

СПОСОБЫ ИНТЕГРАЦИИ И ОБМЕНА ДАННЫМИ СТОРОННИХ ПРИЛОЖЕНИЙ С СИСТЕМОЙ 1С

*А.В. Литасов, студент группы 17В60, научный руководитель: Захарова А.А. д.т.н.
Юргинский технологический институт (филиал) Национального исследовательского
Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
E-mail: litasov19@bk.ru*

Аннотация: в статье приводятся и анализируются возможные способы интеграции и обмена данными сторонних приложений с системой 1С. Приводятся различные подходы к реализации интеграции с системой 1С в зависимости от поставленной задачи. А также разбор наиболее часто применяемых интеграций, реализуемых на основе общепризнанных открытых стандартов и протоколов передачи данных.

Ключевые слова: интеграция, информационная база, конвертация данных, обмен данными, система 1С, формат XML, web-сервис.

Регулярный обмен информацией между 1С и другими внешними системами на разных платформах является значимым звеном в проектировании системы, где необходима своевременная точность, актуальность и возможность вносить желаемые правки по работе с представляемой системой 1С информацией.

Решение интеграционных задач может быть различным. В некоторых случаях достаточно простого интерактивного обмена данными – например, для передачи в банк списка сотрудников для оформления зарплатных пластиковых карт. Для более сложных задач может быть необходим полностью автоматизированный обмен данными, возможно, с обращением к бизнес-логике внешней системы. Есть задачи, носящие специализированный характер, вроде интеграции с внешним оборудованием (например, торговым оборудованием, мобильными сканерами и т.д.) или с унаследованными или узкоспециализированными системами (например, с системами распознавания RFID-меток). Очень важно подобрать для каждой задачи свой механизм интеграции.

Одними из наиболее важных требований, предъявляемые к интегрируемым приложениям, являются следующие:

- Легкость изменения/адаптации логики работы приложения к меняющимся бизнес-задачам.
- Легкость интеграции с другими приложениями.

В зависимости от требований поставленной задачи, существуют различные подходы к реализации интеграции с приложениями 1С.

1. Основой для реализации служат механизмы интеграции, которые предоставляет платформа, свой специализированный API приложения 1С. Достоинства данного подхода – API устойчив к изменению реализации на стороне приложения 1С. Особенностью подхода является требование изменения исходного кода типового решения 1С, что может потребовать усилий при слиянии исходных кодов при переходе на новую версию конфигурации. В таком случае на помощь приходит новая прогрессивная функциональность – расширения конфигурации. Расширения – это, по определению, механизм плагинов, который позволяет создавать дополнения прикладных решений, который не меняет самих прикладных решений. Вынос интеграционного API в расширение конфигурации позволяет избежать трудностей при слиянии конфигураций при переходе на новую версию типового решения.
2. На основе механизмов интеграции платформы, предоставляющих доступ снаружи к объектной модели приложения и не требуют доработки приложения или создания расширения. Плюс такого подхода – не нужно менять приложение 1С. Минус – если приложение 1С было доработано, то могут потребоваться доработки в интегрируемом приложении. Пример такого подхода – использование для интеграции протокола OData, реализованного на стороне платформы 1С: Предприятие.
3. Использование типовых решений 1С и партнеров, реализованных на основе механизмов интеграции, свои собственные прикладные протоколы, ориентированные на конкретные задачи. При использовании таких решений не требуется написание кода в конфигурациях 1С, т.к. мы пользуемся штатными возможностями прикладного решения. На стороне приложения 1С нам нужно лишь выполнить определенные настройки.

Структура дерева метаданных «1С:Предприятие» предоставляет достаточно много объектов, предназначенных для решения задач интеграции, далее приведены одни из часто применяющихся:

- Планы обмена – хранит список узлов, с которыми осуществляется обмен;
- Web-сервисы – предоставляют возможность обращения к текущей базе 1С с помощью Web-сервисов;

- HTTP-сервисы – предоставляют возможность обращения к текущей базе 1С с помощью HTTP-сервисов;
- XDTO-пакеты – объект для описания структуры данных; широко используется при работе с Web-сервисами;
- Внешние источники данных – предоставляет доступ к данным внешних баз, построенным не на 1С.

Итак, приведем яркий пример, стоит задача двунаправленного обмена данными между приложением 1С и произвольным приложением. Например, нам нужно синхронизировать список товаров (справочник Номенклатура) между приложением 1С и произвольным приложением. Для решения такой задачи можно написать расширение, которое выгружает справочник Номенклатура в файл определенного формата (текстовый, XML, JSON, ...) и умеет считывать этот формат.

В платформе реализован механизм сериализации прикладных объектов в XML как напрямую, через методы глобального контекста ЗаписатьXML/ЧтениеXML, так и с помощью вспомогательного объекта XDTO (XML Data Transfer Objects). Любой объект в системе 1С: Предприятие может быть сериализован в XML представление и наоборот.

Далее экспортируем справочник в JSON. Товары будут записаны в массив. Остается только передать данные конечному потребителю. Платформа 1С: Предприятие поддерживает основные интернет-протоколы HTTP, FTP, POP3, SMTP, IMAP, включая их безопасные версии.

Также для передачи данных можно использовать HTTP и/или Web-сервисы. Приложения 1С могут реализовывать свои HTTP- и веб-сервисы, а также вызывать HTTP- и веб-сервисы, реализованные сторонними приложениями.

Функция автоматического формирования REST-интерфейса для всего прикладного решения была включена в версию 8.3.5 платформы 1С:Предприятие. Любой бъект конфигурации можно сделать доступным для получения данных и их модификации через REST-интерфейс. Протоколом доступа на платформе является протокол OData. Для публикации сервиса OData необходимо из меню Конфигуратора «Администрирование -> Публикация на веб-сервере», поставить флажок «Публиковать стандартный интерфейс OData». Поддерживаются форматы atomXML и JSON.

После публикации прикладного решения на веб-сервере, сторонние приложения обращаются к нему через REST-интерфейс с помощью HTTP запросов. Для работы с приложением 1С через протокол OData программирование на стороне 1С не требуется.

В некоторых случаях обмен данными через внешние источники данных может оказаться оптимальным решением. Внешние источники данных – это прикладной объект конфигурации 1С, позволяющий взаимодействовать с любой ODBC-совместимой базой данных как на чтение, так и на запись. Внешние источники данных доступны как в Windows, так и на других ОС.

Механизм обмена данными на данный момент позволяет создавать распределенные территориально системы на 1С:Предприятие, и также организуют обмен данными с другими ИС.

Одно из ключевых понятий в механизме обмена данными – это план обмена. План обмена – это особый тип объекта прикладного платформы 1С, определяющий, в частности, состав данных, которые будут участвовать в обмене (какие именно справочники, документы, регистры и т.п.). План обмена содержит также информацию об участниках обмена (так называемых узлах обмена).

Обмен данными происходит при помощи XML сообщений определенной структуры. Сообщение содержит данные, изменившиеся со времени последней синхронизации с узлом, и некоторую служебную информацию. В структуре сообщений имеется поддержка нумерации сообщений и возможность получения подтверждения от узла-получателя о приеме сообщений. Подтверждение, приходящее от узла-получателя, содержится в каждом сообщении, в виде номера последнего принятого сообщения. Нумерация сообщений позволяет понять платформе, какие данные уже были успешно переданы на узел-получатель, и избежать повторной передачи, передавая лишь данные, измененные со времени приема узлом-отправителем последнего сообщения с квитанцией о полученных узлом-получателем данных. При такой схеме работы обеспечивается гарантированная доставка даже при ненадежных каналах передачи и потере сообщений.

В заключении, можно с уверенностью сказать, что разработчики платформы уделили пристальное внимание функционалу обмена информацией со сторонними приложениями и предоставили разработчикам 1С широкий спектр прикладных решений интегрирования «1С:Предприятие» с другими информационными системами. Обмен между различными базами и конфигурациями 1С – достаточно понятная и относительно не сложная задача. Важно лишь то, чтобы в программе

источнике были все необходимые данные (а также – в необходимом формате) для передачи, а в программе-приемнике – было, куда эти данные загружать.

Список литературы:

1. Способы интеграции с 1С. URL: <https://habr.com/ru/company/1c/blog/308420/> (дата обращения: 16.01.2019).
2. Стандарты и форматы. URL: http://v8.1c.ru/edi/edi_stnd/index.htm (дата обращения: 16.01.2019).
3. Серия уроков по созданию простейшего приложения на платформе Android для взаимодействия с 1С. URL: <https://infostart.ru/public/463225/> (дата обращения: 16.01.2019).
4. Создание и управление интернет магазином на Android в режиме "Online". URL: <https://infostart.ru/public/154962/> (дата обращения: 16.01.2019).
5. Что такое нативное приложение? URL: <https://russia.ibuildapp.com/что-такое-нативное-приложение/> (дата обращения: 16.01.2019).
6. Способы обмена 1С с другими системами. URL: <https://pro1c8.ru/sposobi-integracii-1c/> (дата обращения: 16.01.2019).
7. Радченко М. Г. (фирма "1С"), Хрусталева Е. Ю. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы // Издательство ООО "1С-Паблишинг", ISBN 978-5-9677-2041-3, 965 стр.

ТРЕБОВАНИЯ К СОВРЕМЕННЫМ ИНФОРМАЦИОННЫМ СИСТЕМАМ УПРАВЛЕНИЯ В СФЕРЕ РАБОТЫ С ПЕРСОНАЛОМ

А.Р. Горбачев, студент гр.17В81, научный руководитель Захарова А.А., д.т.н.

Юргинский технологический институт (филиал) Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

E-mail: amir.gorbachyow@yandex.ru

Аннотация. Информационные технологии развиваются семимильными шагами, и сейчас мы можем наблюдать их повсеместное применение. Сфера работы с персоналом (поиск сотрудников, управление кадрами и т. д.) также подверглась влиянию современных программных средств. В этой статье попробуем выяснить, каким требованиям должны соответствовать современные системы управления для эффективного решения проблем в области работы с персоналом.

На данный момент большинство программных продуктов на рынке – это комплексные решения, отвечающие сразу на несколько направлений в управлении персоналом. Это обусловлено тем, что при наличии нескольких мелких сервисов появляется необходимость объединять их в одну систему, что потребует большого количества ресурсов на перенос информации между этими сервисами. Готовые системы позволяют экономить время и человеческие ресурсы, являясь едиными базами данных, помогают избежать ошибок в документах, исключая ручной перевод информации.

Из этого мы можем выделить следующий важный пункт – доступность информации. Все данные хранятся в одной базе данных, что позволяет получать оперативный доступ ко всем документам без необходимости обращаться к нескольким системам. Такие условия позволяют автоматизировать огромное количество кадровых процедур. Руководитель имеет возможность получать полную информацию о конкретном работнике в сжатые сроки (например, информация о заработной плате, графике работы, квалификации и т. д.). Одна из важнейших проблем для компании – удержание высококвалифицированных сотрудников. Быстрое получение полной информации о каждом работнике позволяет отслеживать обучение, отдых, мотивацию важных сотрудников и во время поощрять необходимыми бонусами. Причём доступность информации помогает и сотрудникам. Например, в отслеживании информации о критериях для повышения в должности, что позволит строить карьерный план внутри компании, повышая мотивацию и делая систему более прозрачной.

Опять же если говорить об автоматизации, то современные средства позволяют исключить человека из алгоритма вычисления заработной платы, планирования отпусков, отслеживания состояний проектов и затраченного на них времени. При этом исключение человека даёт возможность не просто упростить, но и исключить ошибки в расчётах, допускаемые людьми. Задавая системе стро-