

Документация по проекту ElOrder

Ревизия 0.1.3 от 09.10.2009
Пономаренко Н.В. (pnv82g@gmail.com)

Краткое описание проекта

ElOrder - проект задавшийся целью внедрить на фармацевтическом рынке Украины общие принципы, методы и стандарты безбумажного документооборота между поставщиками лек. средств и их клиентами.

На первом этапе реализуется и стандартизируется возможность выполнения в онлайн режиме следующих операций:

- получение адресных специализированных прайсов
- отправка заказа
- получение дефектуры(отказа) поставщика

Одним из важнейших преимуществ и новаций этого этапа является работа в онлайн режиме из одного программного продукта с различными поставщиками, ускоряющая процесс формирования заказа.

Для реализации этого этапа поставщику необходимо реализовать [web-сервис](#) работающий по протоколу [SOAP](#) 1.1, который будет выполнять перечисленные операции путем отсылки, приема и обработки xml-файлов заданного формата.

В данном документе приводится техническое описание типового решения и рекомендации по его реализации.

Описание типового веб-сервиса

WSDL схемы вариантов реализаций веб-сервиса проекта EOrder можно получить по адресам:

- <http://elorder.optima.kiev.ua/EOrderService.svc?wsdl>

Обобщенная схема работы

При начале работы с сервисом клиент инициализирует сессию(авторизуется) на сервере посредством вызова метода [Init](#), передавая ему информацию идентифицирующую клиента - код ОКПО и уникальный идентификатор компьютера.

Отдельно нужно отметить, что в данной схеме клиенту не требуется какой-либо специальной регистрации, получения логинов-паролей у поставщика - по умолчанию клиенту доступно только получение общего(публичного) прайс листа. Полный функционал становится доступным когда менеджеры поставщика активируют данного клиента, убедившись в его аутентичности.

Сервер проверяет "активированность" данного ОКПО и компьютера и возвращает идентификатор сессии.

В качестве результата авторизации, помимо сессии, возвращается еще перечень текстовых сообщения пользователю, в которых можно сообщить о статусе ОКПО, причине отказа в авторизации, контактных данных менеджера данного клиента и т.п. Возвращаются они в виде xml-документа [заданного формата](#).

Сразу после авторизации клиент автоматически отправляет на сервер ([UpdateDeliveryAddressesList](#)) список своих новых и изменившихся точек доставки, отсылая пары <уникальный идентификатор> - <полный почтовый адрес>. Эта информация используется поставщиком для более оперативного сопоставления точек доставки имеющихся у него в базе с теми, которые клиент будет указывать при создании заказов.

Этап сопоставления необходим, т.к. гарантировать то, что клиент верно и однозначно укажет адрес нельзя, а полная и проверенная информация о точках доставки уже есть в базе поставщика.

Однозначность сопоставления гарантируется тем, что идентификаторы уникальны (используется [GUID](#)) и генерируются заново при любом изменении адреса пользователем.

Далее клиент продолжает работу, запрашивая прайс-лист, путем вызова метода [GetPriceList](#). Обычно прайс-лист генерируется сразу после запроса пользователя, что позволяет предоставлять клиентам наиболее свежую информацию, но может требовать значительных выч. затрат с пиками в начале дня. В прайс лист, помимо перечня лек. средств, цен и т.п. информации входит секция с перечнем типов цен, специфичных для данного поставщика, и их кодами.

После получения прайса клиент подготавливает заказ, отдельный для каждой точки доставки, и отправляет его на сервер посредством вызова метода [MakeOrder](#). В заказе указывается точка доставки с ее уникальным идентификатором. Для идентификации типа цены используются коды поставщика.

После получения заказа, поставщик может сгенерировать сообщение, информирующее пользователя о том, что его заказ успешно получен и поступил в обработку.

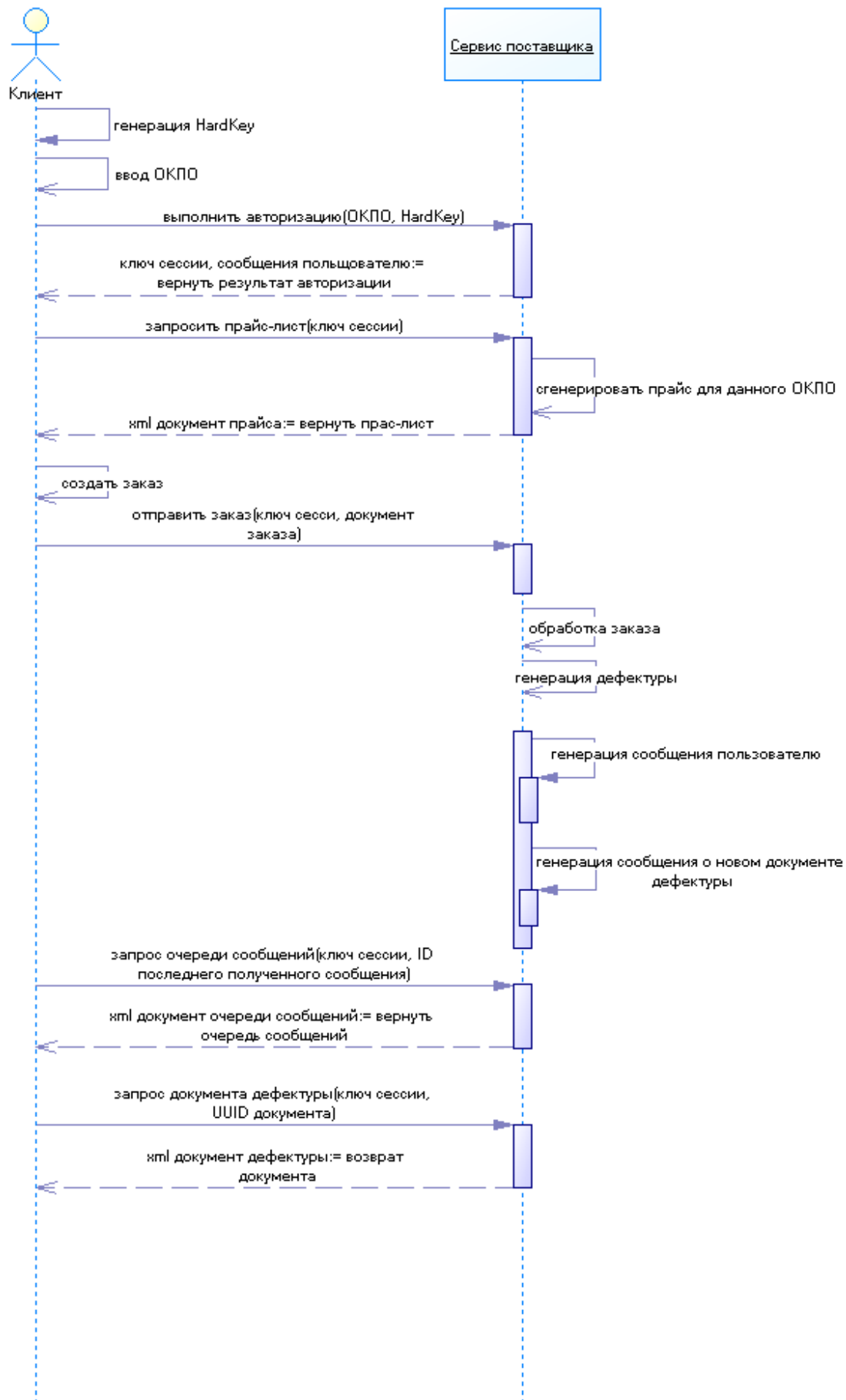
После обработки(автоматической или ручной) заказа сервер может сгенерировать документ дефектуры с перечнем товаров, которые он поставить не может и причиной отказа. Сообщение о наличии данного документа отправляет в очередь клиента и будет им получено при ее обработке.

Опрос очереди сообщений выполняется посредством регулярных вызовов клиентом метода [GetNewNotifications](#) в течении активности сессии.

При обработке очереди сообщений могут выполняться следующие операции:

- показ простого информационного сообщения пользователю
- изменение статуса заказа с показом соотв. сообщения пользователю
- получение нового документа дефектуры с предложением вернуть его в потребность

Документ дефектуры получается путем отдельного вызова метода [GetDocumentByUID](#) с использованием идентификатора из соотв. сообщения.



Авторизация

Для тестирования и отладки на период разработки рекомендуется выделить несколько ОКПО вне реального диапазона:

- 000000000099999001 - эмуляция активированного ОКПО, полный доступ, до 20 рабочих мест
- 000000000099999002 - эмуляция неактивированного ОКПО, ограниченный доступ, до 20 рабочих мест
- 000000000099999003 - заблокированный ОКПО

Использование этих ОКПО позволит исключить попадание тестовых заказов в обработку, резервирование ими остатков и т.п. нежелательное воздействие на внутренние процессы поставщика.

По умолчанию, клиент считает, что время жизни сессии на сервере не превышает 30 мин, и после истечения этого срока отключается, переставая опрашивать очередь сообщений.

Механизм автоматического продления сессии может быть предусмотрен во второй версии протокола.

Очередь сообщений

Формируется на стороне сервиса, для оповещения клиента о происходящих событиях - обработке заказа, появлении нового документа и т.п. Может использоваться для отправки произвольных текстовых сообщений пользователю.

Доставка очереди клиенту осуществляется путем периодического вызова им метода [GetNewNotification](#). Для того, чтобы клиент получал только новые сообщения, в запросе он передает идентификатор последнего обработанного им сообщения, а сервер возвращает лишь сообщения с ID больше указанного.

Поэтому в качестве идентификатора сообщений необходимо использовать возрастающую последовательность 64битных целых - к примеру первичный ключ таблицы сообщений.

Возврат сообщения для ОКПО или для хардкдея?

Типы данных

- UUID - строка длиной 36 символов без концевых фигурных скобок. Пример - 3605A06E-D4E2-4C14-B814-392540E3CE17. [См. UUID](#)
- DataContainer - объект-контейнер для данных со следующими атрибутами:
 - IsCompressed - boolean
Булевый признак описывающий, сжаты ли данные. Алгоритм сжатия - zip.
 - Content - byte array
Массив бинарных данных содержащий соотв. документ в сжатом или распакованном виде. В типах WSDL - Base64Binary.

Описание методов

Init

Функция логина/подключения/авторизации клиента к сервису. Выполняется в начале работы. Иницирует сессию работы.

Параметры

- okpo - string.
Код ОКПО организации. При незаданном ОКПО отдается прайс на общих основаниях
- hardKey - string(32).
Уникальный идентификатор компьютера, генерируется на основе кода установки Windows и CPU ID. Используется для ограничение кол-ва клиентских мест для одного ОКПО.
- clientVersion - string.
Версия клиентского приложения.

Результат

Возвращает объект с атрибутами:

- SessionTicket - UUID.
Идентификатор сессии в пределах которой выполняется вся работа с сервером. При возврате пустой строки (или строки 00000000-0000-0000-0000-000000000000 - особенность одной из реализаций) считается что авторизация не выполнена.
- Content - string.
Дополнительные данные. Представляет собой xml-документ с текстовыми сообщениями пользователю фиксированного формата.
В дальнейшем возможно расширение этого формата для передачи доп. информации: времени жизни сессии, поддерживаемых сервером версий протокола и т.п., без изменения программного интерфейса самого сервиса.

GetPriceList

Функция для получения специализированного прайс-листа. Ввиду большого объема прайс-листа данные обычно передаются в упакованном виде.

Параметры

- sessionTicket - см. [SessionTicket](#)
Идентификатор сессии полученный на этапе авторизации.
- protocol - Integer
Версия протокола, которую поддерживает данный клиент. Пока существует лишь версия 1.

Результат

Возвращает [DataContainer](#) со сжатым xml-файлом прайс-листа.

GetNewNotifications

Функция для получения списка сообщений для данного ОКПО(не решено - только для ОКПО или еще и для HardKey) из очереди сервера. Вызывается клиентом с некоторой регулярностью (обычно раз в 2 минуты).

Параметры

- sessionTicket - см. [SessionTicket](#)
- protocol - см. [protocol](#)
- lastNotificationID - Int64
Идентификатор последнего полученного сообщения от этого клиента. Позволяет ограничить список получаемых в запросе сообщений только теми, которые еще не были обработаны данным клиентом.

Результат

Возвращает [DataContainer](#) с xml-документом со списком разнотипных сообщений.
Если очередь сообщений пуста, то возвращается xml-документ с одним корневым элементом - <EIOrder/>.

GetDocumentByUID

Функция для получения документа произвольного типа по его идентификатору. На текущий момент используется лишь для получения дефектуры.

Параметры

- sessionTicket - см. [SessionTicket](#)
- protocol - см. [protocol](#)
- documentUID - [UUID](#)

Уникальный идентификатор документа о существовании которого на сервере известно. Обычно этот идентификатор берется из сообщений о новом документе.

Результат

Возвращает [DataContainer](#) с соответствующим документом.

UpdateDeliveryAddressesList

Функция для отсылки списка точек доставки на сервер. Играет дублирующую роль, т.к. сервер должен принять заказ даже для точки, которая не была отправлена в этом методе.

Параметры

- sessionTicket - см. [SessionTicket](#)
- protocol - см. [protocol](#)
- addresses - [DataContainer](#)

Список адресов точек доставки в виде xml-файла заданного формата.

Результат

Boolean

MakeOrder

Функция для отправки документа заказа на сервер.

Параметры

- sessionTicket - см. [SessionTicket](#)
 - protocol - см. [protocol](#)
 - order - DataContainer
- Документ заказа в виде xml-файла.

Результат

Boolean

Форматы файлов

Кодировка - UTF-8 или win1251, но с обязательным ее указанием в заголовке xml.

Идентификаторы документов - типа [UUID](#).

Формат дат и дробных фиксирован и описан в файле elorder_types.xsd в виде регулярных выражений. Примеры:

DateTime - 10.01.2009 14:15:[12]

Date - 10.01.2009

Float - 145.45

Результат авторизации

Пример - elorder_content.xml

Схема - elorder_content.xsd

Документ, передаваемый в ответе вместе с идентификатором сессии при вызове функции [Init](#). На текущем этапе содержит перечень информационных сообщений пользователю, которые могут разъяснять причины отказа в авторизации, необходимость активации ОКПО и т.п. В дальнейшем может расширяться доп. информацией о технических параметрах сервиса.

Список сообщений

Пример - elorder_notify.xml

Схема - elorder_notify.xsd

Содержит очередь разнотипных сообщений для клиента. Список типов сообщений:

- TextNews
Текстовое сообщение, которое показывается пользователю.
- ChangesInOrderStatus
Сообщение с изменением статуса документа заказа и текстовым сообщением пользователю.
- DeficiencyInOrder
Сообщение о том, что для документа заказа, отсланного ранее, сформирован документ дефектуры.

Документа прайса

Пример - elorder_price.xml

Схема - elorder_price.xsd

На текущий момент содержит минимум необходимой информации, для уменьшения его размера. Включает в себя список типов цен с их кодами, названиями и длительностью отсрочки.

Как расширение от компании Оптима может идти перечень акций (на данный момент не используется).

Документа заказа

Пример - elorder_order.xml

Схема - elorder_order.xsd

Для каждой точки доставки создается отдельный.

В одном заказе могут быть позиции с разными типами цен. Для идентификации типов цен по

возможности используются коды поставщика(приходят с прайсом).

Документ дефектуры(отказа) поставщика

Пример - elorder_deficiency.xml

Схема - elorder_deficiency.xsd

Документ содержащий перечень позиций из конкретного заказа которые не могут быть поставлены поставщиком.

Документ накладной

Пример - elorder_consignment.xml

Схема - elorder_consignment.xsd

Формат документа накладной находится в процессе согласования и окончательно не утвержден. За основу, при его разработке, брался формат mmo.

Часто задаваемые вопросы

Q:

Информация о ценах из спец-прайсов может попасть к конкурентам.
Как обеспечена конфиденциальность данных?

A:

Q: Какова выгода компании Морион в этом проекте?

A:

Q: Какова выгода поставщика при присоединении к этому проекту?

A:

Q: Зачем поддерживать этот стандарт, если у нас уже есть подобный механизм?

A:

Q: Сколько, чьи сервера задействованы в системе. Кто выполняет их обслуживание, режим обслуживания. Как обеспечивается отказоустойчивость?

A:

Q: Какие продукты (БД, веб сервисы) используются сервером приложений?

A: