Платформа "1С:Предприятие" версии 8.3.5
01.06.2016

Пример создания HTTP-сервисов на платформе "1С:Предприятие"

В этой статье разбираются демонстрационные HTTP-сервисы, созданные в демонстрационной конфигурации "Управляемое приложение" для платформы "1С:Предприятие" версии 8.3.5 и старше.

Цель статьи – помочь разобраться с использованием технологии HTTP-сервисов и показать практическое применение некоторых неочевидных механизмов.

Демонстрационная база "Управляемое приложение" представляет собой простую конфигурацию, в которой создано большинство объектов, которые могут понадобиться при автоматизации деятельности небольшой торговой фирмы. В частности, в ней присутствует справочник "Товары". Элементами этого справочника мы будем управлять при помощи HTTP-сервиса. Такой сценарий может возникнуть, например, при интеграции с интернет-магазином или другой корпоративной ИС, в которую заносится первичная информация о товарах.

В конфигурации создано два HTTP сервиса – "Товары" и "ОписанияТоваров". Первый демонстрирует работу в REST-стиле, второй - в RPC стиле, напоминающем протокол SOAP.

Для удобства изучения описываемых HTTP-сервисов рекомендуется включить авторизацию ОС при публикации на веб-сервере и настроить пользователю с ролью "Администратор" использование windows-аутентификации от имени пользователя ОС, под которым будет проходить изучение.

HTTP-сервис "Товары"

HTTP-сервис "Товары" написан в REST-стиле. Он позволяет получать и удалять элементы и группы в справочнике товаров. Доступ к элементу осуществляется с помощью его пути в иерархии.

Например, для того чтобы получить информацию о товаре "Ряженка" с кодом 000000027, входящем в группу "Молочные" с кодом 000000099, которая входит, в свою очередь в группу "Продукты" с кодом 000000011, в браузере надо будет набрать http://<ПутьПубликацииИБ>/hs/Products/000000011/000000099/000000027. Если база опубликована по пути http://localhost:8090/Platform8Demo/, то путь будет:

http://localhost:8090/Platform8Demo/**hs**/Products/000000011/000000099/000000027.

Из чего состоит путь? Рассмотрим по частям:

- http://localhost:8090/Platform8Demo/ - путь публикации информационной базы
- hs – обязательный сегмент пути, сообщающий серверу, что будет происходить работа с http-сервисами
- Products – строка, идентифицирующая конкретный веб-сервис. Задается свойством "Корневой URL http-сервиса" объекта метаданных веб-сервиса
- 000000011/000000099/000000027 – путь внутри сервиса. Все допустимые пути для данного сервиса определяются совокупностью дочерних объектов метаданных "Шаблон URL".

В нашем случае у сервиса один дочерний объект шаблон URL. В свойстве "Шаблон" этого объекта записана строка "/*". Звездочка – это специальное значение, указывающее на то, что к данному шаблону подходят любые URL. В нашем случае необходимость использования такого шаблона (т.е. по сути отказа от ограничения URL) обусловлена произвольной глубиной иерархии товаров.

У нашего шаблона URL имеются два дочерних объекта, соответствующих HTTP-методам GET (получение) и DELETE (удаление). Именно в них указаны обработчики, которые будут вызываться при обращении к HTTP-сервису.

Для обработки запроса с использованием HTTP-метода GET (а именно такой будет создан, если вставить указанные выше URL в браузер) используется функция *ПутьКТовару**GET**.* Рассмотрим эту функцию немного подробнее:

- Функция принимает единственный параметр "Запрос", это объект типа **HTTPСервисЗапрос**, содержащий всю нужную нам информацию о выполняемом вызове.
- Функция определяет путь к запрашиваемому ресурсу при помощи конструкции *Запрос.ПараметрыURL["*"]* . По сути, она спрашивает: "Что в реальном URL присутствует на том месте, где в шаблоне была звездочка"?
- Дальнейший код функции ищет элемент справочника по цепочке "00000011/000000099/000000027", используя объекты "1С:Предприятия".
- Если в какой-то момент следующее звено цепочки не находится, то формируется ответ сервиса с кодом 404, знакомым каждому пользователю веб-браузера.

Копировать в буфер обмена

```
Ответ = Новый HTTPСервисОтвет(404);
```

- Если цепочка пройдена успешно и конечный элемент найден, то для него формируется XML-представление при помощи встроенной XML-сериализации. Если элемент является группой, то в ответ включается XML-представление всех непосредственных потомков.

  Сформированное XML-представление используется в ответе сервиса:

Копировать в буфер обмена

```
Ответ = Новый HTTPСервисОтвет(200);

Ответ.УстановитьТелоИзСтроки(СтрокаXML);

// Помогает клиенту понять, что за данные к нему пришли

// Браузеры, например, применят удобную подсветку XML-синтаксиса

Ответ.Заголовки.Вставить("Content-type", "application/xml");
```

Обработка HTTP-метода **DELETE** происходит в функции ***ПутьКТовару**DELETE*** и мало чем отличается от обработки запроса с использованием HTTP-метода **GET**. Стоит обратить внимание на то, что в случае **DELETE** используется код ответа 204, а не 200. Код ответа 204 также свидетельствует об успешном выполнении операции на сервере, но сообщает клиенту, что сервер не посылал в ответ никаких данных в теле запроса.

HTTP-сервис "ОписанияТоваров"

HTTP-сервис "ОписанияТоваров" предназначен для получения и редактирования информации о товарах. Он написан в RPC (Remote Procedure Call) стиле, похожем на SOAP. В качестве дополнительного условия также предположим, что заказчик, для которого мы разрабатываем конфигурацию, потребовал предусмотреть наличие нескольких версий API где-то в будущем.

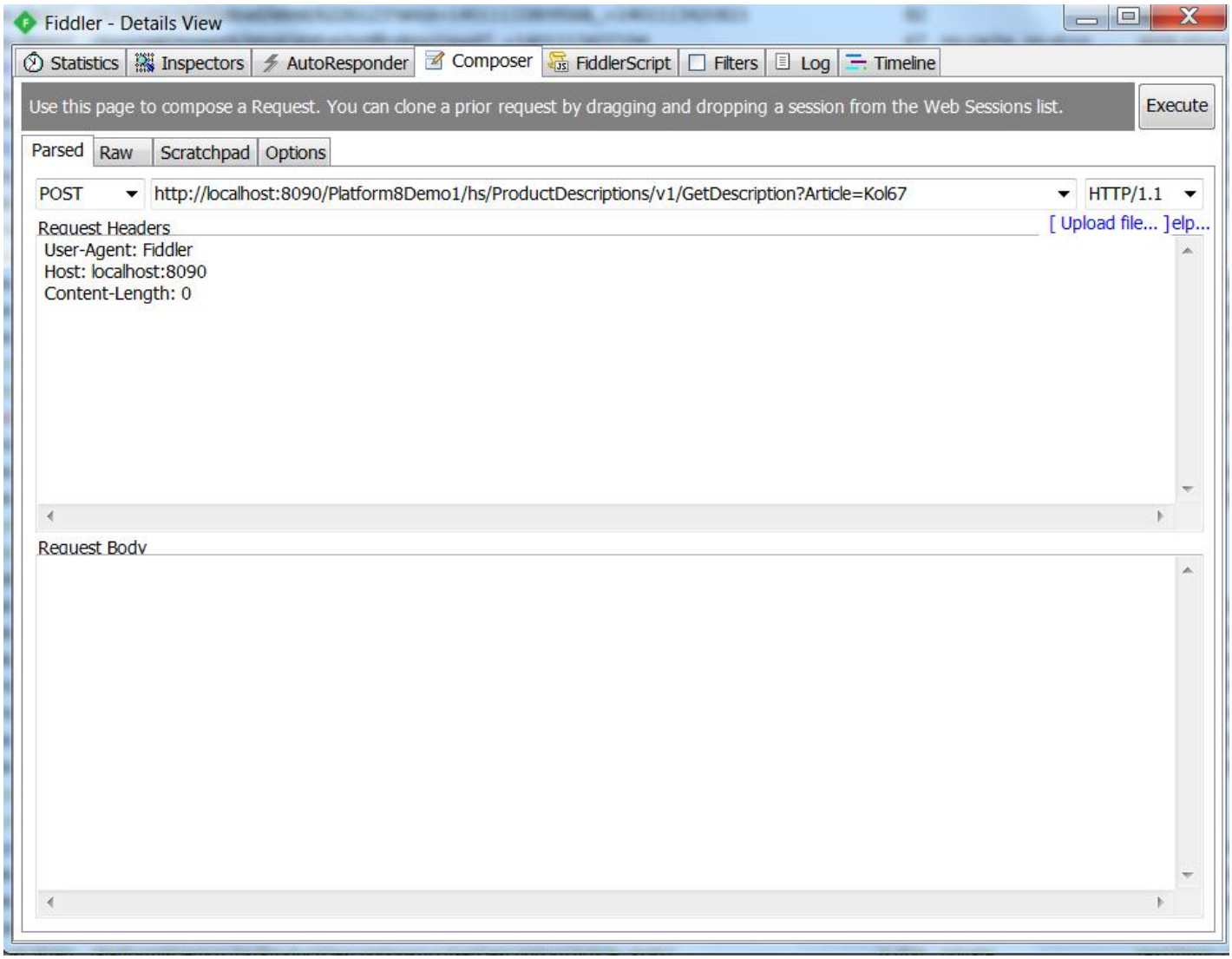
Обращение к сервису выполняется при помощи запросов с использование метода POST к URL следующего вида:

http://localhost:8090/Platform8Demo/hs/ProductDescriptions/v1/SetDescription?Article=Kol67

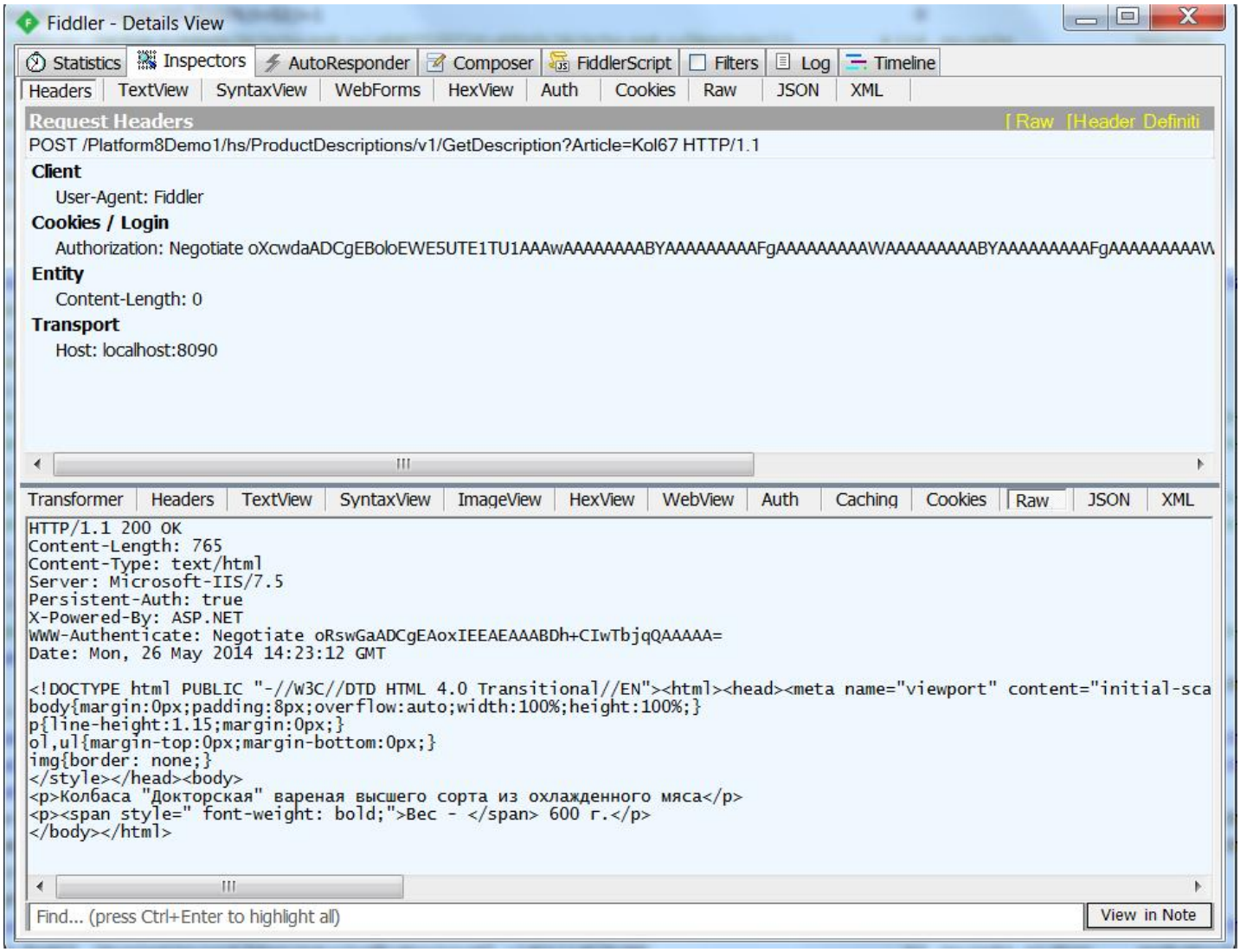
Рассмотрим, из чего состоит путь:

- http://localhost:8090/Platform8Demo/ - путь публикации информационной базы,
- **hs** – обязательный сегмент пути, сообщающий серверу, что будет происходить работа с http-сервисами,
- **ProductDescriptions** –корневой URL http-сервиса,
- **v1** – идентификатор версии. В рассматриваемой реализации всегда v1,
- **SetDescription** - имя метода. В данном случае – установка описания товара. Для получения описания товара предусмотрен метод GetDescription,
- **?Article=Kol67** – строка с параметрами запроса. В общем виде выглядит как "?Параметр1=Значений1&Параметр2=Значение2&Параметр3=Значение3" и т.д. В нашем случае единственный параметр это артикул товара.

Испытаем наш сервис. К сожалению, это нельзя сделать из браузера (браузер, если не предпринять дополнительных усилий, отправляет запрос с использованием метода GET), поэтому мы воспользуемся бесплатной программой Fiddler (http://www.telerik.com/fiddler). Для создания запроса вручную достаточно перейти на вкладку **Composer**.

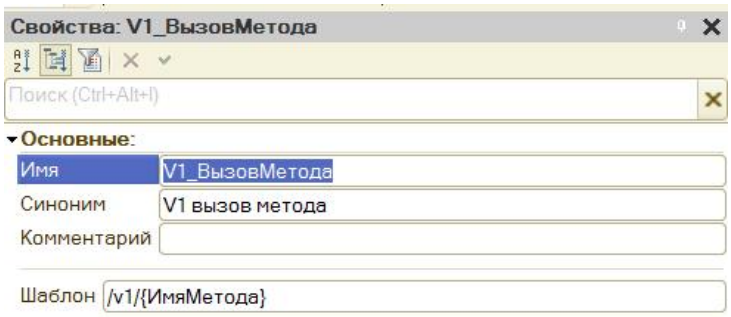


В результате выполнения запроса на вкладке **Inspectors** можно будет увидеть примерно следующее (в верхней части экрана – выполненный запрос, в нижней - ответ сервера):



Видно, что сервер передал описание товара в формате html.

Рассмотрим, как реализован сервис. Объект метаданных HTTP-сервиса имеет единственный дочерний шаблон URL, в котором прописан следующий шаблон:



Т.к. у нас пока нет разных версий сервиса, сегмент с номером версии фиксирован, а вот второй сегмент может принимать разные значения, соответствующие именам методов. В коде получение имени метода выглядит следующим образом:

Копировать в буфер обмена

```
ИмяМетода = Запрос.ПараметрыURL["ИмяМетода"];

Если ИмяМетода = "SetDescription" Тогда

ИначеЕсли ИмяМетода = "GetDescription" Тогда

Иначе

	Ответ = Новый HTTPСервисОтвет(404);

	Ответ.УстановитьТелоИзСтроки("Неизвестное имя метода");

КонецЕсли;
```

Обращаем внимание, что коллекция **"ПараметрыURL"** запроса содержит единственное значение – согласно количеству сегментов, которые могут принимать разные значение.

Аналогично извлекается артикул, только используется свойство **"ПараметрыЗапроса"**, а не **"ПараметрыURL"**.

Копировать в буфер обмена

```
Артикул = Запрос.ПараметрыЗапроса.Получить("Article");
Если Артикул = Неопределено Тогда
      Ответ = Новый HTTPСервисОтвет(400);
      Ответ.УстановитьТелоИзСтроки("Не задан параметр Article");
      Возврат Ответ;
 КонецЕсли;
```

Для возврата описания товара мы устанавливаем тело запроса:

Копировать в буфер обмена

```
ИначеЕсли ИмяМетода = "GetDescription" Тогда
Ответ = Новый HTTPСервисОтвет(200);
Ответ.УстановитьТелоИзСтроки(Товар.Описание);
Ответ.Заголовки["Content-Type"] = "text/html";
```

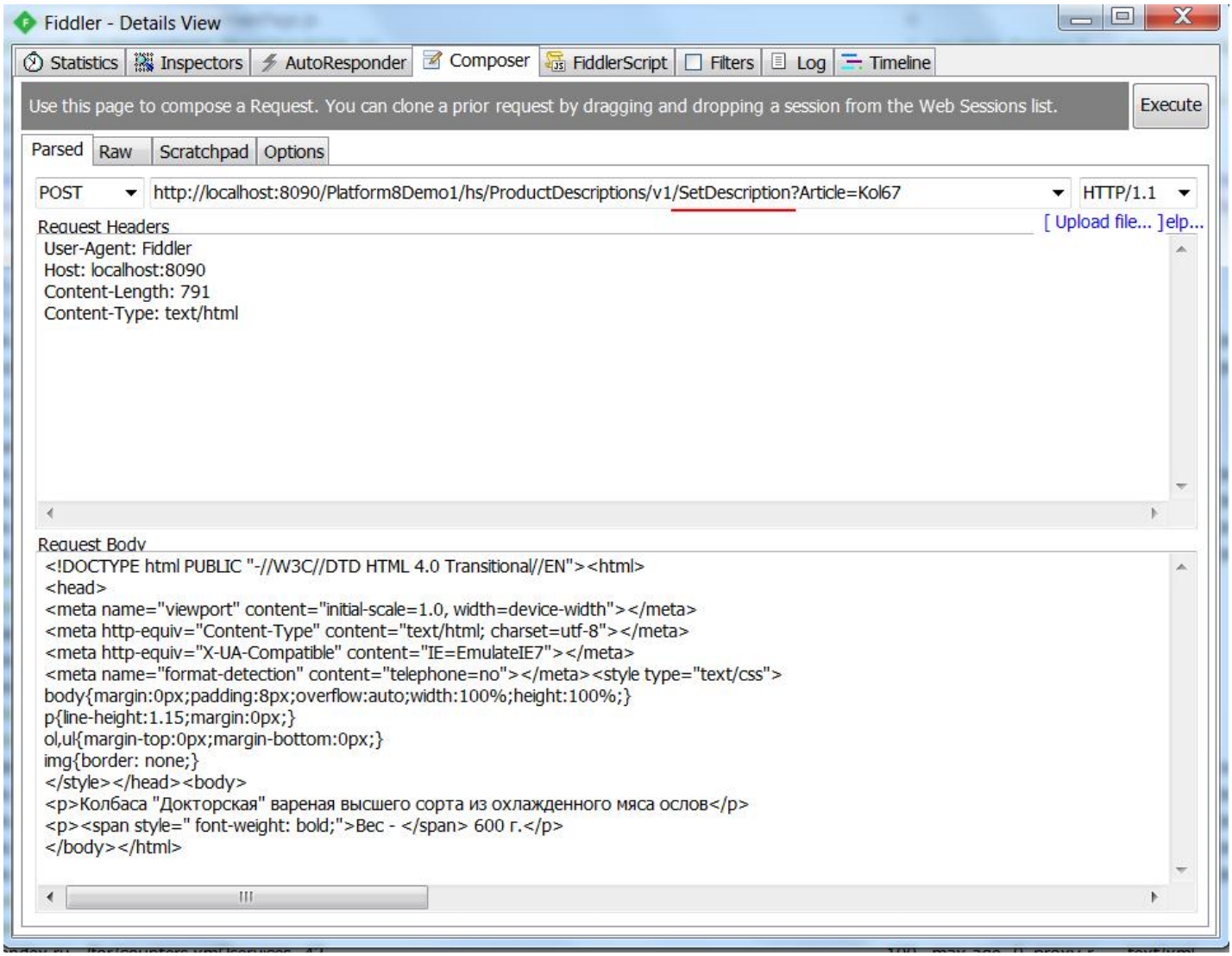
Аналогично, для установки описания товара мы получаем его из запроса:

Копировать в буфер обмена

```
Если ИмяМетода = "SetDescription" Тогда
       ТипСодержимого = Запрос.Заголовки.Получить("Content-Type");
       Если ТипСодержимого <> "text/html" И  ТипСодержимого <> "text/plain" Тогда
             // Сообщаем клиенту, что не поддерживаем такой тип содержимого
             Ответ = Новый HTTPСервисОтвет(415);
      Иначе
             Товар.Описание = Запрос.ПолучитьТелоКакСтроку();
             Товар.Записать();
             Ответ = Новый HTTPСервисОтвет(204);
      КонецЕсли;
```

При установке описания из тела запроса мы проводим минимальную проверку корректности того, что прислал нам клиент, в данном случае – только типа содержимого, изучая заголовок "Content-type".

Для того чтобы протестировать установку тела запроса достаточно заполнить его в Fiddler:



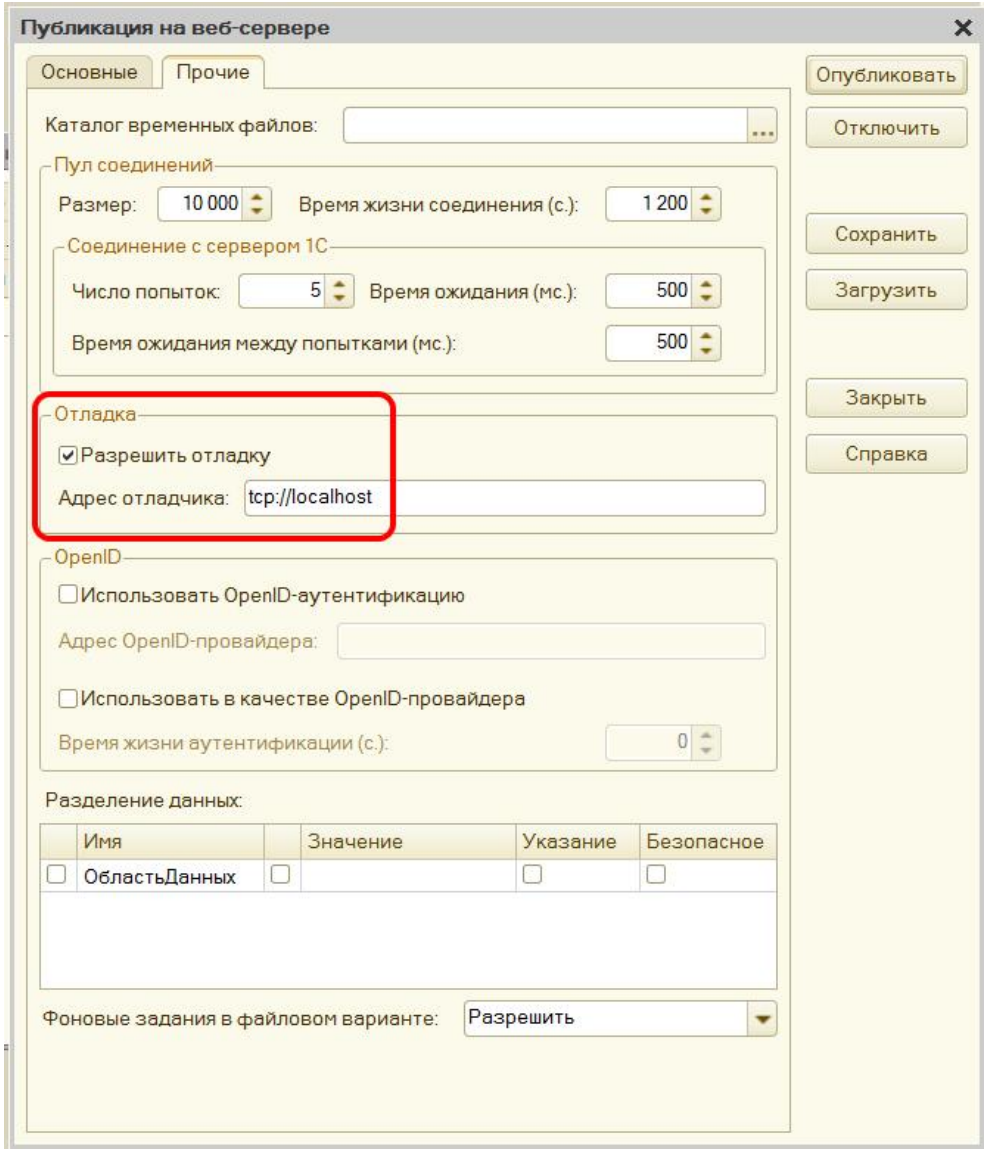
Отладка кода HTTP-сервиса

Отладка кода HTTP-сервиса аналогична отладке код SOAP веб-сервиса. Для включения отладки нужно:

- Разрешить отладку при публикации на веб-сервере,
- Включить автоматическое подключение к HTTP-сервисам,
- Установить точку останова в интересующем месте.

Разрешение отладки на веб-сервере

Для разрешения отладки на веб-сервере нужно перейти на вкладку **"Прочие"** диалога публикации на веб-сервере, установить флаг "разрешить отладку" и указать адрес отладчика. Для локальной отладки можно указать **tcp://localhost**



То же самое можно сделать вручную, исправив vrd-файл, см документацию.

Включение автоматического подключения

Для того чтобы платформа автоматически подключалась для отладки к вызываемым HTTP-сервисам нужно:

- Открыть окно настроек автоматического подключения с помощью команды главного меню **"Отладка – Автоматическое подключение"**.
- Установить флажок **"HTTP-сервисы на сервере"**.

Помните, что флажок следует устанавливать при каждом запуске конфигуратора, в котором требуется отладка HTTP-сервисов.

Заключение

В статье рассмотрены основные аспекты программирования HTTP-сервисов в "1С:Предприятии", в частности:

- Шаблоны URL и параметры URL,
- Параметры запроса,
- Работа с заголовками,
- Получение тела запроса,
- Формирование ответа.

Также показано, как можно их тестировать при помощи программы Fiddler. Более полные справочные материалы можно найти в ИТС по постоянному адресу.