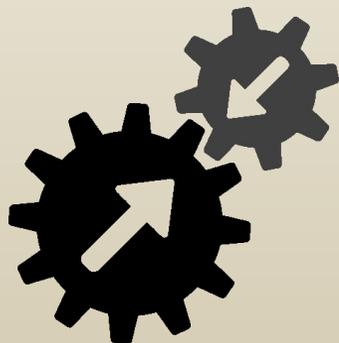


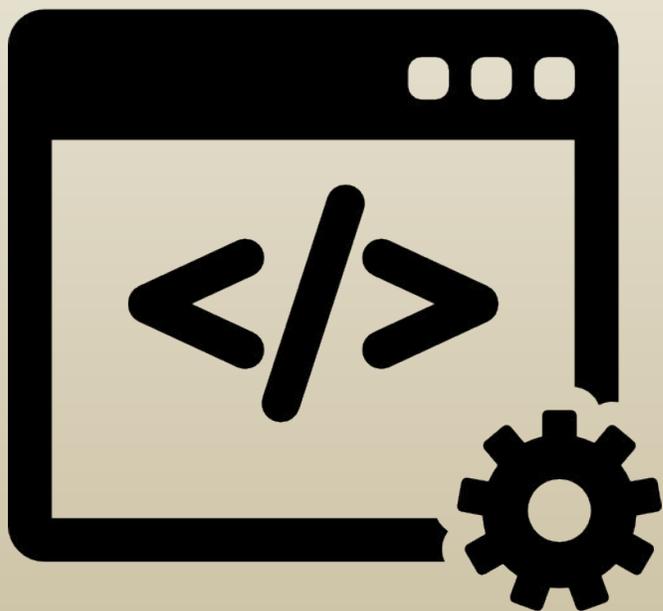
# Хакатон DevTime 4 in 1



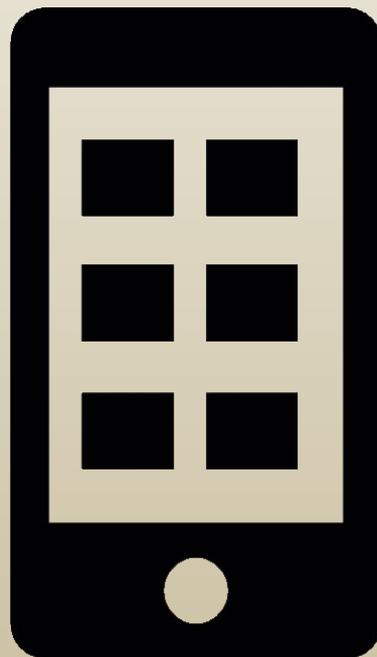
ИНТЕГРАЦИЯ 1С

# Сегодня рассмотрим следующие возможности интеграции 1с с другими стеками технологий

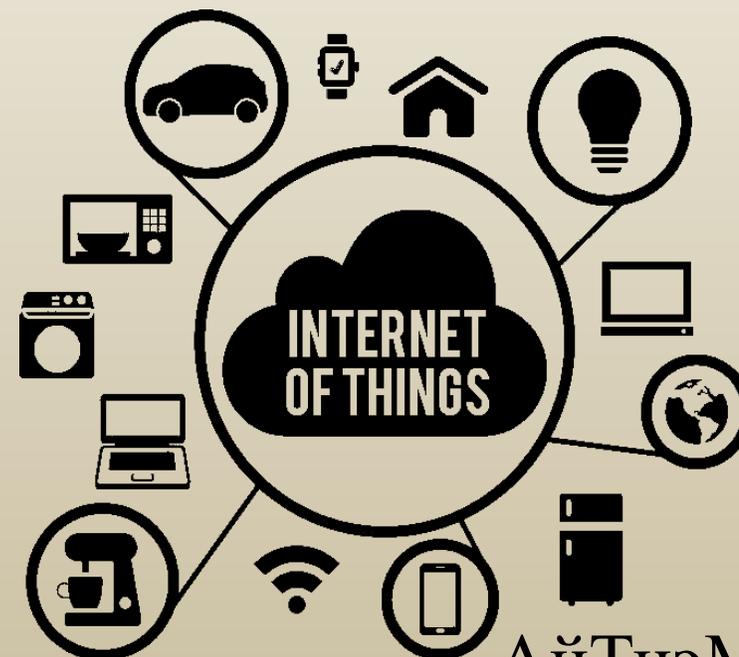
Веб разработка

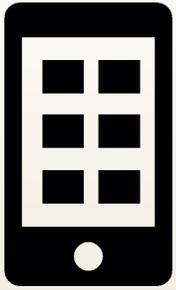


Мобильные приложения



Интернет вещей





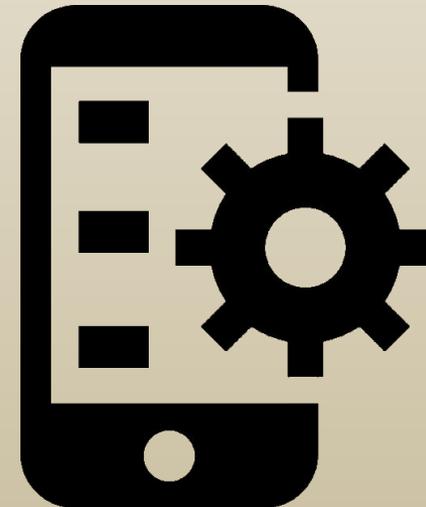
# МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ

Есть 2 варианта работы с мобильными приложениями:

Мобильное приложение 1с

# 1С

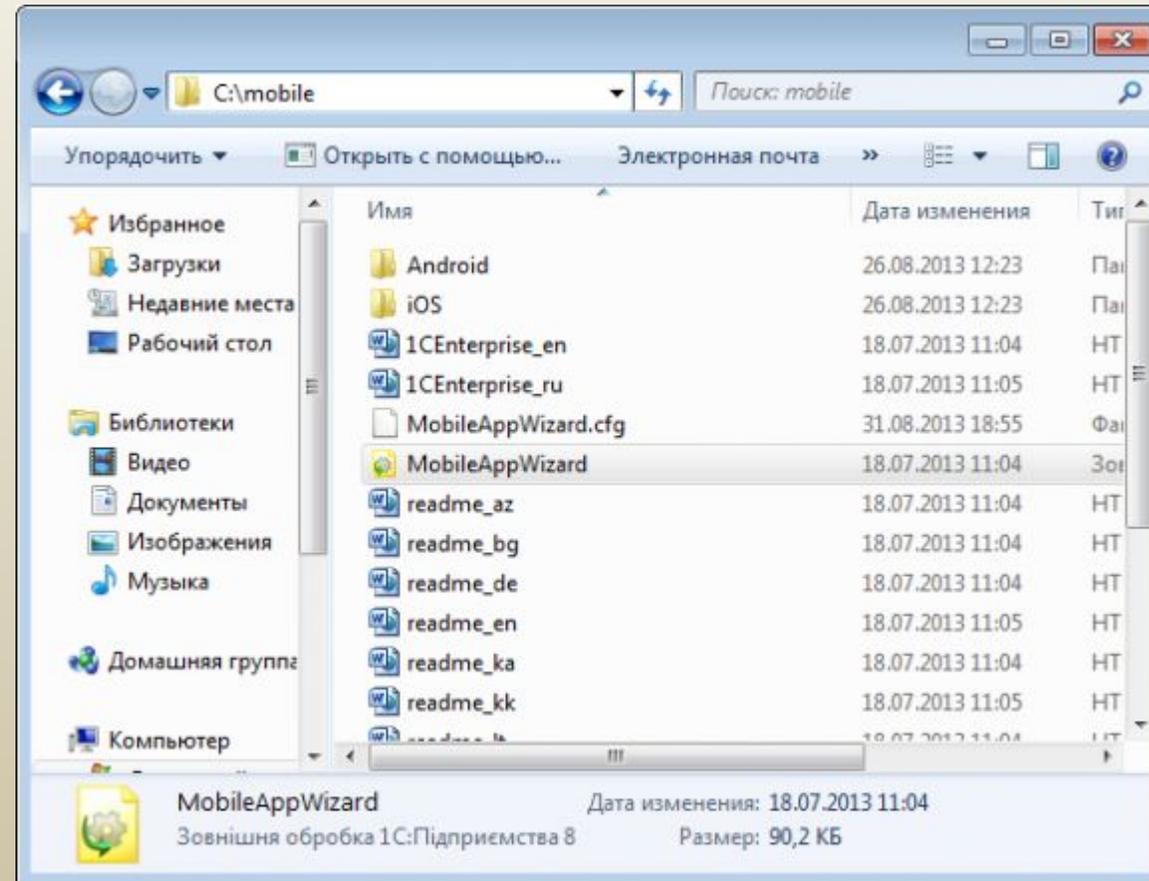
Интеграция



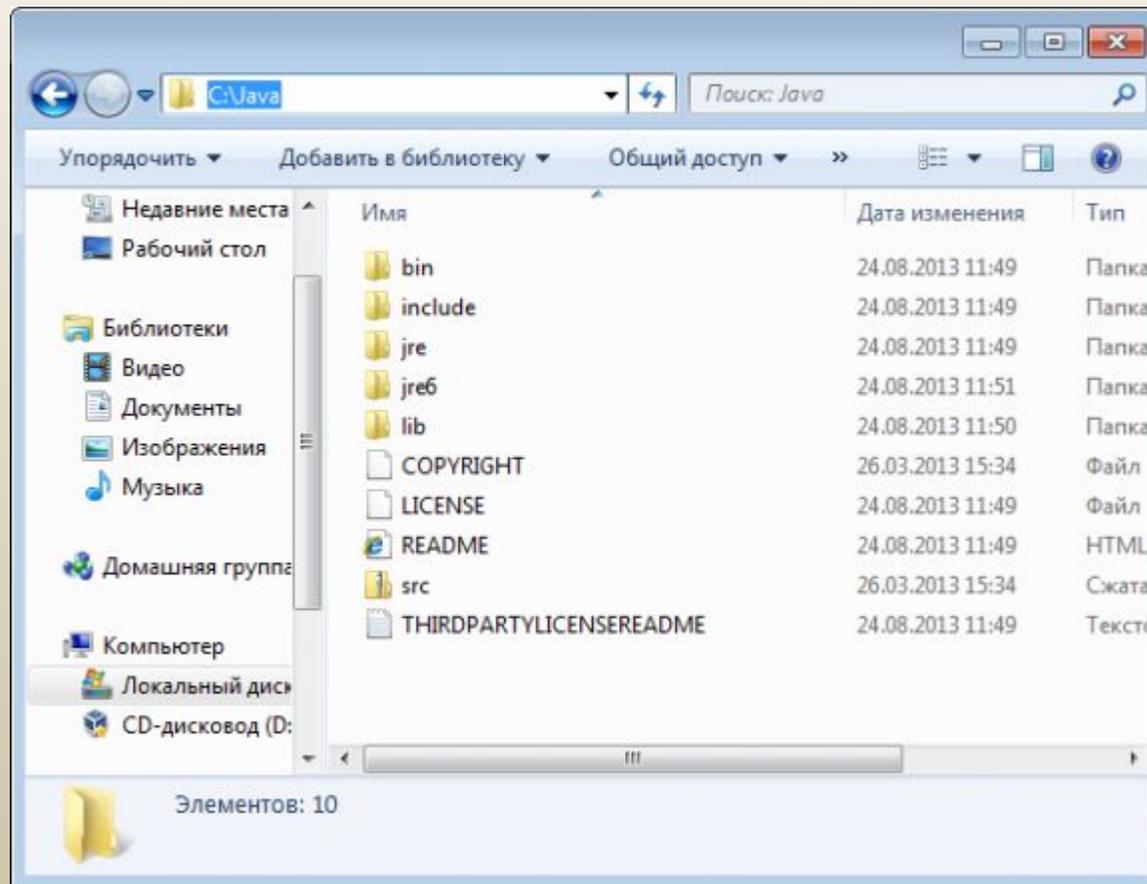
Мобильное приложение 1с

1с

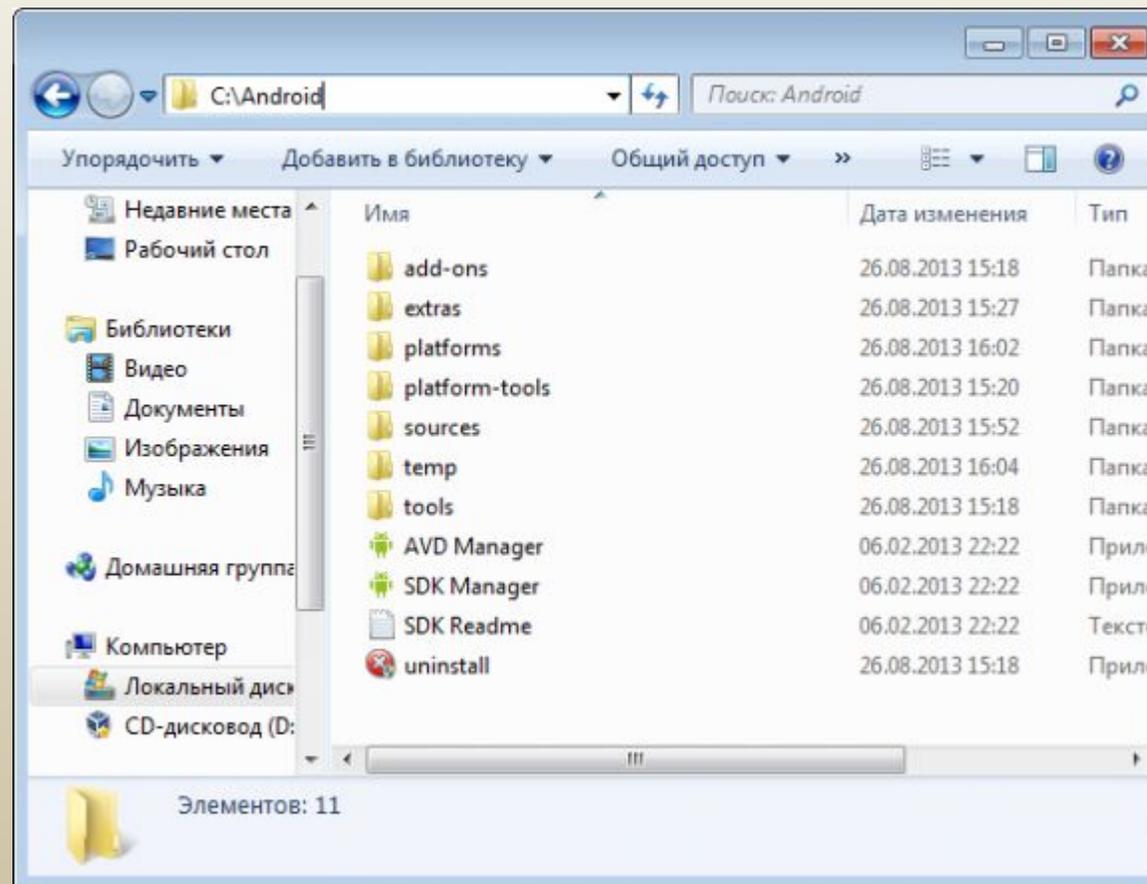
Первое, что нам нужно – технологическая платформа 1С 8.3.3.715.  
и следующий инструмент, который нам потребуется – мобильная платформа 8.3.3.35. Компания 1С предусмотрела обработку MobileAppWizard с помощью которой, мы сможем легко собирать приложения под ОС Android.



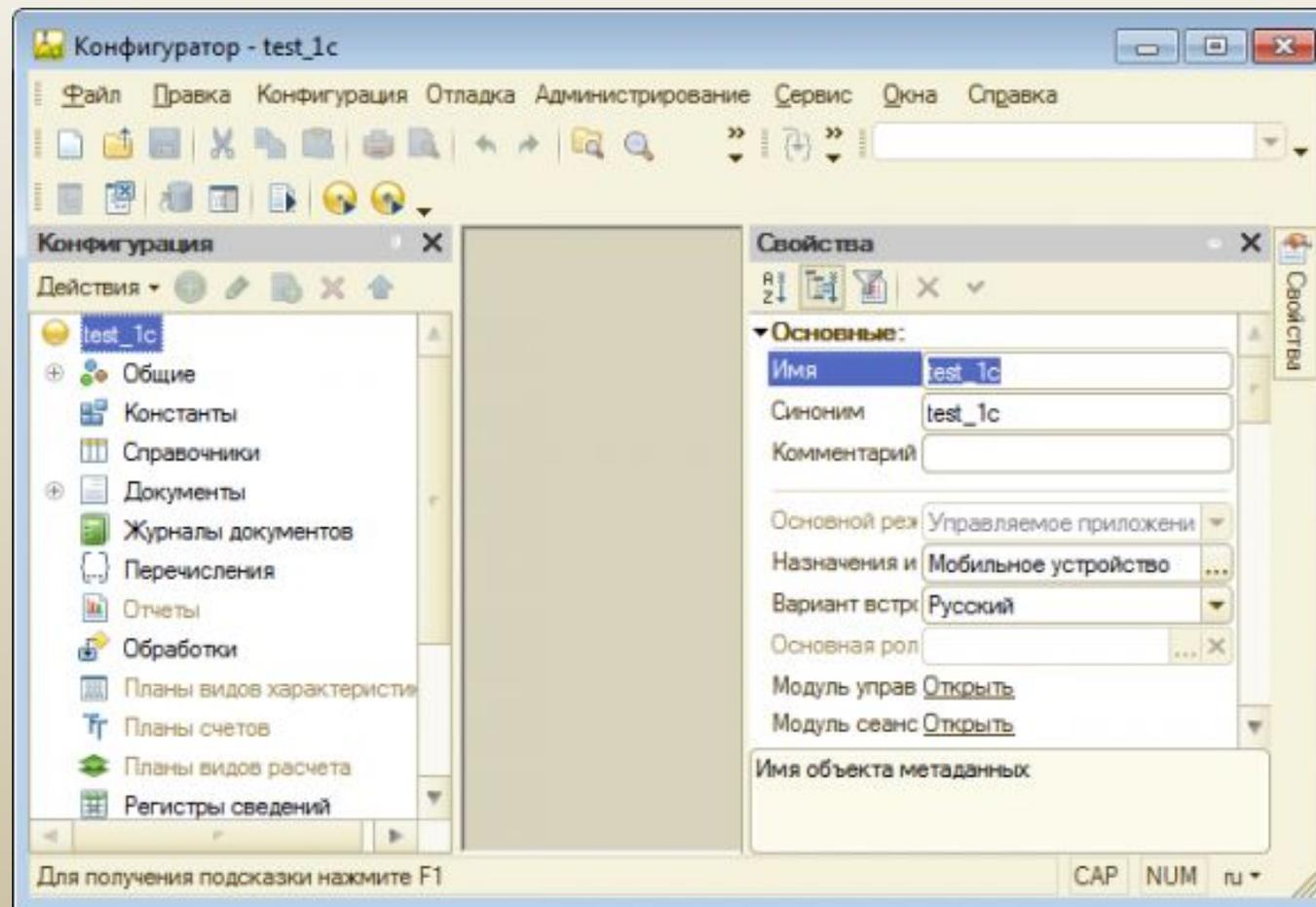
Следующий шаг – установка Java SDK, хочу сразу предупредить читателей блога, мобильная платформа 8.3.3.35 работает только из версией SDK не выше 1.6. Устанавливать рекомендуется в корневой каталог компьютера.



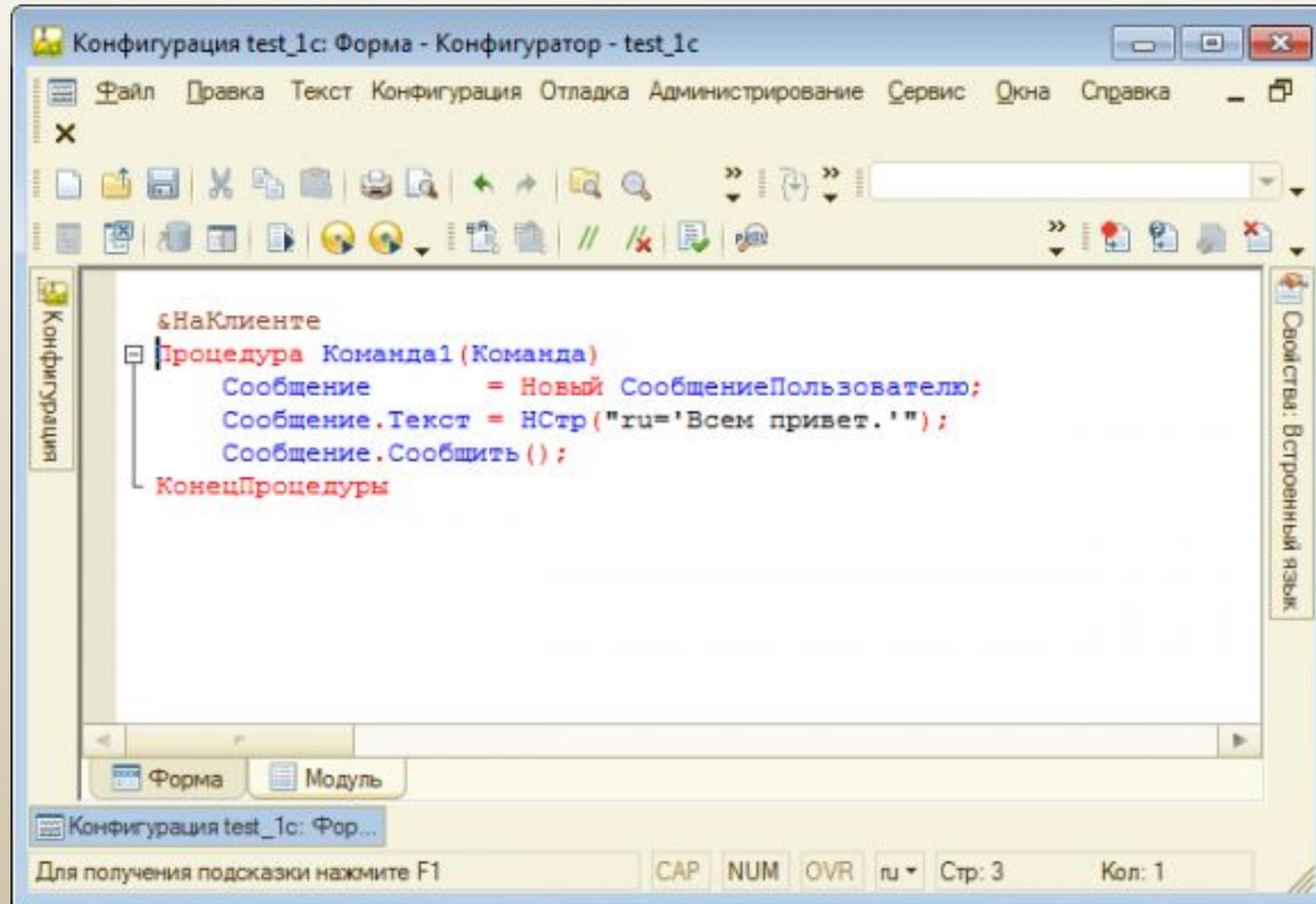
И последний шаг – установка Android SDK. Мобильная платформа 8.3.3.35 работает из Android SDK Tools версии 16.0. Если вы установите версию больше 16.0 – приложение придется собирать вручную. Устанавливать приложение нужно в корневой каталог



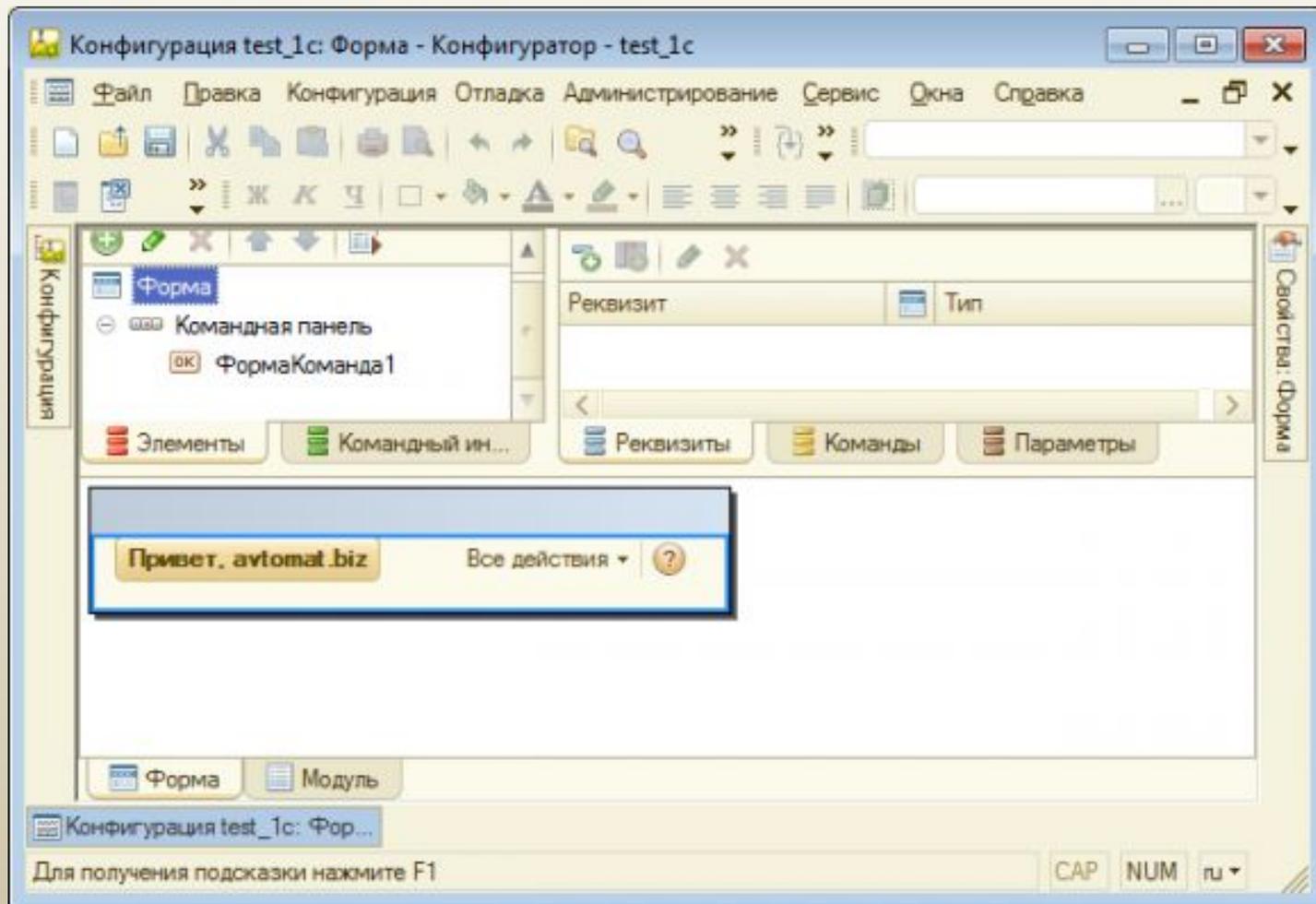
Запустим 1С, создадим пустую конфигурацию. Свойство «Назначение использования» необходимо установить «Мобильное устройство».



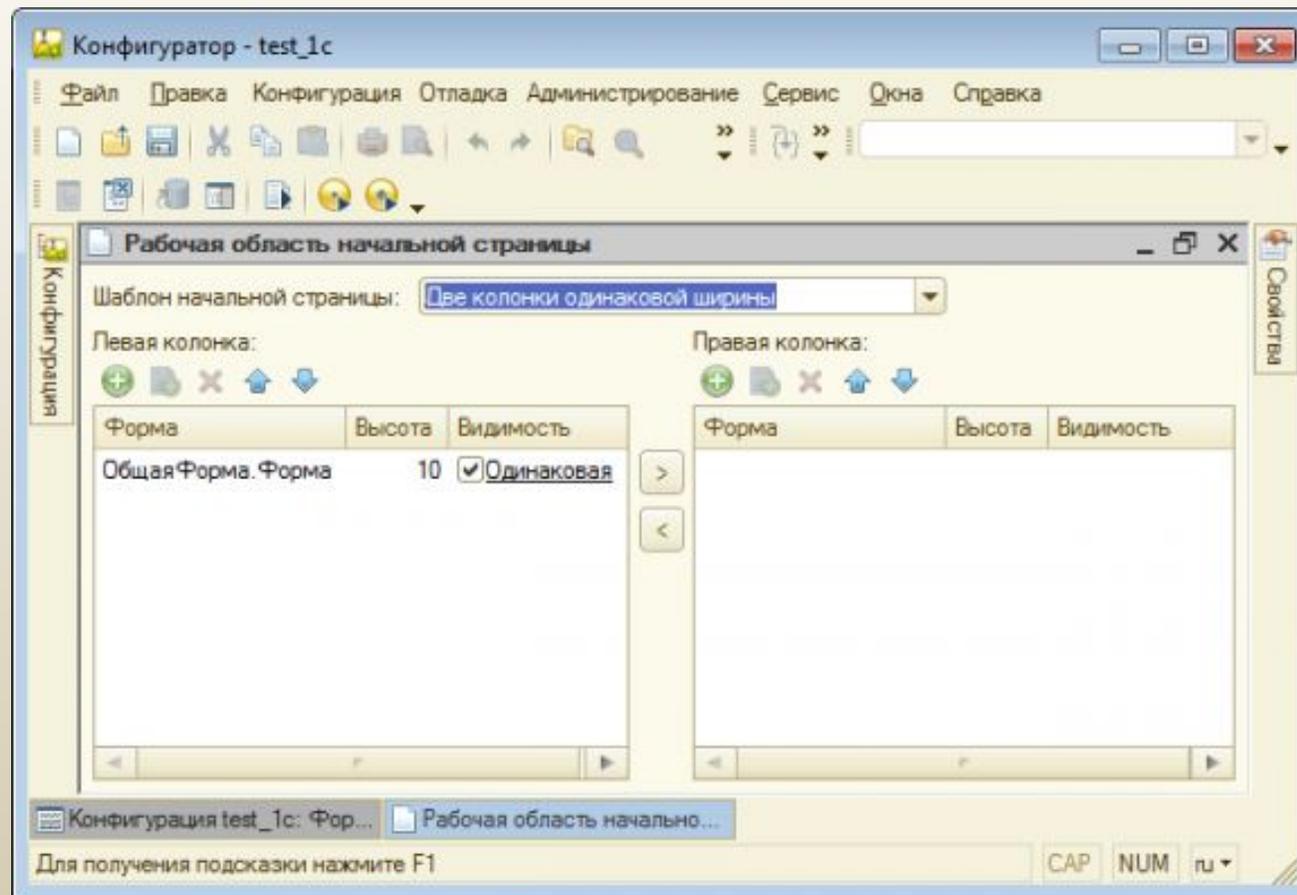
Создадим общую форму. Добавим команду формы которая выводит текст «Привет всем»



Выведем команду на командную панель формы.



Выведем форму на рабочую область начальной страницы.



Далее необходимо обновить информационную базу. После обновления необходимо записать мобильное приложение в файл. Выполним команду «Конфигурация» - «Мобильное приложение» - «Записать в файл». Полученный XML-файл сохраним на диск в любой удобный каталог.

Запустим обработку MobileAppWizard в режиме 1С Предприятия. Для начала нам необходимо сгенерировать ключ для подписи нашего предложения. Укажем файл ключа, псевдоним и пароль. Если мы ключ генерируем, необходимо также указать ФИО, Организацию, Подразделение, Страну, Город и Область.

Настройки (1С:Предприятие)

### Настройки

Параметры размещения | Параметры ключа для ОС Android

Файл ключа: C:\key\key.key

Пароль ключа: 111111

Псевдоним ключа: test

Хеш ключа (SHA1): 27:25:A4:7B:1D:3F:63:61:BF:39:8D:E6:05:D9:0A:F9:ED:4...

Параметры нового ключа

ФИО:

Организация:

Подразделение:

Страна:

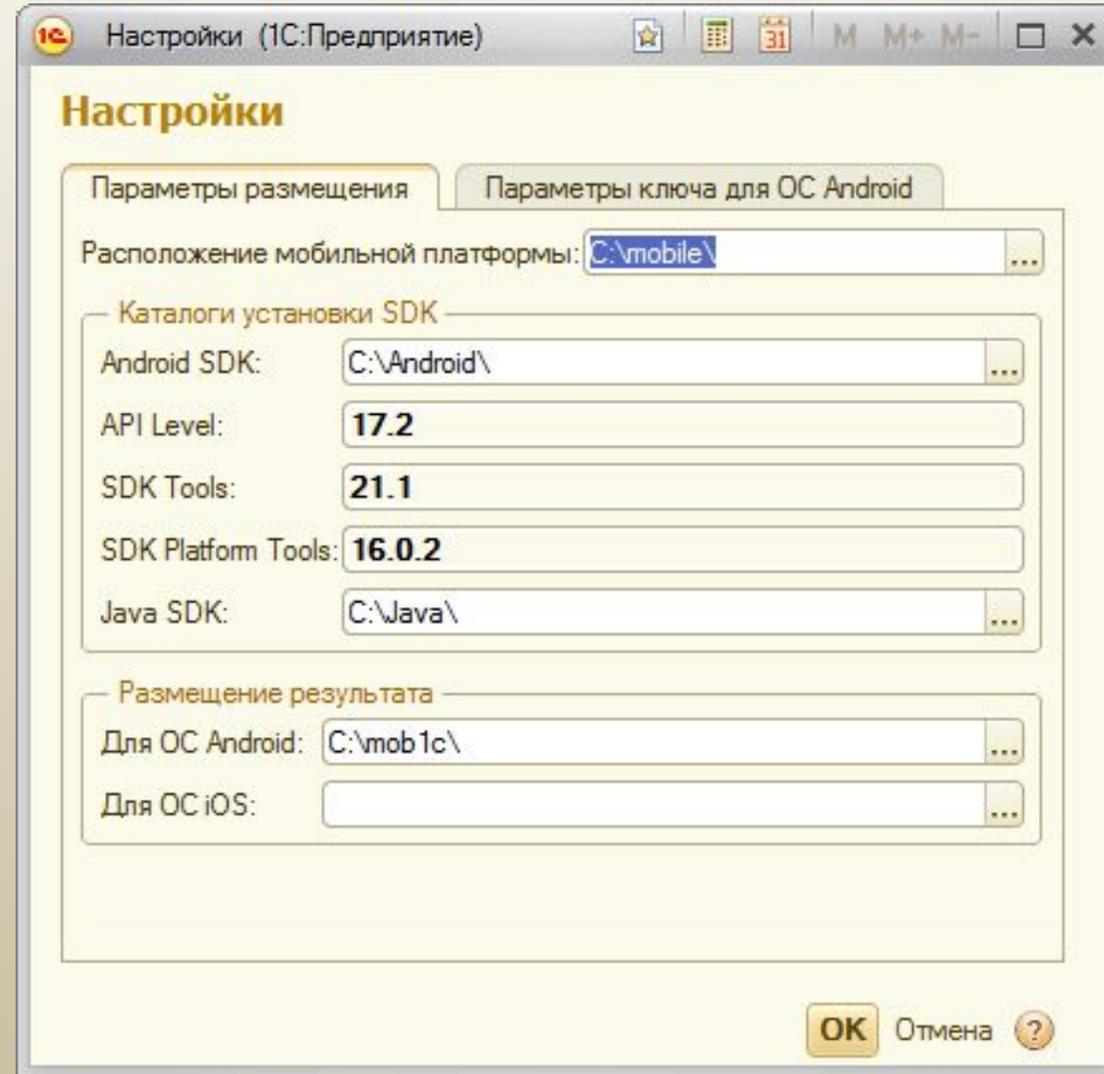
Город:

Область:

Сформировать файл ключа | Получить хеш SHA1

OK Отмена ?

В параметрах размещения необходимо заполнить настройки согласно подсказкам программы, окно выглядит примерно следующим образом.



Сохраним настройки и перейдем непосредственно к компиляции приложения. Заполним реквизит «Конфигурация мобильного приложения». Должно получиться примерно так, как показано на рисунку ниже.

Помощник созд... (1С:Предприятие)

### Помощник создания мобильного приложения \*

Создать Прочитать Записать Настройки Все действия ?

Параметры мобильного приложения

Создавать для Android:

Создавать для iOS:

Представление приложения: test\_1c

Язык по умолчанию: Русский (ru)

Идентификатор приложения: com.e1c.test\_1c

Параметр получения ключа для работы с Google Maps: 27:25:A4:7B:1D:3F:63:61:BF:39:8D:E

Ключ для работы с Google Maps: |

Версия приложения: 1 Номер сборки: 5

[Задать иконки и логотипы...](#)

Конфигурация мобильного приложения: C:\mob1c\1cema.xml

[Использовать несколько конфигураций](#)

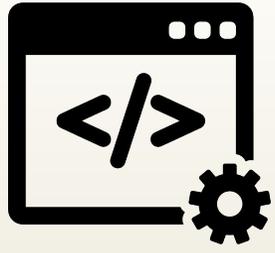
C:\mobile\MobileAppWizard.cfg

Все подготовительные настройки сделаны, осталось собрать приложение (нажать кнопку «Создать»).

Если Вы все настроили правильно файл с именем com.e1c.test\_1c.apk должен появиться в каталоге, указанном в разделе «Размещение результата». Нам осталось только установить программу на телефон, например с помощью утилиты ipk.

скриншоты приложения с телефона



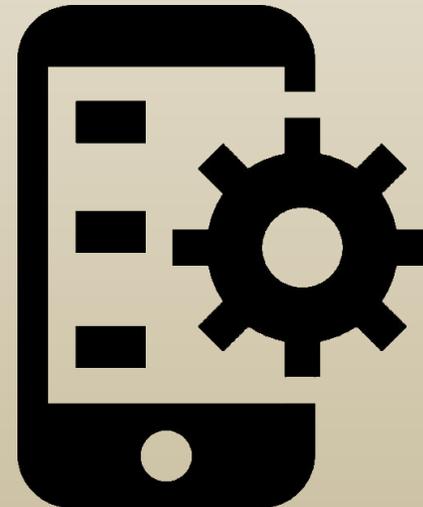


Веб разработка

web-сервер 1с

1С

Интеграция



web-сервер 1с

1с

Необходимо скачать openserver. Заходим на официальный сайт [open-server.ru](http://open-server.ru)



The screenshot shows the homepage of the Open Server website. At the top left is the Open Server logo, which includes a terminal window icon and the text "OPEN SERVER" in large blue letters, with the tagline "ЛУЧШИЙ ИНСТРУМЕНТ ДЛЯ РАЗРАБОТКИ ПОД WINDOWS" below it. To the right of the logo is a list of three features, each with a checkmark: "Простое и понятное управление", "Уникальные возможности по настройке", and "Open Server скачали уже 787 471 раз!". Below the logo and features is a blue navigation bar with six buttons: "Главная", "Новости", "Скачать", "Мануал", "Форум", and "Спасибо". The "Скачать" button is highlighted with a red underline. Below the navigation bar is a section titled "Что это такое?". To the left of the text is a graphic of an Open Server software box with a globe and people icons. The text describes Open Server as a portable server platform and programming environment designed for web developers, highlighting its rich set of server software, user-friendly interface, and powerful configuration capabilities.

**OPEN SERVER**  
ЛУЧШИЙ ИНСТРУМЕНТ ДЛЯ РАЗРАБОТКИ ПОД WINDOWS

- ✓ Простое и понятное управление
- ✓ Уникальные возможности по настройке
- ✓ Open Server скачали уже **787 471** раз!

Главная    Новости    **Скачать**    Мануал    Форум    Спасибо

### Что это такое?

**Open Server** — это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а так же для предоставления веб-сервисов в локальных сетях.

Нажимаем Скачать. На загрузившейся странице Выбираем версию Basic. Вводим числовой код с картинки и нажимаем кнопку Скачать.

## Скачать Open Server 5.2.3

Пожалуйста, введите код с картинки и выберите дистрибутив:

**2**

Код с картинки...

**1**

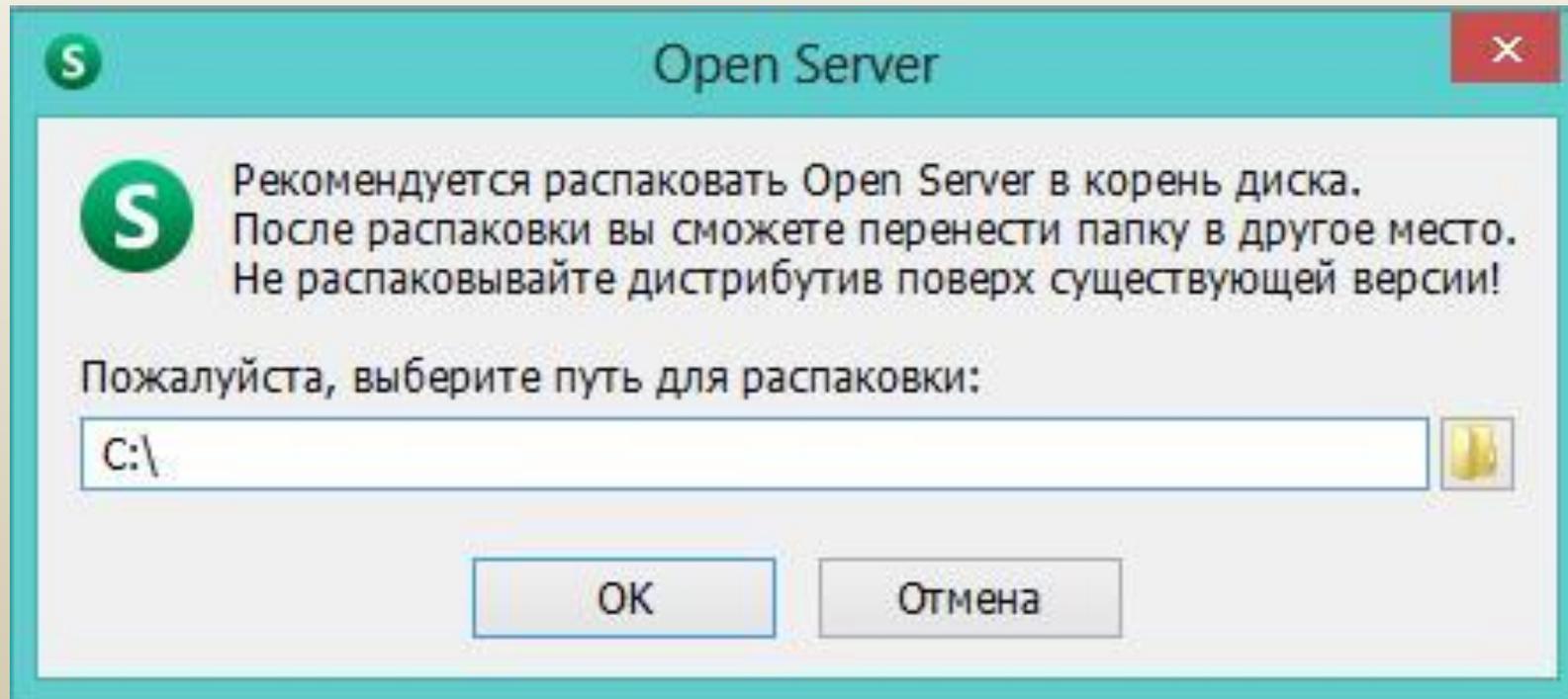
- Ultimate версия - 543 МБ
- Ultimate версия - 543 МБ
- Premium версия - 268 МБ
- Basic версия - 134 МБ**

**2**

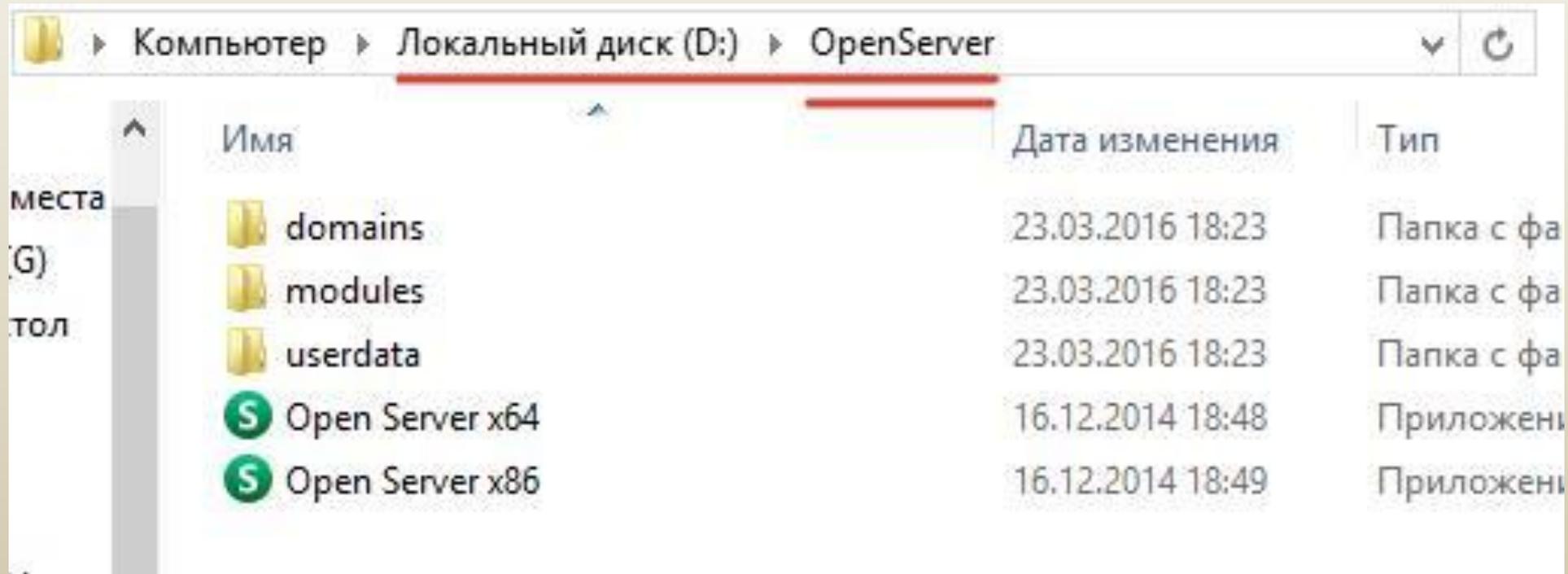
↓ Скачать

Не распаковывайте файлы этой версии!

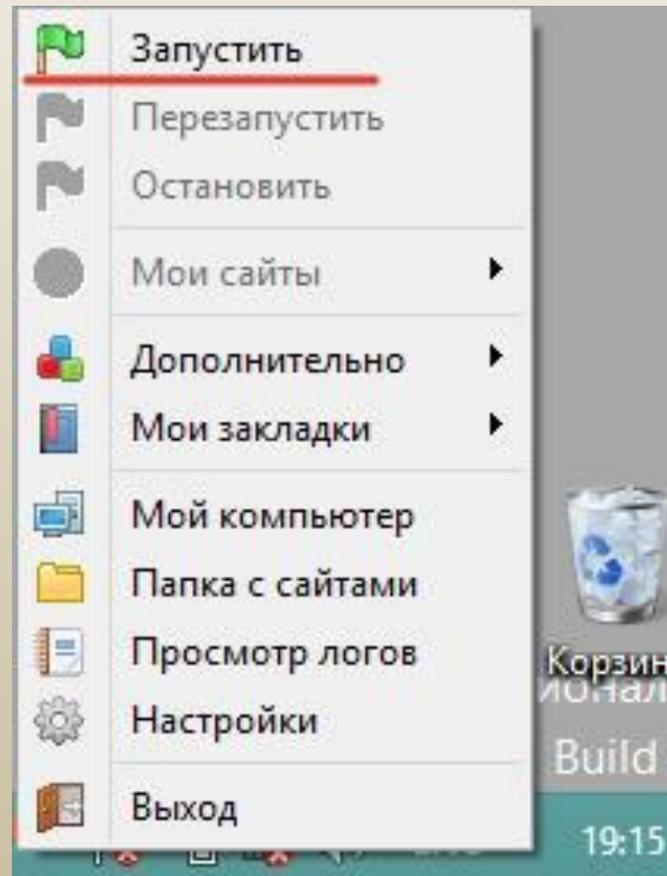
Двойной клик по файлу. Откроется окошко с предложением выбрать место, куда распаковать файлы из архива.



Заходим в папку, которую выбрали на 3-ем шаге.



После установки повторно нажимаем на Open Server x64

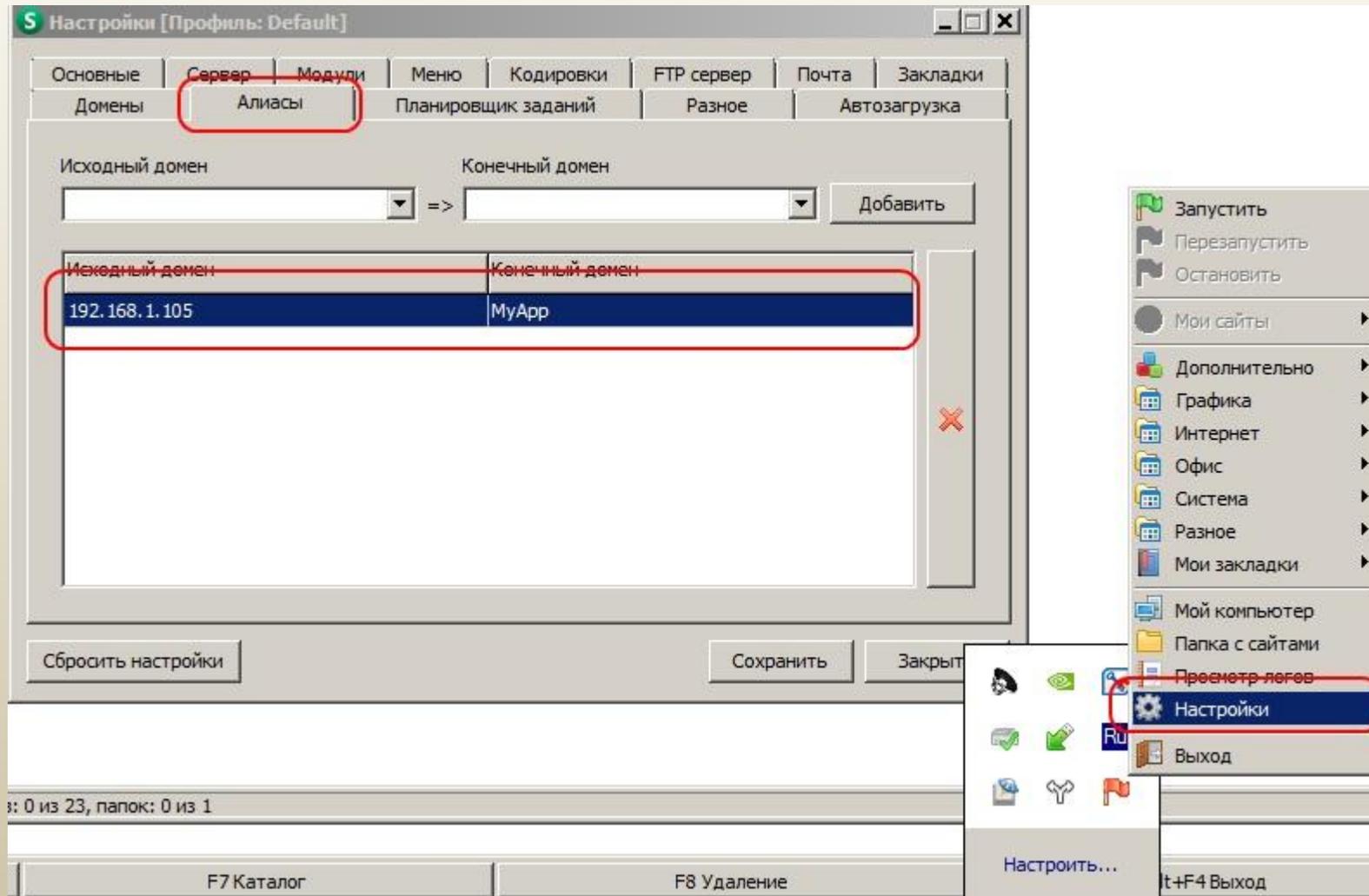


Для того чтобы 1С увидела веб-сервер Open Server-а нужно после инсталляции Open Server-а зарегистрировать службу веб-сервера из командной строки.

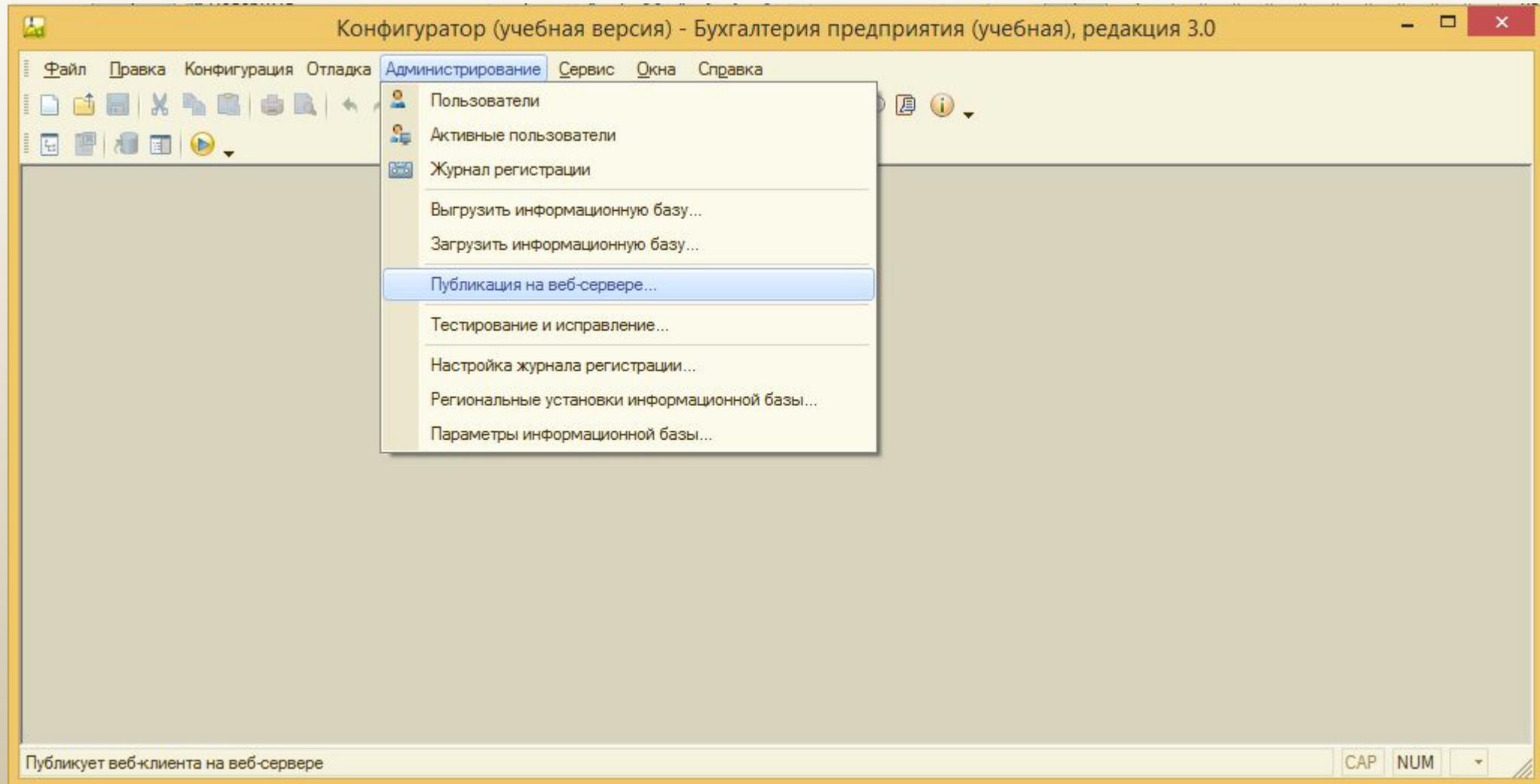
Важно! Open Server перед этим необходимо остановить

```
C:\OpenServer\modules\http\Apache-2.2\bin\httpd.exe -k install
```

Теперь рассмотрим, как настроить Open Server так, чтобы к веб-серверу можно было подключиться на другом устройстве по IP-адресу



Откроем конфигурацию нашей базы и выбираем пункт «Администрирование-Публикация на веб-сервере»



Заполняем поля. Указываем имя базы, выбираем веб-сервер и выбираем каталог

Публикация на веб-сервере

Основные Прочие

Имя: localhost

Веб-сервер: Apache 2.2

Каталог: C:\OpenServer\OpenServer\domains\localhost\

Публиковать тонкий клиент и веб-клиент

Публиковать стандартный интерфейс OData

Web-сервисы HTTP сервисы

Публиковать Web-сервисы по умолчанию

Публиковать Web-сервисы:

	Имя	Адрес
<input checked="" type="checkbox"/>	Exchange	exchange.1cws
<input checked="" type="checkbox"/>	Exchange_2_0_1_6	exchange_2_0_1_6.1cws
<input checked="" type="checkbox"/>	InterfaceVersion	InterfaceVersion.1cws
<input checked="" type="checkbox"/>	ManagedApplication_1_0_0_1	ManagedApplication_1_0_0_1...
<input checked="" type="checkbox"/>	MessageExchange	messageexchange.1cws
<input checked="" type="checkbox"/>	MessageExchange_2_0_1_6	messageexchange_2_0_1_6.1...
<input checked="" type="checkbox"/>	RemoteAdministrationOfExchange	RemoteAdministrationOfExcha...
<input checked="" type="checkbox"/>	RemoteAdministrationOfExchang...	RemoteAdministrationOfExcha...
<input checked="" type="checkbox"/>	RemoteAdministrationOfExchang...	RemoteAdministrationOfExcha...

Публиковать дистрибутив

Расположение публикуемого дистрибутива:

Опубликовать

Отключить

Сохранить

Загрузить

Закреть

Справка

# Результат

ИТ управление 2012.2.1 / Администратор (1С:Предприятие)

Рабочий стол | Складской учет | Заявки поставщикам | Заявки пользователей | Ремонт и обслуживание | Ответственные сотрудники | Бюджетирование | Нормативно-справочная информация | Администрирование

### Места хранения

Создать | Найти... | Все действия

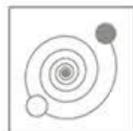
Наименование	Код	Ном
Места хранения		
Бухгалтерия	12	
ИТ отдел	6	
Бухгалтерия	2	
Бух 1	3	
Общий принтер	8	
Место ремонта	4	
Основной склад	1	
Сетевое оборудование	9	
Маркетинг	14	
Отдел маркетинга	7	
Тех. отдел	15	

Номенклатура | Отборы и свойства | Схемы расположения | Схемы сетей

Изменение состояния/МОЛ | Все действия

№	Номенклатура	Состояние	Инвентарный	Количество	Сумма
1	Mitsumi classic			1,000	500,00
2	Корпус станд			1,000	
1	Intel(R) Core(TM			1,000	
2	Gigabyte Techr			1,000	
3	DDR3 SDRAM			4,000	
4	OCZ Technolog			1,000	
5	Western Digital			1,000	
6	PS/2-совместн			1,000	
7	Mitsumi classic			1,000	
8	Western Digital			1,000	

История...



**ИТ-Терминал**

Продажа, внедрение и сопровождение программ фирмы «1С»  
г. Пермь М. Горького 65а, (342) 246-91-13, [www.it-terminal.ru](http://www.it-terminal.ru)

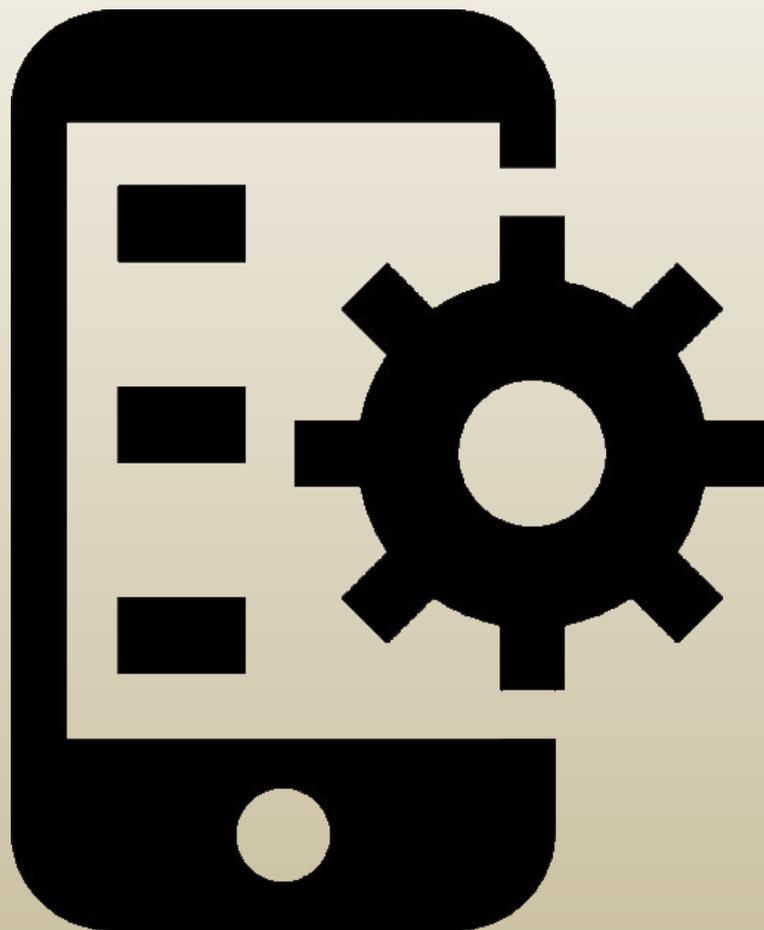


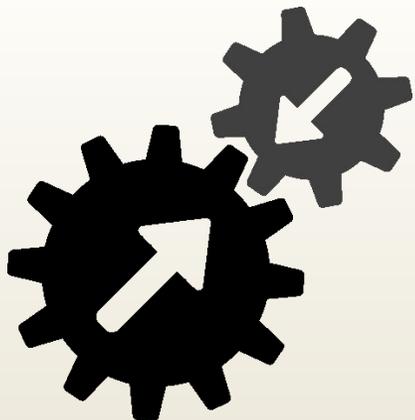
## **УСТАНОВКА И НАСТРОЙКА АРАСНЕ 2.2 И ПУБЛИКАЦИЯ БАЗЫ 1С НА WEB-СЕРВЕР**

---

1. |

# Интеграция





## Прочие возможности интеграции

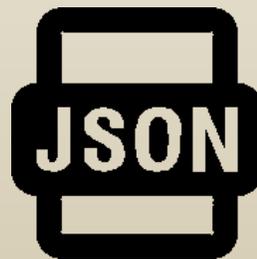
HTTP сервисы



XML



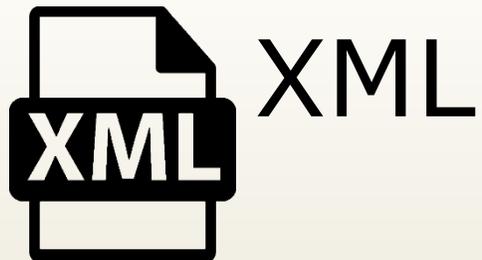
JSON



REST интерфейс

**REST API**

И это еще не все



XML

Выгрузка объектов в строку xml

**Функция** ВXML(ДокументСсылка) **Экспорт**

ЗаписьXML = **Новый** ЗаписьXML;

ЗаписьXML.УстановитьСтроку();

ЗаписатьXML(ЗаписьXML, ДокументСсылка.

ПолучитьОбъект());

**Возврат** ЗаписьXML.Закреть();

**КонецФункции**

Загрузка объектов из строки xml

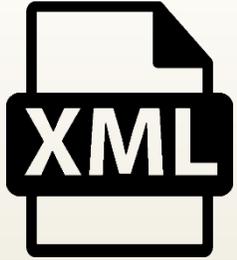
**Функция** ИзXML(Стр) **Экспорт**

ЧтениеXML = **Новый** ЧтениеXML;

ЧтениеXML.УстановитьСтроку(Стр);

**Возврат** ПрочитатьXML(ЧтениеXML);

**КонецФункции**



# XML

Рассмотрим на примере выгрузки реестра начислений в Сбербанк на основании ведомости в конфигурации «Бухгалтерия Предприятия 8.2»

1. Все начинается с инициализации класса «ЗаписьXML», предназначенного для организации последовательной записи документов и фрагментов XML.

```
ЗаписьXML = Новый ЗаписьXML;
```

```
ЗаписьXML.ОткрытьФайл(Путь+«\»+«1548z.xml», «windows-1251»); //Открываем файл из переменной  
«Путь», указанной на форме обработки
```

```
ЗаписьXML.ЗаписатьОбъявлениеXML(); // Так как мы создаем документ-XML ,а не фрагмент, то  
необходимо в начало его поместить объявление
```

2. Начинаем заполнять XML-документ. Указанная последовательность записей сделана исходя из образца файла для Клиент-Банка от Сбербанка. Ваша же последовательность будет отличаться, но принцип один и тот же.

```
ЗаписьXML.ЗаписатьНачалоЭлемента(«СчетаПК»); //записываем первый элемент
  ЗаписьXML.ЗаписатьАтрибут(«ДатаФормирования», Формат(ДатаПлатежка,
«ДФ=уууу-ММ-dd»)); //указываем все атрибуты первого элемента в документе
  ЗаписьXML.ЗаписатьАтрибут(«НомерДоговора», НомерДоговора);
  ЗаписьXML.ЗаписатьАтрибут(«НаименованиеОрганизации», Организация.Наименование);
  ЗаписьXML.ЗаписатьАтрибут(«ИНН», Организация.ИНН+«\«+Организация.КПП);
  ЗаписьXML.ЗаписатьАтрибут(«РасчетныйСчетОрганизации», РасчетныйСчет);

  ЗаписьXML.ЗаписатьНачалоЭлемента(«ЗачислениеЗарплаты»); //записываем следующий элемент
....И.Т.Д.
...
```

3. Пишем небольшой запрос для выборки данных

```
ЗапросВыплата = Новый Запрос;  
    ЗапросВыплата.УстановитьПараметр(«ТекущийДокумент», Ведомость);
```

```
    ЗапросВыплата.Текст =
```

```
«ВЫБРАТЬ
```

```
| ЛицевыеСчета.Банк,  
| ЛицевыеСчета.ФизЛицо КАК Сотрудник,  
| ЛицевыеСчета.НомерЛицевогоСчета,  
| ЛицевыеСчета.Организация
```

```
|ПОМЕСТИТЬ Лицевые
```

```
|ИЗ
```

```
| РегистрСведений.ЛицевыеСчетаРаботниковОрганизации КАК ЛицевыеСчета
```

```
;
```

(Продолжение на следующем слайде)

```
|  
|/////////////////////////////////////////////////////////////////  
|ВЫБРАТЬ  
| Зарплата.Сумма + Зарплата.КомпенсацияЗаЗадержкуЗарплаты КАК Сумма,  
| Лицевые.НомерЛицевогоСчета,  
| Лицевые.Организация,  
| Зарплата.Физлицо  
|ИЗ  
| Документ.ЗарплатаКВыплатеОрганизаций.Зарплата КАК Зарплата  
| ЛЕВОЕ СОЕДИНЕНИЕ Лицевые КАК Лицевые  
| ПО Зарплата.Физлицо = Лицевые.Сотрудник  
| И Зарплата.Ссылка.Организация = Лицевые.Организация  
|ГДЕ  
| Зарплата.Ссылка = &ТекущийДокумент»;  
ТаблицаВыплата = ЗапросВыплата.Выполнить().Выгрузить();  
ТаблицаВыплата.Сортировать(«Физлицо возр»);
```

4.Перебираем таблицу значений, полученную из результата запроса

Для Каждого Стр Из ТаблицаВыплата Цикл

счетчик=счетчик+1;

ПорядковыйНомер = строка(счетчик);

Сотрудник = Стр.Физлицо;

НомерЛицевогоСчета = Стр.НомерЛицевогоСчета;

Сумма = Стр.Сумма;

... И.Т.Д

5. В цикле, описанном выше заполняем новые элементы и атрибуты

ЗаписьXML.ЗаписатьНачалоЭлемента(«Сотрудник»);

ЗаписьXML.ЗаписатьАтрибут(«Нпп», строка(счетчик));

...

ЗаписьXML.ЗаписатьНачалоЭлемента(«ОтделениеБанка»);

ЗаписьXML.ЗаписатьТекст(ОтделениеБанка);

ЗаписьXML.ЗаписатьКонецЭлемента();

ЗаписьXML.ЗаписатьНачалоЭлемента(«ЛицевойСчет»);

ЗаписьXML.ЗаписатьТекст(НомерЛицевогоСчета);

ЗаписьXML.ЗаписатьКонецЭлемента();

ЗаписьXML.ЗаписатьНачалоЭлемента(«Сумма»);

ЗаписьXML.ЗаписатьКонецЭлемента();

...

ЗаписьXML.ЗаписатьКонецЭлемента();

6. После всех записей необходимо закрыть XML файл

ЗаписьXML.Закреть();

Сообщить(«Создан файл «+Путь+«\»+ «1548z.xml»);

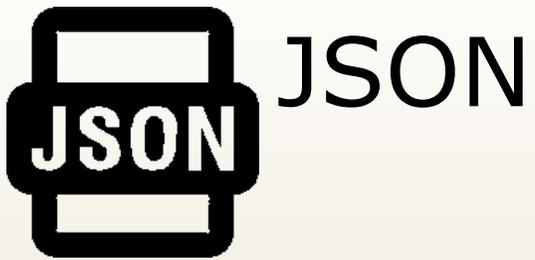
## Результат

```
<?xml version="1.0" encoding="windows-1251" ?>
<СчетаПК ДатаФормирования="2016-01-14" НомерДоговора="65432" ДатаДоговора="2015-01-12"
НаименованиеОрганизации="Общество с ограниченной ответственностью 'Восход'" ИНН="61258795003"
РасчетныйСчетОрганизации="4210649946421000006" БИК="004568451" ИдПервичногоДокумента="6f817478-
b9ca-11e5-be3d-94de805e6f02">
<ЗачислениеЗарплаты>
<Сотрудник Нпп="1">
<Фамилия>Иванов</Фамилия>
<Имя>Дмитрий</Имя>
<Отчество>Владимирович</Отчество>
<ОтделениеБанка>5221</ОтделениеБанка>
<ФилиалОтделенияБанка>7773</ФилиалОтделенияБанка>
<ЛицевойСчет>408.....802</ЛицевойСчет>
<Сумма>60900.00</Сумма>
</Сотрудник>
<Сотрудник Нпп="2">
<Фамилия>Петров</Фамилия>
<Имя>Иван</Имя>
<Отчество>Минеханович</Отчество>
<ОтделениеБанка>5221</ОтделениеБанка>
<ФилиалОтделенияБанка>7773</ФилиалОтделенияБанка>
<ЛицевойСчет>56468...41654</ЛицевойСчет>
<Сумма>10066.00</Сумма>
</Сотрудник>
<Сотрудник Нпп="3">
<Фамилия>Гончаров</Фамилия>
<Имя>Дмитрий</Имя>
<Отчество>Олегович</Отчество>
<ОтделениеБанка>5221</ОтделениеБанка>
<ФилиалОтделенияБанка>7773</ФилиалОтделенияБанка>
<ЛицевойСчет>750688....49845</ЛицевойСчет>
<Сумма>33833.89</Сумма>
</Сотрудник>
```

АйТиЭМ

Внедренческая компания





## Выгрузка данных в строку JSON

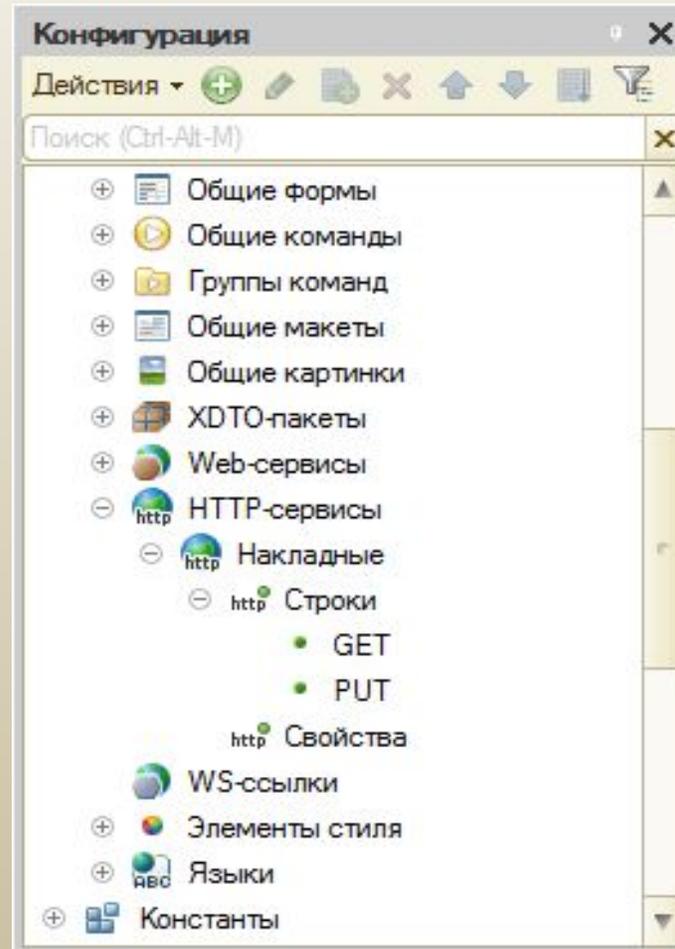
**Функция** ВJSON(СтруктураДанных) **Экспорт**  
Ответ = **Новый** ЗаписьJSON ;  
Ответ.ПроверитьСтруктуру = **Ложь** ;  
ПараметрыЗаписи = **Новый**  
ПараметрыЗаписиJSON( , Символы.Таб) ;  
Ответ.УстановитьСтроку(ПараметрыЗаписи) ;  
ЗаписатьJSON(Ответ, СтруктураДанных) ;  
**Возврат** Ответ.Закреть() ;  
**КонецФункции**

## Выгрузка данных в строку JSON

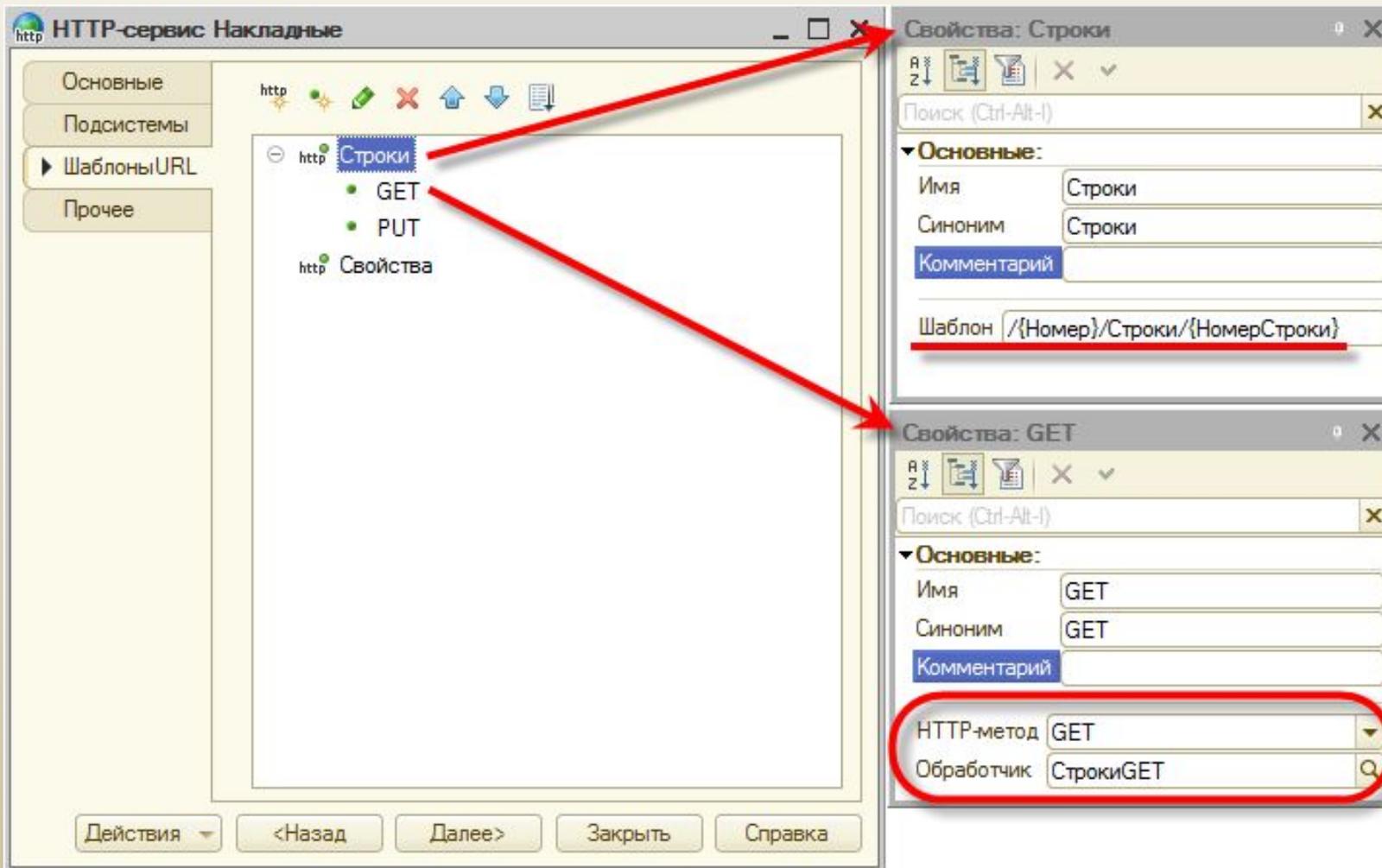
**Функция** ИзJSON(СтрокаJSON) **Экспорт**  
ТекТоварыJSON = **Новый** ЧтениеJSON ;  
ТекТоварыJSON.УстановитьСтроку(СтрокаJSON) ;  
**Возврат** ПрочитатьJSON(ТекТоварыJSON) ;  
**КонецФункции**

# HTTP сервисы

По своему «конструктивному исполнению» HTTP-сервисы очень напоминают web-сервисы, имеющиеся в платформе. Точно так же есть специальный объект конфигурации **HTTP сервис**. Такие объекты добавляются в ветку **Общие - HTTP-сервисы**:



Каждый HTTP-сервис может содержать в себе один или несколько шаблонов. Для каждого шаблона можно создать один или несколько методов, выполняющих обработку данных:



Шаблон задаёт путь, по которому может происходить обращение к HTTP-сервису. В шаблоне можно использовать определённый набор символов, в том числе параметризованные сегменты вида **{какой-то текст}**.

Для каждого метода указывается, во-первых, обрабатываемый HTTP метод, а также создаётся процедура на встроенном языке, которая и будет выполнять обработку данных. Также можно указать, что будет обрабатываться не какой-то конкретный, а любой HTTP-метод из доступных.

При обращении к такому HTTP-сервису платформа сначала попытается сопоставить URL, по которому произошло обращение, с одним из имеющихся шаблонов и методов. Если сопоставить не удалось, то платформа выдаст код ответа **404 Not Found**. Если подходящий метод будет найден, то платформа начнёт выполнение его обработчика, передав в него все имеющиеся в запросе данные в виде объекта встроенного языка **HTTPСервисЗапрос**: (следующий слайд)

Из этого объекта можно легко получить, например, параметры, содержащиеся в исходном URL, и использовать их для извлечения из базы нужных данных.

Полученные данные можно вернуть в разных форматах. Например, их можно преобразовать в XML, как на картинке выше, или даже просто в текстовую строку с разделителями.

Ответ сервиса формируется специальным объектом встроенного языка **HTTPСервисОтвет**, в тело которого можно поместить подготовленные данные.

Публикация HTTP-сервисов выполняется аналогично тому, как публикуются web-сервисы. Также аналогичным образом для них работает аутентификация, использование разделения данных и отладка.

```
HTTP-сервис Накладные: Модуль
// функция возвращает данные N-ой строки расходной накладной.
// Номер нужной накладной и номер строки содержатся в полученном HTTP запросе.
//
// Параметры:
// Запрос - HTTPСервисЗапрос
// Возвращаемое значение:
// - HTTPСервисОтвет
функция СтрокиGET(Запрос)

    // Пример запроса:
    // HTTP://test.server.ru/hs/Накладные.hs/000000012/Строки/1

    // Разобрать URL запроса.
    ЗапросНомерДокумента = Запрос.ПараметрыURL["Номер"];
    ЗапросНомерСтроки = Запрос.ПараметрыURL["НомерСтроки"];

    НужныйДокумент = Документы.РасходТовара.НайтиПоНомеру(ЗапросНомерДокумента);

    // Обработка ошибочных ситуаций: документ не найден, номер не задан.
    Если НужныйДокумент = Неопределено ИЛИ НужныйДокумент.Пустая() Тогда
        Ответ = Новый HTTPСервисОтвет(404);

        Возврат Ответ;

    КонецЕсли;

    // Вернуть данные строки.
    СтрокаДокумента = НужныйДокумент.Товары[Число(ЗапросНомерСтроки) - 1];

    // Преобразовать данные строки в XML.
    ЗаписьXML = Новый ЗаписьXML;
    ЗаписьXML.УстановитьСтроку();
    ЗаписьXML.ЗаписатьОбъявлениеXML();
    ЗаписьXML.ЗаписатьНачалоЭлемента("answer");
    ЗаписьXML.ЗаписатьНачалоЭлемента("СтрокаНакладной");
    ЗаписьXML.ЗаписатьАтрибут("Товар", СтрокаДокумента.Товар.Наименование);
    ЗаписьXML.ЗаписатьАтрибут("Цена", Строка(СтрокаДокумента.Цена));
    ЗаписьXML.ЗаписатьАтрибут("Количество", Строка(СтрокаДокумента.Количество));
    ЗаписьXML.ЗаписатьАтрибут("Сумма", Строка(СтрокаДокумента.Сумма));
    ЗаписьXML.ЗаписатьКонецЭлемента();
    ЗаписьXML.ЗаписатьКонецЭлемента();
    XMLСтрока = ЗаписьXML.Закрыть();

    // Передать строку в HTTP ответ.
    Ответ = Новый HTTPСервисОтвет(200);
    Ответ.УстановитьТелоИзСтроки(XMLСтрока);

    Возврат Ответ;

Конецфункции
```

Начиная с версии 8.3.5 платформа 1С:Предприятие может автоматически формировать REST интерфейс для всего прикладного решения. Любой объект конфигурации (справочник, документ, регистр сведений и т.д.) можно сделать доступным для получения и модификации данных через REST интерфейс. В качестве протокола доступа платформа использует протокол OData версии 3.0. Публикация сервисов OData выполняется из меню Конфигуратора «Администрирование -> Публикация на веб-сервере», флажок «Публиковать стандартный интерфейс OData» должен быть помечен. Поддерживаются форматы XML и JSON. После того, как прикладное решение опубликовано на веб-сервере, сторонние системы могут обращаться к нему через REST интерфейс с помощью HTTP запросов. Для работы с приложением 1С через протокол OData программирование на стороне 1С не требуется.

Использовать стандартный интерфейс OData прикладного решения просто:

- В конфигураторе вы публикуете REST интерфейс - флажок Опубликовать стандартный интерфейс OData;
- После этого объекты прикладного решения становятся доступны через этот интерфейс;
- Способы аутентификации OData клиентов полностью совпадают со способами, используемыми для веб-сервисов;
- OData клиенты могут запросить через HTTP документ метаданных, описывающий доступные объекты прикладного решения;
- OData клиенты выполняют операции создания, чтения, модификации и удаления данных прикладного решения.

По умолчанию после публикации объекты конфигурации не доступны. Прежде чем обращаться к ним, необходимо разрешить доступ, например с помощью типовой обработки "Настройка автоматического REST сервиса". В обработке можно задать отдельного пользователя REST сервиса и указать доступные объекты конфигурации

Так, URL вида «http://<сервер>/<конфигурация>/odata/standard.odata/Catalog\_Номенклатура» вернет нам содержимое каталога Номенклатура в формате XML — коллекцию элементов entry (заголовок сообщения пропущен для краткости)

```
<entry>
  <id>http://server/Config/odata/standard.odata/Catalog_Номенклатура(guid'35d1f6e4-289b-11e6-8ba4-e03f49b16074')</id>
  <category term="StandardODATA.Catalog_Номенклатура" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  <title type="text"/>
  <updated>2016-06-06T16:42:17</updated>
  <author/>
  <summary/>
  <link rel="edit" href="Catalog_Номенклатура(guid'35d1f6e4-289b-11e6-8ba4-e03f49b16074')" title="edit-link"/>
  <content type="application/xml">
    <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <d:Ref_Key>35d1f6e4-289b-11e6-8ba4-e03f49b16074</d:Ref_Key>
      <d:DataVersion>AAAAAgAAAA=</d:DataVersion>
      <d:DeletionMark>>false</d:DeletionMark>
      <d:Code>000000001</d:Code>
      <d:Description>Кондиционер Mitsubishi</d:Description>
      <d:Описание>Мощность 2,5 кВт, режимы работы: тепло/холод</d:Описание>
    </m:properties>
  </content>
</entry>
<entry>
  <id>http://server/Config/odata/standard.odata/Catalog_Номенклатура(guid'35d1f6e5-289b-11e6-8ba4-e03f49b16074')</id>
  <category term="StandardODATA.Catalog_Номенклатура" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"/>
  ...
```

Прибавляя к URL-у строку «?\$format=application/json», получим содержимое каталога Номенклатура в формате JSON (URL вида «[http://<сервер>/<конфигурация>/odata/standard.odata/Catalog\\_Номенклатура?\\$format=application/json](http://<сервер>/<конфигурация>/odata/standard.odata/Catalog_Номенклатура?$format=application/json)» )

```
{
  "odata.metadata": "http://server/Config/odata/standard.odata/$metadata#Catalog_Номенклатура",
  "value": [{
    "Ref_Key": "35d1f6e4-289b-11e6-8ba4-e03f49b16074",
    "DataVersion": "AAAAAgAAAA=",
    "DeletionMark": false,
    "Code": "000000001",
    "Description": "Кондиционер Mitsubishi",
    "Описание": "Мощность 2,5 кВт, режимы работы: тепло/холод"
  }, {
    "Ref_Key": "35d1f6e5-289b-11e6-8ba4-e03f49b16074",
    "DataVersion": "AAAAAwAAAA=",
    "DeletionMark": false,
    "Code": "000000002",
    "Description": "Кондиционер Daikin",
    "Описание": "Мощность 3 кВт, режимы работы: тепло/холод"
  }, ...
}
```

Спасибо за внимание!

**АйТиЭМ**  
Внедренческая компания



- ◆ Учебный центр 1С
- ◆ Продажа, внедрение, сопровождение 1С
- ◆ Автоматизация торговли: магазинов, автосервисов, производственных предприятий

- 📍 г. Кемерово, пр-т Кузнецкий, 102, оф. 215
- ☎ (3842) 44-11-98
- 📞 8 952 168 4901
- ✉ itm.office@yandex.ru
- 👤 vk.com/itm.office
- 📘 facebook.com/itm.kemerovo
- 🐦 twitter.com/itm\_kemerovo
- 🌐 itm.vk.ru