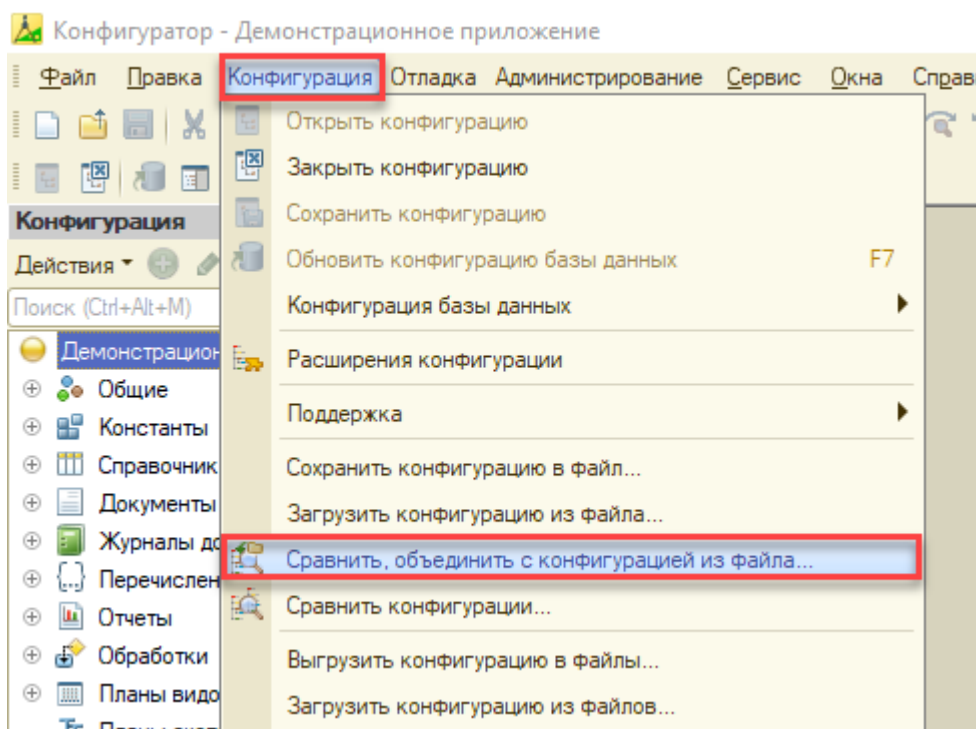


Инструкция по внедрению механизма «Контроль HTTP-сервисов»

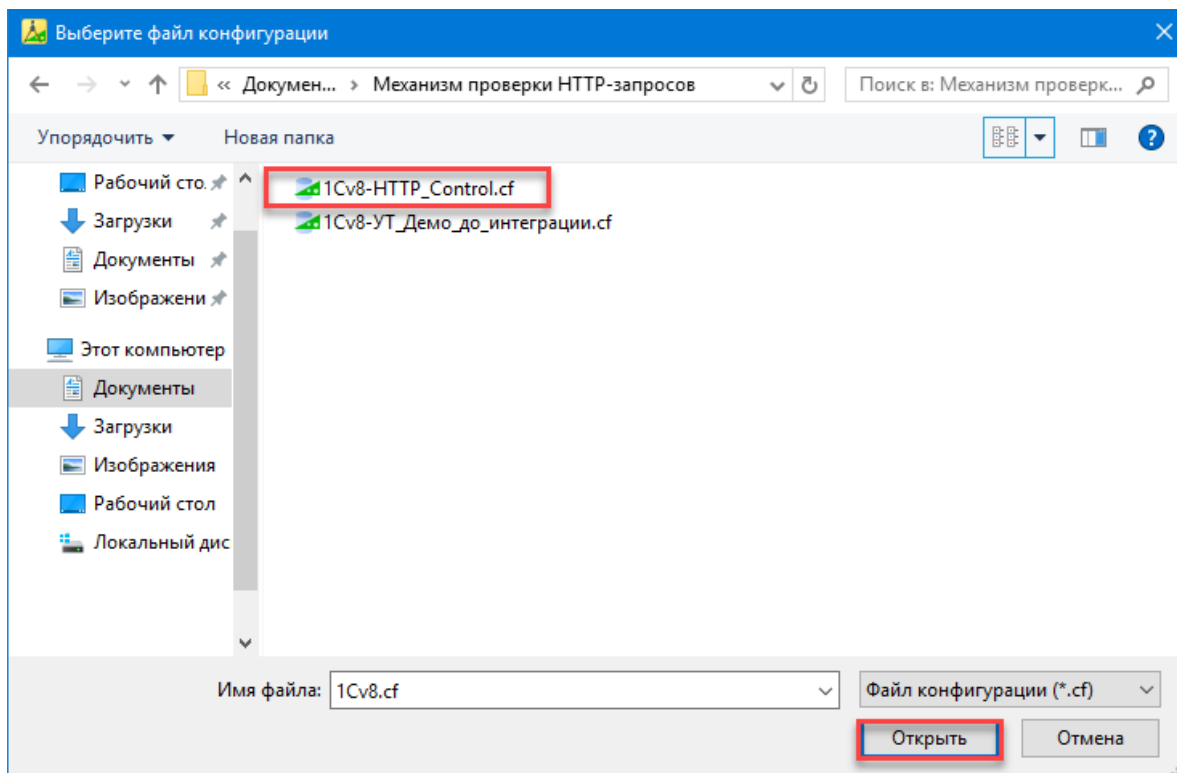
1. Интеграция механизма в существующую конфигурацию

Ниже описан процесс интеграции инструмента на примере конфигурации 1С «Демонстрационное приложение» версии 1.0.32.4. Скачать конфигурацию можно по адресу: <https://its.1c.ru/db/metod8dev/content/5028/hdoc>

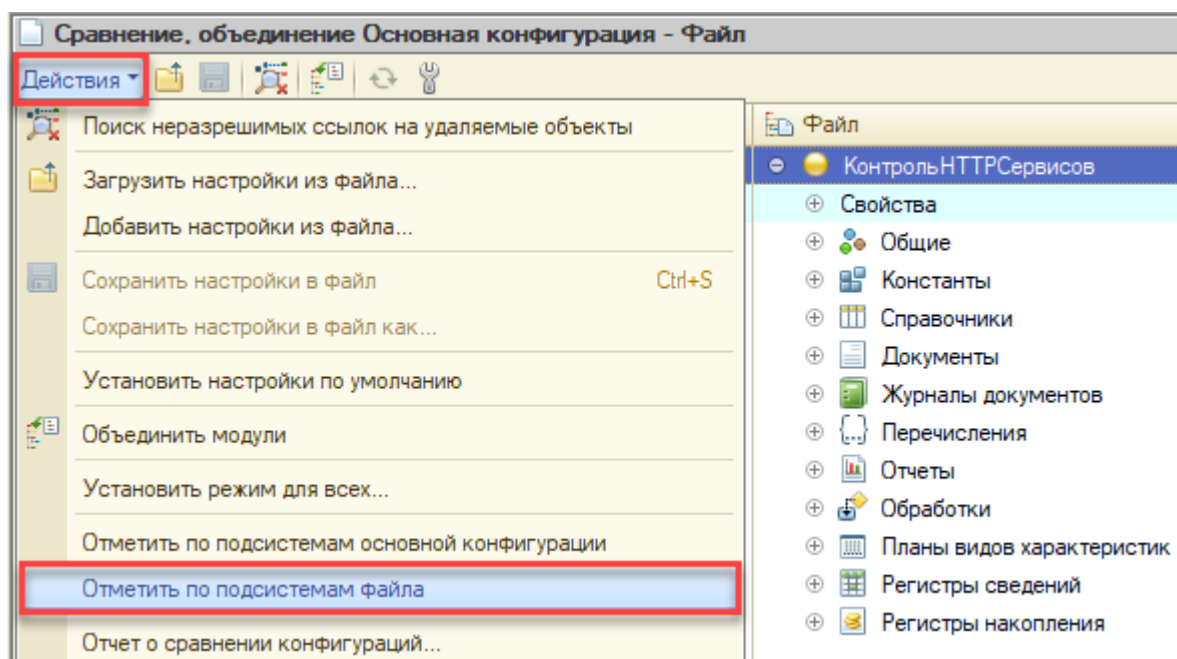
Механизм «Контроль HTTP-сервисов» поставляется в виде файла конфигурации. Для интеграции механизма в существующую конфигурацию необходимо в конфигураторе выполнить команды **«Конфигурация» – «Сравнить, объединить с конфигурацией из файла...»**



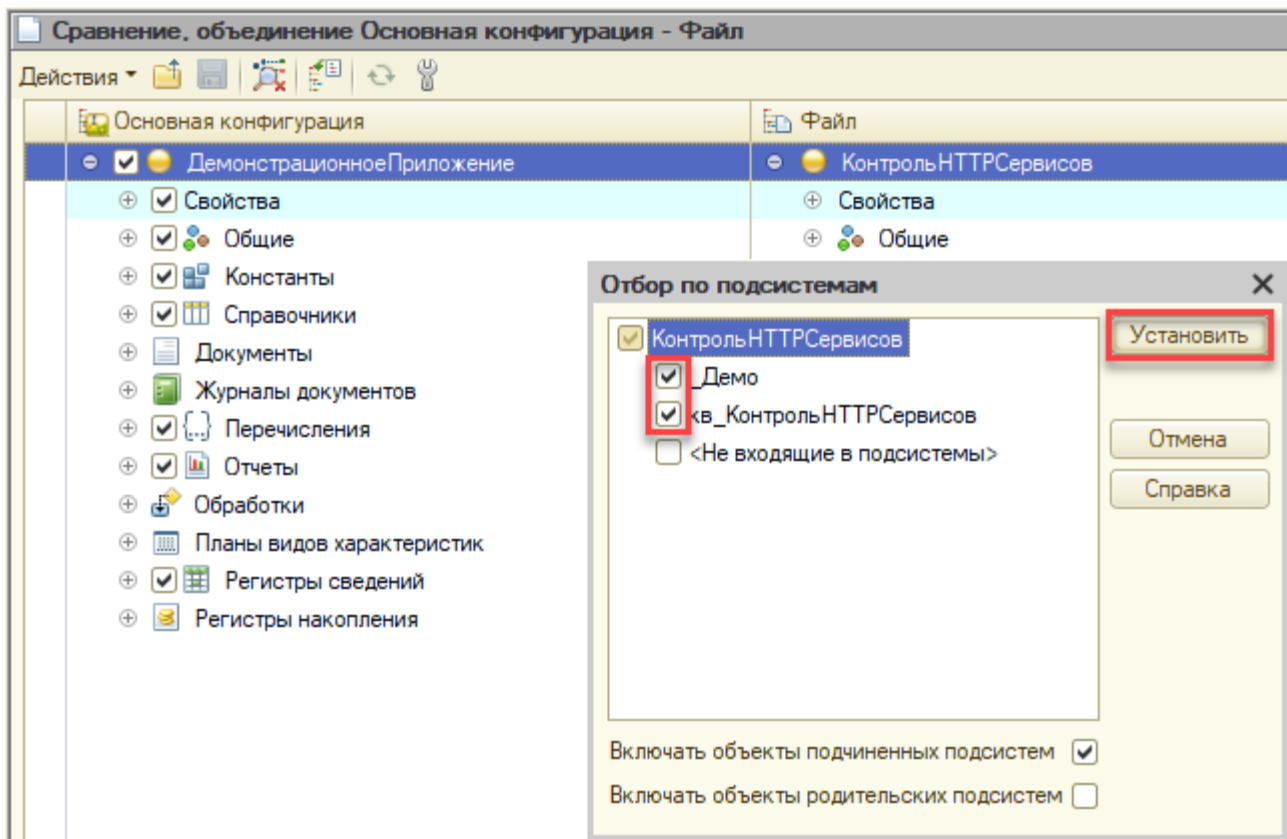
В окне выбора файла конфигурации необходимо указать поставляемый файл.



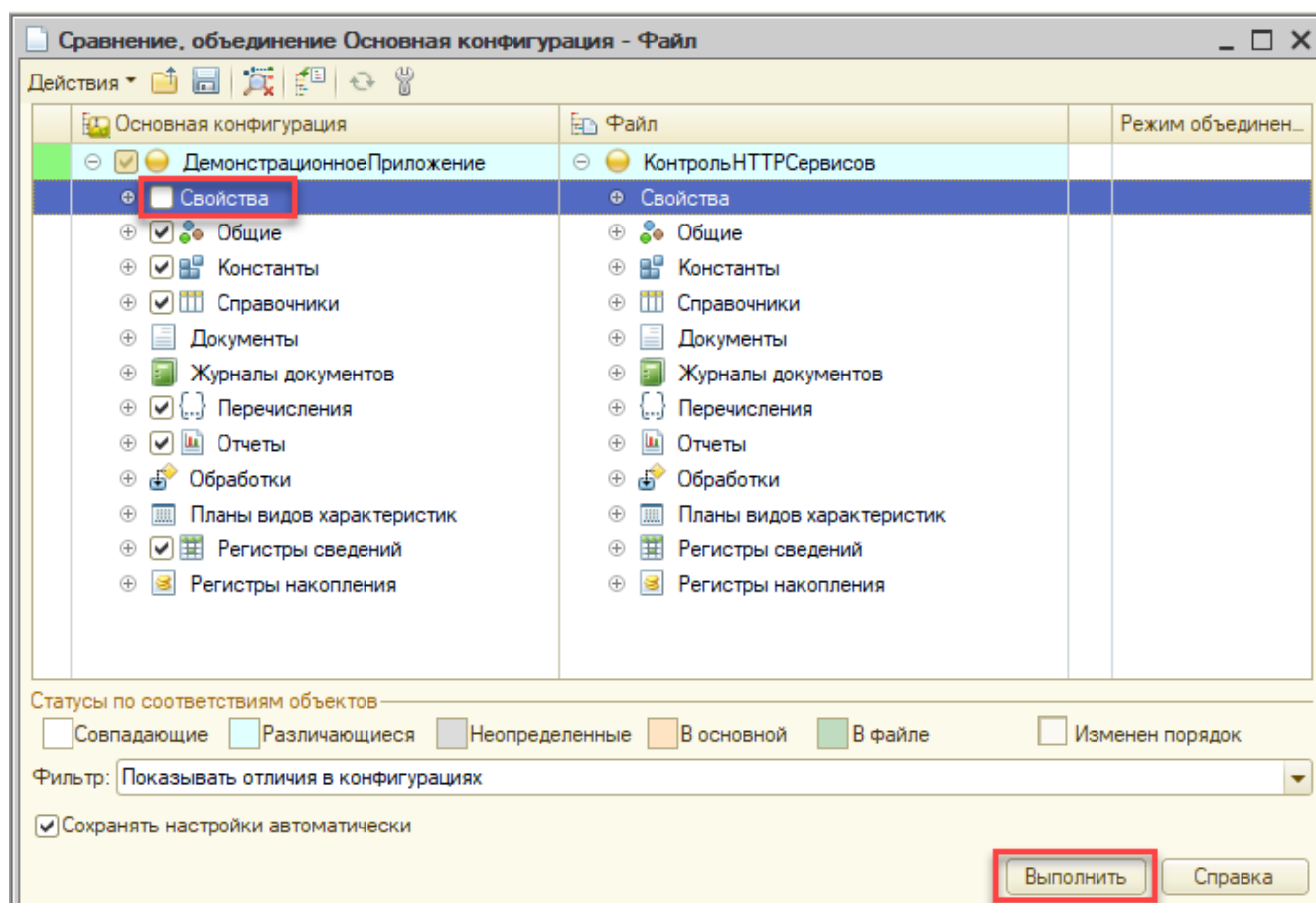
В окне «Сравнение, объединение» выполнить команду «Отметить по подсистемам файла».



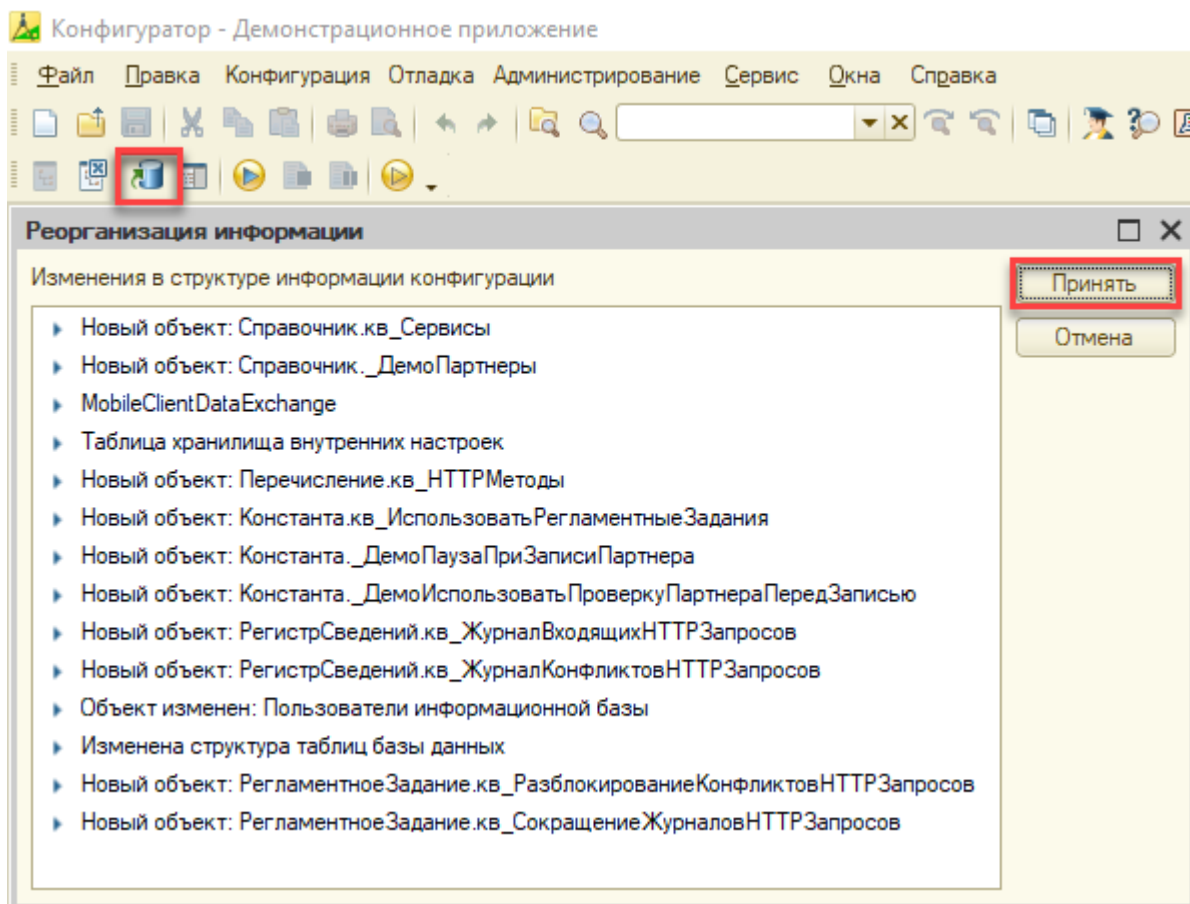
В открывшемся окне отметить флагом нужные подсистемы. Если объекты подсистемы «**Демо**» не нужны в конфигурации рабочей базы, флаг напротив её имени необходимо оставить не установленным.



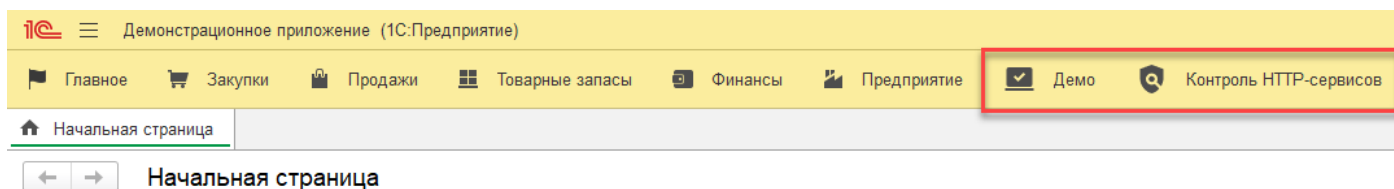
После выбора подсистем, в окне «Сравнение, объединение» необходимо снять флаг с раздела «Свойства» и нажать кнопку «Выполнить», завершив процесс объединения конфигураций.



После завершения объединения необходимо применить изменения командой конфигуризатора «Обновить конфигурацию базы данных».



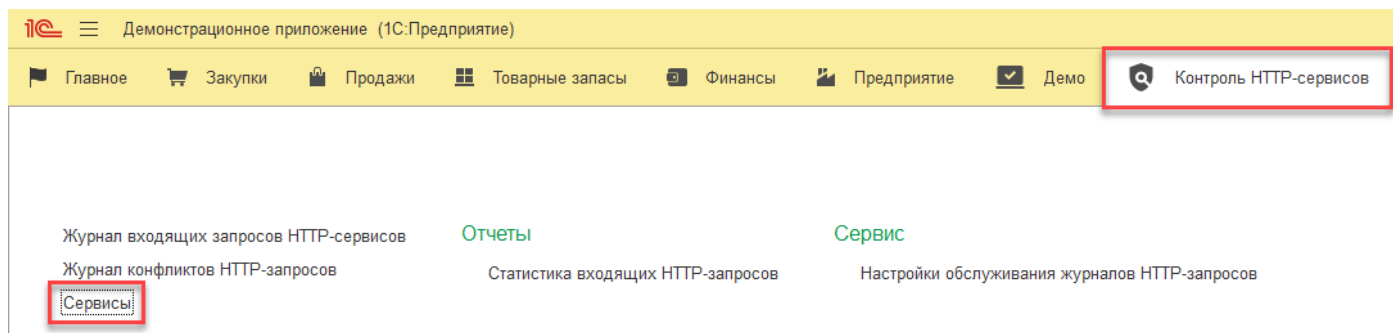
После применения изменений в пользовательском режиме 1С:Предприятие становятся доступны команды выбранных подсистем инструмента. На этом первый этап интеграции можно считать успешно завершенным.



Следующим этапом необходимо выполнить подключение и настройку HTTP-сервиса к механизму в режиме 1С:Предприятие.

2. Подключение HTTP-сервиса к механизму в режиме 1С:Предприятие

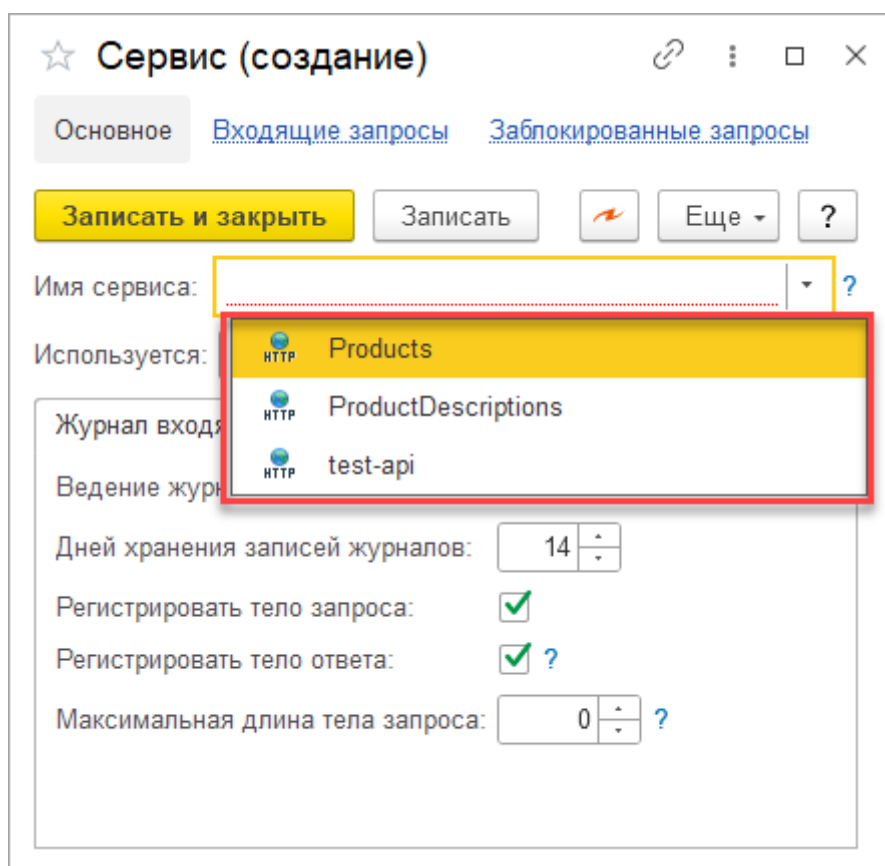
В подсистеме «Контроль HTTP-сервисов» доступна команда перехода к списку элементов справочника «Сервисы».



В форме списка справочника «Сервисы» добавляем новый элемент справочника сервисы по кнопке «Создать». Поле «Имя сервиса» заполняется путем выбора нужного HTTP-сервиса из выпадающего списка.

В списке доступны HTTP-сервисы из метаданных конфигурации.

Вручную текст поля «Имя сервиса» не редактируется, оно автоматически устанавливается по значению свойства «Корневой URL» подключаемого HTTP-сервиса.



В форме справочника доступны несколько настроек, назначение которых подробно описано во встроенной справке, вызываемой по клавише **F1**.

Необходимо выставить нужную комбинацию настроек под особенности конкретного сервиса. Эти настройки будут впоследствии доступны разработчику в **run-time**, в контексте обработки каждого метода этого сервиса.

☆ **ProductDescriptions (Сервис)**

Основное
Входящие запросы
Заблокированные запросы

Записать и закрыть
Записать
Еще
?

Имя сервиса: ProductDescriptions

Используется: Включено
Выключено

Журнал входящих запросов

Контроль запросов

Описание

Обеспечение идемпотентности: ☒

Контроль конфликтов параллельных запросов: ☒

Повторов до разблокировки конфликтов: 2

Порог разблокировки конфликтов: 5

Период проверки порога разблокировки: 5

Период гарантированной разблокировки: 0

После записи нового элемента справочника, имя сервиса и его основные настройки отражаются в форме списка справочника «Сервисы».

На этом завершен этап пользовательской настройки подключаемого к механизму HTTP-сервиса.

Демонстрационное приложение (1С:Предприятие)

Главное

Закупки

Продажи

Товарные запасы

Финансы

Предприятие

Демо

Контроль HTTP-сервисов

Начальная страница

Сервисы x

←

→

☆ Сервисы

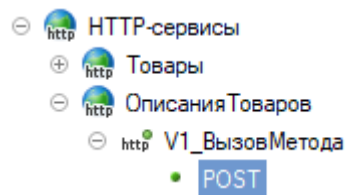
Создать

Имя сервиса ↓	Используется	Журналирование	Обеспечение идемпотентности	Контроль конфликтов запросов
<div><div></div>ProductDescriptions</div>	✓	✓	✓	✓

Следующим этапом необходимо в коде конфигурации подключить нужные методы HTTP-сервиса к механизму контроля.

3. Подключение методов HTTP-сервисов к механизму контроля в коде конфигурации

Выбираем в конфигурации метод сервиса, который необходимо подключить к механизму контроля.



Обработчик выбранного метода до подключения к механизму выглядит как на скриншоте ниже.

```
функция ВызовМетодаPOST (Запрос)

    Переименовать Ответ;

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

    Товар = ПолучитьТоварПоАртикулу (Артикул);
    Если Товар = Неопределено Тогда
        Ответ = Новый HTTPСервисОтвет (404);
        Ответ.УстановитьТелоИзСтроки ("Товар не найден");
        Возврат Ответ;
    КонецЕсли;

    ИмяМетода = Запрос.ПараметрыURL ["ИмяМетода"];

    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;
    ИначеЕсли ИмяМетода = "GetDescription" Тогда
        Ответ = Новый HTTPСервисОтвет (200);
        Ответ.УстановитьТелоИзСтроки (Товар.Описание);
        Ответ.Заголовки ["Content-Type"] = "text/html";
    Иначе
        Ответ = Новый HTTPСервисОтвет (404);
        Ответ.УстановитьТелоИзСтроки ("Неизвестное имя метода");
    КонецЕсли;

    Возврат Ответ;
Конецфункции
```


Для подключения метода к механизму контроля необходимо выполнить некоторый объем рефакторинга кода его обработчика.

С примером того, как должен выглядеть код обработчика метода, подключенного к механизму, можно ознакомиться в HTTP-сервисе «_ДемоТестовыйСервис», входящего в состав подсистемы «_Демо» инструмента.

После рефакторинга код обработчика подключенного к механизму метода «ВызовМетодаPOST» стал выглядеть следующим образом:

```
[-] функция ВызовМетодаPOST (Запрос)

    // Получаем перечень параметров извлекаемых из запроса
    ПараметрыМетода = ПараметрыМетодаВызовМетодаPOST ();

    // Регистрируем запрос в подсистеме контроля HTTP-сервисов
    Контекст = кв_HTTPСервисы.КонтекстЗапроса (Запрос, Истина, Истина, ПараметрыМетода);

    // Проверяем запрос на идемпотентность
    Если Контекст.ГотовОтветНаЗапрос Тогда
        Возврат кв_HTTPСервисы.ОтветНаЗапрос (Контекст);
    КонецЕсли;

    // Выполняем обработку бизнес логики запроса
    ОбработатьЗапросВызовМетодаPOST (Контекст);

    // Регистрируем ответ сервиса и статус обработки
    Возврат кв_HTTPСервисы.ОтветНаЗапрос (Контекст);

[-] Конецфункции

[-] // Формирует описание получаемых параметров запроса ВызовМетодаPOST. [//...]
[-] функция ПараметрыМетодаВызовМетодаPOST ()

    ПараметрыМетода = Новый Структура;
    ПараметрыМетода.Вставить ("Article", Истина);

    Возврат ПараметрыМетода;

[-] Конецфункции

[-] // Обработчик входящего запроса ВызовМетодаPOST [//...]
[-] Процедура ОбработатьЗапросВызовМетодаPOST (Контекст)

    Артикул = Контекст.ПараметрыЗапроса ["Article"];

    Товар = ПолучитьТоварПоАртикулу (Артикул);
    Если Товар = Неопределено Тогда
        ТекстОшибки = СтрШаблон ("Товар с артикулом '%1' не найден", Артикул);
        кв_HTTPСервисы.ДобавитьОписаниеОшибки (Контекст, 400, , ТекстОшибки);
        Возврат;
    КонецЕсли;

    ИмяМетода = Контекст.ПараметрыЗапроса ["ИмяМетода"];

    Если НЕ ИзвестныйМетод (ИмяМетода) Тогда
        ТекстОшибки = СтрШаблон ("Неизвестное имя метода: %1", ИмяМетода);
        кв_HTTPСервисы.ДобавитьОписаниеОшибки (Контекст, 400, , ТекстОшибки);
        Возврат;
    КонецЕсли;

    УстановитьОписаниеТовара (ИмяМетода, Товар, Контекст);

    ПередатьОписаниеТовара (ИмяМетода, Товар, Контекст);

[-] КонецПроцедуры
```


Вся бизнес-логика вызывается теперь в новом методе «ОбработатьЗапросВызовМетодаPOST». Остальные метода, которые следуют до и после указанного метода, являются обязательными техническими вызовами самого механизма.

В методе «ПараметрыМетодаВызовМетодаPOST» собираем коллекцию параметров метода, которые будут извлечены из тела запроса, если тип содержимого запроса = "application/json".

Ключ структуры «ПараметрыМетода» – имя извлекаемого параметра, Значение с типом Булево – признак обязательности заполнения параметра.

Полный текст модуля HTTP-сервиса «ОписанияТоваров» демонстрационного приложения после рефакторинга можно скачать по ссылке: [ссылка на файл «HTTP-сервис ОписанияТоваров_ Модуль»](#).

Если подключение метода сервиса выполнено успешно, и в настройках сервиса включена опция журналирования запросов, то после обработки входящего запроса в журнале будут зафиксированы параметры такого запроса и ответа на него.

В примере ниже тело запроса представлено в виде HTML-текста и передается в реквизит карточки товара «как есть», без преобразования. То есть так, как это и задумано в демонстрационном приложении 1С.

☆ POST: ProductDescriptions/v1/SetDescription

Сервис: ProductDescriptions

Ресурс: /v1/SetDescription

Ключ идемпотентности: 26a2c017-0ad5-4f56-b408-e0b6ba244fd3

Дата запроса: 12.01.2025 23:34:34

Метод: POST

Код состояния: 204

Основное

Подобные

Дополнительно

Размер запроса: 950

Размер ответа: 48

Длительность: 54,35500

Запрос:

URL: https://localhost/UT_integration/hs/ProductDescriptions,
HTTP-метод: POST,
Заголовки: "Idempotency-Key": "26a2c017-0ad5-4f56-b408-e0b6ba244fd3"; "Authorization": "Basic X3Ric3RhcGk6MTIz"; "Connection": "keep-alive"; "X-Forwarded-For": "::1"; "Accept": "*/"; "Postman-Token": "6786b4e1-4d12-4605-b47d-9400a6bef24e"; "User-Agent": "PostmanRuntime/7.43.0"; "Accept-Encoding": "gzip, deflate, br"; "Content-Type": "text/plain"; "Host": "localhost"; "Content-Length": "950"; "Cache-Control": "no-cache",
Параметры ресурса:
- "ИмяМетода": "SetDescription",
Параметры запроса:
- "Article": "NT02",
Тело запроса:
"<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"><html><head><meta name=""viewport"" content=""initial-scale=1.0, width=device-width""></meta><meta http-equiv=""Content-Type"" content=""text/html; charset=utf-8""></meta><meta http-equiv=""X-UA-Compatible"" content=""IE=EmulateIE7""></meta><meta name=""format-detection"" content=""telephone=no""></meta><style type=""text/css"">
body{margin:0px;padding:8px;overflow:auto;width:100%;height:100%;}
p{line-height:1.15;margin:0px;}
ol,ul{margin-top:0px;margin-bottom:0px;}
img{border: none;}
</style></head><body>
<p>Торт ""Праздничный"".</p>
<p>Три бисквитных коржа со вкусом шоколада щедро залиты банановым кремом.
</p>
<p>Вес - 1кг.</p>
<p>Страна-производитель - Россия.</p>
</body></html>"

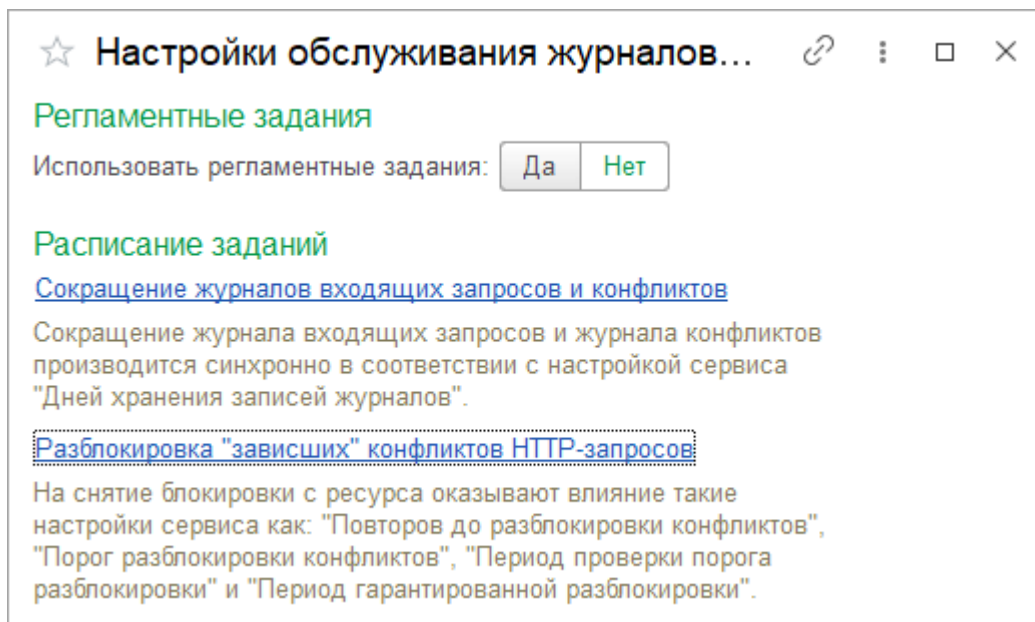
Ответ:

Код состояния: 204. Причина:
Заголовки: "Content-Type": "application/json; charset=utf-8"
Тело ответа: {
"succes": true,
"Article": "NT02"
}

4. Настройка служебных параметров механизма после его интеграции

В механизме предусмотрены регламентные задания, форму настроек которых можно вызвать из раздела «Сервис» подсистемы «Контроль HTTP-сервисов» командой «Настройки обслуживания журналов HTTP-севрисов»:

- Сокращение журналов входящих HTTP-запросов и конфликтов HTTP-запросов;
- Разблокирование конфликтов HTTP-запросов.



Переключателем «Использовать регламентные задания» регулируется, будут ли регламентные задания включены в механизм.

Каждое регламентное задание поставляется как предопределенное с предустановленным расписанием по умолчанию.

Перейдя по ссылке, пользователь может самостоятельно установить расписание каждого регламентного задания под свои потребности.

На этом процесс интеграции и выполнения базовых настроек можно считать завершенным. Тонкую настройку каждого сервиса, политики журналирования, обнаружения и разблокирования конфликтов, пользователь выполняет в зависимости от поведения конкретного сервиса и предъявляемых к нему требований.

Те методы существующих на момент интеграции HTTP-сервисов, которые не нуждаются в использовании механизма, не требуется подвергать рефакторингу.