

Отправка уведомлений на клиент при наступлении событий на сервере

В этом разделе мы рассмотрим случай, когда при наступлении каких-то событий на сервере сервер через служебное обсуждение информирует об этом клиентское приложение. И оно, в свою очередь, реагирует на это сообщение.

Например, в тот момент, когда менеджер интернет-магазина формирует и проводит заказ со статусом *В работе*, курьеру, указанному в заказе, должно автоматически прийти уведомлении об этом. После этого на клиенте у соответствующего курьера должна быть автоматически открыта форма проведенного заказа.

В альтернативном варианте эту функциональность можно реализовать как взаимодействие менеджера и курьера или как взаимодействие бота и курьера, которое мы рассматривали в предыдущей главе. Но в данном случае нас интересует именно аспект программного взаимодействия сервера и клиентского приложения.

Чтобы реализовать нашу задачу, нужно создать неотображаемое служебное обсуждение, участниками которого будут не все сотрудники интернет-магазина, а только курьеры, то есть пользователи с соответствующей ролью. Затем каждый раз при запуске клиентского приложения оно должно подписываться на это служебное обсуждение. При этом к обсуждению будет подключаться обработчик, который вызывается после появления новых сообщений в обсуждении. В этом обработчике после получения сообщения о проведении заказа у курьера, которому предназначается этот заказ, будет появляться уведомление и открываться форма заказа.

ПРИМЕЧАНИЕ

Пример отправки уведомлений курьерам с сервера можно посмотреть в конфигурации «ПрограммноеВзаимодействиеНапоминаниеКурьерам», которая прилагается к книге на компакт-диске.

Итак, сначала создадим служебное обсуждение для получения курьерами информации с сервера. Сделаем это в момент программного создания пользователей системы взаимодействия. Для упрощения примера будем реализовывать соответствующий код с помощью общей команды *НачатьРаботуССистемойВзаимодействия*, которую нужно вызвать всего один раз (листинг 4.10).

Листинг 4.10. Процедура «ОбработкаКоманды»

```

&НаКлиенте
Процедура ОбработкаКоманды (ПараметрКоманды, ПараметрыВыполненияКоманды)

    СлужебныеСообщения.Инициализация ();
    СлужебныеСообщенияКлиент.ПриНачалеРаботы ();

КонецПроцедуры
  
```

В обработчике команды мы вызываем процедуру *Инициализация()* общего серверного модуля *СлужебныеСообщения* (листинг 4.11).

Листинг 4.11. Процедура «Инициализация»

```

Процедура Инициализация () Экспорт

    Если НЕ СистемаВзаимодействия.ИнформационнаяБазаЗарегистрирована () Тогда
        Возврат;

    КонецЕсли;

    УстановитьПривилегированныйРежим (Истина);

    КлючСлужебногоОбсуждения = СлужебныеСообщения.ПолучитьКлючСлужебногоОбсуждения ();

    // Создаем пользователей системы взаимодействия
    Для Каждого ПользовательИБ ИЗ ПользователиИнформационнойБазы.ПолучитьПользователей () Цикл

        Попытка

            ИдентификаторПользователяСВ =
                СистемаВзаимодействия.ПолучитьИдентификаторПользователя (ПользовательИБ.УникальныйИдентификатор);
            ПользовательСВ = СистемаВзаимодействия.ПолучитьПользователя (ИдентификаторПользователяСВ);

        Исклучение

            ПользовательСВ = СистемаВзаимодействия.СоздатьПользователя (ПользовательИБ);
            ПользовательСВ.Записать ();

        КонецПопытки;

        Если ПользовательИБ.Роли.Содержит (Метаданные.Роли.Курьер) Тогда
  
```

```

// создаем обсуждение служебное обсуждение для курьеров
СлужебноеОбсуждение = СистемаВзаимодействия.ПолучитьОбсуждение (КлючСлужебногоОбсуждения) ;
Если СлужебноеОбсуждение = Неопределено Тогда
    СлужебноеОбсуждение = СистемаВзаимодействия.СоздатьОбсуждение () ;
    СлужебноеОбсуждение.Отображаемое = Ложь ;
    СлужебноеОбсуждение.Ключ = КлючСлужебногоОбсуждения ;

КонецЕсли ;

Если НЕ СлужебноеОбсуждение.Участники.Содержит (ПользовательСВ.Идентификатор) Тогда
    СлужебноеОбсуждение.Участники.Добавить (ПользовательСВ.Идентификатор) ;

КонецЕсли ;

СлужебноеОбсуждение.Записать () ;

КонецЕсли ;

КонецЦикла ;

КонецПроцедуры

```

В этой процедуре сначала создаются пользователи системы взаимодействия на основе пользователей информационной базы. Этот вопрос подробно рассматривался в предыдущей главе, поэтому мы не будем еще раз на нем останавливаться. Нужный нам фрагмент выделен в листинге процедуры жирным шрифтом.

Ключ служебного обсуждения получаем с помощью функции *ПолучитьКлючСлужебногоОбсуждения()* общего модуля *СлужебныеСообщения* как произвольную строку (листинг 4.12).

В процессе создания пользователей системы взаимодействия для пользователей с ролью *Курьер* мы создаем служебное обсуждение с ключом *КлючСлужебногоОбсуждения*, если обсуждение с таким ключом еще не существует. И затем добавляем созданного пользователя системы взаимодействия в состав участников этого обсуждения. Чтобы обсуждение не отображалось в интерфейсе, мы устанавливаем свойство обсуждения *Отображаемое* в значение *Ложь*.

Листинг 4.12. Функция «ПолучитьКлючСлужебногоОбсуждения»

```

Функция ПолучитьКлючСлужебногоОбсуждения () Экспорт

    Возврат "Служебное обсуждение для курьеров" ;

КонецФункции

```

Теперь из модуля приложения в обработчике события *ПриНачалеРаботыСистемы* вызовем процедуру *ПриНачалеРаботы()* общего клиентского модуля *СлужебныеСообщенияКлиент* (листинг 4.13).

Листинг 4.13. Обработчик события «ПриНачалеРаботыСистемы» в модуле приложения

```

Процедура ПриНачалеРаботыСистемы ()

    СлужебныеСообщенияКлиент.ПриНачалеРаботы () ;

КонецПроцедуры

```

В результате при каждом запуске клиентского приложения оно будет подписываться на получение сообщений в служебном обсуждении. Кроме того, эта же процедура вызывается и в обработчике команды *НачатьРаботуССистемойВзаимодействия* после создания служебного обсуждения (см. листинг 4.10).

Процедуру *ПриНачалеРаботы()* общего модуля *СлужебныеСообщенияКлиент* заполним следующим образом (листинг 4.14).

Листинг 4.14. Процедура «ПриНачалеРаботы»

```

Процедура ПриНачалеРаботы () Экспорт

    Если НЕ СистемаВзаимодействия.ИнформационнаяБазаЗарегистрирована () Тогда
        Возврат ;

    КонецЕсли ;

    Если НЕ СлужебныеСообщения.ЭтаПодпискаДляКурьера () Тогда
        Возврат ;
    КонецЕсли ;

```

```

КонецЕсли;

КлючСлужебногоОбсуждения = СлужебныеСообщения.ПолучитьКлючСлужебногоОбсуждения();
ОбработкаСообщенийСервера = Новый ОписаниеОповещения("ОбработкаСообщенийСервера",
СлужебныеСообщенияКлиент);
СистемаВзаимодействия.НачатьПодключениеОбработчикаНовыхСообщений(, КлючСлужебногоОбсуждения,
ОбработкаСообщенийСервера);

КонецПроцедуры

```

В этой процедуре мы сначала должны определить, «кем» по своей роли является текущий пользователь клиентского приложения. Потому что никто, кроме курьеров, не является участником служебного обсуждения и не должен на него подписываться, иначе будет получена ошибка. Эту проверку мы выполняем с помощью функции *ЭтаПодпискаДляКурьера()* общего модуля *СлужебныеСообщения* (листинг 4.15).

Затем мы получаем ключ служебного обсуждения и вызываем метод менеджера системы взаимодействия *НачатьПодключениеОбработчикаНовыхСообщений()*. Во втором параметре метода мы передаем ключ служебного обсуждения. В третьем параметре мы передаем описание обработчика оповещения – процедуры *ОбработкаСообщенийСервера()*, которая будет вызвана после появления новых сообщений в обсуждении с ключом *КлючСлужебногоОбсуждения*.

Таким образом, при получении новых сообщений в нашем служебном обсуждении будет выполняться процедура *ОбработкаСообщенийСервера()*, в которой и производится обработка данных, содержащихся в сообщениях. Сам обработчик мы рассмотрим позже, после того как рассмотрим процедуру обработки проведения документов *Заказ*, в которой эти сообщения отправляются с сервера (см. листинг 4.16).

Листинг 4.15. Функция «ЭтаПодпискаДляКурьера»

```

Функция ЭтаПодпискаДляКурьера() Экспорт

ПользовательИБ = ПользователиИнформационнойБазы.ТекущийПользователь();
Если ПользовательИБ.Роли.Содержит(Метаданные.Роли.Курьер) Тогда
    Возврат Истина;

Иначе
    Возврат Ложь;

КонецЕсли;

КонецФункции

```

Теперь в процедуру *ОбработкаПроведения()* в модуле документа *Заказ* нам нужно добавить код для отправки уведомлений курьерам в служебное обсуждение, в случае если заказ имеет статус *ВРаботе* (листинг 4.16).

Листинг 4.16. Процедура «ОбработкаПроведения» в модуле документа «Заказ»

```

Процедура ОбработкаПроведения(Отказ, Режим)

Если СтатусЗаказа = Перечисления.СтатусыЗаказов.ВРаботе Тогда

    // регистр ОстаткиТоваров Приход
    Движения.ОстаткиТоваров.Записывать = Истина;
    Для Каждого ТекСтрокаТовары Из Товары Цикл
        Движение = Движения.ОстаткиТоваров.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Товар = ТекСтрокаТовары.Товар;
        Движение.Цвет = ТекСтрокаТовары.Цвет;
        Движение.Размер = ТекСтрокаТовары.Размер;
        Движение.Склад = Склад;
        Движение.Заказано = ТекСтрокаТовары.Количество;

    КонецЦикла;

    КлючСлужебногоОбсуждения = СлужебныеСообщения.ПолучитьКлючСлужебногоОбсуждения();
    СлужебныеСообщения.ОтправкаУведомленийКурьерам(КлючСлужебногоОбсуждения, Ссылка, Курьер);

ИначеЕсли СтатусЗаказа = Перечисления.СтатусыЗаказов.Выполнен Тогда

    ...

КонецЕсли;

...

```

КонецПроцедуры

В выделенном фрагменте процедуры мы получаем ключ служебного обсуждения. И затем вызываем процедуру *ОтправкаУведомленийКурьерам()* общего модуля *СлужебныеСообщения* и передаем в нее этот ключ, ссылку на заказ и ссылку на курьера, указанного в заказе (листинг 4.17).

Листинг 4.17. Процедура «ОтправкаУведомленийКурьерам»

```
Процедура ОтправкаУведомленийКурьерам(КлючСлужебногоОбсуждения, Заказ, Курьер) Экспорт

    СлужебноеОбсуждение = СистемаВзаимодействия.ПолучитьОбсуждение(КлючСлужебногоОбсуждения);
    СтруктураДанных = Новый Структура();
    СообщениеКлиенту = СистемаВзаимодействия.СоздатьСообщение(СлужебноеОбсуждение.Идентификатор);

    СтруктураДанных.Вставить("Текст", "Сформирован новый заказ");
    СтруктураДанных.Вставить("Заказ", Заказ);
    СтруктураДанных.Вставить("Курьер", Курьер);
    СообщениеКлиенту.Данные = СтруктураДанных;
    СообщениеКлиенту.Записать();

КонецПроцедуры
```

В этой процедуре по ключу служебного обсуждения, переданному в процедуру в первом параметре, мы получаем служебное обсуждение, в которое нужно отправлять сообщения для курьеров. Затем создаем сообщение в этом обсуждении.

После этого создаем структуру данных с полями *Текст*, *Заказ* и *Курьер*. Полям структуры *Заказ* и *Курьер* присваиваем ссылку на проводимый заказ и ссылку на курьера, переданные в процедуру во втором и третьем параметрах. И помещаем структуру *СтруктураДанных* в данные сообщения.

Теперь пришло время рассмотреть, как осуществляется обработка полученных сообщений на клиенте, подписанном на служебное обсуждение.

Эта обработка выполняется в процедуре *ОбработкаСообщенийСервера()* общего модуля *СлужебныеСообщенияКлиент* (листинг 4.18). Первым параметром в обработчик оповещения передается копия объекта *СообщениеСистемыВзаимодействия*, то есть того сообщения, которое было получено в служебном обсуждении.

Листинг 4.18. Процедура «ОбработкаСообщенийСервера»

```
Процедура ОбработкаСообщенийСервера(Сообщение, ДопПараметры) Экспорт

    ИдентификаторПользователяСВКурьер =
        СлужебныеСообщения.ПолучитьИдентификаторСВКурьера(Сообщение.Данные.Курьер);

    Если СистемаВзаимодействия.ИдентификаторТекущегоПользователя() = ИдентификаторПользователяСВКурьер
    Тогда
        ПоказатьЗначение(, Сообщение.Данные.Заказ);
        ПоказатьОповещениеПользователя(Сообщение.Данные.Текст,,,, СтатусОповещенияПользователя.Важное);

    КонецЕсли;

КонецПроцедуры
```

Сначала с помощью функции *ПолучитьИдентификаторСВКурьера()* общего модуля *СлужебныеСообщения* (см. листинг 4.19) по переданной ссылке на курьера (*Сообщение.Данные.Курьер*) мы получаем идентификатор курьера в системе взаимодействия. И если он совпадает с идентификатором текущего пользователя, то, значит, это клиентское приложение курьера, указанного в заказе, при проведении которого было послано сообщение в служебное обсуждение.

В этом случае мы открываем форму заказа (*Сообщение.Данные.Заказ*) и показываем оповещение пользователя с текстом, переданным в сообщении (*Сообщение.Данные.Текст*).

Листинг 4.19. Функция «ПолучитьИдентификаторСВКурьера»

```
Функция ПолучитьИдентификаторСВКурьера(Курьер) Экспорт

    ПользовательИБКурьер = ПользователиИнформационнойБазы.НайтиПоИмени(СокрЛП(Курьер.Код));
    Если ПользовательИБКурьер = Неопределено Тогда
        Возврат "";
    КонецЕсли;
```

```

ИдентификаторПользователяСВКурьер =
СистемаВзаимодействия.ПолучитьИдентификаторПользователя (ПользовательИБКурьер.УникальныйИдентификатор);

Возврат ИдентификаторПользователяСВКурьер;

КонецФункции

```

Таким образом, сначала администратор с помощью команды *Начать работу с системой взаимодействия* создает служебное обсуждение, в котором курьеры будут получать сообщения от сервера. Затем каждый раз при запуске клиентские приложения всех курьеров подписываются на это обсуждение. И когда менеджер формирует и проводит заказ со статусом *В работе*, курьеру, указанному в заказе, приходит уведомление об этом и форма самого заказа автоматически открывается (рис. 4.2).

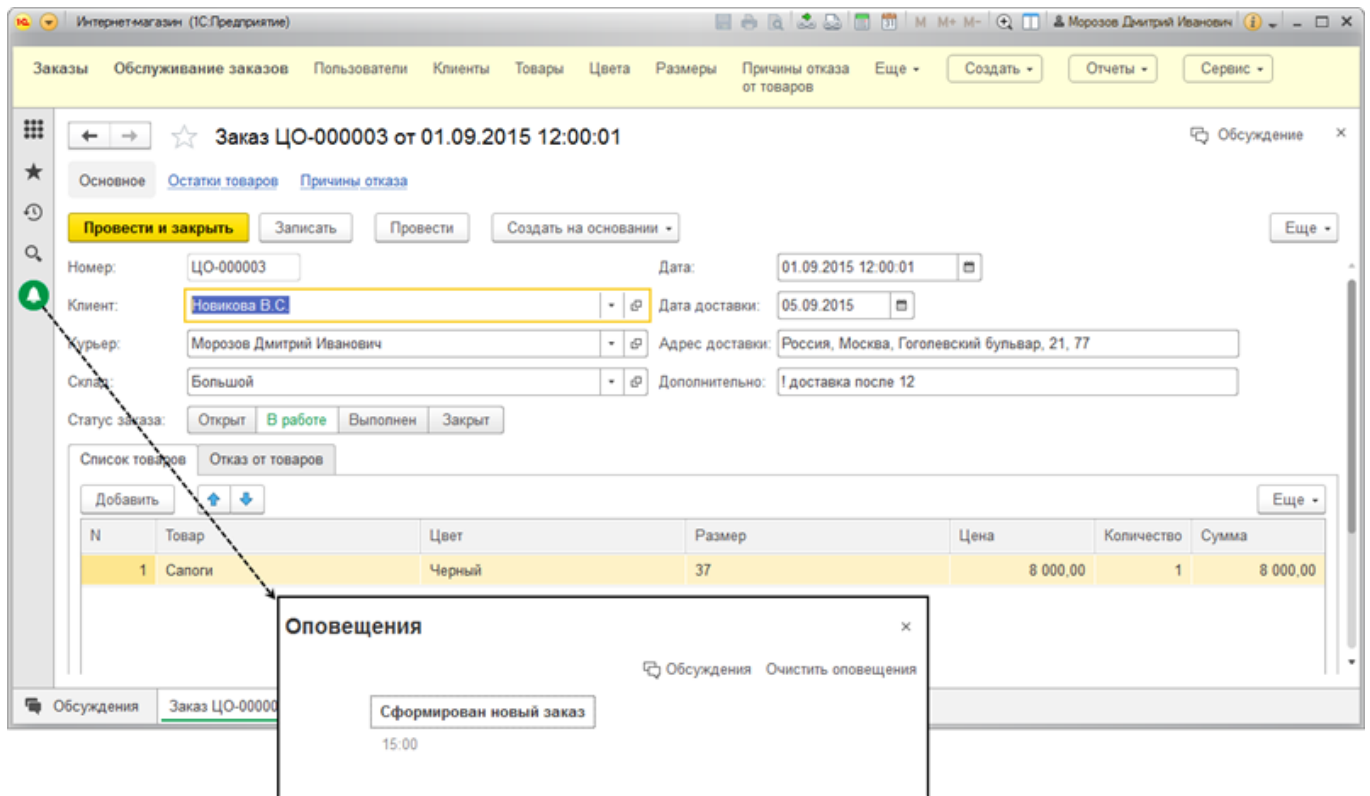


Рис. 4.2. Клиентское приложение курьера