повысить скорость обмена информацией, оптимизировать бизнес-процессы и, как

следствие, повысить экономическую эффективность деятельности организации. В данной работе проанализирован опыт создания интеграционного решения в отделе подключений службы сопровождения ключевых клиентов СКБ Контур, результаты этого анализа могут быть полезны для дальнейших разработок в данной сфере.

Работа состоит из трех глав, введения, заключения, списка литературы и приложений.

В первой главе проведен анализ предметной области, рассмотрены бизнес-процессы отдела, разработаны диаграммы, которые показывают деятельность отдела, разработаны требования к информационному решению, которое должно оптимизировать текущие бизнес-процессы.

Во второй главе описано то, как будет реализована интеграция между системой управления проектами и сервисом электронной почты, какие функции разработанное решение будет выполнять, реализовано логическое моделирование системы с помощью диаграмм UML.

Третья глава посвящена разработке интеграции и автоматизации рутинных действий, созданию на основе полученного решения отчета о выполненных задачах с учетом заранее разработанных критериев.

Выпускная квалификационная работа представлена на 56 страницах и включает 18 рисунков и список литературы из 30 наименований.

Глава 1 Постановка задачи на разработку интеграционного решения для синхронизации данных

1.1 Анализ бизнес-процессов в отделе подключений службы сопровождения ключевых клиентов

Отдел подключений относится к управлению внедрений проекта Контур.EDI. Контур.EDI является системой обмена формализованными документами между поставщиками, торговыми сетями, дистрибьюторами или производителями [24]. Отдел подключений занимается настройкой этого взаимодействия. Взаимодействие осуществляется через цепочку EDI-сообщений (ORDERS, ORDRSP, DESADV, RECADV, INVOIC, COINVOIC), сотрудники настраивают основную цепочку и осуществляют дополнительные настройки в случае, если это предусмотрено регламентом подключения. Настройки осуществляются в сервисе администрирования EDI и в справочниках для коннекторов. На основе выполненных настроек контрагенты могут работать в системах Контур.EDI, Диадок или с помощью интеграции в информационной системе, которую уже использует контрагент (чаще всего 1С) [18]. Настройка осуществляется на основании задачи в системе управления проектами [15] или письма в сервисе электронной почты [13]. В результате выполненной работы в отделе подключений появляется настроенная взаимосвязь ключевого клиента и поставщика, и в подтверждение этого имеется ответ на электронное письмо или выполненная задача в системе управления проектами. Таким образом, основным бизнес-процессом отдела подключений является настройка взаимосвязи между поставщиком и ключевым клиентом. Бизнес-процесс отражен на диаграмме в нотации IDEF0 и представлен на рисунке 1.

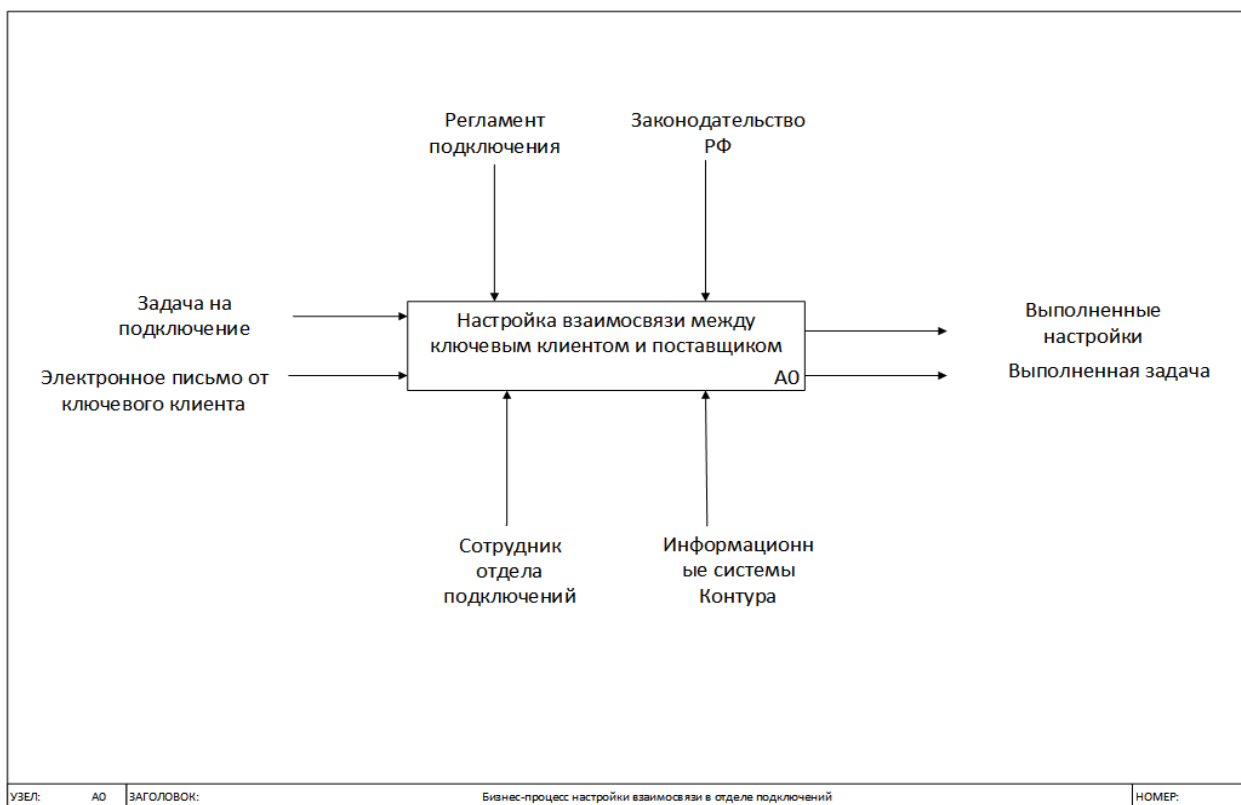


Рисунок 1 – Диаграмма бизнес-процесса настройки взаимосвязи

Рассмотрим бизнес-процесс более подробно для его дальнейшей декомпозиции. Для того чтобы в отделе подключений были выполнены настройки для поставщика и торговой сети необходимо, чтобы в отдел подключений обратился сотрудник отдела внедрения управления внедрений или менеджер по продажам или представитель ключевого клиента. Обращение может быть оформлено в виде задачи в системе управления проектами, если это обращение сотрудника СКБ Контур, или в виде письма в сервисе электронной почте, если это обращение представителя контрагента.

Рассмотрим случай обращения сотрудником СКБ Контур. Как правило, это сотрудник отдела внедрений, у него уже есть задача, связанная с осуществлением настроек от поставщика, и он создает задачу в системе управления проектами для сотрудников отдела подключений. Задача должна быть заполнена в соответствии с регламентом подключения, приложены необходимые документы для настроек. Если все заполнено корректно, то сотрудник отдела подключений берет в работу задачу и осуществляет

настройку необходимой цепочки сообщений в соответствии с регламентом подключения, в зависимости от ключевого клиента добавляет новые юридические лица или GLN в сервисе администрирования EDI, производит дополнительные настройки в справочнике, добавляет поставщика на коннектор, далее проверяет счет на оплату в Биллинге и активирует трафик в сервисе администрирования EDI [18]. После того как необходимые настройки выполнены на стороне СКБ Контур, такие же настройки должны быть сделаны и на стороне ключевого клиента. Для этого сотрудники отдела подключений на основании задачи в системе управления проектами отправляют заявку в торговую сеть с необходимостью настройки на их стороне. Заявка отправляется с помощью сервиса электронной почты. После того, как торговая сеть рассмотрит заявку, сотрудники отвечают на письмо с предоставлением информации о том, выполнены настройки или нет. Соответствующий ответ сотрудник отдела подключений транслирует в задачу и закрывает задачу с комментарием настроен / не настроен.

В случае, если за настройкой обращается представитель ключевого клиента, то он присылает письмо с заявкой на адрес электронной почты отдела подключений [6], сотрудник отдела подключений также выполняет все настройки в соответствии с регламентом и напрямую отправляет ответ на письмо ключевому клиенту, не ожидая обратной связи о настройке, так как предполагается, что на стороне ключевого клиента настройки уже выполнены. В данной ситуации задача в системе управления проектами не создается, работа производится на основании письма, что снижает эффективность, так как сотрудникам приходится работать в разных системах: сотрудник вынужден переключаться между окнами приложений, при этом он не видит весь объем работы. Кроме того, повышается риск возникновения ситуации, вызванной вмешательством человеческого фактора: сотрудник может забыть проверить электронную почту, и обработка заявки будет затянута. Декомпозиция бизнес-процесса отображена на рисунке 2.

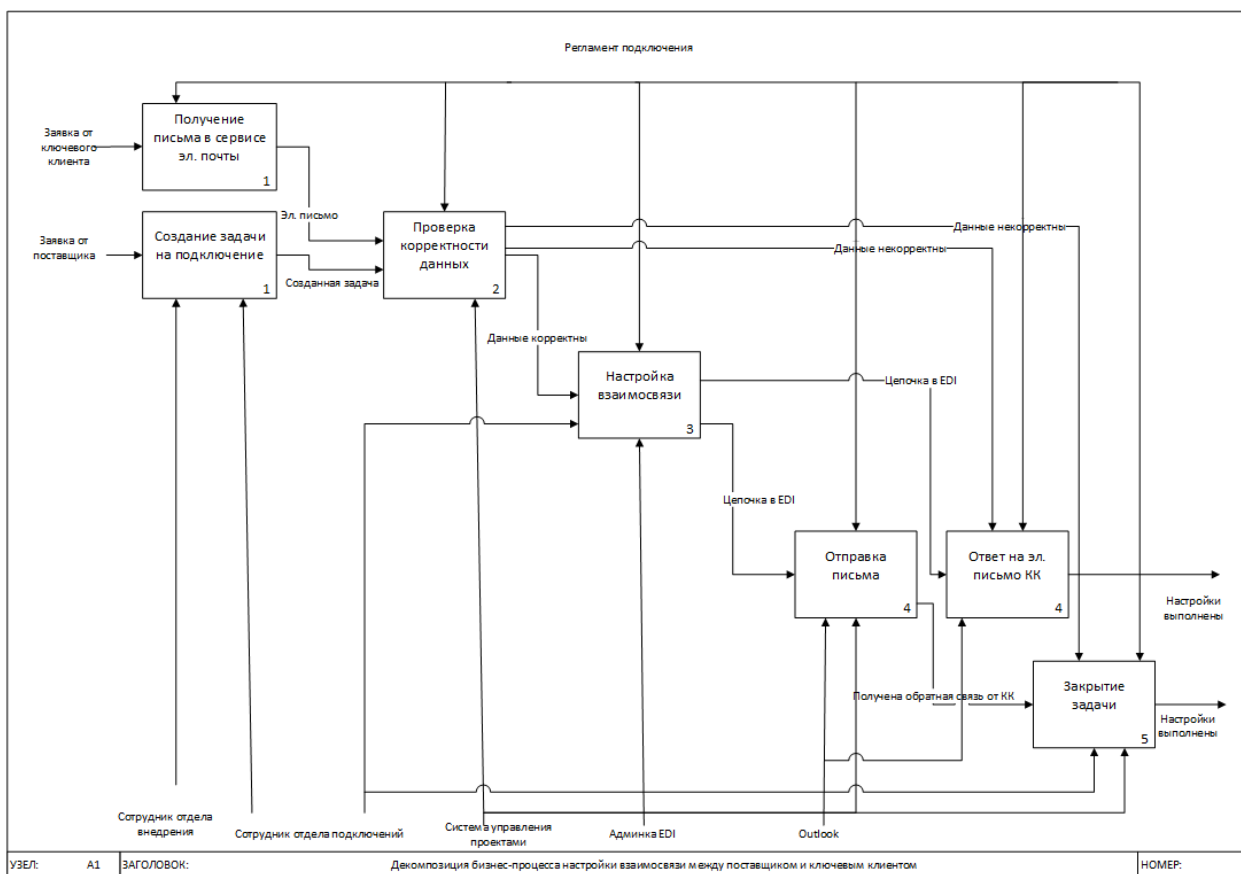


Рисунок 2 – Декомпозиция бизнес-процесса настройки взаимосвязи

Как видно на диаграмме, в группе прямых подключений имеется два входящих информационных потока, в первом заявки на подключение поступают в системе управления проектами, а во втором потоке заявки на подключение поступают в сервисе электронной почты, из этого следуют, что сотрудник работает в двух разных информационных системах.

Помимо основного бизнес-процесса, существует и вспомогательный внутренний бизнес-процесс руководителей отдела – формирование отчетности и показателей эффективности работы. Имеющиеся два информационных потока не позволяют автоматизировать сбор статистики. Система управления проектами позволяет собрать данные о количестве обработанных задач за месяц одним сотрудником. Однако количество заявок, которые поступили по электронной почте, необходимо считать в ручном режиме. Кроме того, в отделе подключений не все заявки одинаковы по

сложности, трудовым и временным затратам. В связи с этим, к каждой задаче применяется коэффициент сложности. В случае с письмами в электронной почте руководителю необходимо прочитать каждое письмо, определить сложность задачи и применить соответствующий коэффициент.

Таким образом, более рациональной представляется ситуация, когда все входящие заявки на подключение и обратная связь по настройке контрагентов отображалась бы только в одном ресурсе: в системе управления проектами. Это позволит избежать работы в разных сервисах и автоматизировать сбор статистики показателей эффективности отдела подключений.

Найти способ работы только из системы управления проектами – такая задача была поставлена руководителем отдела подключений. Решение проблемы видится в разработке интеграционного решения [9, 21] по синхронизации данных между сервисом электронной почты и системой управления проектами.

1.2 Разработка требований к интеграционному решению по синхронизации системы управления проектами и сервисом электронной почты

Руководителем отдела подключений была поставлена задача, чтобы вся работа с задачами на подключение происходила только в системе управления проектами. Это было мотивировано проблемами с автоматическим сбором статистики и желанием повысить эффективность работы с задачами.

Так, руководителем отдела были сформулированы бизнес-требования к системе. В первую очередь интеграция с почтой [1] должна происходить в отдельном проекте в системе управления проектами. Для этого должен быть создан проект SSKK request и к нему были сформулированы требования, которые выражались в ряде задач на разработку. В целом требования к получившемуся решению можно разделить на два типа: первый тип –

автоматизация работы с задачами на подключение, второй – автоматизированный сбор статистики.

Для автоматизации работы с задачами требования также можно разделить на две категории: работа с исходящими письмами и работа с входящими письмами. Работа с исходящими письмами предполагает, что из системы управления проектами есть возможность отправлять исходящие электронные письма. Для реализации этого функционала система управления проектами должна быть интегрирована с сервисом электронной почты по протоколу SMTP (Simple Mail Transfer Protocol) [18], иметь соответствующую кнопку в задаче, по нажатию на которую будет открываться окно с письмом. Для отправки исходящих писем с почтой должны быть интегрированы все проекты, которые есть в отделе подключений, то есть в каждом проекте должна быть кнопка для отправки письма. Для автоматизации отправки заявки из общего ящика должен быть шаблон заявки, в который в соответствующие строки подставляются значения из полей задачи в системе управления проектами, подпись того сотрудника, который является исполнителем задачи, автоматически заполняется тема письма и подставляются адреса получателей в зависимости от значения поля «Ключевой клиент» в задаче.

Работа с входящей почтой предполагает, что также существует возможность получать новые электронные письма от контрагентов в системе управления проектами. Система управления проектами должна иметь возможность при получении любого нового входящего письма создавать новую задачу в проекте, который интегрирован с сервисом электронной почты. Для получения входящей почты предназначен отдельный проект и каждое новое письмо должно создавать новую задачу. В соответствии с бизнес-процессом в отдел подключений поступают два типа электронных писем: новые заявки и ответы на исходящие заявки. Система должна уметь определять, какой тип у этого письма. Необходимо, чтобы была проверка по теме сообщения. Каждое отправленное письмо в теме имеет номер задачи в

системе управления проектами, и система должна осуществлять проверку. Если в системе управления проектами уже существует задача с таким номером, то задачи должны связаться между собой и письмо должно создать новый комментарий в задаче, при этом задача должна сменить свой статус на «Открыт» при получении нового комментария, а задача в проекте, который связан с сервисом электронной почты, должна сменить статус на «Хлам». В связи с этим проект должен быть интегрирован не только с сервисом электронной почты, но и со всеми остальными проектами отдела подключений, иначе система не сможет найти задачу, если она существует. В случае, если система не находит задачу с номером существующей задачи, то такая задача остается в проекте, связанным с системой электронной почты в статусе «Открыт».

Таким образом, интеграционное решение должно обеспечивать работу с входящим и исходящим информационным потоком, что может имитировать работу из одного «окна», сотрудник не должен пользоваться сервисом электронной почты. Для получения информации о смене статуса задачи, о получении нового комментария, об упоминании сотрудника в задаче в системе управления проектами настроены уведомления, что избавляет от необходимости использования сервиса электронной почты.

Система управления проектами и сервис электронной почты должны обмениваться данными по протоколам электронной почты. Отправка исходящих писем осуществляется по протоколу SMTP. Протокол работает в клиент-серверной архитектуре [18]. В отделе подключений для отправки электронной почты используется Microsoft Outlook, который использует SMTP для подключения к email-серверу и передачи сообщения на сервер получателя. Данный почтовый протокол поддерживает аутентификацию и шифрование для обеспечения безопасной передачи сообщений. Но поскольку данный протокол поддерживает только отправку сообщений, для обеспечения получения электронных сообщений используется протокол IMAP [8]. Протокол позволяет получать доступ к сообщениям электронной

почты непосредственно на сервере электронной почты, он работает путем поддержания постоянного соединения между клиентом и сервером: когда клиент запрашивает доступ к электронному сообщению, сервер отправляет копию сообщения клиенту. Для соединения должен быть использован порт 993 для зашифрованных соединений.

Перейдем к требованиям, касающиеся автоматизации сбора статистики. После интеграции все задачи попадают систему управления проектами и обрабатываются в ней, для удобства сбора статистики должен быть настроен отчет по выполненным задачам. В отделе подключений все задачи имеют разный тип сложности, не одинаковы по затратам рабочего времени, все сотрудники имеют разный грейд, что означает, что разным сотрудникам регламентируется разное время выполнения одной задачи. Так, предполагается, что сотрудник с грейдом «Junior» затрачивает на одну задачу 20 минут, а сотрудник с грейдом «Middle» затрачивает 15 минут. В связи с этим норма выполненных задач у сотрудника с грейдом «Middle» увеличивается. Руководителю отдела подключений необходимо предоставить данные по эффективности и производительности сотрудников с учетом вышеуказанных нюансов. В отделе разработана определенная система оценки сотрудников, при которой применяются коэффициенты сложности задачи и учитывается грейд сотрудников. Отчет в системе управления проектами должен автоматизировать подсчет SP сотрудника, чтобы руководителю отдела не требовалось выгружать задачи вручную и делить их по типам и умножать на коэффициенты. Кроме того, при формировании отчета должна быть фильтрация, то есть система не должна добавлять в отчет все задачи, а только задачи конкретного сотрудника за текущий месяц и только задачи, находящиеся в конечных статусах. Таким образом, отчет в системе управления проектами должен суммировать задачи по определенным критериям и учитывать коэффициент их сложности, став основой для подсчета статистики, которая уже учитывает грейд сотрудников.

Вывод по первой главе

В главе были проанализированы основные бизнес-процессы группы прямых подключений, указаны проблемные места: работа в разных информационных системах и необходимость автоматизировать сбор статистики. Сделан вывод о том, что для решения указанных проблем необходимо разработать интеграцию между информационными системами. Для проектирования информационной системы были разработаны бизнес-требования к единой системе, которая позволяла бы эффективным образом работать с задачами и создавать автоматический отчет по выполненным задачам каждого сотрудника группы прямых подключений.

Глава 2 Проектирование интеграционного решения для синхронизации данных между сервисом электронной почты и системы управления проектами

2.1 Описание интеграционного решения в отделе подключений службы сопровождения ключевых клиентов

Система управления проектами состоит из клиентской части в виде графического интерфейса и серверной части с базой данных (система использует встроенную базу данных Xodus [29] для записи и хранения данных) и используется во всей компании разными командами. Поэтому с помощью REST API можно производить различные действия программно, в том числе импорт существующих проектов и задач из других систем отслеживания ошибок, создание, изменение, получение всех атрибутов задач и управление проектами, пользователями, группами и ролями; для этого через REST API используется Hub – система управления пользователями и разрешениями [27].

Система управления проектами разграничивает права доступа сотрудников, для того чтобы зайти в систему, необходимо использовать Контур.Паспорт и присутствовать в одной из ролей в пандоре. Для того чтобы осуществлять доработки в системе управления проектами, у сотрудника должна быть роль «developer» [16].

В связи с тем, что система управления проектами используется во всей компании, помимо разграничения прав доступа, она должна позволять производить доработки под потребности конкретного проекта или структурного подразделения.

Так, разрабатываемая нами интеграция с почтовыми ящиками позволяет настроить соединение между системой управления проектами и почтовым сервером. После того, как соединение установлено, есть возможность преобразовывать входящие сообщения электронной почты в

задачи или комментарии в систему управления проектами. Разрабатываемая интеграция позволяет осуществить настройки так, чтобы каждое письмо создавало новую задачу, при создании новой задачи система управления проектами использует следующую информацию из сообщения электронной почты для создания новой задачи: тема сообщения копируется в заголовок задачи без специальных символов, тело сообщения копируется в описание задачи без HTML кода, с помощью парсера данных заполняются поля с реквизитами (наименование ключевого клиента, ИНН, КПП, GLN) в задаче, если эти реквизиты указаны в письме.

Когда система обнаруживает, что входящее сообщение связано с существующей задачей, текст сообщения добавляется к существующей задаче в виде комментария и задаче присваивается статус «Открыт», что позволяет сотруднику оперативнее реагировать на какие-либо изменения, происходящие с задачей. Система управления проектами сопоставляет входящее сообщение с существующей задачей при следующих случаях: во-первых, если значения свойств в заголовке сообщения соответствуют ранее обработанному сообщению электронной почты (это стандартный метод разбора сообщений в MIME-кодировке; в данном случае эти значения используются для цепочки комментариев, когда получатель отвечает на уведомления, которые отправляются при создании или обновлении задачи в проекте); во-вторых, если тема сообщения содержит идентификатор задачи в том же проекте.

Система управления проектами поддерживает два варианта управления электронной почтой. Первый – это переадресация электронной почты, в этом случае электронное письмо, отправленное на один адрес электронной почты, автоматически перенаправляется на другой адрес электронной почты. Например, с адреса электронной почты из домена компании можно перенаправить все входящие письма в проект системы. Этот вариант подходит для компаний, у которых уже есть учетные записи электронной

почты для получения запросов клиентов, и он относительно прост в настройке.

Второй вариант, который и был использован, это подключения к почтовому серверу, что позволяет обрабатывать сообщения, отправленные на рабочий адрес электронной почты, размещенный в домене компании. Интеграция с почтовыми ящиками в системе управления проектами поддерживает только протоколы POP/POPS и IMAP/IMAPS. В соответствии с требованиями необходимо было использовать протокол IMAP. Но в отделе подключений используется почтовый сервер Microsoft Exchange. Для него протоколом по умолчанию является MAPI, этот протокол не поддерживается системой управления проектами. Для того чтобы подключиться к Microsoft Exchange, необходимо включить и настроить протокол IMAP в Microsoft Exchange и настроить параметры SMTP с проверкой подлинности. Для Microsoft Exchange Server интеграция с почтовыми ящиками поддерживает подключения к общим почтовым ящикам, именно общий ящик используется в отделе подключений для отправки и получения электронных писем. Для проверки подлинности можно использовать имя пользователя и пароль любого пользователя, являющегося участником общего почтового ящика [28].

Для интеграции с электронной почтой в соответствии с требованиями был создан отдельный проект SSKK request. Данный проект в системе управления проектами имеет стандартный интерфейс, состоит из перечня задач. В задаче в проекте SSKK request проведены следующие настройки. В задаче созданы следующие поля: «Статус задачи для учета», «GLN», «ИНН», «Ключевой клиент», «e-mail отправителя», «SP из справочника», «SP учтено», «Тип задачи», «Дата завершения», «В работе», «Assignee». У поля «Статус задачи для учета» типом поля является состояние, есть возможность выбрать только один вариант, варианты выбора выпадающего списка: «Новый», «В работе», «Выполнено», «Не настроен», «Хлам», поле является обязательным, видимо для всех пользователей. Поля «GLN», «ИНН» и «e-

mail отправителя» имеют строковое значение, пользователь вводит значение вручную, видимы для всех. Поле «Ключевой клиент» имеет тип поля перечисление, можно выбрать только одиночное значение, из выпадающего списка можно выбрать только одно наименование ключевого клиента, поле видимо для всех пользователей. Поле «Тип задачи» имеет тип перечисление, можно выбрать только одно значение из выпадающего списка, варианты выбора: «Подключения (I уровень)», «Подключения (II уровень)», «Подключения (обратный роуминг)», «Отключения от КК», «Изменения настроек в EDI у организации», «Обработка партин», «Инициация/упрощенное тестирование», «Выводы в продуктив (старты массовых подключений)», «Доработки YТ/расширения». Поля «В работе» и «Дата завершения» имеют тип дата, с возможностью изменения, видимы для всех. Поле «Assignee» имеет тип пользователь, одиночное значение, которое можно выбрать из выпадающего списка всех пользователей, видимо для всех. Поля «SP из справочника» и «SP учтено» имеют тип значение с плавающей точкой, видны только для команды «Руководители ОП», поле «SP из справочника» является неизменяемым, поле «SP учтено» можно редактировать вручную.

Что касается сбора статистики, то в системе управления проектами нами разработан отчет, в котором отражаются Story Points (SP) каждого сотрудника. На основании этих данных считается эффективность каждого сотрудника. SP в отчете разделены по уровням сложности задач. В отчет попадают только те задачи, которые находятся в конечном статусе и были выполнены в текущем месяце. На основании поля «SP учтено» каждая задача имеет свой вес и в отчете суммируются уже не единицы (чистое количество задач), а SP задачи. В итоге в отчете получается сумма SP. Для подсчета статистики используются все проекты в отделе подключений, для этого у всех проектов должно быть общее поле, на основе которых производится сбор статистики. Для этого в проект SSKK request было добавлено поле «Тип задачи», которое есть во всех остальных проектах. Поле «Тип задачи»

заполняется на основании значения поля «Тип подключения», которое зависит от содержания самого запроса. Добавление поля «Тип задачи» позволяет привести значение уникального поля к стандартному и получить значение поля «SP из справочника», которое добавится автоматически.

В задачах нами настроено поле, в котором указывается статус задачи – «Статус задачи для учета», это поле заполняется автоматически на основании заполненного поля «Статус задачи», в отчет попадают задачи только со значением «Выполнено». Также в отчете отдельно отражаются задачи первого и второго уровней подключения. Уровень подключения зависит от поля «Ключевой клиент», на основании этого поля в отчете происходит разделение задач на первый и второй уровень. Это разделение реализовано на уровне правила, в котором перечислены все ключевые клиенты, которые относятся к подключению второго уровня.

Кроме того, для сбора статистики используются поля SP из справочника и SP учтено, которые необходимы для проставления SP для каждой задачи, то есть вес задачи, который зависит от ее сложности. В стандартном случае они должны быть равны, в нетипичных задачах обычно значение поля «SP учтено» больше, чем значение поля «SP из справочника», то есть сотруднику учитывается больше SP за одну задачу. Справочник – это перечень возможных вариантов сочетаний значений полей и соответствие SP за это сочетание: например, сочетание конкретного вида задачи и названия проекта, за каждый тип задачи указано разное стандартное значение SP. В проекте SSKK request количество SP за задачу в основном зависит от полей «Вид задачи» и «Ключевой клиент», тег «Запрос из почты» имеет повышенный вес задачи.

Кроме того, как уже отмечалось, в отделе подключений сотрудники имеют грейды: Junior, Middle, Senior. В задачах не отражается грейд, и задача имеет стандартный SP, который соответствует грейду Middle, то есть отчет настроен для сотрудника с грейдом Middle. Для сотрудников с отличными грейдами в окончательном отчете, отражающем эффективность,

проставляются другие коэффициенты сложности. Норма выполненных задач у всех сотрудников одинакова, но, учитывая повышенный коэффициент у сотрудников с низким грейдом, SP у них выше при одинаковом количестве задач.

Таким образом, в системе управления проектами нами настроен отчет, который реализует механизм подсчета SP задач конкретного сотрудника за текущий месяц в конечном статусе, разделенных по уровню подключения. На основании него руководитель отдела может посчитать эффективность каждого сотрудника, избегая ручных методов подсчета количества задач.

2.2 Логическое моделирование интеграционного решения

Для проектирования интеграционного решения [10, 23, 25] будет использована нотация UML [12], диаграммы UML существуют двух типов: структурные диаграммы и диаграммы поведения. Диаграммы позволяют с помощью унифицированного языка моделирования показать различные аспекты системы, такие как взаимодействие классов системы, поведение системы и взаимодействие системы и пользователя.

Интеграционное решение будет реализовано с помощью нового проекта в системе управления проектами, в который будут попадать все новые электронные письма. Рассмотрим варианты использования проекта для интеграции с почтовым сервисом. После того, как появилось новое письмо в сервисе электронной почты, почтовый клиент [19] создает новую задачу в связанном проекте в системе управления проектами. Соответственно появляясь новая запись в базе данных системы управления проектами и в проекте создается новая задача в статусе «Открыт». Таким образом, видно, что основная сущность проекта – это задача. Все новые письма в электронной почте будут создавать новую задачу в проекте. Задача будет являться новой, в ней основные поля будут пустыми, они должны быть заполнены сотрудниками отдела подключений или в автоматическом

режиме. То есть над задачей будет производиться анализ в соответствии с настроенными правилами, и если данные в задаче удовлетворяют определенным условиям, то над задачей будут производиться определенные действия. Например, если в заголовке задачи есть номер какой-либо предыдущей задачи, то происходит связывание двух задач и создается комментарий в существующей задаче или автоматически заполняется поле ключевой клиент на основании информации об отправителе письма.

На этапе существования новой задачи с ней могут работать пользователи и производить определенные действия с ней. Все изменения, происходящие с задачей, включают работу с данными, которые сохраняются в базе данных системы управления проектами.

Пользователи, которые могут работать с задачей делятся на две группы сотрудников, они имеют разный уровень доступа: группа сотрудников отдела подключений (имеет доступ для изменения задачи, редактирования данных, заполнения полей, смена значений полей статуса задачи) и группа руководителей отдела подключений (имеет доступ к тем же действиям, что и сотрудник отдела подключений, но также имеет доступ к просмотру и редактированию служебных полей для подсчета статистики в Story Points).

Рассмотрим более подробно действия, которые могут производить с задачей сотрудники отдела подключений. Сотрудник отдела подключений может выбирать вид задачи, менять ее статус, оставлять комментарии, редактировать описание, отправлять письма из задачи и выбирать исполнителя задачи. Руководители отдела подключений имеют права на все эти действия, но, как правило, только меняют исполнителя задачи. Со стороны руководителя вариант использования задачи такой: руководитель видит поле «SP учтено», меняет значение поля «SP учтено из справочника» и формирует отчет, то есть пользуется инструментами для автоматизированного сбора статистики. Диаграмма вариантов использования представлена на рисунке 3.

основные настройки. Основой бизнес-процесса является задача, которая наследует от проекта поля, то есть поля могут быть разными в разных проектах. Кроме полей задача также имеет комментарии и описание. Поля зависят друг от друга, и изменение значения одного поля меняет значение другого. Так, изменение поля задачи «Статус» с «Открыт» на «В работе» автоматически изменит значение поле «Assignee», автоматически будет проставлено значение текущего пользователя, в ручном режиме имеется возможность проставить другого пользователя из выпадающего списка. Также поле «Assignee» зависит от значения поля «Ключевой клиент», а от значения поля «Assignee» зависит значение поля «Тип задачи», так как каждый сотрудник имеет свою зону ответственности и распределение ключевых клиентов, то есть каждый сотрудник работает с определенными ключевыми клиентами и определенными типами задач. Это взаимодействие отображено на диаграмме классов на рисунке 4.

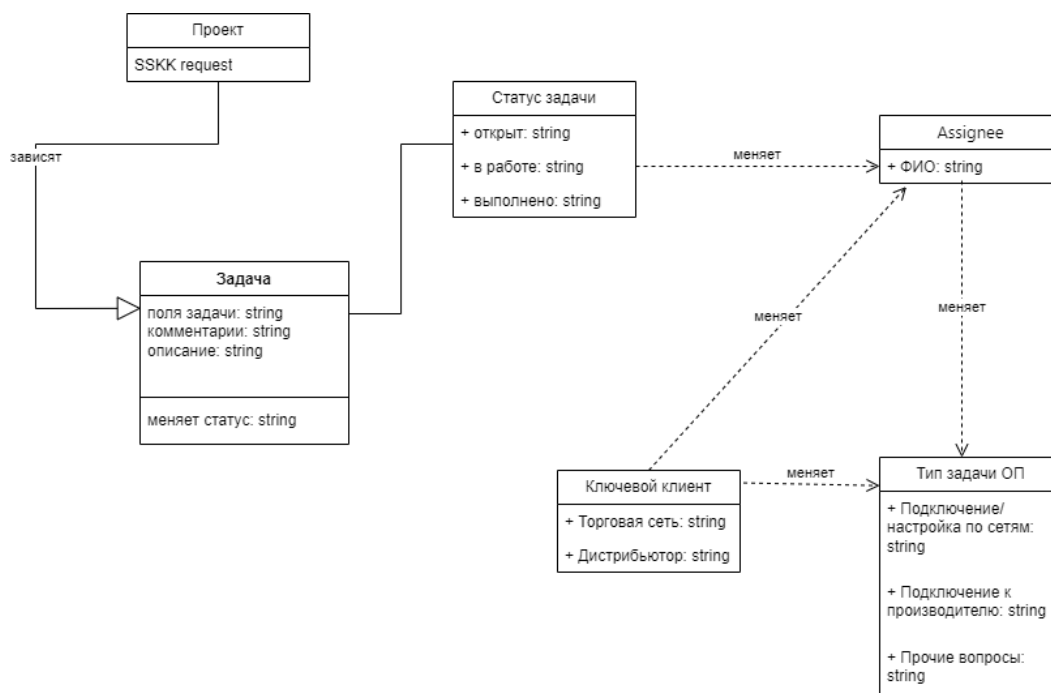


Рисунок 4 – Диаграмма классов

Далее рассмотрим поведение системы, в каких состояниях она может находиться и как меняются состояния в интеграционном решении в ответ на определенные события. Как уже было отмечено, основной Сущностью в проекте в интеграционном решении для синхронизации данных с почтовым сервисом является задача. У каждой задачи есть определенный жизненный цикл. Рассмотрим его более подробно. Изначально проект в системе управления проектами для интеграции с сервисом электронной почты подключен почтовому сервису и находится в состоянии ожидания нового электронного письма. Как только сервис электронной почты получает новое письмо, в системе управления проектами создается задача. Происходит внешнее событие и меняется состояние информационной системы, в ней появляется новая задача. Жизненный цикл задачи в информационной системе начинается с ее состояния «Открыт», то есть задача появилась в очереди задач, поле «Статус задачи» имеет значение «Открыт», и задача находится в ожидании смены статуса, то есть смены значения поля «Статус задачи» с «Открыт» на значение «В работе». Далее, чтобы произошла смена состояния системы, необходимо внешнее воздействие сотрудника на систему, а именно смена значения поля «Вид задачи ОП», если поле заполнено, то состояние системы меняется, если поле не заполнено, то статус задачи остается прежним (происходит в том случае, если с задачей поработали, но значение поля «Вид задачи ОП» осталось пустым, то смена состояния системы не произойдет). Если значение поля «Статус задачи» сменен на «В работе», то далее происходит заполнение основных полей задачи на основе описания задачи через инструмент парсинга данных. После этого система находится в состоянии ожидания действий пользователя. В зависимости от успешности действий меняется значение поля «Статус задачи» либо на «Выполнен», либо на «Не настроен». Оба эти статуса являются конечными и жизненный цикл задачи завершается. Состояния системы отражены на диаграмме состояний на рисунке 5.

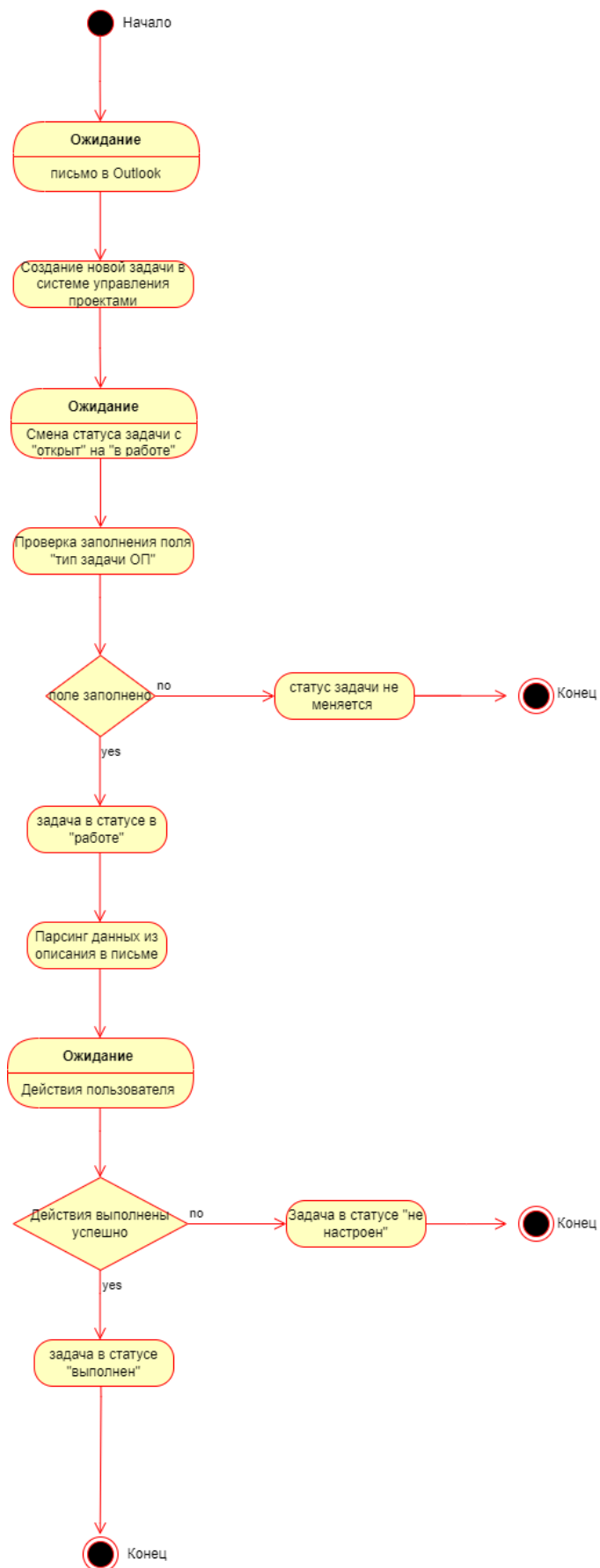


Рисунок 5 – Диаграмма состояний

И, наконец, рассмотрим взаимодействие интеграционного решения с пользователем.

Система управления проектами взаимодействует с почтовым сервером через протокол электронной почты IMAP. Почтовый сервис, который используется в СКБ Контур, Microsoft Exchange, протоколом по умолчанию которого является MAPI. Данный протокол не поддерживается системой управления проектами, поэтому необходимо настроить протокол IMAP в Microsoft Exchange и настроить параметры аутентифицированного SMTP. Для Microsoft Exchange Server интеграция почтовых ящиков поддерживает подключения к общим почтовым ящикам, используется имя пользователя и пароль для аутентификации любого пользователя, который является членом общего почтового ящика. Таким образом, система управления проектами получает сообщение от почтового сервера через протокол IMAP и создает новую задачу. Данные сохраняются в базе данных системы [2]. Используется встроенная база данных Xodus. Xodus является универсальной базой данных и используется в системе управления проектами, задействованной в СКБ Контур. Указанную базу данных можно использовать как транзакционное хранилище ключей и значений, которое обеспечивает высокую производительность, или выбрать его объектно-ориентированную модель данных для более сложных структур. Эта двойственная природа достигается посредством многоуровневой архитектуры, которая поддерживает как транзакционное хранилище ключей и значений (среды), так и набор типизированных сущностей с именованными свойствами и отношениями (хранилища сущностей). По сравнению с аналогичными базами данных, такими как SQLite и MongoDB, Xodus выделяется благодаря своему механизму хранения только с добавлением, что снижает случайные накладные расходы ввода-вывода, вызванные обновлениями на месте. Xodus работает на уровне изоляции моментального снимка, гарантируя согласованный моментальный снимок всей базы данных для всех чтений в рамках транзакции. В отличие от других баз данных, которые предоставляют

несколько уровней изоляции, что приводит к потенциальной несогласованности данных, Xodus сохраняет целостность данных. Эта уникальная функция вытекает из лог-структурированного дизайна Xodus, где все изменения последовательно записываются в журнал. Этот журнал, серия файлов .xd, служит неизменяемым хранилищем записей данных. Каждая зафиксированная транзакция создает новый моментальный снимок (версию) базы данных, позволяя последующим транзакциям ссылаться на этот моментальный снимок. Этот дизайн эффективно преобразует базу данных в постоянную функциональную структуру данных, обеспечивающую управление параллелизмом с несколькими версиями [29].

Рассмотрим общую структуру интеграционного решения, которая показана на диаграмме компонентов на рисунке 6 и состоит из двух крупных компонент – сервиса электронной почты и системы управления проектами, внутри системы управления проектами существует две зависимости в виде проекта необходимого для обработки задач и модуля формирования отчетности в автоматическом режиме.

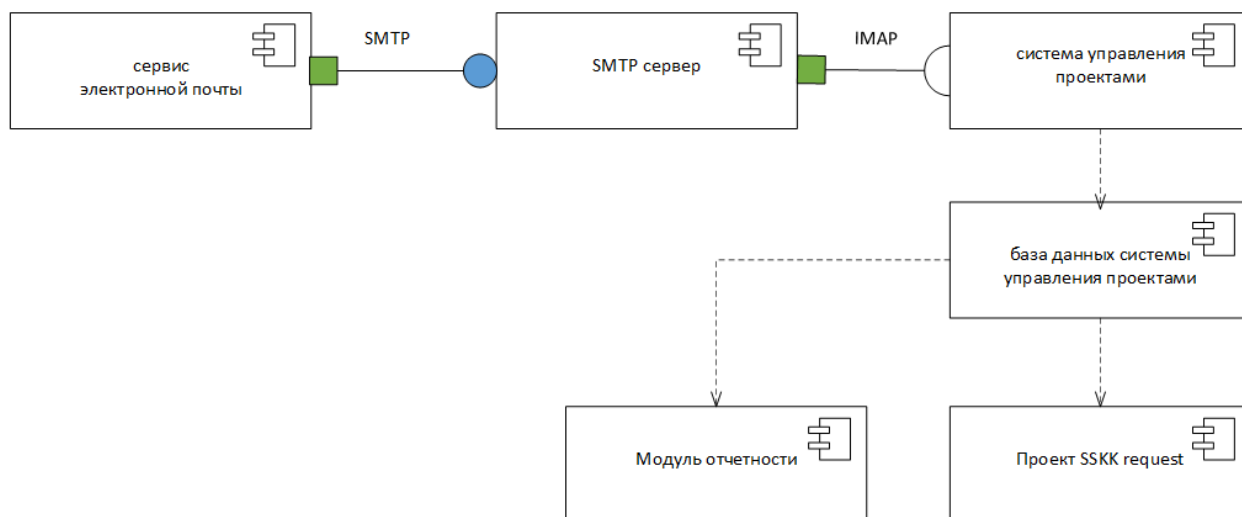


Рисунок 6 – Диаграмма компонентов

Вывод по второй главе

В главе приведено описание разрабатываемой системы интеграции между системой управления проектами и сервисом электронной почты: описано само интеграционное решение, каой функционал она должна выполнять и каким образом должна быть реализована. Описан новый проект и указаны его функции, перечислена последовательность действий системы, а также разработана логическая модель интеграционного решения в виде диаграмм UML. С помощью диаграмм показаны компоненты системы, способ их взаимодействия, показаны состояния, в которых может находиться система и действия, меняющие состояние системы, кроме того, показаны классы, из которых состоит система и их зависимость друг от друга, а также варианты использования системы, какие действия могут быть произведены над системой разными группами пользователей.

Глава 3 Разработка и тестирование интеграционного решения для обмена данными между системой управления проектами и сервисом электронной почты

3.1 Реализация интеграционного решения для отдела подключений службы сопровождения ключевых клиентов

Разработка интеграционного решения заключалась в том, чтобы настроить обмен данными между сервисом электронной почты и системой управления проектами [11]. Система управления проектами является довольно гибким инструментом и позволяет дорабатывать реализованный функционал под конкретные потребности компании. Для решения этой задачи система управления проектами имеет встроенный редактор скриптов, что позволяет дорабатывать функционал с помощью кода, написанного на языке JavaScript [26].

В отделе подключений существует два информационных потока: входящий и исходящий. Для комфортной работы нами было предложено внедрить в систему управления проектами оба этих потока.

Рассмотрим работу системы управления проектами с входящей почтой. Для того, чтобы автоматизировать сбор статистики, необходимо, чтобы все задачи были только в системе управления проектами. Для этого необходимо, чтобы все заявки, попадающие в сервис электронной почты, создавали новую задачу в системе управления проектами. Чтобы реализовать данный функционал, к системе управления проектами нами было предложено подключить сервис электронной почты (в СКБ Контур используются Microsoft Exchange) [20]. У сотрудников отдела подключений нет прав для интеграции систем, поэтому само подключение осуществляли сотрудники смежных отделов. Сложность заключалась в том, что сервер Microsoft Exchange использует протокол MAPI, а система управления проектами передает данные по протоколу IMAP, для решения проблемы на сервере

Microsoft Exchange был включен прокол IMAP. В системе управления проектами был указан защищенный порт передачи данных 993.

После того, как соединение было установлено, нами был настроен проект, в котором создаются все задачи из писем электронной почты. Был создан отдельный проект SSKK request, созданы индивидуальные поля для этого проекта и поля, которые есть в других проектах для связывания этого проектами с другими, чтобы была возможность автоматизировать сбор статистики с помощью реализации автоматического подсчета SP.

Работа с входящей почтой нами была доработана с помощью правил рабочих процессов, написанных на языке JavaScript, внутри системы управления проектами [30].

Базовая настройка системы управления проектами осуществляется с помощью настройки полей через правила. Правило (rule) – это фрагмент кода (скрипт), который позволяет выполнять определенные действия с задачами в системе управления проектами. Правила настраиваются через рабочий процесс. Рабочий процесс (workflow) – это сборник правил, которые можно подключить как к одному проекту, так и ко всем сразу [14]. Рабочие процессы в системе управления проектами позволяют настраивать и автоматизировать работу с задачами в проектах, уведомлять о событиях, обеспечивать соблюдение регламентов при оформлении задач, выполнять периодические задачи [16].

Существуют несколько видов правил. Первый вид – это правила при изменении – они запускаются автоматически только при изменении определенного поля или тега в задаче, при создании правила указывается команда, которую необходимо выполнить с задачей при изменении заданных условий. Такие типы правил используются при валидации (проверке полей на корректность заполнения), проставлении даты или комментария, создания напоминаний, автозаполнении полей, отправки уведомлений, заполнении описания задач. Следующий вид правил – действие. Данное правило создает дополнительный пункт в выпадающем меню «три точки» в системе

управления проектами, в котором указывается команда, которую необходимо выполнить с задачей (может запускаться как для одной задачи, так и массово для списка задач. Данный тип правил используется для массовой обработки задач и для принудительного выполнения команд. Третий вид правил – по расписанию. Этот вид правил запускается автоматически в определенное время, с определенной периодичностью, в этих правилах указывается команда, которую необходимо выполнить с задачей при наступлении определенного времени. Правила используются для напоминаний в день наступления даты и для автоматического перевода задач в статус «Архив». Четвертый вид правил – настраиваемый. Они содержат код, который можно использовать в других правилах, в отделе подключений они используются как справочник со значениями полей. Последний вид правил – конечного автомата. В отделе подключений эти правила не используются, но такие правила могут регулировать переходы от одного значения к другому для поля и отправлять напоминания по графику.

Структура правила состоит из трех частей: определение вида правила, условие правила и действие, которое необходимо выполнить при наступлении условия.

Виды правил определяются следующим образом:

exports.rule

При изменении:

= entities.Issue.onChange

Действие:

= entities.Issue.action

По расписанию

= entities.Issue.onSchedule

Конечный автомат

=entities.Issue.stateMachine

Следующий блок правила – это заголовок:

title – заголовок правила (как правило будет называться в рабочем процессе). Для правил Действия – это еще и название кнопки для запуска правила.

Условия, которые должны быть выполнены для запуска правила, указываются в разделе guard.

Условия, которые могут запускать правила:

- поле (issue.fields) – изменилось, значение равно, значение не равно, не заполнено;
- тег (hasTag) – есть / нет тега;
- свойство задачи – создана, изменена, название проекта;
- время – разница во времени, наступление даты.

Следующим обязательным блоком правила является action – действия, которые применяются при выполнении условий guard.

Действия, которые используются в задачах отдела подключений: написать комментарий, сделать напоминание, проставить дату, заполнить описание задачи, добавить тег, проверить заполнение поля, отправить письмо, заполнить поле.

Для того, чтобы добавить правило, необходимо добавить новый рабочий процесс – либо через окно редактирования правил, либо через настройки. В рабочем процессе в редакторе добавляется код правила на языке JavaScript. Кроме того, система управления проектами позволяет добавлять правила с помощью конструктора, но это имеет ряд ограничений по действиям, можно создать ограниченный набор типовых правил.

Рассмотрим правила, которые нами настроены для проекта, интегрированного с сервисом электронной почты – SSKK request. Условно правила для указанного проекта можно разделить на два вида: правила, автоматизирующие работу с задачей, и правила, настроенные для подсчета SP.

Правила, которые автоматизируют работу с задачей, необходимы для исключения повторяющихся действий сотрудников, направлены на

сокращение затрат времени обработки задачи и сокращения ошибок, связанных с человеческим фактором.

Так, для каждой задачи, которая создается по входящему письму из почты, добавляется тег «запрос из почты». Код правила представлен на рисунке 7.

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');

exports.rule = entities.Issue.onChange({

  title: 'Автопроставление тега',
  guard: (ctx) => {
    return ctx.issue.becomesReported;
  },
  action: (ctx) => {
    const issue = ctx.issue;
    ctx.issue.addTag("Запрос из почты");
  },
  requirements: {

  }

});
```

Рисунок 7 – Правило для проставления тега «запрос из почты»

Далее для проекта реализована возможность автоматического заполнения пустых полей в новой задаче из данных, которые есть в письме. В каждой новой задаче автоматически заполняется поле «Ключевой клиент», реализовано это с помощью парсинга письма по домену электронной почты ключевого клиента. Используемые поля в системе управления проектами «Ключевой клиент» и «e-mail отправителя». Код правила представлен на рисунке 8.

```

const entities = require('@jetbrains/youtrack-scripting-api/entities');

exports.rule = entities.Issue.onChange({

  title: 'Парсинг:Заполнение поля Ключевой клиент по домену',
  guard: (ctx) => {
    return ctx.issue.becomesReported;
  },
  action: (ctx) => {
    try {
      if (ctx.issue.fields.PostAdress.includes("@test.com")) {
        ctx.issue.fields.NamePole = 'ООО Тестовый дистр';
      } else if (ctx.issue.fields.PostAdress.includes("@primer.ru")) {
        ctx.issue.fields.NamePole = 'ТС Пример';
      }
    } catch (err) {
      console.log("e-mail отправителя не заполнен");
    }
  },
  requirements: {
    PostAdress: {
      name: 'e-mail отправителя',
      type: entities.Field.stringType
    },
    NamePole: {
      name: 'Ключевой клиент',
      type: entities.EnumField.fieldType
    },
  },
});

```

Рисунок 8 –Правило для заполнения поля «Ключевой клиент».

Таким образом, с помощью указанных выше настроек интеграции и написанных нами правил, в системе управления проектами создается новая задача, которая имеет специальный тег и в которой автоматически заполняется поле «Ключевой клиент». Само содержание письма отображается в описании задачи. В письме ключевой клиент указывает, какие настройки ему нужны и с каким контрагентом. В задаче необходимо, чтобы были заполнены реквизиты поставщика: наименование, GLN, ИНН, КПП. Для автоматического заполнения указанных полей реализовано правило парсинга письма. Это необходимо для ускорения обработки писем,

исключения заполнения полей вручную. Правило нами было реализовано с помощью кнопки в системе управления проектами «парсинг данных». При нажатии на кнопку запускается правило, которое позволяет найти совпадения и заполнить поля задачи. Если данных в письме нет, то возникает предупреждение, что в письме нет ИНН, КПП или GLN. Правило работает как для юридических лиц, так и для индивидуальных предпринимателей, так как у них отличается количество знаков в ИНН. Код правила приведен в приложении А.

После того, как все поля, касающиеся контрагентов, в задаче заполнены, необходимо, чтобы была проверка на заполнение остальных полей, которые касаются определения типа задачи, в случае, если они не заполнено, то при смене значения поля «Статус задачи для учета» на «Выполнено» будет появляться блокирующее сообщение «Необходимо указать Вид задачи ОП». Код правила представлен на рисунке 9.

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
const workflow = require('@jetbrains/youtrack-scripting-api/workflow');

exports.rule = entities.Issue.onChange({
  title: 'Валидация: Обязательность поля Вид задачи при переводе в Выполнено',
  guard: (ctx) => {
    return (ctx.issue.fields.isChanged('Статус задачи для учета') && ctx.issue.fields.becomes('Статус задачи для учета', 'Выполнено'));
  },
  action: (ctx) => {
    if(ctx.issue.fields.becomes('Статус задачи для учета', 'Выполнено'))
    {
      workflow.check(ctx.issue.fields["Вид задачи ОП"]!=null, "Необходимо указать Вид задачи ОП");
    }
  },
  requirements: {
  }
});
```

Рисунок 9 – Код правила на валидацию заполнения поля «Вид задачи»

Также нами было настроено правило на валидацию данных: поле должно быть обязательным при определенных условиях. Если поле «Вид задачи ОП» заполнено и имеет одно из значений: отключение, подключение к сверке, подключение по страхованию, подключение к производителю, то при смене поля «Статус для учета» на «Выполнено» проверяется заполнение поля «Наименование организации», в случае, если поле не заполнено, появляется ошибка «Поле 'Наименование организации' обязательно для заполнения»). Код правила представлен на рисунке 10.

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
const workflow = require('@jetbrains/youtrack-scripting-api/workflow');

exports.rule = entities.Issue.onChange({

  title: 'Валидация: Обязательность поля Наименование организации',
  guard: (ctx) => {
    return (ctx.issue.fields.isChanged('Статус задачи для учета') && ctx.issue.fields.becomes("Статус задачи для учета", 'Выполнено'));
  },
  action: (ctx) => {

    if (ctx.issue.is('Вид задачи ОП', "Отключение")||ctx.issue.is('Вид задачи ОП', "Подключение по Сверке")||ctx.issue.is('Вид задачи ОП', "Подключение по Страхованию")||ctx.issue.is('Вид задачи ОП', "Подключение к производителю"))
    {
      workflow.check(ctx.issue.fields["Наименование организации"]!=null, "Поле 'Наименование организации' обязательно для заполнения");
    }

  },
  requirements: {
  }

});
```

Рисунок 10 – Правило на валидацию заполнения поля

После того, как все поля, необходимые для сотрудника, заполнены, необходимо автоматизировать рутинные действия сотрудника. Так, если сотрудник работает над задачей, то он должен в поле «Assignee» указать себя, за день в отделе подключений каждый сотрудник обрабатывает в среднем 50 задач и необходимо сделать так, чтобы сотруднику не нужно было проставлять себя исполнителем вручную. Для этого нами разработано

правило автопроставления исполнителя задачи, оно реализует заполнение поля «Assignee» логином текущего пользователя, при соблюдении определенных условий. Код правила представлен на рисунке 11.

```
// Правило заполняет поле Assignee логином текущего пользователя, при соблюдении
// определенных условий.

var entities = require('@jetbrains/youtrack-scripting-api/entities');
const workflow = require('@jetbrains/youtrack-scripting-api/workflow');

exports.rule = entities.Issue.onChange({
  title: "Автозаполнение поля Assignee",
  guard: function(ctx) {
    // Запускается при смене статуса задачи на В работе
    return (ctx.issue.fields.isChanged('Статус задачи') && ctx.issue.fields.becomes("Статус задачи", 'В
работе'));
  },

  action: function(ctx) {
    // Если поле Assignee пустое - заполняется при взятии в работу.
    if (ctx.issue.fields.is(ctx.Assignee, null) && ctx.issue.fields.is(ctx.Assignee, undefined)) {
      ctx.issue.fields.Assignee = ctx.currentUser;
      console.log("Было пустое - заполнилось");
    }
    else {
      if (ctx.issue.comments.isEmpty() || !(ctx.issue.comments.last().author.login == ctx.user_0.login) )
        {ctx.issue.fields.Assignee = ctx.currentUser;
        console.log("Assignee изменен");}
      if (ctx.issue.comments.isNotEmpty() && (ctx.issue.comments.last().author.login == ctx.user_0.login)
      && !(ctx.issue.fields.Assignee.login == ctx.currentUser.login))
        {workflow.message("Поле Assignee не изменено. Если нужно, скорректируй его
самостоятельно.");
        ctx.issue.addTag("Отпуск Assignee");
        }
      else {ctx.issue.fields.Assignee = ctx.currentUser;}
    }
  },
  requirements: {
    Assignee: {
      type: entities.User.fieldType
    },
    user_0: {
      type: entities.User,
      login: 'Почта_edi_request'
    }
  }
});
```

Рисунок 11 – Правило для автоматизации заполнения поля «Assignee»

Правило запускается при изменении значения поля «Статус задачи» на «В работе». И если поле «Assignee» пустое, то оно заполнится логином текущего пользователя; если поле «Assignee» уже заполнено и нет комментариев или автор последнего комментария не Почта_edi_request, то поле «Assignee» будет заполнено также логином текущего пользователя. Но в случае, если последний комментарий от пользователя Почта_edi_request и логин пользователя не совпадает со значением поля «Assignee», то значение поля «Assignee» не изменится и в системе управления проектами появится сообщение: «Поле Assignee не изменено. Если нужно, скорректируй его самостоятельно».

На рисунке 12 представлен код правила, которое переоткрывает задачу при появлении нового комментария.

После интеграции системы управления проектами и сервиса электронной почты сотрудники отдела подключений работают с задачами только в системе управления проектами, и после отправки электронного письма ключевому клиенту сотрудник переводит задачу в статус «Выполнено», но часто возникает такая ситуация, что ключевой клиент предоставляет какую-либо обратную связь, этот ответ прикрепляется в задачу в комментарий, но сотрудник не знает об этом. Для решения этой проблемы существует правило, которое меняет значение поля «Статус задачи» на «Открыт» после получения нового комментария. Это правило реализовано не только для комментариев от пользователя электронной почты, но и для комментариев от любого сотрудника, который не входит в отдел подключений. Это правило позволяет оперативно реагировать на задачи, с которыми необходимо произвести какие-либо действия.

```

// Переоткрывает задачу при появлении новых комментариев.

const entities = require('@jetbrains/youtrack-scripting-api/entities');

exports.rule = entities.Issue.onChange({

  title: 'Перевод в открыт при новом комментарии',
  guard: (ctx) => {
    var myTask = ctx.issue.id === "SSKKSS-6361";
    return ctx.issue.comments.added.isNotEmpty() && !myTask;
  },

  action: (ctx) => {
    if (ctx.issue.comments.added.isNotEmpty() && !ctx.issue.comments.last().author.isInGroup("Отдел
подключений") && !(ctx.issue.comments.last().author.login === ctx.user_0.login)) {
      ctx.issue.fields.Status = "Открыт";
      console.log(ctx.issue.id + " правило включилось ");
    }
  },

  // данные о полях
  requirements: {
    Status: {
      type: entities.EnumField.fieldType,
      name: "Статус задачи",
    },
    user_0: {
      type: entities.User,
      login: 'YouTrack.Workflow'
    }
  }
});

```

Рисунок 12 – Правило для смены значения поля

При выполнении настроек на стороне ключевого клиента, ключевые клиенты предоставляют обратную связь о том выполнены настройки или не выполнены, а также могут запрашивать какую-либо дополнительную информацию. Такие письма с обратной связью прикрепляются к исходной задаче на подключение в виде комментария, но эти комментарии не должны быть видны коллегам из смежных подразделений, потому что часто это бывает рабочая информация и коллеги могут выполнять лишнюю работу. Для того, чтобы сотрудники не видели комментарии от ключевых клиентов в отделе подключений нами разработано правило, которое скрывает комментарии,

полученные от пользователя электронной почты `edi_request`, от всех пользователей, которые не входят в отдел подключений. Комментарий появляется в задаче и у него стоит ограничение на область видимости и указана группа пользователей «Отдел подключений», которые могут видеть оставленный комментарий. Однако получение данного комментария также меняет статус задачи и она переходит в статус «Открыт». Код правила представлен на рисунке 13.

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
exports.rule = entities.Issue.onChange({
  title: 'Скрыть комментарий от edi_request',
  guard: (ctx) => {
    // запускается, когда появится новый комментарий не пустой
    return ctx.issue.comments.added.isNotEmpty();
  },
  action: (ctx) => {
    var issue = ctx.issue;
    if (ctx.issue.comments.added.isNotEmpty() && (ctx.issue.comments.last().author.login ===
    ctx.user_0.login))
    {
      issue.comments.added.forEach(function(comment) {
        comment.permittedGroup = ctx.usergroup_0;
      });
    }
  },
  requirements: {
    usergroup_0: {
      type: entities.UserGroup,
      name: 'Отдел подключений'
    },
    user_0: {
      type: entities.User,
      login: 'edi_request'
    }
  }
});
```

Рисунок 13 – Правило для отображения комментариев

Далее необходимо, чтобы были заполнены также поля для автоматизированного сбора статистики. Рассмотрим второй тип правил, которые действуют в проекте SSKK Request.

Основой для формирования отчета являются поля «SP учтено» и «SP из справочника». Именно их значение показывает вес одной задачи, которая зависит от ее сложности. Поэтому важно автоматизировать подсчет сложности задачи и автопроставление SP задачи. На первом этапе должно заполниться поле «SP из справочника». Правило позволяет при изменении значения поля «Статус задачи для учета» на «Выполнено» в поле SP учтено указать значение поля «SP из справочника», это значение подставляется из справочника, в котором описаны разные комбинации значения полей, и в зависимости от этих комбинаций указаны значения SP для задачи. Код правила представлен на рисунке 14.

Но в отчет входят только задачи, находящиеся в конечном статусе, поэтому правило запускается только при значении поля «Статус задачи для учета» «Выполнено». Важно отметить, что поле «Статус задачи для учета» является служебным и видимо только для руководителей отдела подключений. Сотрудники заполняют поле «Статус задачи», и это поле имеет больше конечных статусов. Поэтому существует правило, которое приравнивает разные конечные статусы задачи к одному значению «Выполнено» для поля «Статус задачи для учета», то есть при смене значения поля «Статус задачи» автоматически меняется значение поля «Статус задачи для учета». Соответствие полей является следующим: значению поля «статус задачи для учета» «В работе» соответствуют значения «В работе» и «Письмо отправлено» поля «Статус задачи»; новой задаче в поле «Статус задачи для учета» соответствуют значения «На стадии создания» и «Открыт» поля «Статус задачи»; значению «Выполнено» поля «Статус задачи для учета» соответствуют все остальные значения поля «Статус задачи». Код правила представлен в приложении Б.

```

const entities = require('@jetbrains/youtrack-scripting-api/entities');
exports.rule = entities.Issue.onChange({
  title: 'Заполнение sp учтено',
  guard: (ctx) => {
    return ctx.issue.isChanged(ctx.StageTask);
  },
  action: (ctx) => {
    console.log(ctx.issue.id);
    const issue = ctx.issue;
    var oldValue = "";
    if (issue.fields.oldValue(ctx.StageTask) != null) {
      oldValue = issue.fields.oldValue(ctx.StageTask);
      console.log(oldValue);
    }
    console.log(issue.fields.is(ctx.StageTask));
    //проверяем, что поле в задаче меняется на значение, которое обозначение что задача выполнена
    if ((issue.fields.is(ctx.StageTask, ctx.StageTask.Production)) && oldValue !=
    ctx.StageTask.Production) {
      issue.fields.SpAccounted = issue.fields.SpIsdirectory;
    }
  },
  requirements: {
    StageTask: {
      name: "Статус задачи для учета",
      type: entities.State.fieldType,
      Production: {
        name: 'Выполнено'
      },
    },
    SpAccounted: {
      name: 'SP учтено',
      type: entities.Field.floatType
    },
    SpIsdirectory: {
      name: 'SP из справочника',
      type: entities.Field.floatType
    },
    ProjectEDI: {
      type: entities.Project,
      name: 'EDIImplementation'
    },
  },
});

```

Рисунок 14 – Правило для заполнения поля «SP из справочника»

Для сбора статистики важно заполнить поле «Тип задачи», от значения которого зависит количество SP, тип задачи в свою очередь зависит от значения поля «Ключевой клиент», так как разным ключевым клиентам необходимо производить разные настройки, и от поля «Вид задачи ОП», так как разный тип задач имеет разную сложность. Соответственно правило

запускается при изменении значения поля «Вид задачи ОП» и при изменении значения поля «Ключевой клиент», и в зависимости от их значения заполняется поле «Тип задачи», которое хранит данные для отчета. Код правила приведен в приложении В.

Однако для подсчета SP основным является другое правило (test_directory_2). Оно запускается после того, как проставлено значение в поле «SP учтено» при изменении значения поля «Статус задачи для учета» на «В работе», при изменении значения поля «Ключевой клиент». Это единое правило, действующее во всех проектах, для автоматизации сбора статистики по задачам в SP. Оно проставляет значения в поля «Тип задачи» и «SP из справочника», то есть в зависимости от названия проекта, значения поля «Ключевой клиент» и уже заполненного Типа задачи (для проекта Обратный роуминг) заполняется поле «Тип задачи» и «SP из справочника» данными из справочника test_directory. Справочник – это отдельное правило, в котором прописано сочетание значений полей и количество SP за каждый тип сочетания (это количество SP за каждый тип сочетания является стандартным значением SP за типовую задачу). Соответственно в поле «SP из справочника» подставляются эталонные значения, которые прописаны в справочнике, в стандартных ситуациях значение в поле «SP из справочника» и значение поля «SP учтено» равны, в нестандартных ситуациях значение поля «SP учтено» проставляется вручную. Код правила приведен в приложении Г.

И последнее правило, которое нами было разработано для автоматизированного сбора статистики, проставляет в значение поля «Дата завершения» текущую дату. В отчет должны попасть только те задачи, которые находятся в конечном статусе и завершены в отчетный период конкретным пользователем. Это правило является важным, так как часто происходят ситуации, когда задача переоткрывается сотрудником смежных подразделений для уточнения какой-либо информации или в ситуации, когда у поставщика или ключевого клиента система для обмена сообщениями

перестала работать, или в случае, когда основной сотрудник в отпуске и его замещает коллега, соответственно меняется исполнитель у задачи, но основная работа, выполнение всех настроек уже были выполнены сотрудником в каком-либо прошлом отчетном периоде и задача должна попасть именно тому сотруднику, который выполнил настройки и за тот период, когда они были выполнены.

Данное правило запускается при изменении значения поля «Статус задачи для учета» на «Выполнено». Но необходимо учитывать, что при этом новое значение поля «Статус задачи» не должно быть значением «Трафик активирован». И в поле «Дата завершения» проставляется значение текущей даты. Следует отметить, что дата обновляется каждый раз при переводе задачи в конечный статус «Выполнено». Код представлен на рисунке 15.

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
exports.rule = entities.Issue.onChange({
  // TODO: give the rule a human-readable title
  title: 'Дата завершения',
  guard: (ctx) => {
    //При переводе задачи в любой завершающий статус
    return ctx.issue.fields.becomes(ctx.Status, ctx.Status.Assign) &&
    !ctx.issue.fields.becomes(ctx.StateOP, ctx.StateOP.Traffic);
  },
  action: (ctx) => {
    ctx.issue.fields.AssignDate = Date.now();
  },
  requirements: {
    Status: {
      type: entities.EnumField.fieldType,
      name: "Статус задачи для учета",
      Assign: {
        name: 'Выполнено'
      }
    },
    StateOP: {
      type: entities.EnumField.fieldType,
      name: "Статус задачи",
      Traffic: {
        name: 'Трафик активирован'
      },
    },
    AssignDate: {
      type: entities.Field.dateType,
      name: "Дата завершения"
    }
  }
});
```

Рисунок 15 – Правило для проставления даты завершения

Таким образом, через разработку правил была автоматизирована работа с входящей почтой и сбором статистики по эффективности работы сотрудников отдела подключений.

В отделе подключений есть и исходящий информационный поток, работа с ним стала возможной через разработанное нами браузерное расширение. Расширение для браузера является возможностью расширить базовый функционал веб-приложения, в данном случае системы управления проектами. В отделе подключения возможности расширений для браузера используются для отправки исходящих заявок ключевым клиентам. Заявка на подключение, которая формируется в электронном письме, должна иметь определенную структуру и содержать данные, необходимые ключевому клиенту для подключения. Кроме того, у разных ключевых клиентов, как правило, наиболее крупных, существуют свои требования для оформления заявок и их собственные шаблоны. Также различаются письма в зависимости от типа проделанной работы. После формирования электронного письма его необходимо отправить на определенный адрес электронной почты: у каждого ключевого клиента уникальный адрес, у крупных ключевых клиентов несколько адресов. Очевидно, что формировать электронное письмо вручную довольно трудоемко. Для решения этой проблемы в отделе подключений разработано расширение для браузера, которое автоматизирует отправку писем по шаблонам. Расширение написано на языке JavaScript и постоянно дорабатывается, так как у ключевых клиентов могут меняться адреса электронной почты, добавляться новые проекты, для которых необходимо разработать новый шаблон электронной заявки на подключение. Расширение позволяет формировать и отправлять электронные письма с заявками на подключение. В расширении прописаны шаблоны писем и адреса электронной почты ключевых клиентов, на которые необходимо отправлять заявки. Код расширения представлен в приложении Д. Таким образом, расширение позволяет существенным образом упростить отправку электронных писем для ключевых клиентов.

3.2 Тестирование интеграционного решения для синхронизации данных системы управления проектами и сервиса электронной почты

Для осуществления тестирования применялись только методы ручного тестирования, так как большой нагрузки на систему не оказывается, задачи являются типовыми и достаточно было протестировать отправку одного письма из каждого вида шаблонов для отправки или отправить типовую заявку от ключевого клиента, чтобы убедиться, что система работает корректно [17].

Тестирование интеграции было проведено в тестовой среде системы управления проектами, то есть создан тестовый проект, который никак не связан с продуктивными проектами. В этом проекте создаются тестовые задачи с тестовыми данными, но не в больших количествах, так как задачи являются типовыми.

Далее необходимо в тестовом проекте, который имеет схожую структуру с проектом, интегрированным с электронной почтой, настроить тестовые рабочие процессы, сходные с продуктивными, которые действуют в проекте. После этого в код правила необходимо добавить, что созданное правило должно действовать для конкретной тестовой задачи. Это позволяет проверить, корректно ли работает правило, и быстро отследить ошибки, если они появляются. Фрагмент кода, который позволяет настроить работу правила только для одной задачи, представлен на рисунке 16.

```
//Задать номер своей тестовой задачи  
var myTask = ctx.issue.id === "SSKKSS-6361";  
return myTask && ctx.issue.comments.added.isEmpty();
```

Рисунок 16 – Фрагмент кода для тестирования задачи

Кроме того, есть правило, которое позволяет настроить работу тестового правила только для определенного пользователя, тестирующего задачу, код указанного правила представлен на рисунке 17.

```
//Логин текущего пользователя - это логин тестировщика(вписать себя)
return (ctx.currentUser || {}).login === (ctx.user_0 || {}).login ;
+прописать в requirements
user_0: {
  type: entities.User,
  login: 'логин тестировщика'
}
```

Рисунок 17 – Фрагмент кода для тестирования задачи

Эти правила позволяют работать только с задачей с определенным номером и от определенного пользователя. Сотрудник, тестирующий задачу, может проводить проверки, не нарушая рабочий процесс других сотрудников.

После того, как тестовая среда была настроена, необходимо было протестировать, создается ли задача в системе управления проектами по входящему письму в сервисе электронной почты. Для этого был создан тестовый адрес электронной почты и настроено правило, которое позволяет создавать задачи только от определенного адреса электронной почты. Код правила представлен на рисунке 18.

```
//Подходит для тестирования в SSKK request, прописываем свою почту
return ctx.issue.becomesReported &&
ctx.issue.fields.ProviderAddress.includes("@skbkontur.ru"); //
requirements: {
  ProviderAddress: {
    name: 'e-mail отправителя',
    type: entities.Field.stringType
  }
}
```

Рисунок 18 – Фрагмент кода для тестирования задачи

После того, как тестирование было проведено в тестовой среде и все задачи были созданы корректно, рабочие процессы с настроенными правилами были подключены в продуктивные проекты [16]. Интеграция с почтой показала свою эффективность. Сотрудники стали быстрее обрабатывать задачи, удалось сократить время обработки одной задачи на 20%. Кроме того, удалось сократить влияние человеческого фактора на возникновение ошибок за счет автоматизации рутинных действий: исключен риск того, что задача останется без исполнителя, так как сотрудник забыл заполнить поле, – поле с исполнителем заполняется автоматически, это означает, что 100% задач сотруднику попадут в статистику. Исключен риск того, что сотрудник забудет сменить статус задачи с «В работе» на любой конечный статус, так как действует правило, что задача попадает в статус «Хлам» через 4 или 10 дней и приходит напоминание на почту о такой задаче. Настроенные шаблоны электронных заявок позволяют существенно экономить время на формировании писем и исключают возникновение ошибок, что письмо будет отправлено не на тот электронный адрес. Кроме того, автоматизация заполнения полей задачи и обработка всех задач в одном сервисе позволили существенным образом автоматизировать сбор статистики работы отдела. Система автоматически формирует отчет по сотруднику грейда «Middle» с подсчитанными SP, руководителю отдела необходимо только исправить коэффициенты сложности для сотрудников других грейдов, которых в отделе меньшее количество.

Вывод по третьей главе

В главе описывается реализация разработанного интеграционного решения, указано, что интеграция имеет две основные стороны, во-первых, это интеграция самих сервисов по протоколу электронной почты IMAP и во-вторых, это настройка самой системы управления проектами для того, чтобы задачи из электронных писем были созданы корректно и разработка автоматизации работы с задачами с помощью разработки правил поведения

системы, написанными в виде скриптов на JavaScript. Кроме того, описано тестирование разработанного решения. Отмечено, что было проведено ручное тестирование системы в тестовом проекте с тестовыми данными и применением небольшого количества тестовых скриптов для проверки корректности работы автоматизированных действий.

Таким образом, можно отметить, что опыт применения интеграционного решения в отделе подключений службы сопровождения ключевых клиентов является положительным и, помимо решения основных проблем со сбором статистики и отказа от работы в разных информационных системах, имеет дополнительные положительные эффекты в виде исключения ошибок под влиянием человеческого фактора и увеличения скорости обработки задачи на подключение.

Заключение

В ходе работы было рассмотрено разработанное интеграционное решение по синхронизации данных между сервисом электронной почты и системой управления проектами в отделе подключений службы сопровождения клиентов СКБ Контур. Основная цель работы достигнута, разработанное решение позволяет собирать все задачи на подключение в одной системе и формировать отчет по выполненным задачам на каждого сотрудника. В ходе выполнения работы были описаны бизнес-процессы отдела, выявлены недостатки и предложен способ решения проблемы с помощью разработки интеграции. Сформулированы требования к разрабатываемому решению и составлены диаграммы для его проектирования. На основе требований разработано и описано само интеграционное решение между системой управления проектами и сервисом электронной почты, которое работает на основе написанных правил в системе управления проектами для входящей почты и браузерного расширения для исходящей почты. В работе подробно описана реализация этого решения и сделан вывод о его эффективности.

Таким образом, с помощью доступных технологий была решена задача, поставленная руководителем отдела подключений, чтобы все задачи обрабатывались в одной системе и на основе выполненных задач формировался отчет об эффективности в SP. В работе показан и проанализирован опыт внедрения этого решения, который может быть полезен для дальнейших разработок в данной сфере.

Список используемой литературы и используемых источников

1. Багаутдинов К. Ш. Методы интеграции информационных систем на основе универсального анализатора онлайн-информации // Вестник кибернетики. 2019. №3 (35). С. 52-60.
2. Божинский И. А. Методы и технологии интеграции информационных систем и распределенных баз данных // Радиоэлектроника и информатика. 2015. №2. С. 30-36.
3. Борреманс А. Д., Воронова О. В., Ильин И. В. Совершенствование системы документооборота сетевых компаний FMCG-ритейла на основе проектирования архитектуры данных // Экономика и управление. 2019. №9 (167). С. 87-96.
4. Воронова О. В, Харева В. А. Документооборот в системе управления сетевых торговых компаний // Научный вестник ЮИМ. 2019. №2. С. 5-11.
5. Грудинина Е. А., Мкоян Г. В. Оптимизация электронного документооборота на примере компании «Связьтранзит» // Бизнес-образование в экономике знаний. 2021. №1 (18). С. 32-35.
6. Едалина Д. И., Горячев Н. В., Юрков Н. К. Электронная почта в сети Интернет // Надежность и качество. 2017. С. 364-365.
7. Зыбин А. Я., Каменева М. В. Электронный документооборот в финансовой сфере Российской Федерации // Финансовые исследования. 2021. №2 (71). С. 27-33.
8. Иванова Н. А., Кубанских О. В. Создание корпоративной почтовой системы с использованием системы управления конфигурациями // Прикаспийский журнал: управление и высокие технологии. 2022. №1 (57). С. 17-26.
9. Иванюгин М. А. Интеграция информационных систем // Международный журнал гуманитарных и естественных наук. 2020. №5-1. С. 80-83.

10. Ишутин Д. В., Козырева Н. Е. Проектирование информационной системы управления продажами предприятия // Бизнес и дизайн ревю. 2022. №2 (26). С. 44-54.
11. Каюмова Н. А. Методика проектирования информационных систем // Экономика и социум. 2023. №1-2 (104). С. 627-630.
12. Макеева О. В., Сартаков М. В., Чернов Е. А. Моделирование информационных процессов с помощью UML // Инновации и инвестиции. 2021. №9. С. 121-125.
13. Маркова Т.И. Роль электронной почты в информационном обмене. Анализ рисков, связанных с использованием электронной почты // Вестник ВУиТ. 2010. №16. С. 30-36.
14. Описание рабочих процессов отдела подключений [Электронный ресурс]: регламент по доработкам расширений/правил в отделе подключений. URL: <https://wiki.skbbkontur.ru/pages/viewpage.action?pageId=754473066> (дата обращения: 06.07.2024).
15. Поникаровских А. Е. Система управления проектами как инструмент повышения эффективности компании // Инновационные аспекты развития науки и техники. 2020. №2. С. 105-109.
16. Доработки YT / расширения [Электронный ресурс]: регламент по доработкам расширений / правил в отделе подключений. URL: <https://wiki.skbbkontur.ru/pages/viewpage.action?pageId=710678846> (дата обращения: 06.07.2024).
17. Тестирование правил [Электронный ресурс]: регламент по тестированию в отделе подключений. URL: <https://wiki.skbbkontur.ru/pages/viewpage.action?pageId=796124082> (дата обращения: 06.07.2024).
18. Подключение поставщиков [Электронный ресурс]: регламент подключения поставщиков в отделе подключений. URL:

<https://wiki.skbkontur.ru/pages/viewpage.action?pageId=495279099> (дата обращения 06.07.2024).

19. Савватеев М. Е. Особенности работы почтовых клиентов // Ученые записки Брянского государственного университета. 2020. №2 (18). С. 21-24.

20. Сафонов В. О. Электронная почта (email) – инструмент программиста // КИО. 2008. №1. С.19-28.

21. Семерханов И. А., Муромцев Д. И. Интеграция информационных систем на основе технологии связанных данных // Научно-технический вестник информационных технологий, механики и оптики. 2013. №5 (87). С. 123-127.

22. Турищева Т. Б., Мигачева Е. Н. Роль электронного документооборота для организаций в работе с покупателями и заказчиками // Молодежный сборник научных статей «Научные стремления». 2019. №26. С. 90-93.

23. Шевченко О. В. Анализ современных подходов проектирования информационных систем // Известия ЮФУ. Технические науки. 2013. №1 (138). С. 89-93.

24. Электронный документооборот для ритейла URL: <https://kontur.ru/edi> (дата обращения 06.07.2024).

25. Язханова Х. Д. Особенности проектирования информационных систем // Вестник науки. 2023. №4 (61). С. 210-213.

26. JavaScript Workflow Quick Start Guide [Электронный ресурс]: Documentation. URL: <https://www.jetbrains.com/help/youtrack/devportal/Quick-Start-Guide-Workflows-JS.html> (дата обращения: 06.07.2024).

27. Hub: Identity management service [Электронный ресурс]: Documentation. URL: <https://www.jetbrains.com/ru-ru/hub/> (дата обращения: 06.07.2024).

28. Mailbox Integration [Электронный ресурс]: Documentation. URL: <https://www.jetbrains.com/help/youtrack/cloud/mailbox-integration.html> (дата обращения: 06.07.2024).

29. Xodus [Электронный ресурс]: Transactional schema-less embedded database. URL: <https://github.com/JetBrains/xodus/wiki> (дата обращения: 06.07.2024).

30. Workflow Rule Types [Электронный ресурс]: Documentation. URL: <https://www.jetbrains.com/help/youtrack/devportal/Workflow-Rules.html>

Приложение А

Код правила для парсинга данных

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
const workflow = require('@jetbrains/youtrack-scripting-api/workflow');

exports.rule = entities.Issue.action({

  title: 'Парсинг письма',
  userInput: null,
  command: 'testparsinginn',
  guard: (ctx) => {
    return ctx.issue.isReported;
  },
  action: (ctx) => {
    const issue = ctx.issue;
    let oldstr = issue.description.replace(/<[^>]+>/g, "");
    let str = oldstr.replace(/&quot;/g, "");
    // let str = issue.description; //переменная str - это поле Описание

    // Пробуем найти GLN
    try {const regex = /\b\d{13}\b/; // делаем фильтр цифра длиной в 13 символов в окружении
    символов
      let result = str.match(regex); //применяем фильтр к описанию
      let itogo = result.join(""); //преобразуем result массив в строку
      issue.fields.GLNnumber = itogo;} // выводим itogo в поле GLN
      catch (err) {
        workflow.message("В письме нет GLN");}

    //Пробуем найти ИНН, сначала 10 символов, потом 12
    try {const regex2 = /\b\d{10}\b/; // делаем фильтр цифра длиной в 12 символов
      let result2 = str.match(regex2); //применяем фильтр к описанию
      let itogo2 = result2.join(""); //преобразуем массив result2 в строку
      issue.fields.INNnumber = itogo2; // выводим itogo2 в поле ИНН
    } // если не нашел 12 цифр, ищет 10
      catch (err) {
        try {const regex4 = /\b\d{12}\b/; // делаем фильтр цифра длиной в 10 символов
          let result4 = str.match(regex4); //применяем фильтр к описанию без 13 цифр
          let itogo4 = result4.join(""); //преобразуем массив result2 в строку
          issue.fields.INNnumber = itogo4;
        } // выводим itogo2 в поле ИНН
        catch (err) {
          workflow.message("В письме нет ИНН");}}

    //Пробуем найти КПП
    try {const regex3 = /\b\d{9}\b/; // делаем фильтр цифра длиной в 9 символов
      let result3 = str.match(regex3); //применяем фильтр к описанию
      let itogo3 = result3.join(""); //преобразуем result3 массив в строку
      issue.fields.KPPnumber = itogo3;} // выводим itogo3 в поле КПП
      catch (err) {
```

Продолжение Приложения А

```
workflow.message("В письме нет КПП");}

// Пробуем найти GUID
try { const regex4 = /2[A-ZBM]{2}[\dA-Za-z\-\-]+/;
    let result4 = str.match(regex4);
    let itogo4 = result4.join("");
    issue.fields.GUIDnumber = itogo4;
  } catch (err) {
    workflow.message("В письме возможно нет GUID");
  }
},
requirements: {
  GLNnumber: {
    name: 'GLN',
    type: entities.Field.stringType
  },
  INNnumber: {
    name: 'ИИН',
    type: entities.Field.stringType
  },
  KPPnumber: {
    name: 'КПП',
    type: entities.Field.stringType
  },
  GUIDnumber: {
    name: 'GUID (идентификатор участника ЭДО)',
    type: entities.Field.stringType
  },
}
});
```

Приложение Б

Код правила для автоматического проставления значения поля «SP из справочника»

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');

exports.rule = entities.Issue.onChange({
  title: 'Автоматизация: Заполнение статуса для учета',
  guard: (ctx) => {
    return (ctx.issue.fields.isChanged(ctx.StateOP));
  },
  action: function(ctx) {
    const issue = ctx.issue;
    if (issue.is(ctx.StateOP, ctx.StateOP.Vrabote.name) || issue.is(ctx.StateOP, ctx.StateOP.Pismo.name)) {
      issue.fields.StateNEW = ctx.StateNEW.Vrabote;
    }
    else if (issue.fields.is(ctx.StateOP, ctx.StateOP.Otcryt.name) || issue.is(ctx.StateOP,
ctx.StateOP.Nesozdan.name)) {
      issue.fields.StateNEW = ctx.StateNEW.Newtask;
    }
    else if (issue.is(ctx.StateOP, ctx.StateOP.Hlam.name)) {
      issue.fields.StateNEW = ctx.StateNEW.Hlam;
    }
    else {
      issue.fields.StateNEW = ctx.StateNEW.Done;
    }
  },
  requirements: {

    StateNEW: {
      type: entities.State.fieldType,
      name: "Статус задачи для учета",
      Vrabote: {
        name: 'В работе'
      },
      Hlam: {
        name: 'Хлам'
      },
      Newtask: {
        name: 'Новый'
      },
      Done: {
        name: 'Выполнено'
      },
    },
    StateOP: {
      type: entities.EnumField.fieldType,
      name: "Статус задачи",
      Vrabote: {
        name: 'В работе'
      },
    },
  },
});
```

Продолжение Приложения Б

```
Pismo: {  
  name: 'Письмо отправлено'  
},  
Otcryt: {  
  name: 'Открыт'  
},  
Nesozdan: {  
  name: 'На стадии создания'  
},  
Hlam: {  
  name: 'Хлам'  
},  
}  
});
```

Приложение В

Код правила для заполнения поля «Тип задачи»

```
//при смене Вида задачи ОП проставляется Тип задачи. Если статус задачи "в работе", то есть уже
проставились SP-учтено, то еще плюсом запустится основное правило автоматизации
(test_directory_2).
const entities = require('@jetbrains/youtrack-scripting-api/entities');
const directory_2 = require("sskkrequest/test_directory_2new");

exports.rule = entities.Issue.onChange({

  title: 'Автоматизация: Автопроставление Типа задач ',
  guard: (ctx) => {

    return (ctx.issue.fields.isChanged('Вид задачи ОП')|| ctx.issue.fields.isChanged('Ключевой клиент'));
  },

  action: (ctx) => {
    var SecondLevel = ctx.issue.is('Ключевой клиент', 'ТС для примера 1') || ctx.issue.is('Ключевой
клиент', 'ТС для примера 2'));

    if(ctx.issue.is('Вид задачи ОП', 'Запросы статусов от ТС')){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Подключение/настройка по сетям')&& !SecondLevel){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Подключение по Страхованию')){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Подключение по Сверке')){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Маркировка')){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Запрос для ОСКК')){
      ctx.issue.fields.TypeOP = "Подключения (I уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Подключение к производителю')){
      ctx.issue.fields.TypeOP = "Подключения (II уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Подключение/настройка по сетям')&& SecondLevel){
      ctx.issue.fields.TypeOP = "Подключения (II уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Изменение регламентов/знаний')){
      ctx.issue.fields.TypeOP = "Подключения (II уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Прочие вопросы')){
      ctx.issue.fields.TypeOP = "Подключения (II уровень)"; }

    if(ctx.issue.is('Вид задачи ОП', 'Отключение')){
```

Продолжение Приложения В

```
ctx.issue.fields.TypeOP = "Отключения от КК"; }

if(ctx.issue.is('Вид задачи ОП', 'Вопросы по партинам')){
ctx.issue.fields.TypeOP = "Обработка партий"; }

if(ctx.issue.is('Вид задачи ОП', 'Вопросы по тарификации')){
ctx.issue.fields.TypeOP = "Изменения настроек в EDI у организаций"; }

if(ctx.issue.is('Вид задачи ОП', 'Смена реквизитов')){
ctx.issue.fields.TypeOP = "Изменения настроек в EDI у организаций"; }

if (ctx.issue.fields.is('Статус задачи для учета', 'В работе')) {
    directory_2.rule.action (ctx);}

},

//данные о полях
requirements: {

SpIsdirectory: {
    name: 'SP из справочника',
    type: entities.Field.floatType
},
TypeOP: {
    type: entities.EnumField.fieldType,
    name: "Тип задачи",
},
    ProjectEDI: {
        type: entities.Project,
        name: 'SSKK dorabotki YT'
    },
    ProjectEDI2: {
        type: entities.Project,
        name: 'SSKK smena rekvizitov'
    },
    ProjectEDI30: {
        type: entities.Project,
        name: 'SSKK otkljuchenie'
    },
    ProjectEDI4: {
        type: entities.Project,
        name: 'SSKK produktiv'
    },
    ProjectEDI5: {
        type: entities.Project,
        name: 'SSKK sverka'
    },
    ProjectEDI6: {
        type: entities.Project,
        name: 'SSKK strahovanie'
    },
    ProjectEDI7: {
        type: entities.Project,
```

Продолжение Приложения В

```
name: 'SSKK otchety'
},
ProjectEDI8: {
  type: entities.Project,
  name: 'SSKK faktoring'
},
ProjectEDI9: {
  type: entities.Project,
  name: 'SSKK obratnyj rouming'
},
ProjectEDI10: {
  type: entities.Project,
  name: 'SSKK produktiv obratnyj rouming'
},
ProjectEDI11: {
  type: entities.Project,
  name: 'SSKKretail'
},
// Добавила новый проект SSKK request
ProjectEDI12: {
  type: entities.Project,
  name: 'SSKK request'
},
SP_Directory: {
  type: entities.Field.floatType,
  name: 'SP из справочника',
},
TC_manufacture: {
  type: entities.EnumField.fieldType,
  name: 'Ключевой клиент',
  X5: {
    name: 'TC X5'
  },
},
TypeTask: {
  type: entities.EnumField.fieldType,
  name: 'Тип задачи'
}
});
```

Приложение Г

Код правила для заполнения поля «SP учтено»

```
const entities = require('@jetbrains/youtrack-scripting-api/entities');
const Directory = require("sskk_all_project/test_directory");

exports.rule = entities.Issue.onChange({

  title: 'Заполнение Типа задачи и SP из справочника',
  guard: (ctx) => {
    var issue = ctx.issue;
    return (
      (issue.fields.isChanged(ctx.StateOP) && issue.fields.becomes(ctx.StateOP,
        ctx.StateOP.Vrabote.name)) ||
      (issue.fields.isChanged(ctx.TC_manufacture) && issue.fields.is(ctx.StateOP,
        ctx.StateOP.Vrabote.name))
    );
  },
  action: (ctx) => {

    const issue = ctx.issue;

    console.log(issue.id);

    var ProektEDIIM = "";
    var elementDirectory = "";
    var TC_manufactureName = "";
    var Vid = "";
    var TagSSKK = "";
    // Сюда заносим КК второго уровня
    var SecondLevel = ctx.issue.is('Ключевой клиент', 'ТС для примера 1') || ctx.issue.is('Ключевой
    клиент', 'ТС для примера 2'));

    // Прописываем в логах названия проектов
    //SSKK dorabotki YT Доработки
    if (ctx.issue.project.name == ctx.ProjectEDI.name) {
      ProektEDIIM = ctx.ProjectEDI.name;
      TC_manufactureName = "";
      TagSSKK = "";
      Vid = "";
    }
    //SSKK smena rekvizitov Смена реквизитов
    if (ctx.issue.project.name == ctx.ProjectEDI2.name) {
      ProektEDIIM = ctx.ProjectEDI2.name;
      TC_manufactureName = "";
      TagSSKK = "";
      Vid = "";
    }
    //SSKK otkljuchenie Отключение
    if (ctx.issue.project.name == ctx.ProjectEDI30.name) {
      ProektEDIIM = ctx.ProjectEDI30.name;
      TC_manufactureName = "";
    }
  }
});
```

Продолжение Приложения Г

```
TagSSKK = "";
Vid = "";
}
//SSKK produktiv Продуктив
if (ctx.issue.project.name == ctx.ProjectEDI4.name) {
    ProektEDIIM = ctx.ProjectEDI4.name;
    TC_manufactureName = "";
    TagSSKK = "";
    Vid = "";
}
//SSKK sverka Сверка
if (ctx.issue.project.name == ctx.ProjectEDI5.name) {
    ProektEDIIM = ctx.ProjectEDI5.name;
    TC_manufactureName = "";
    TagSSKK = "";
    Vid = "";
}
//SSKK strahovanie Страхование
if (ctx.issue.project.name == ctx.ProjectEDI6.name) {
    if (ctx.issue.hasTag("Запрос из почты")) {
        ProektEDIIM = ctx.ProjectEDI6.name;
        TagSSKK = "Запрос из почты";
        TC_manufactureName = "";
        Vid = "";
    } else {
        TC_manufactureName = "";
        ProektEDIIM = ctx.ProjectEDI6.name;
        Vid = "";
        TagSSKK = "";
    }
}
//SSKK otchety Отчеты
if (ctx.issue.project.name == ctx.ProjectEDI7.name) {
    ProektEDIIM = ctx.ProjectEDI7.name;
    TC_manufactureName = "";
    Vid = "";
    TagSSKK = "";
}
//SSKK faktoring Факторинг
if (ctx.issue.project.name == ctx.ProjectEDI8.name) {
    if (!issue.fields.is(ctx.TC_manufacture, null) && !issue.fields.is(ctx.TC_manufacture, undefined) &&
        issue.fields.is(ctx.TC_manufacture, ctx.TC_manufacture.X5.name)) {
        TC_manufactureName = issue.fields.TC_manufacture.name;
        ProektEDIIM = ctx.ProjectEDI8.name;
        Vid = "";
        TagSSKK = "";
    } else {
        TC_manufactureName = "";
        ProektEDIIM = ctx.ProjectEDI8.name;
        Vid = "";
        TagSSKK = "";
    }
}
```

Продолжение Приложения Г

```
//SSKK obratnyj rouming Обратный роуминг
if (ctx.issue.project.name == ctx.ProjectEDI9.name) {

if (ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Iniciation.name)) {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "";
    Vid = "Инициация"; //в этой переменной пропишем признак того, что Вид задачи ОР
Инициация
    TagSSKK = "";
} else if (ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Testirovanie.name)) {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "";
    Vid = "Тестирование";
    TagSSKK = "";
} else if (ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Otklyuchenie.name)) {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "";
    Vid = "Отключение"; //в этой переменной пропишем признак того, что Вид задачи ОР
Отключение
    TagSSKK = "";
} else if (!ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Otklyuchenie.name) &&
!ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Iniciation.name) && !ctx.issue.is(ctx.TypeTask,
ctx.TypeTask.Partin.name) && (SecondLevel)) {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "Сложнотень";
    Vid = "";
    TagSSKK = "";
} //
else if (ctx.issue.is(ctx.TypeTask, ctx.TypeTask.Partin.name)) {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "";
    Vid = "Партин";
    TagSSKK = "";
} //в этой переменной пропишем признак того, что Вид задачи ОР Партин
else {
    ProektEDIIM = ctx.ProjectEDI9.name;
    TC_manufactureName = "";
    Vid = "";
    TagSSKK = "";
}
}
//SSKK produktiv obratnyj rouming Продуктив Обратный роуминг
if (ctx.issue.project.name == ctx.ProjectEDI10.name) {
    ProektEDIIM = ctx.ProjectEDI10.name;
    TC_manufactureName = "";
    Vid = "";
    TagSSKK = "";
}
//SSKKretail Прямые подключения
if (ctx.issue.project.name == ctx.ProjectEDI11.name) {
    if (!issue.fields.is(ctx.TC_manufacture, null) && !issue.fields.is(ctx.TC_manufacture, undefined) &&
(SecondLevel)) {
        TC_manufactureName = "Сложнотень";
        ProektEDIIM = ctx.ProjectEDI11.name;
    }
}
```

Продолжение Приложения Г

```
Vid = "";
TagSSKK = "";

} else {
    TC_manufactureName = "";
    ProektEDIIM = ctx.ProjectEDI11.name;
    Vid = "";
    TagSSKK = "";
}
}
//SSKK request Запросы из почты
if (ctx.issue.project.name == ctx.ProjectEDI12.name) {
    if (ctx.issue.is('Тип задачи', 'Подключения (I уровень)')) {
        ProektEDIIM = ctx.ProjectEDI12.name;
        TC_manufactureName = "";
        Vid = "Подключения (I уровень)";
        TagSSKK = "";
    } //в этой переменной пропишем признак того, что Вид задачи ОП может быть любой из
Подключения (I уровень)
    else if (ctx.issue.is('Тип задачи', 'Обработка партий')) {
        ProektEDIIM = ctx.ProjectEDI12.name;
        TC_manufactureName = "";
        TagSSKK = "";
        Vid = "Обработка партий";
    } //в этой переменной пропишем признак того, что Вид задачи ОП может быть любой из
Обработка партий
    else if (ctx.issue.is('Тип задачи', 'Отключения от КК')) {
        ProektEDIIM = ctx.ProjectEDI12.name;
        TC_manufactureName = "";
        TagSSKK = "";
        Vid = "Отключения от КК";
    } //в этой переменной пропишем признак того, что Вид задачи ОП может быть любой из
Отключения от КК
    else if (ctx.issue.is('Тип задачи', 'Подключения (II уровень)')) {
        ProektEDIIM = ctx.ProjectEDI12.name;
        TC_manufactureName = "";
        TagSSKK = "";
        Vid = "Подключения (II уровень)";
    } //в этой переменной пропишем признак того, что Вид задачи ОП может быть любой из
Подключения (II уровень)
    else if (ctx.issue.is('Тип задачи', 'Изменения настроек в EDI у организаций')) {
        ProektEDIIM = ctx.ProjectEDI12.name;
        TC_manufactureName = "";
        TagSSKK = "";
        Vid = "Изменения настроек в EDI у организаций";
    } //в этой переменной пропишем признак того, что Вид задачи ОП может быть любой из
Изменения настроек в EDI у организаций
}
// ctx.issue.project.name === project_0.name
console.log(' TC_manufacture ' + TC_manufactureName + 'ProektEDIIM' + ProektEDIIM + 'Vid' +
Vid + 'TagSSKK' + TagSSKK);

elementDirectory = Directory.SP_directory(TC_manufactureName, ProektEDIIM, Vid, TagSSKK);
console.log(elementDirectory);
```

Продолжение Приложения Г

```
if (elementDirectory !== undefined) {
  issue.fields.TypeTask = elementDirectory.TypeTask;
  console.log('TypeTask = ' + issue.fields.TypeTask.name);
  issue.fields.SP_Directory = elementDirectory.SP_directory;
  console.log('SP_Directory = ' + issue.fields.SP_Directory);
} else {
  issue.fields.TypeTask = undefined;
  issue.fields.SP_Directory = 0;
}
},
requirements: {
  ProjectEDI: {
    type: entities.Project,
    name: 'SSKK dorabotki YT'
  },
  ProjectEDI2: {
    type: entities.Project,
    name: 'SSKK smena rekvizitov'
  },
  ProjectEDI30: {
    type: entities.Project,
    name: 'SSKK otkljuchenie'
  },
  ProjectEDI4: {
    type: entities.Project,
    name: 'SSKK produktiv'
  },
  ProjectEDI5: {
    type: entities.Project,
    name: 'SSKK sverka'
  },
  ProjectEDI6: {
    type: entities.Project,
    name: 'SSKK strahovanie'
  },
  ProjectEDI7: {
    type: entities.Project,
    name: 'SSKK otchety'
  },
  ProjectEDI8: {
    type: entities.Project,
    name: 'SSKK faktoring'
  },
  ProjectEDI9: {
    type: entities.Project,
    name: 'SSKK obratnyj rouming'
  },
  ProjectEDI10: {
    type: entities.Project,
    name: 'SSKK produktiv obratnyj rouming'
  },
  ProjectEDI11: {
    type: entities.Project,
    name: 'SSKKretail'
  }
}
```

Продолжение Приложения Г

```
},
ProjectEDI12: {
  type: entities.Project,
  name: 'SSKK request'
},
SP_Directory: {
  type: entities.Field.floatType,
  name: 'SP из справочника',
},
TC_manufacture: {
  type: entities.EnumField.fieldType,
  name: 'Ключевой клиент',
  X5: {
    name: 'TC X5'
  },
},
StateOP: {
  type: entities.EnumField.fieldType,
  name: "Статус задачи",
  Vrabote: {
    name: 'В работе'
  },
},
TypeTask: {
  type: entities.EnumField.fieldType,
  name: "Тип задачи",
  Initiation: {
    name: 'Инициация/упрощенное тестирование'
  },
  Otklyuchenie: {
    name: 'Отключения от КК'
  },
  Partin: {
    name: 'Обработка партий'
  },
  Podkyuchenie: {
    name: 'Подключения (обратный роуминг)'
  },
  Testirovanie: {
    name: 'Тестирование роуминга'
  }
},
}

});
```

Приложение Д

Код расширения для браузера, автоматизирующего отправку электронных заявок ключевым клиентам

```
// Стандартное письмо EDI
function first(){
    var INN = orgInfo.INN; // ИНН
    var KPP = orgInfo.KPP; // КПП
    var GLN = orgInfo.GLN; // GLN
    var type = orgInfo.type; // Тип подключения/отключения
    var kk = orgInfo.kk; // ключевой клиент
    var maker = orgInfo.maker; // подпись
    var task = orgInfo.idTask; // номер запроса в YT
    var GUID = orgInfo.GUID; // GUID
    var orgname = orgInfo.orgName; // наименование организации
    var contacts = orgInfo.contacts; // Контакты поставщика или дистрибьютора
    var sup = orgInfo.orgName;
    var subj = 'Заявка на подключение ' + kk + ' - ' + orgname // тема письма
    var arrymail = orgInfo.arrymail; // контакты отправки письма
    var ccmail = orgInfo.ccmail; // контакты из копии
    var dopinfo = orgInfo.dopinfo; // дополнительные поля

    var texts =
    '<p>Добрый день</p>' +
    '<p>Прошу настроить и подключить поставщика по ' + '<strong>' + type + '</strong>' + ' к ' + kk
    + '</p>' +
    '<p>Наименование организации: ' + orgname + '<br>' +
    'ИНН: ' + INN + '<br>' +
    'КПП: ' + KPP + '<br>' +
    'GLN: ' + GLN + '<br>' +
    'GUID: ' + GUID + '<br>' +
    dopinfo + '<br>' +
    '<p>Контакты поставщика: ' + contacts + '</p>' +
    '<p>Прошу подтвердить завершение настройки ' + '<strong>' + type + '</strong>' + '</p>' +
    '<p>С уважением,<br>' +
    maker + '<br>' +
    'Отдел подключений <br>' +
    '<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a
    href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

    mailbody = texts;

    let ediservise = "";
    if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
    orgInfo.typeIssue == "Самостоятельное" )
        && orgInfo.versionPlatformYT != "другое")
```

Продолжение Приложения Д

```
&& orgInfo.retailer == "Задача на интеграцию"
&& orgInfo.project == "EDI"
&& orgInfo.ModuleSend == "")
{ediservise = ";edi-in@skbkontur.ru"}
var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + ' (' + task + ') ' + subj + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + '/' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}

//Доподключения EDI-сообщений
function second(){
    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var GLN = orgInfo.GLN;
    var type = orgInfo.type;
    var kk = orgInfo.kk;
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var GUID = orgInfo.GUID;
    var orgname = orgInfo.orgName;
    var contacts = orgInfo.contacts;
    var ediMessage = orgInfo.ediMessage
    var sup = orgInfo.orgName;
    var subj = 'Заявка на доподключение ' + ediMessage + ' - ' + kk;
    var arrymail = orgInfo.arrymail;
    var ccmail = orgInfo.ccmail;

    var texts =
    '<p>Добрый день</p>' +
    '<p>Прошу доподключить поставщику ' + '<strong>' + ediMessage + '</strong>' + ' в ' + kk + '</p>' +
    '<p>Наименование организации: ' + orgname + '<br>' +
```

Продолжение Приложения Д

```
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'GLN: ' + GLN + '<br>' +
'GUID: ' + GUID + '<br>' +

'Контакты поставщика: ' + contacts + '</p>' +
'<p>Ожидаем подтверждения настройки в системе.</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел подключений <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a
href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное " )
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + ' (' + task + ' ) ' + subj + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + ' / ' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}

//Доподключение PRICAT
function third(){
```

Продолжение Приложения Д

```
var INN = orgInfo.INN;
var KPP = orgInfo.KPP;
var GLN = orgInfo.GLN;
var type = orgInfo.type;

var kk = orgInfo.retailer;
var maker = orgInfo.maker;
var task = orgInfo.idTask;
var GUID = orgInfo.GUID;
var orgname = orgInfo.orgName;//Наименование организации
var sup = orgInfo.orgName;
var time = orgInfo.time;//Дата начала работы
var newNomer = orgInfo.newNomer;//Номер и дата договора поставки
var subj = 'Регистрация признака для работы с Pricat КА ' + orgname
var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;

var texts =
'<p>Добрый день</p>' +
'<p>Прошу зарегистрировать признак для работы с Pricat ' + '<strong>' + time + '</strong>' + '</p>' +
'<p>Наименование организации: ' + orgname + '<br>' +
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'Номер и дата договора поставки: ' + newNomer + '<br>' +
'GUID: ' + GUID + '<br>' +
'Оператор ЭДО: Контур <br>' +
'<p>Форма заявления на использование Pricat во вложении.' + '</p>' +
'<p>Ожидаем обратную связь.</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел подключений <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное ")
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
```

Продолжение Приложения Д

```
emlContent += 'Subject: ' + '(' + task + ')' + subj + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';

emlContent += " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + '/' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}
```

```
//АШАН Ритейл Россия подключение
function quarter(){
```

```
    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var GLN = orgInfo.GLN;
    var type = orgInfo.type;
    var kk = orgInfo.kk;
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var GUID = orgInfo.GUID;
    var orgname = orgInfo.orgName;
    var sup = orgInfo.orgName;
    var time = orgInfo.time;
    var nomer = orgInfo.nomer;
    var sap = orgInfo.sap;
    var dopSap = orgInfo.dopSap;
    var subj = 'Заявка EDI '
    var arrymail = orgInfo.arrymail;
    var ccmail = orgInfo.ccmail;

    var texts =
    '<p>Торговая сеть ' + kk + '</p>' +
    '<p>Настройка поставщика</p>' +
    '<p>GLN поставщика: ' + GLN + '</p>' +
    '<p>Код(ы) поставщика: ' + sap + '</p>' +
    '<p>Наименование поставщика: ' + orgname + '</p>' +
```

Продолжение Приложения Д

```
'<p>ИНН поставщика: ' + INN + '</p>' +  
'<p>Провайдер: Контур<br>' +  
'Подпроект: МЕТИ</p>' +  
'<p></p>' +
```

```
'<p>С уважением,<br>' +  
maker + '<br>' +  
'Отдел подключений <br>' +  
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a  
href="www.r-kontur.ru">www.r-kontur.ru</a></p>';
```

```
mailbody = texts;
```

```
let ediservise = "";  
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||  
orgInfo.typeIssue == "Самостоятельное " )  
    && orgInfo.versionPlatformYT != "другое"  
    && orgInfo.retailer == "Задача на интеграцию"  
    && orgInfo.project == "EDI"  
    && orgInfo.ModuleSend == "")  
    {ediservise = ";edi-in@skbkontur.ru"}
```

```
var emlContent = "data:message/rfc822 eml;charset=utf-8,";  
emlContent += 'To: ' + arrymail + '\n';  
emlContent += 'CC: ' + ccmail + '\n';  
emlContent += 'Subject: ' + ' (' + task + ' ) ' + subj + '\n';  
emlContent += 'X-Unsent: 1' + '\n';  
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';  
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';  
emlContent += " " + '\n';  
emlContent += mailbody;
```

```
var encodedUri = encodeURI(emlContent); //encode spaces etc like a url  
var a = document.createElement('a'); //make a link in document  
var linkText = document.createTextNode("fileLink");  
a.appendChild(linkText);  
a.href = encodedUri;  
a.id = 'fileLink';  
a.download = subj + '_' + sup + '/' + INN + '.eml';  
a.style = "display:none;"; //hidden link  
var child = document.body.appendChild(a);  
child.click(); //click the link
```

```
}
```

```
//Нетоварные подключения  
function eighth() {
```

Продолжение Приложения Д

```
var INN = orgInfo.INN;
var KPP = orgInfo.KPP;
var GLN = orgInfo.GLN;
var type = orgInfo.type;
var kk = orgInfo.retailer;

var maker = orgInfo.maker;
var task = orgInfo.idTask;
var GUID = orgInfo.GUID;
var orgname = orgInfo.orgName;
var sup = orgInfo.orgName;
var time = orgInfo.time;
var nomer = orgInfo.nomer;
var newNomer = orgInfo.newNomer; //Номер и дата договора поставки
var sap = orgInfo.sap;
var typeIssue = orgInfo.typeIssue;
var contacts = orgInfo.contacts;
var dopSap = orgInfo.dopSap;
var date = orgInfo.date;
var subj = 'Заявка на подключение к ТС Магнит ';
var emailsend = orgInfo.emailsend;
var gcp = orgInfo.gcp;
if (orgInfo.emailsend == "Noemail") {
    emailsend = "";
}
var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;

var texts =
'<p>Добрый день</p>' +
'<p>Наименование организации: ' + orgname + '<br>' +
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'GLN: ' + GLN + '<br>' +
'Оператор ЭДО: Контур<br>' +
'Идентификатор участника ЭДО: ' + GUID + '<br>' +
'Направление: Исходящий<br>' +
'Очередь: ' + type + '<br>' +
'Вид документа: ' + type + '<br>' +
'Номер и дата договора поставки: ' + newNomer + '<br>' +
'Начало действия/Дата расторжения: ' + time + '</p>' +
'<p>Прошу подтвердить завершение настройки в системе.</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел сопровождения ключевых клиентов <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a href="www.r-kontur.ru">www.r-kontur.ru</a></p>';
```

Продолжение Приложения Д

```
mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное " )

    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + '(' + task + ')' + subj + orgname + ' ' + type + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + ' / ' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link
}

//Доподключение УПД СЧФДОП

function eleventh() {

    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var GLN = orgInfo.GLN;
    var type = orgInfo.type;
    var kk = orgInfo.retailer;
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var GUID = orgInfo.GUID;
    var orgname = orgInfo.orgName;
```

Продолжение Приложения Д

```
var sup = orgInfo.orgName;
var time = orgInfo.time;
var nomer = orgInfo.nomer;
var sap = orgInfo.sap;
var typeIssue = orgInfo.typeIssue;
var contacts = orgInfo.contacts;

var dopSap = orgInfo.dopSap;
var date = orgInfo.date;
var sap = orgInfo.sap;
var post = orgInfo.post;
var subj = 'Настроить поставщика на СЧФДОП ';
var emailsend = orgInfo.emailsend;
var gcp = orgInfo.gcp;
var contactsMetro = orgInfo.contactsMetro; //E-mail для получения PDF-статусных на INVOIC
if (orgInfo.emailsend == "Noemail") {
    emailsend = "";
}
var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;

var texts =
'<p>Добрый день</p>' +
'<p>Прошу настроить поставщика на УПД СЧФДОП.</p>' +
'<p>Наименование организации: ' + orgname + '<br>' +
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'GLN: ' + GLN + '<br>' +
'<p>Номер(а) договора(ов) с Метро Cash&Carry: ' + sap + '</p>' +
'Контакты поставщика: ' + '<strong>' + contactsMetro + '</strong>' + ' в системе.</p>' +
'<p>Ожидаем обратную связь.<br>' +
'Спасибо.</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел сопровождения ключевых клиентов <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное ")
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}
```

Продолжение Приложения Д

```
var emlContent = "data:message/rfc822 eml;charset=utf-8,";
    emlContent += 'To: ' + "mcoe.docflow@metro-cc.ru" + '\n';
    emlContent += 'CC: ' + ccmail + '\n';
    emlContent += 'Subject: ' + ' (' + task + ') ' + subj + '\n';
    emlContent += 'X-Unsent: 1' + '\n';
    emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';

    emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
    emlContent += " " + '\n';
    emlContent += mailbody;

    var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
    var a = document.createElement('a'); //make a link in document
    var linkText = document.createTextNode("fileLink");
    a.appendChild(linkText);
    a.href = encodedUri;
    a.id = 'fileLink';
    a.download = subj + '_' + sup + ' / ' + INN + '.eml';
    a.style = "display:none;"; //hidden link
    var child = document.body.appendChild(a);
    child.click(); //click the link
}
//Оповещение (не ждем ответа от ТС)
function fifteenth() {

    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var GLN = orgInfo.GLN;
    var type = orgInfo.type;
    var kk = orgInfo.kk; // ключевой клиент
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var GUID = orgInfo.GUID;
    var orgname = orgInfo.orgName; //Наименование организации
    var sup = orgInfo.orgName;
    var time = orgInfo.time;
    var nomer = orgInfo.nomer;
    var sap = orgInfo.sap;
    var typeIssue = orgInfo.typeIssue;
    var contacts = orgInfo.contacts;
    var dopSap = orgInfo.dopSap;
    var date = orgInfo.date;
    var kat = orgInfo.kat
    var post = orgInfo.post;
    var subj = 'Уведомление о выполнении настроек по поставщику ';
    var emailsend = orgInfo.emailsend;
    var gcp = orgInfo.gcp;
```

Продолжение Приложения Д

```
var dopinfo = orgInfo.dopinfo; // дополнительные поля
if (orgInfo.emailsend == "Noemail") {
    emailsend = "";
}
var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;

var texts =
'<p>Добрый день</p>' +
'<p>Уведомляем вас, что на нашей стороне поставщик настроен.</p>' +
'<p>Наименование организации: ' + orgname + '<br>' +
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'GLN: ' + GLN + '<br>' +
dopinfo + '<br>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел сопровождения ключевых клиентов <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a' +
href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное " )
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + ' (' + task + ' ) ' + subj + orgname + ' с ' + kk + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
```

Продолжение Приложения Д

```
a.download = subj + '_' + sup + ' / ' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link
}
```

//Шаблон письма для СВЕРКА

```
function fortytwo(){
```

```
    var INN = orgInfo.INN; //ИНН
    var KPP = orgInfo.KPP; //КПП
    var GLN = orgInfo.GLN; //GLN
    var kk = orgInfo.kk; //Ключевой клиент
    var maker = orgInfo.maker; //Assignee
    var task = orgInfo.idTask; //номер задачи
    // var type = orgInfo.type;
    // var GUID = orgInfo.GUID;
    // var kat = orgInfo.kat;
    // var sap = orgInfo.sap;
    // var dopSap = orgInfo.dopSap;
    // var gsp = orgInfo.gsp;
    // var tov = orgInfo.tov;
    var orgname = orgInfo.orgName; //Наименование организации
    var sup = orgInfo.orgName;
    var time = orgInfo.time;
    var nomer = orgInfo.nomer;
    var typeIssue = orgInfo.typeIssue;
    var contacts = orgInfo.contacts; //Контакты поставщика или дистрибьютора
    var conts = orgInfo.conts;
    var date = orgInfo.date;
    var subj = 'Сверка. Уведомление о начале работы поставщика'; //Тема письма
    var arrymail = orgInfo.arrymail; //e-mail КК
    var ccmail = orgInfo.ccmail; //e-mail в копии
    var UR = orgInfo.UR; // Список юридических лиц ключевых клиентов
    var dobro = orgInfo.dobro; //Юр.лица Доброцен
    var karavan = orgInfo.karavan;

    var texts =
    '<p>Добрый день!</p>' +
    '<p>Уведомляем Вас о том, что поставщик готов производить сверку в модуле Взаиморасчеты с ' +
    kk + '</p>' +
```

Продолжение Приложения Д

'<p>Необходимо сопоставить его в настройках модуля Взаиморасчеты и ответить "Готово" на данное письмо.</p>' +

'<p>Наименование организации: ' + orgname + '
' +

'ИНН: ' + INN + '
' +

'КПП: ' + KPP + '
' +

'Контакты ответственного со стороны поставщика: ' + contacts + '
' +

'Список ЮЛ: ' + UR + dobro + '</p>' +

'<p>Ожидаем обратную связь.
' +

'Спасибо.</p>' +

'<p>С уважением,
' +

maker + '
' +

'Отдел подключений
' +

'edi_request@skbkontur.ru | тел. 8 800 500-33-51 | www.r-kontur.ru</p>;

mailbody = texts;

let ediservise = "";

if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" || orgInfo.typeIssue == "Самостоятельное ")

&& orgInfo.versionPlatformYT != "другое"

&& orgInfo.retailer == "Задача на интеграцию"

&& orgInfo.project == "EDI"

&& orgInfo.ModuleSend == "")

{ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml; charset=utf-8,";

emlContent += 'To: ' + arrymail + '\n';

emlContent += 'CC: ' + ccmail + '\n';

emlContent += 'Subject: ' + ' (' + task + ') ' + subj + orgname + ' с ' + kk + '\n';

emlContent += 'X-Unsent: 1' + '\n';

emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';

emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';

emlContent += " " + '\n';

emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url

var a = document.createElement('a'); //make a link in document

var linkText = document.createTextNode("fileLink");

a.appendChild(linkText);

a.href = encodedUri;

a.id = 'fileLink';

a.download = subj + '_' + sup + '/' + INN + '.eml';

a.style = "display:none;"; //hidden link

var child = document.body.appendChild(a);

child.click(); //click the link}

Продолжение Приложения Д

```
//Списки застрахованных
function fiftyone(){
    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var strah_comp = orgInfo.strah_comp;
    var maker = orgInfo.maker;

    var task = orgInfo.idTask;
    var sup = orgInfo.orgName;
    var orgname = orgInfo.orgName;
    var arrymail = orgInfo.arrymail;
    var ccmail = orgInfo.ccmail;
    var subj = 'Настройки клиники ';
    var medi = orgInfo.medi;
    var lpu = orgInfo.lpu;

    var texts =
    '<p>Добрый день.</p>' +
    '<p>Пробьба настроить на обмен списками через К.Страхование следующую ЛПУ:</p>' +
    '<p>Наименование ЛПУ: ' + lpu + '<br>' +
    'ИНН: ' + INN + '<br>' +
    'КПП: ' + KPP + '<br>' +
    'MediID: ' + medi + '</p>' +
    '<p>Ожидаем подтверждения настроек. <br>' +
    'Спасибо.</p>' +
    '<p>С уважением,<br>' +
    maker + '<br>' +
    'Отдел подключений <br>' +
    '<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

    mailbody = texts;

    let ediservise = "";
    if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
    orgInfo.typeIssue == "Самостоятельное " )
        && orgInfo.versionPlatformYT != "другое"
        && orgInfo.retailer == "Задача на интеграцию"
        && orgInfo.project == "EDI"
        && orgInfo.ModuleSend == "")
        {ediservise = ";edi-in@skbkontur.ru"}

    var emlContent = "data:message/rfc822 eml;charset=utf-8,";
    emlContent += 'To: ' + arrymail + '\n';
    emlContent += 'CC: ' + ccmail + '\n';
    emlContent += 'Subject: ' + ' (' + task + ' ) ' + subj + orgname + ' на обмен Список застрахованных
    через К.Страхование ' + '\n';
```

Продолжение Приложения Д

```
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document

var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + ' / ' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}
//Страхование
function fiftytwo(){

    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var strah_comp = orgInfo.strah_comp;
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var sup = orgInfo.orgName;
    var orgname = orgInfo.orgName;
    var arrymail = orgInfo.arrymail;
    var ccmail = orgInfo.ccmail;
    var subj = 'Настройки клиники ';
    var medi = orgInfo.medi;
    var lpu = orgInfo.lpu;

    var texts =
    '<p>Добрый день.</p>' +
    '<p>Просьба настроить на обмен списками через К.Страхование следующую ЛПУ:</p>' +
    '<table border = "1" cellpadding = "4" cellspacing = "0"><tr
align="center"><th>ЛПУ</th><th>Наименование</th><th>ИНН</th><th>КПП</th><th>MediID</th>
<th>"Наименование" с карточки ЛПУ</th><th>Форма доступа</th><th>Код ЛПУ</th></tr><tr
align = "center"><td>' + lpu + '</td><td>' + lpu + '</td><td>' + INN + '</td><td>' + KPP + '</td><td>' +
medi + '</td><td></td><td></td><td></td></tr></table>' +
    '<p>Ожидаем подтверждения настроек. <br>' +
    'Спасибо.</p>' +
    '<p>С уважением,<br>' +
    maker + '<br>' +
    'Отдел подключений <br>' +
```

Продолжение Приложения Д

'edi_request@skbkontur.ru | тел. 8 800 500-33-51 | www.r-kontur.ru</p>';

mailbody = texts;

let ediservise = "";

if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" || orgInfo.typeIssue == "Самостоятельное ")

&& orgInfo.versionPlatformYT != "другое"

&& orgInfo.retailer == "Задача на интеграцию"

&& orgInfo.project == "EDI"

&& orgInfo.ModuleSend == "")

{ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";

emlContent += 'To: ' + arrymail + '\n';

emlContent += 'CC: ' + ccmail + '\n';

emlContent += 'Subject: ' + ' (' + task + ') ' + subj + orgname + ' на обмен Список застрахованных через К.Страхование ' + '\n';

emlContent += 'X-Unsent: 1' + '\n';

emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';

emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';

emlContent += " + '\n';

emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url

var a = document.createElement('a'); //make a link in document

var linkText = document.createTextNode("fileLink");

a.appendChild(linkText);

a.href = encodedUri;

a.id = 'fileLink';

a.download = subj + '_' + sup + '/' + INN + '.eml';

a.style = "display:none;"; //hidden link

var child = document.body.appendChild(a);

child.click(); //click the link

}

//Расширение цепочки

function sixty(){

var INN = orgInfo.INN;

var KPP = orgInfo.KPP;

var strah_comp = orgInfo.strah_comp;

var maker = orgInfo.maker;

var task = orgInfo.idTask;

var sup = orgInfo.orgName;

Продолжение Приложения Д

```
var orgname = orgInfo.orgName;
var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;
var kk = orgInfo.kk;
var subj = "";
var guid = orgInfo.guid;
var factoring = orgInfo.factoring;
var GLN = orgInfo.GLN;
var ediMessage0 = orgInfo.ediMessage;

var ediMessage2 = ediMessage0.replace(/,\s/g, '-');
var provider = orgInfo.provider;
var vids = orgInfo.vids; //виды документов
var kam = orgInfo.kam; //кам
var pm = orgInfo.pm; // пм
var reg = orgInfo.reg; // регламент
var ogr = orgInfo.ogr; // ограничения по продажам
var tarif = orgInfo.tarif; // тариф
var PilotPo = orgInfo.PilotPo; //По сообщению(-ям) или документа (-у)

var texts =

'<p>Коллеги, добрый день!</p>' +
'<p>По ' + '<strong>' + kk + '</strong>' + ' (GLN: ' + GLN + ', ИНН: ' + INN + ', КПП: ' + KPP + ')
завершился пилотный проект по <strong>' + PilotPo + '</strong> и начинается массовое
подключение поставщиков.</p>' +
'<p>' + '<strong>' + 'Цепочка сообщений: ' + '</strong>' + ediMessage2 + vids + '</p>' +
'<p>' + '<strong>' + 'КАМ: ' + '</strong>' + kam + '<br>' +
'<strong>' + 'ПМ: ' + '</strong>' + pm + '</p>' +
'<p>' + '<strong>' + 'Актуальный регламент - ' + '</strong>' + reg + '</p>' +
'<p>' + '<strong>' + 'Ограничения по продажам и подключениям: ' + '</strong>' + ogr + '</p>' +
'<p>' + '<strong>' + 'Тариф: ' + '</strong>' + tarif + '</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел подключений <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a
href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное" )
&& orgInfo.versionPlatformYT != "другое"
&& orgInfo.retailer == "Задача на интеграцию"
&& orgInfo.project == "EDI"
&& orgInfo.ModuleSend == "")
```

Продолжение Приложения Д

```
{ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + ' (' + task + ') ' + kk + ' - расширение цепочки сообщений и начало
массового подключения ' + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';

emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + '/' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}
//Согласование подключения
function seventyseven() {

    var INN = orgInfo.INN;
    var KPP = orgInfo.KPP;
    var GLN = orgInfo.GLN;
    var type = orgInfo.type;
    var kk = orgInfo.retailer;
    var maker = orgInfo.maker;
    var task = orgInfo.idTask;
    var GUID = orgInfo.GUID;
    var orgname = orgInfo.orgName;
    var sup = orgInfo.orgName;
    var time = orgInfo.time;
    var nomer = orgInfo.nomer;
    var sap = orgInfo.sap;
    var typeIssue = orgInfo.typeIssue;
    var contacts = orgInfo.contacts;
    var dopSap = orgInfo.dopSap;
    var date = orgInfo.date;
    var kat = orgInfo.kat
```

Продолжение Приложения Д

```
var post = orgInfo.post;
var dateYPD = orgInfo.dateYPD;
var reg = orgInfo.reg;
var conts = orgInfo.conts;
var ad = orgInfo.ad;
var subj = 'Согласование подключения поставщика: ';
var emailsend = orgInfo.emailsend;
var gcp = orgInfo.gcp;
if (orgInfo.emailsend == "Noemail") {
    emailsend = "";
}

var arrymail = orgInfo.arrymail;
var ccmail = orgInfo.ccmail;
var texts =
'<p>Добрый день</p>' +
'<p>Прошу согласовать подключение: </p>' +
'<p>Наименование организации: ' + orgname + '<br>' +
'ИНН: ' + INN + '<br>' +
'КПП: ' + KPP + '<br>' +
'GLN: ' + GLN + '</p>' +
'<p>Ожидаем обратную связь.<br>' +
'Спасибо.</p>' +
'<p>С уважением,<br>' +
maker + '<br>' +
'Отдел подключений <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a
href="www.r-kontur.ru">www.r-kontur.ru</a></p>';

mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное ")
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}
```

```
var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + arrymail + '\n';
emlContent += 'CC: ' + ccmail + '\n';
emlContent += 'Subject: ' + ' (' + task + ') ' + subj + orgname + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Reply-To:edi_request@skbkontur.ru' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " + '\n';
```

Продолжение Приложения Д

```
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_ ' + sup + ' / ' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);

child.click(); //click the link
}

// Ответ провайдеру
function eightysix(){
    var INN = orgInfo.INN; // ИИН
    var KPP = orgInfo.KPP; // КПП
    var GLN = orgInfo.GLN; // GLN
    var type = orgInfo.type; // Тип подключения/отключения
    var kk = orgInfo.kk; // Ключевой клиент
    var maker = orgInfo.maker; // подпись
    var task = orgInfo.idTask; // номер запроса в YT
    var GUID = orgInfo.GUID; // GUID
    var orgname = orgInfo.orgName; // наименование организации
    var sup = orgInfo.orgName;
    var arrymail = orgInfo.arrymail; // адресаты письма
    var ccmail = orgInfo.ccmail; // адреса из копии писем
    var dopinfo = orgInfo.dopinfo; // дополнительная информация в шаблоны
    var providerpost = orgInfo.providerpost; // e-mail отправителя
    var copyproviderpost = orgInfo.copyproviderpost; // e-mail копии отправителя
    var actualcomment = orgInfo.ActualComment;
    var summary = orgInfo.summary;
    var descriptionsskk = orgInfo.descriptionsskk;
    var kosushka = descriptionsskk.replace(/#/g, "");
    var finaldiscription = kosushka.replace(/&#43/g, "");
    var subj = summary + ' / запрос' //тема письма

    var texts =
    '<p>Добрый день!<br>'+
    '<br>' +

    '<p>С уважением,<br>' +
    maker + '<br>' +
    'Отдел подключений <br>' +
    '<a href="edi_roaming@skbkontur.ru">edi_roaming@skbkontur.ru</a></p>'+
    '<p>'+finaldiscription+'</p>';
```

Продолжение Приложения Д

```
mailbody = texts;

let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное ")
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + providerpost + '\n';
emlContent += 'CC: ' + " " + '\n';
emlContent += 'Subject: ' + subj + ' ' + task + ' ' + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
a.download = subj + '_' + sup + '/' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}
//Пустое письмо (Реквест)
function ninetytwo(){
    var INN = orgInfo.INN; // ИНН
    var KPP = orgInfo.KPP; // КПП
    var GLN = orgInfo.GLN; // GLN
    var type = orgInfo.type; // Тип подключения/отключения
    var kk = orgInfo.kk; // Ключевой клиент
    var maker = orgInfo.maker; // подпись
    var task = orgInfo.idTask; // номер запроса в YT
    var GUID = orgInfo.GUID; // GUID
    var orgname = orgInfo.orgName; // наименование организации
    var sup = orgInfo.orgName;
    var arrymail = orgInfo.arrymail; // адресаты письма
    var ccmail = orgInfo.ccmail; // адреса из копии писем
```

Продолжение Приложения Д

```
var dopinfo = orgInfo.dopinfo; // дополнительная информация в шаблоны
var providerpost = orgInfo.providerpost; // e-mail отправителя
var copyproviderpost = orgInfo.copyproviderpost; // e-mail копии отправителя
var actualcomment = orgInfo.ActualComment;
var summary = orgInfo.summary;
var descriptionsskk = orgInfo.descriptionsskk;
var kosushka = descriptionsskk.replace (/&#43/g, "");
var finaldiscription = kosushka.replace (/#/g, "");
var subj = summary //тема письма

var texts =
'<p>Добрый день.<br>'+

'<br>' +
'<p>Спасибо.<br>'+

'<p>С уважением,<br>' +
maker + ' <br>' +
'Отдел подключений <br>' +
'<a href="edi_request@skbkontur.ru">edi_request@skbkontur.ru</a> | тел. 8 800 500-33-51 | <a href="www.r-kontur.ru">www.r-kontur.ru</a></p>'+
'<p>'+finaldiscription+'</p>';

mailbody = texts;
let ediservise = "";
if ((orgInfo.typeIssue == "Типовое, консультации (API,FTP)" || orgInfo.typeIssue == "Кастомное" ||
orgInfo.typeIssue == "Самостоятельное " )
    && orgInfo.versionPlatformYT != "другое"
    && orgInfo.retailer == "Задача на интеграцию"
    && orgInfo.project == "EDI"
    && orgInfo.ModuleSend == "")
    {ediservise = ";edi-in@skbkontur.ru"}

var emlContent = "data:message/rfc822 eml;charset=utf-8,";
emlContent += 'To: ' + providerpost + '\n';
emlContent += 'CC: ' + " " + '\n';
emlContent += 'Subject: ' + ' (' + task + ' ) ' + subj + '\n';
emlContent += 'X-Unsent: 1' + '\n';
emlContent += 'Content-Type: text/html; charset=UTF-8' + '\n';
emlContent += " " + '\n';
emlContent += mailbody;

var encodedUri = encodeURI(emlContent); //encode spaces etc like a url
var a = document.createElement('a'); //make a link in document
var linkText = document.createTextNode("fileLink");
a.appendChild(linkText);
a.href = encodedUri;
a.id = 'fileLink';
```

Продолжение Приложения Д

```
a.download = subj + '_' + sup + '/' + INN + '.eml';
a.style = "display:none;"; //hidden link
var child = document.body.appendChild(a);
child.click(); //click the link

}

// конец шаблонов

// вывод списка шаблонов в таске для Ритейл
function addLettersLink() {

    // создадим селект и сразу добавим его
    let letterSelect = document.createElement('select');
    letterSelect.id = 'letterSelect';
    letterSelect.style = 'top: 0px; padding-top: 0px;padding-bottom: 0px; color: var(--ring-link-color);
height: 22px; width: 250px;margin-bottom: 1px;';
    letterSelect.className = 'button_7f4 light_dd0';
    document.getElementById('lettersLink').append(letterSelect);

    // создадим кнопку
    let letterSubmit = document.createElement('input');
    letterSubmit.id = 'letterSubmit';
    letterSubmit.type = 'submit';
    letterSubmit.value = '→';
    letterSubmit.style = 'color: var(--ring-link-color);padding-left: 6px;padding-right: 6px;top:
0px;height: 22px; width: 34px;';
    letterSubmit.className = 'button_7f4 light_dd0';
    document.getElementById('lettersLink').append(letterSubmit);

    // создадим кнопку скачивания
    let letterSubmitD = document.createElement('input');
    letterSubmitD.id = 'letterSubmitDown';
    letterSubmitD.type = 'submit';
    letterSubmitD.value = '↓';
    letterSubmitD.style = 'color: var(--ring-link-color);padding-left: 6px;padding-right: 6px;top:
0px;height: 22px; width: 34px;';
    letterSubmitD.className = 'button_7f4 light_dd0';
    document.getElementById('lettersLink').append(letterSubmitD);

    // вызываем функцию send у option, а ее назначим ниже
    letterSubmit.onclick = () => {
        console.log("selectedIndex", letterSelect.selectedIndex);
        letterSelect.options[letterSelect.selectedIndex].send();
    };
    letterSubmitD.onclick = () => {
        console.log("selectedIndex", letterSelect.selectedIndex);
```

Продолжение Приложения Д

```
letterSelect.options[letterSelect.selectedIndex].send(true);
};

// добавляем опции для Ритейла
let newOption;

newOption = new Option('Стандартное письмо EDI', '');
newOption.send = () => {first(); };
letterSelect.append(newOption);

newOption = new Option('Доподключения EDI-сообщений', '');
newOption.send = () => {second(); };
letterSelect.append(newOption);

newOption = new Option('Доподключение PRICAT', '');
newOption.send = () => {third(); };
letterSelect.append(newOption);

newOption = new Option('Атак/Ашан подключение', '');
newOption.send = () => {quarter(); };
letterSelect.append(newOption);

newOption = new Option('Метро доподключение УПД СЧФДОП', '');
newOption.send = () => {eleventh(); };
letterSelect.append(newOption);

newOption = new Option('Оповещение (не ждем ответа от ТС)', '');
newOption.send = () => {fifteenth(); };
letterSelect.append(newOption);

newOption = new Option('X5 Вторсырье', '');
newOption.send = () => {seventyeight(); };
letterSelect.append(newOption);

newOption = new Option('Пустое письмо', '');
newOption.send = () => {ninetytwo(); };
letterSelect.append(newOption);

// вывод списка шаблонов в таске для реквеста
function addLettersLinkforRequest() {

    // создадим селект и сразу добавим его
```

Продолжение Приложения Д

```
let letterSelect    = document.createElement('select');
letterSelect.id     = 'letterSelect';
letterSelect.style  = 'top: 0px; padding-top: 0px;padding-bottom: 0px; color: var(--ring-link-color);
height: 22px; width: 250px;margin-bottom: 1px;';
letterSelect.className = 'button_7f4 light_dd0';
document.getElementById('lettersLink').append(letterSelect);

// создадим кнопку
let letterSubmit    = document.createElement('input');
letterSubmit.id     = 'letterSubmit';
letterSubmit.type    = 'submit';
letterSubmit.value   = '→';

letterSubmit.style  = 'color: var(--ring-link-color);padding-left: 6px;padding-right: 6px;top:
0px;height: 22px; width: 34px;';
letterSubmit.className = 'button_7f4 light_dd0';
document.getElementById('lettersLink').append(letterSubmit);

// создадим кнопку скачивания
let letterSubmitD    = document.createElement('input');
letterSubmitD.id     = 'letterSubmitDown';
letterSubmitD.type    = 'submit';
letterSubmitD.value   = '↓';
letterSubmitD.style  = 'color: var(--ring-link-color);padding-left: 6px;padding-right: 6px;top:
0px;height: 22px; width: 34px;';
letterSubmitD.className = 'button_7f4 light_dd0';
document.getElementById('lettersLink').append(letterSubmitD);

// вызываем функцию send у option, а ее назначим ниже
letterSubmit.onclick = () => {
    console.log("selectedIndex", letterSelect.selectedIndex);
    letterSelect.options[letterSelect.selectedIndex].send();
};

letterSubmitD.onclick = () => {
    console.log("selectedIndex", letterSelect.selectedIndex);
    letterSelect.options[letterSelect.selectedIndex].send(true);
};

// добавляем опции для реквеста
let newOption

newOption = new Option('Пустое письмо', '');
newOption.send = () => {ninetytwo(); };
letterSelect.append(newOption);

}}
```