



Martin Heigl

ODATA Service with Powerbuilder

A modern ERP application needs an ODATA interface.

You will learn about experiences with a native PowerBuilder solution with examples like

- Rest Service in a WEB application and
- "Middle Tier Layer" between PowerBuilder and different database systems like MSSQL and HANA

✉ mah@boyum-it.com



About me

Powerbuilder developer since 1998

Working for boyum it alias „beas GmbH“

alias „Weber Datentechnik“ since 1998

Working on actual project „beas

Manufacturing“ since 1998





About boyum It

Existing since 2001

Since 2005 SAP Partner

Since 2016 acquires beas Group AG

Deliver AddOns for SAP Business One

152 418 daly users

7258 Customers

608 Reseller Partners

100 countries installed



About beas manufacturing

- Existing since 1997
- ~ 1000 active customers
- between 2 and 200 Clients / Customer
- up to 100 WEB Clients / Customer



Example SAP Program with beas addon

The screenshot displays the SAP user interface with three overlapping windows:

- Main Menu:** A vertical sidebar on the left containing navigation options such as Administration, Financials, CRM, Opportunities, Sales - A/R, Purchasing - A/P, Business Partners, Banking, and Inventory.
- Sales Order SAP Window:** A window titled "Sales Order" showing customer details (Customer: C001, Name: Customer 001) and a table of items. The table has columns for Item No., Variante einfach, Item Description, and Whse. Item 2 is highlighted with item number 0815 and description "<p>Complete>08FF0".
- Item master data for 0815 Beas window:** A window titled "Item master data for 0815" showing various data tabs. The "Master Data" tab is active, displaying fields for Item number (1112), Description (Import Test), Item type (Item), and Material Group (HM). The "Manufacturing data" section includes Procurement Method (Buy), Breakdown (No Breakdown), and Unit bom (Pcs).
- Material Group HM Dialog Box:** A smaller dialog box titled "Material Group HM" is open, showing details for group HM, including Additional Info (Dangerous Goods), Overhead Costs (Surcharge Marginal Cost %: 0.00, Surcharge Full Cost %: 0.00), Picture (bmp\project_blue.png), and Color (99).

Beas Manufacturing solution improve production efficiency & increase productivity

Work Orders

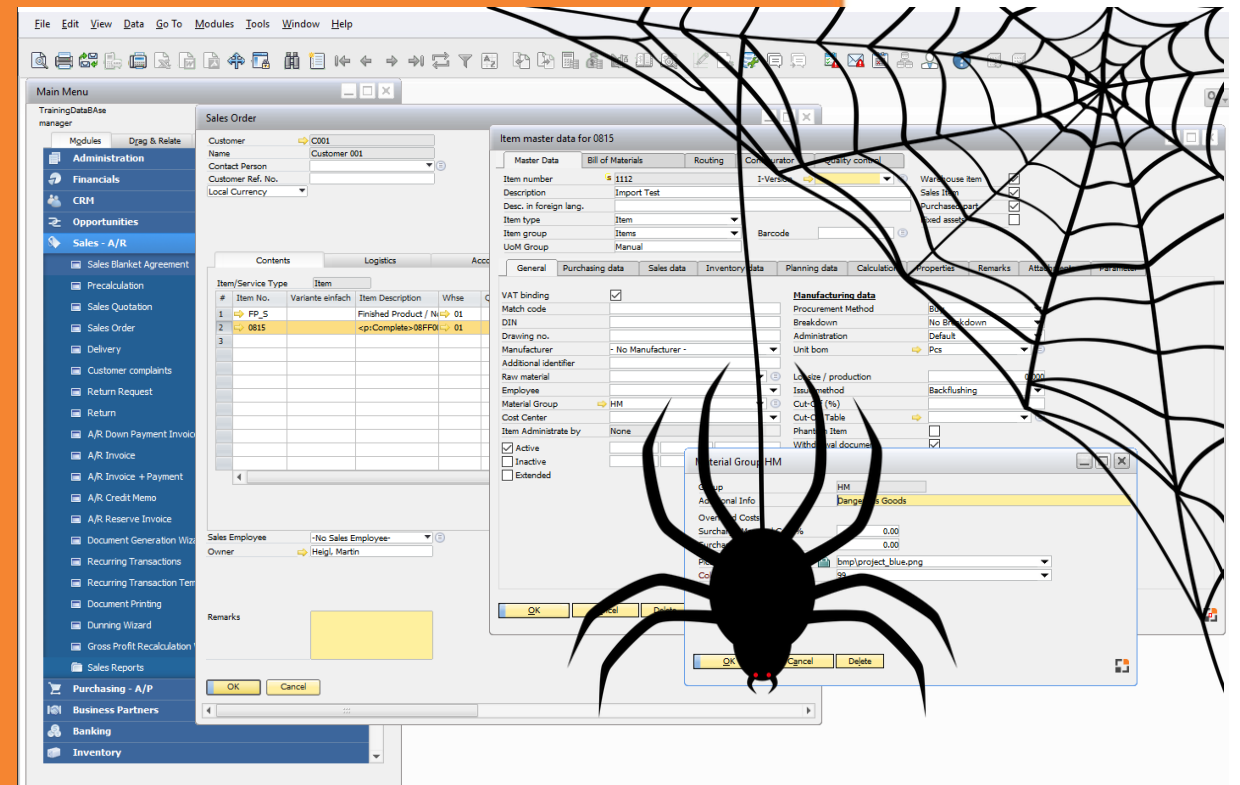
Work Orders List Assembly

Document	Sales Order	Date	Customer	Name	From	To	Item	Plan	Actual	Drawing number	Type / Scrap
1326	WH000385	29/04/19			25/04/19	29/04/19	SerialAssembly				Storage Order
1323	VA:Prod. PlannedCos	26/04/19			01/01/00	26/04/19	FPCOST				Storage Order
1320	WH000382	24/04/19			02/05/19	02/05/19	ATP				Storage Order
1319	ABC-TestImport2	09/04/19	customer-name		09/04/19	26/04/19	781-A				muster
1318	ATP	09/04/19			11/04/19	11/04/19	ATP				planned_work_order
1317	WH000359	08/04/19			03/04/19	08/04/19	FP				Storage Order
10	FP	Finished Product / Normal			03/04/19	08/04/19		5.000	0.000 Pcs		0.000 / 0.000 Pcs
10	A RM	Raw Material / Normal						5.000	0.000 Pcs	295.000 Pcs	
20	RM_B	Raw Material / Batch						5.000	0.000 Pcs	298.000 Pcs	
	RM_B	20190323 01			10/04/19			5.000	5.000 Pcs	298.000 Pcs	
30	RM_S	Raw Material / Serial						5.000	0.000 Pcs	5.000 Pcs	
10	3133	Internal Operation With QC - Standard (default Reso			03/04/19	03/04/19		130.000	5.000 Min.		1.000
20	1201	Assembling			03/04/19	05/04/19		1,010.000	Min.		
10	1500	Painting						0.000	Min.		
10	T2	Test Tool 2						1,010.000	Min.		
30	R-01-1T.STD.08H	Internal Operation STND001 - Standard (default Res			08/04/19	08/04/19		130.000	1.000 Min.		
1316	QC(pc_20190408)	08/04/19			08/04/19	08/04/19	pc_20190408_FG				Storage Order
1315	TFS19430	08/04/19	C40001	Earthshaker Corp.	08/04/19	08/04/19	PROJECT				Project
10	PROJECT	Default Item for Projects			08/04/19	08/04/19		1.000	0.000 Pcs		
1314	WH000358	23/03/19			29/12/99	23/03/19	FP				Storage Order
1313	WH000355	19/03/19			13/09/18	19/03/19	abc				Storage Order
1312	WH000354	13/03/19			27/03/19	01/04/19	P_ParallelResourceTest				Storage Order
1311	WH000353	07/03/19			27/03/19	26/04/20	abc				Storage Order
1310	WH000352	07/03/19			10/04/19	13/04/20	abc				Storage Order

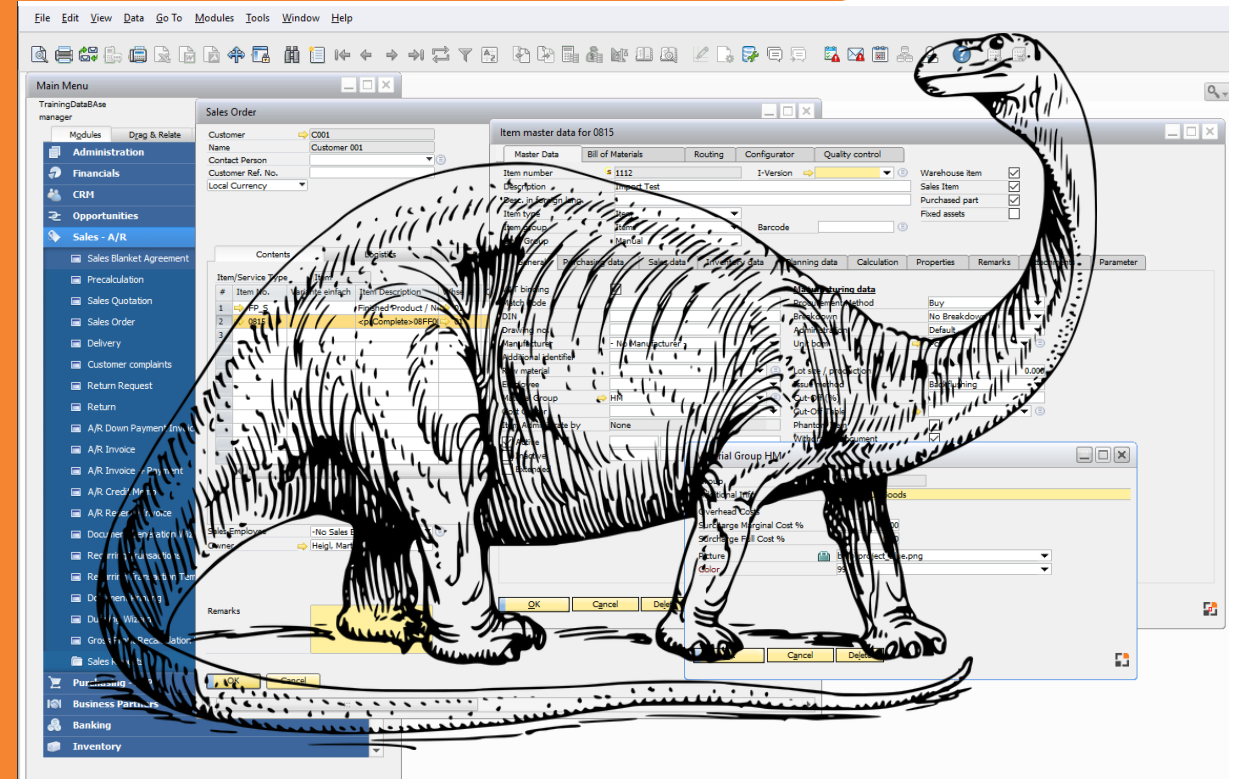
Search Edit End New Delete Call up Product Costing Analysis

A 20-year-old program is not “modern”
Usability no longer corresponds to the
current time.

It's like the end of MS DOS time and the new
Windows graphical user interface ~1998



But SAP and beas are a “Bontosaurus”
Too big to make it new
Too many customers use it
Too many expensive customizing's are
existing





What we need is a correct “modernization” for the System

Not good Idea:

Display Current User interface in a browser

Effekt: 0.

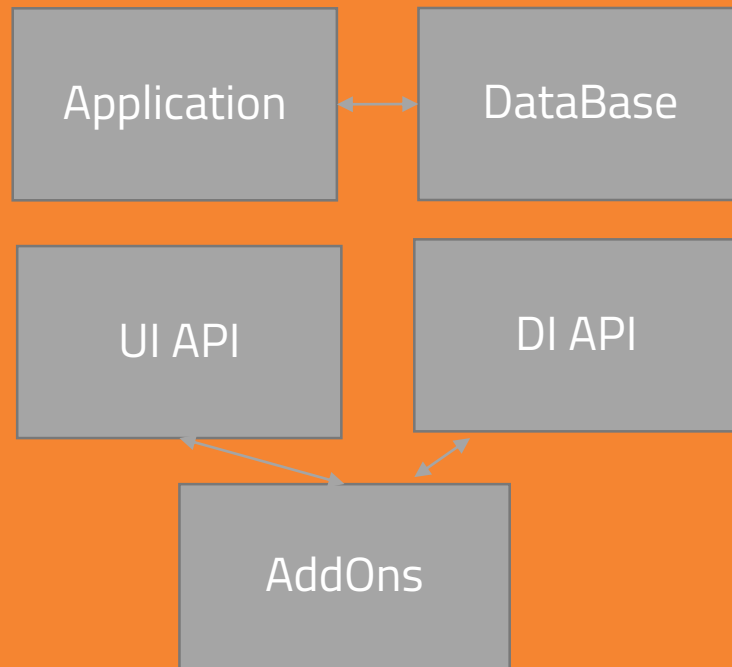
- Not modern interface
- Not running on every Hardware

Modern is

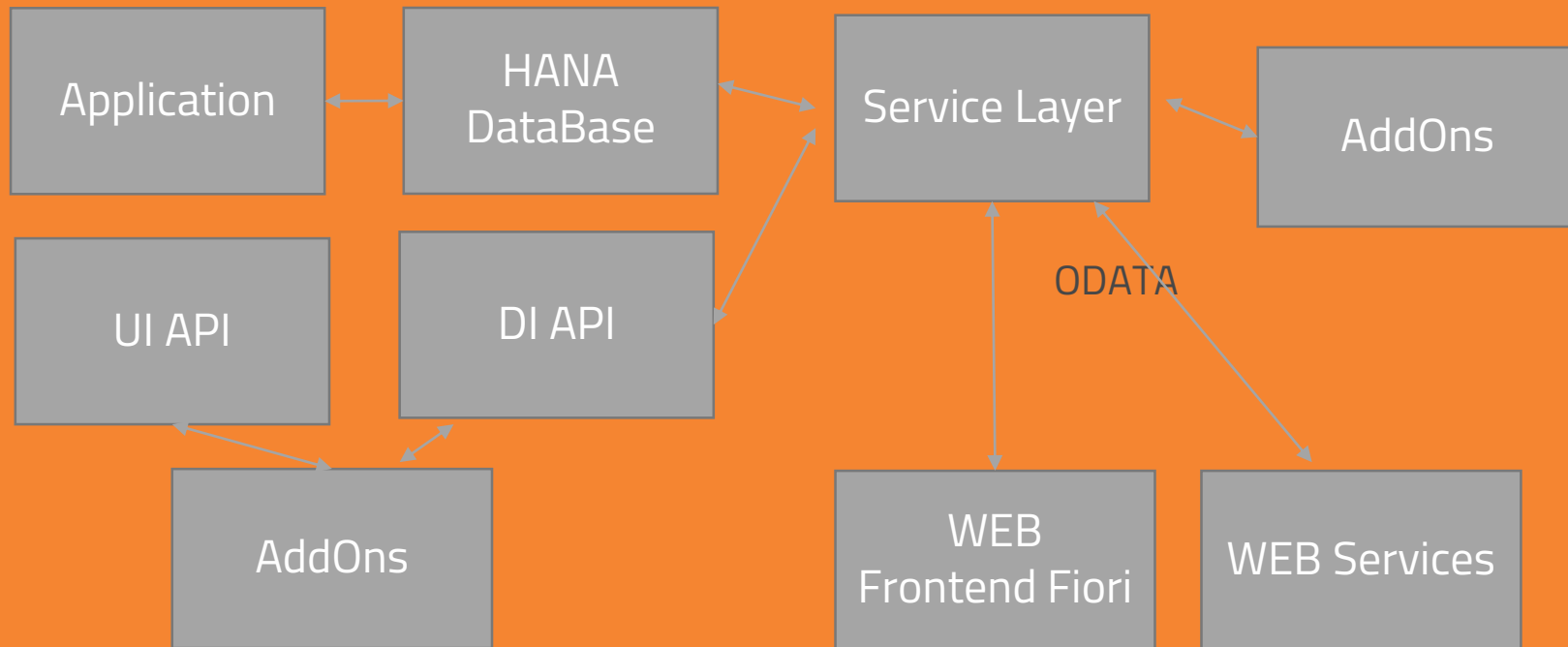
Example

- Internet of Things
- Machine – Machine communication (Industry 4.0)
- Simple Front end interface for Employees and End-user
- Running on every Hardware
- Apps for Android and IOS

How SAP is working?



How SAP is working?



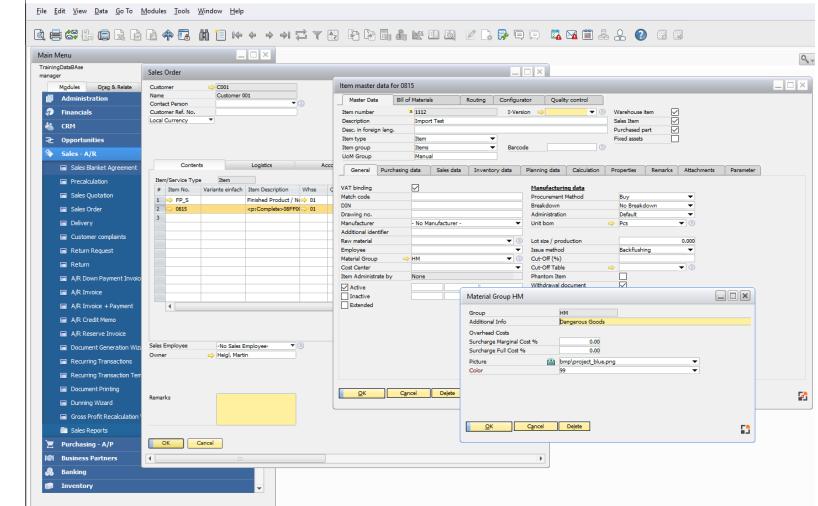
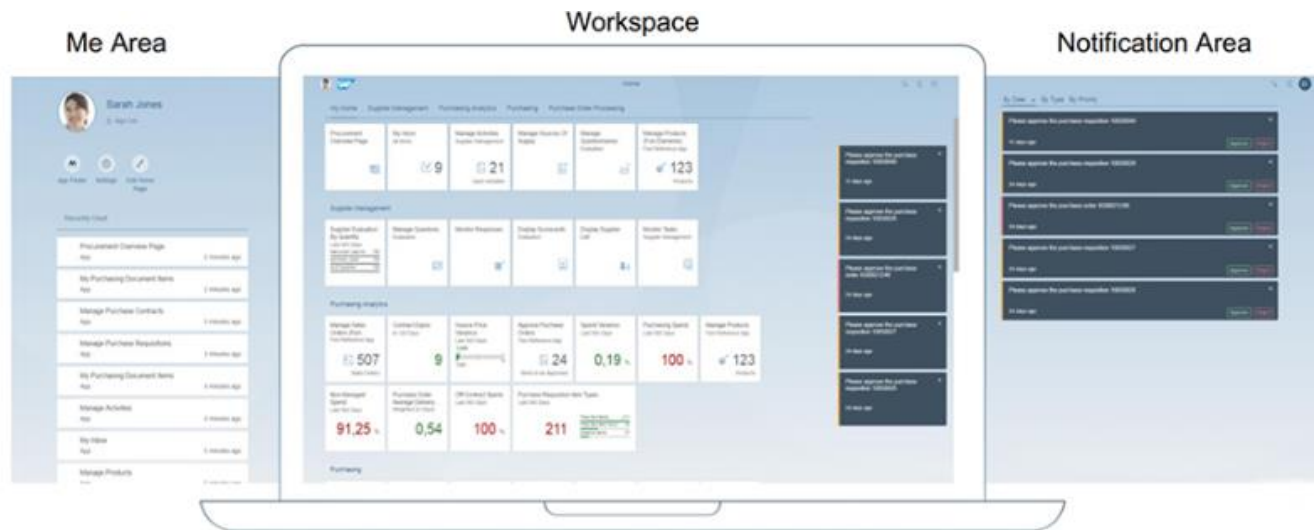
How SAP is working?



Old and new style are available in parallel

User can switch part by part to new application and can decide self, in which department they using which front end

In Backend only new: The ODATA Wrapper





Is it possible to create **WEB Server** with native Powerbuilder?

Always answer: No

But one of my employee delivered an running HTTP Object



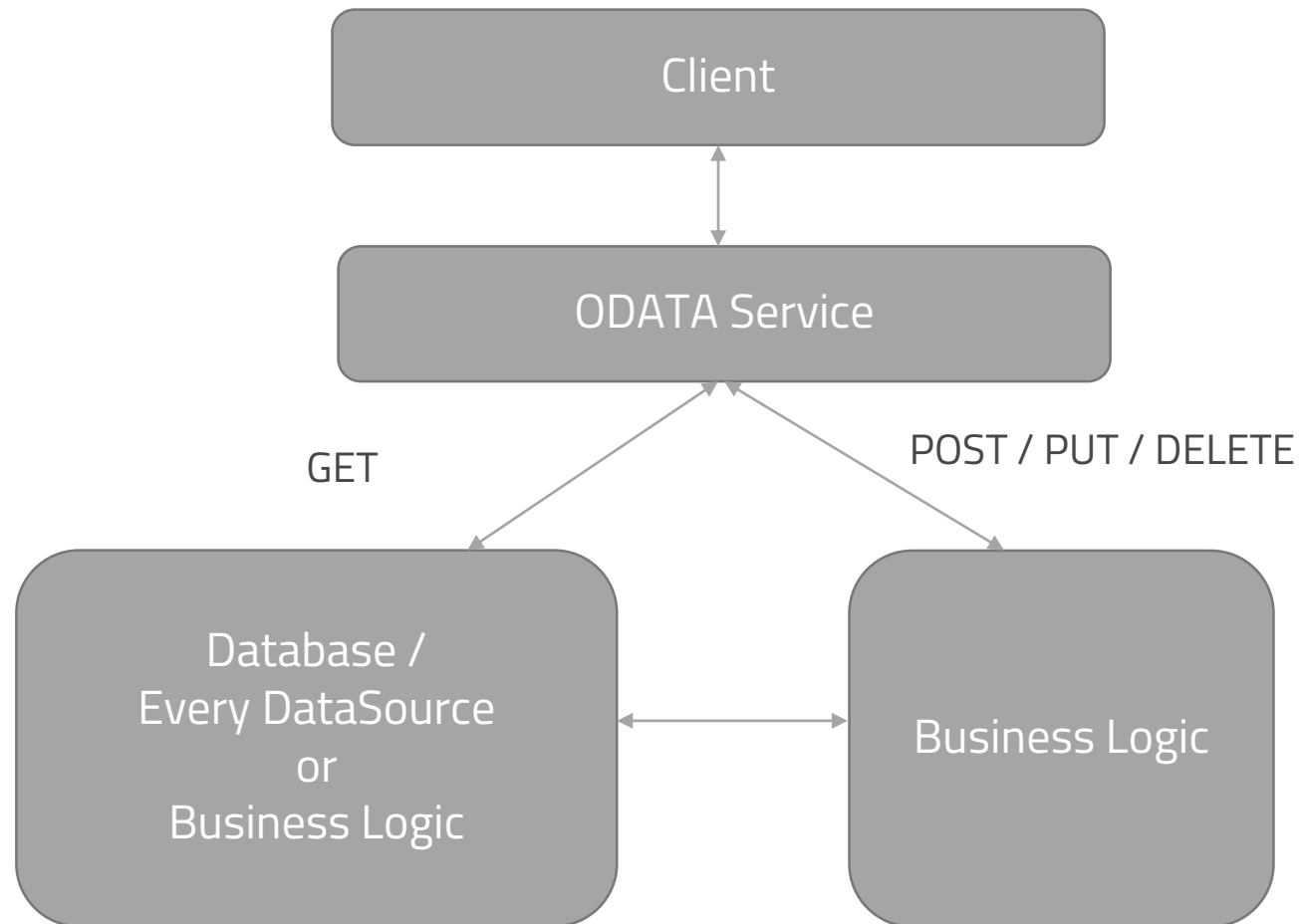
Is it possible to create **ODATA Service** with Powerbuilder?

always answer "NO"

I asked me self:

Answer: what is a ODATA Service ?

What is ODATA



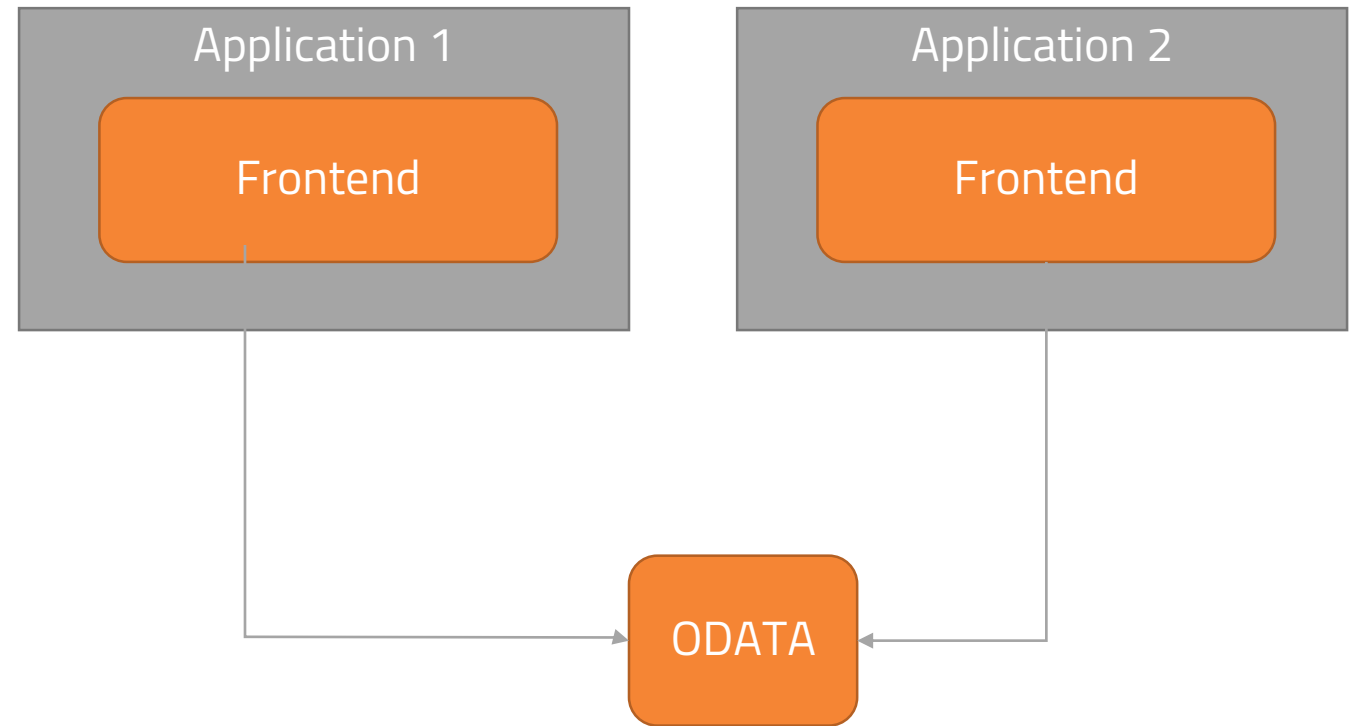
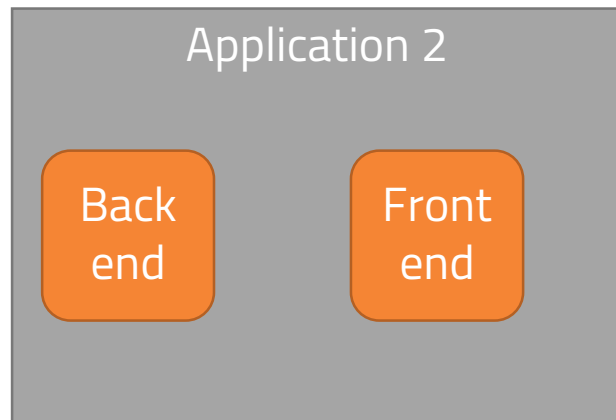
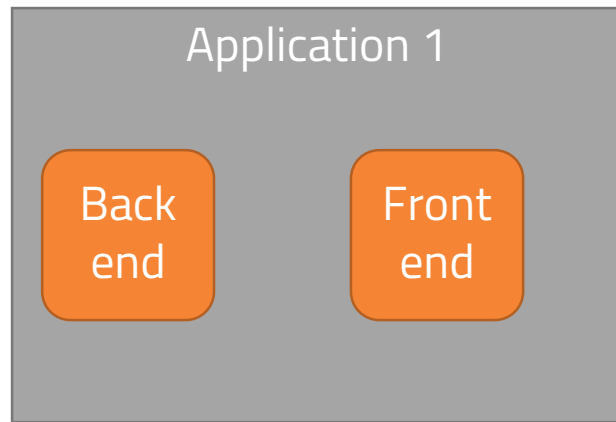
Middle Layer between Client
and Database / Business Logic


Do we need a HTTP Server?

No!

ODATA is a communication language

What is the difference between a normal „RestService“ and ODATA?





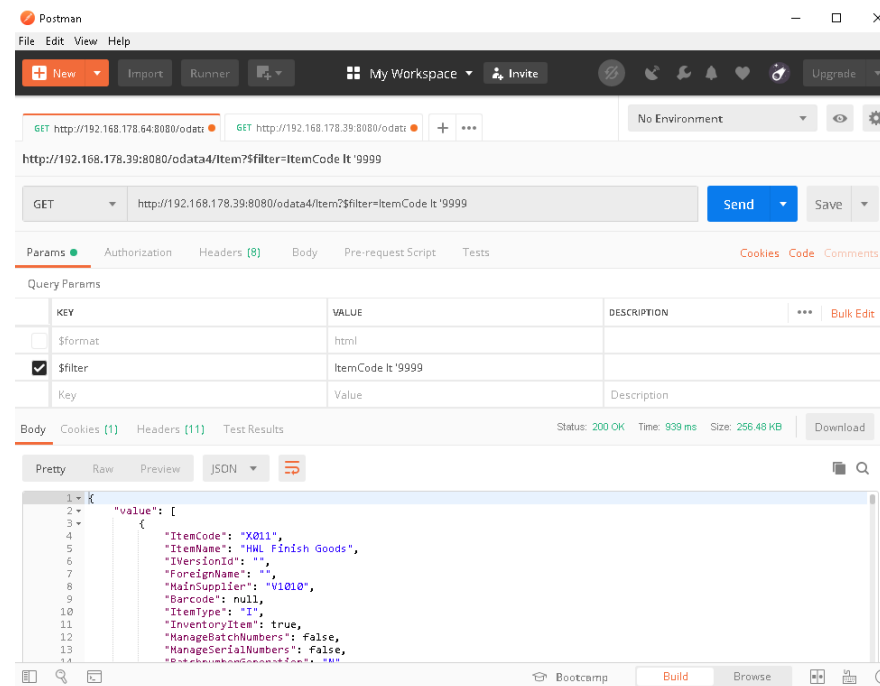
A simple function
MUST BE POSSIBLE



How to learn ODATA?

Develop a ODATA Service. If ready, you're a ODATA specialist

40 hours development time later the first version was running over HTTP

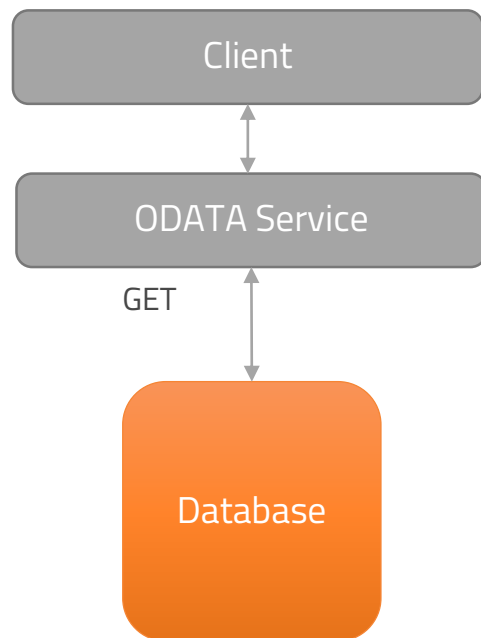




Examples

ODATA as database Middle Layer

Simple Language for single source solution
for different Database



ODATA

`get Item?$select=ItemCode&$filter=ItemCode eq '11'`

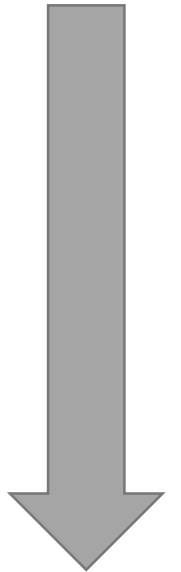
Translate this to used Database:

HANA Database

`select "ItemCode" from "OITM" where "ItemCode"='11'`

MSSQL

`select "ItemCode" from "OITM" where "ItemCode"=N'11'`



ODATA as database Middle Layer

English fieldname with compatibility to
20 years old german databases

ODATA

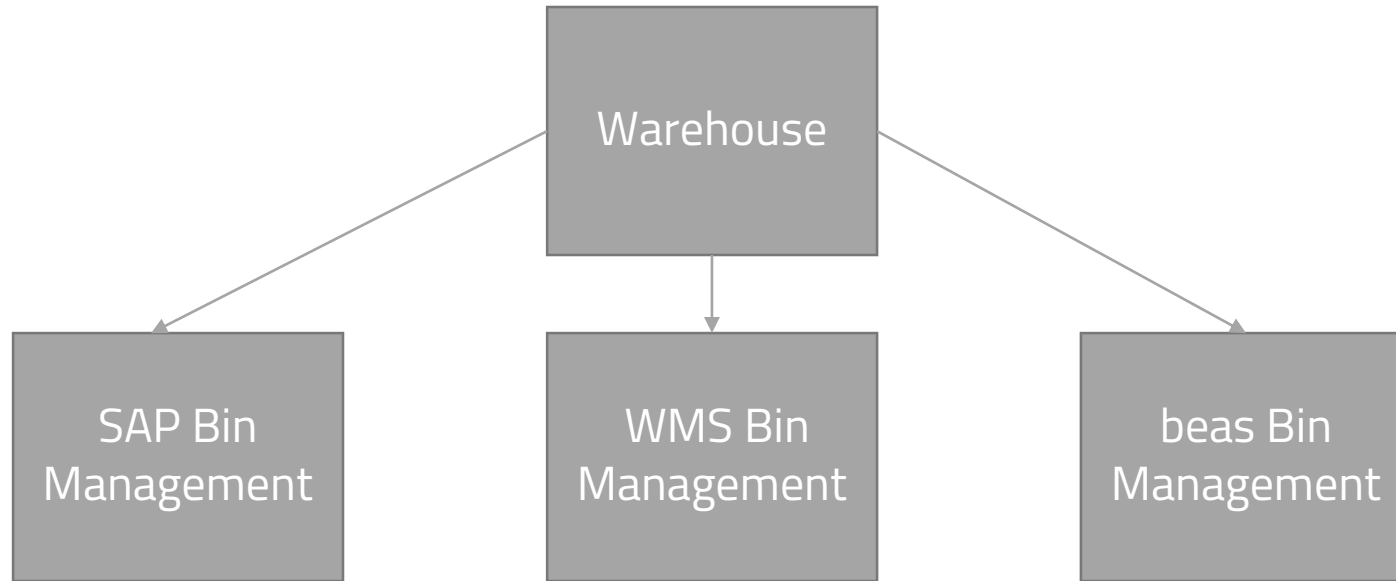
`get Workorder/DocumentNumber`



Translate this to SQL Statement

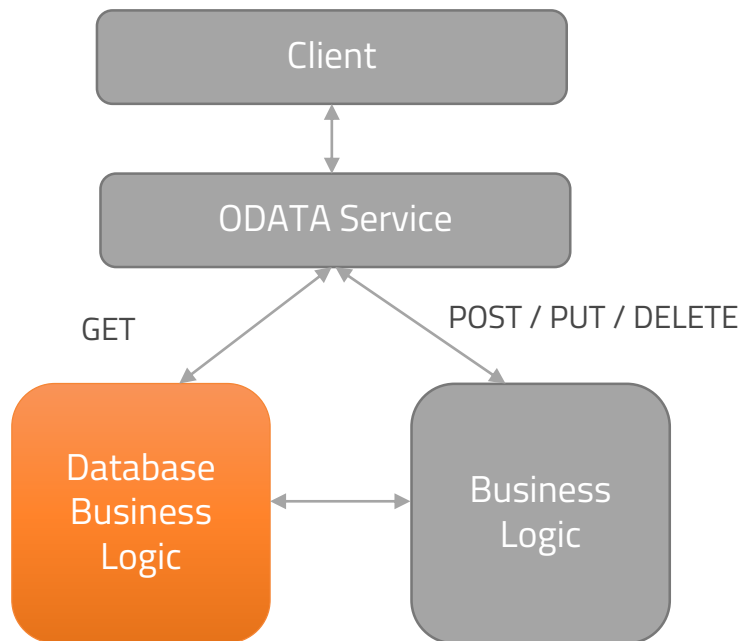
`select "FertigungsAuftraege"."BelegNummer"`

Bin Management



ODATA as database Middle Layer

One Object from Different DataSources



ODATA

get Warehouse/Stock

Warehouse is managed by SAP

Select "Stock" from "SAP-Table"

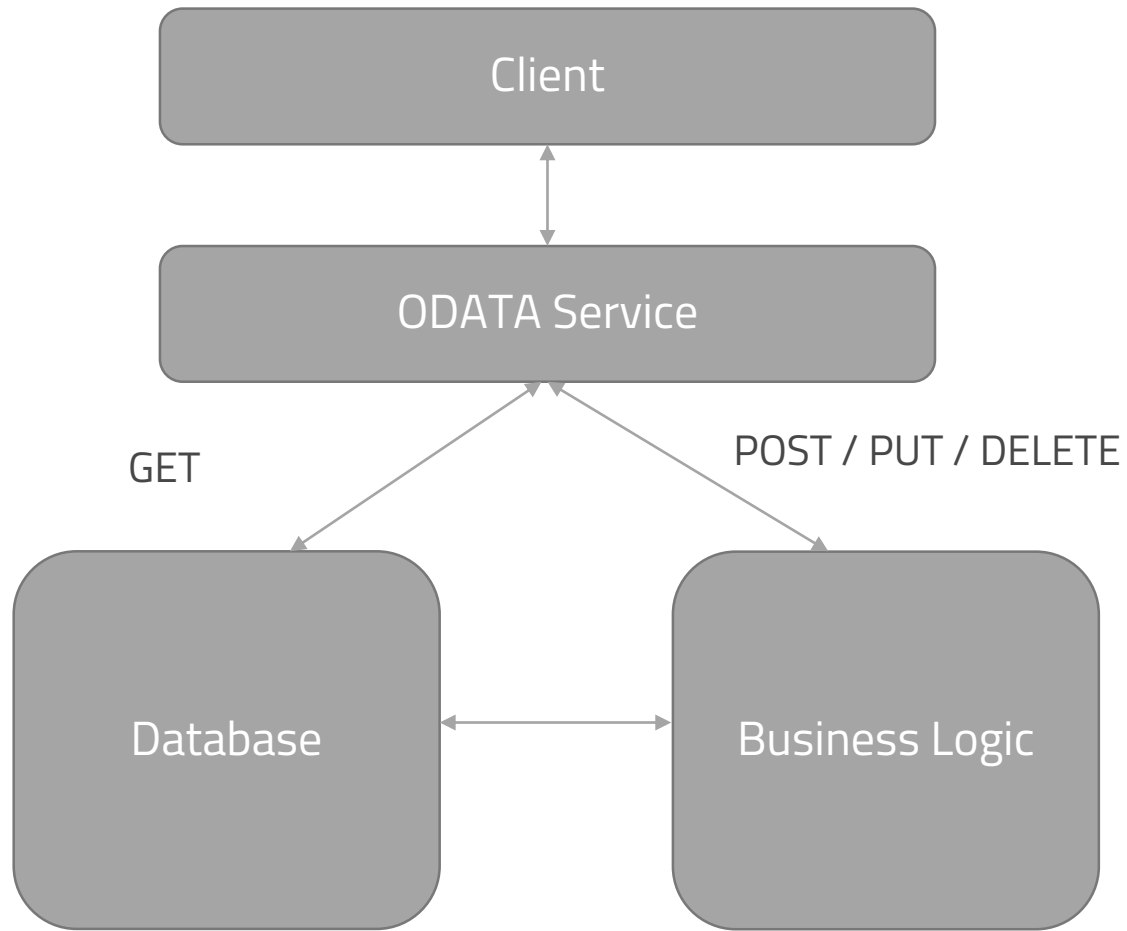
Warehouse is managed by beas

Select "Bestand" from "BEAS-Table"

The screenshot shows a SAP Goods Receipt form. The header includes a home icon, the text "Goods Receipt", and a "Save" button. Below the header, there are tabs for "Line" and "Lines (1)". The form contains several input fields with search icons:

- Item No: Simple Version Admin Item 1
- 1111
- I-Version: aaa
- Warehouse: Warehouse
- Bin Location: Bin Location

What is the Client



Communication:

- Simple language
- Standard
- Compatible
- Modern

WEB Frontend

IOS/Android App

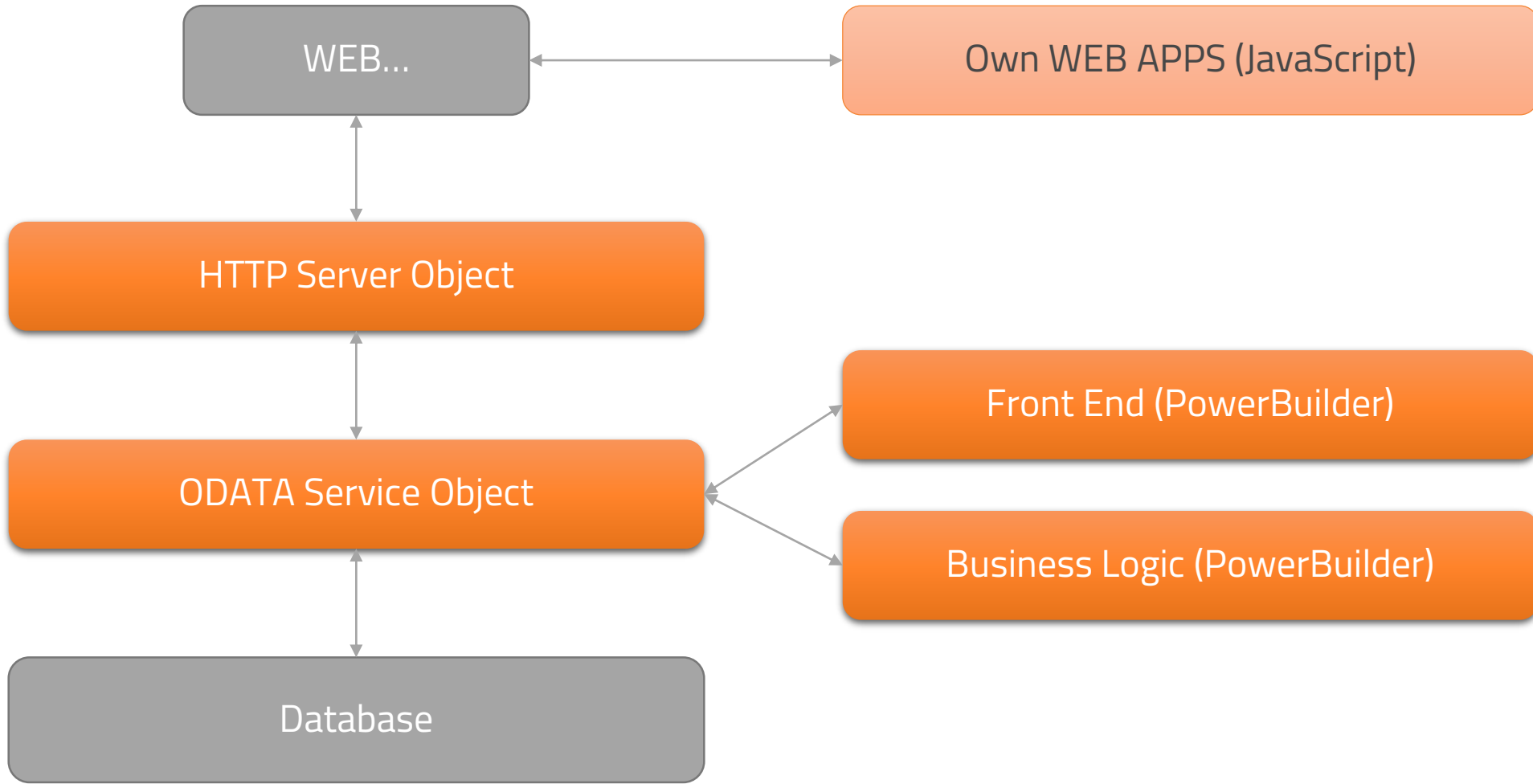
Excel

Other Languages (example C#)

Machine / Machine (Industry 4.0)

Own PowerBuilder Program

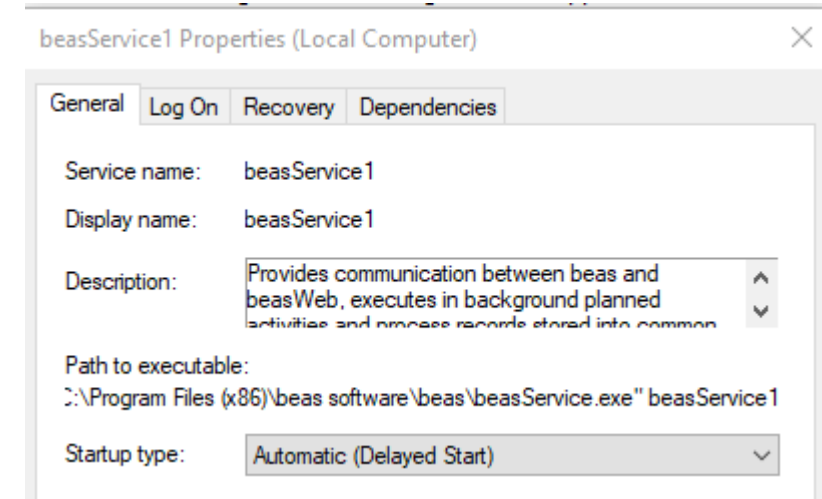
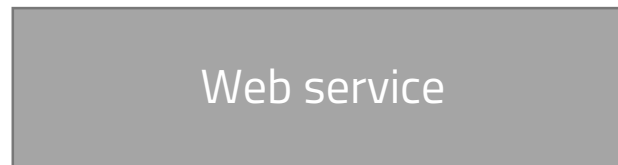
Native PowerBuilder Solution



Application is Service and Desktop solution

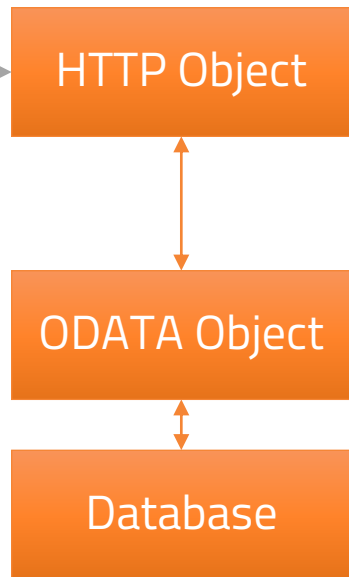
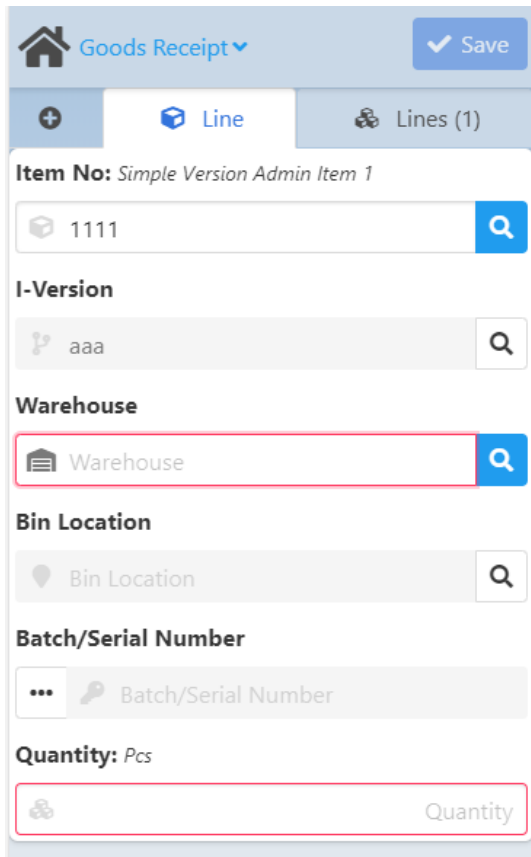
Only one source code base and only one compilation

Register beas.exe as service

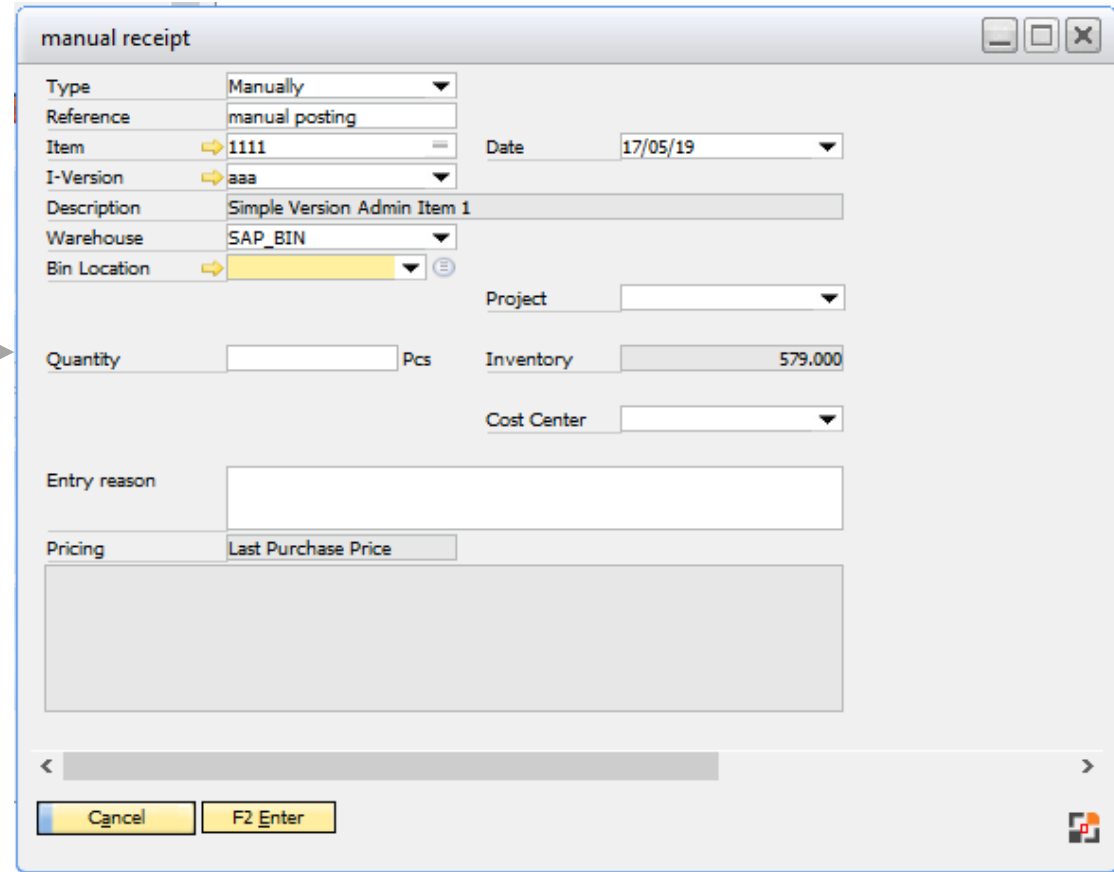


Real Example

WEB Application (JavaScript)



Powerbuilder Window





How to created this in PowerBuilder

Definition of a ODATA object (Entity Collection)

Example Object "Item" (= Items/Material – SAP Table)

`ue_EntityCollection.of_addObject(Name,Table,Primary)`

Property	
Table	OITM (SAP Table for Items)
Primary	"ItemCode"
(...)	(...)

Fields

`ue_EntityCollection.of_AddField("Item","ItemCode",' "OITM"."ItemCode" ', "Edm.String(50)", (... additional properties...)`

Property		Type
ItemCode	"OITM"."ItemCode"	Edm.String(50)
ItemName	"OITM"."ItemName"	Edm.String(100)
(...)		

Object array

EntityCollections[]

EntityCollection

Properties

ColumnDefinition[]



ODATA Object

ue_restservice object

Instance: DataStore

ue_restservice.of_get(as_odataString) return String

ODATA4 String

get [object]?\$select=[fields]&\$filter=[filter rules]&\$orderby=[fields]

get [object]?\$select=[fields]&\$filter=[filter rules]&\$orderby=[fields]

get Item?\$select=ItemCode,ItemName&\$filter=ItemCode eq "0815"&\$orderby=ItemCode

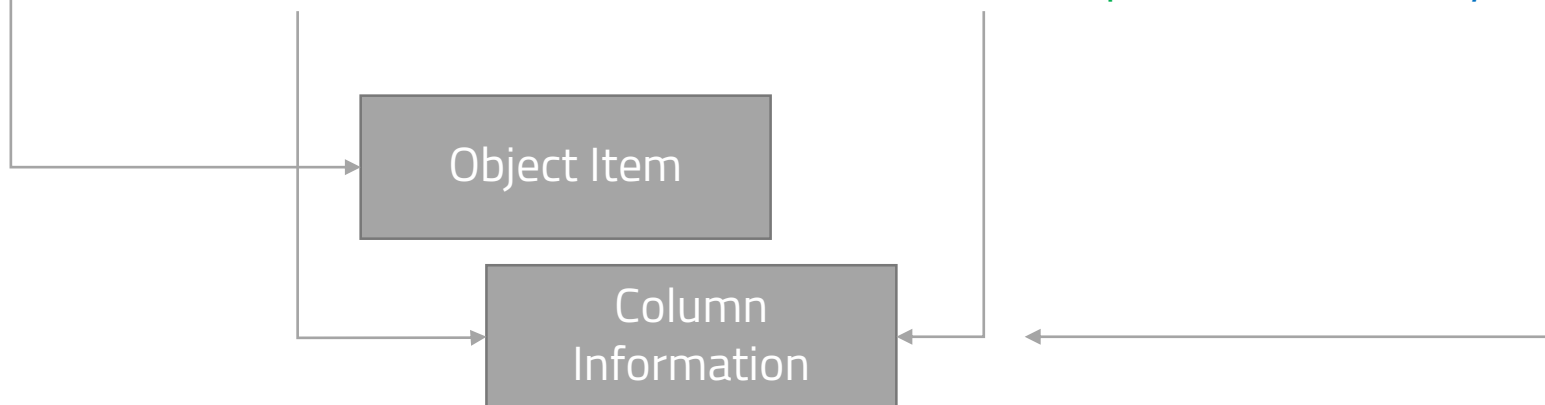
Convert to MSSQL SQL Script

Select "OITM"."ItemCode","OITM"."ItemName" from "OITM" where "OITM"."ItemCode"=N'0815' order by
"OITM"."ItemCode"

Or HANA

Select "OITM"."ItemCode","OITM"."ItemName" from "OITM" where lower("OITM"."ItemCode")=lower('0815')
order by "OITM"."ItemCode"

get Item?\$select=ItemCode,ItemName&\$filter=ItemCode eq "0815"&\$orderby=ItemCode



Select information

Typ	Value
is_fromtable	OITM
is_select	"OITM"."ItemCode" as ItemCode,"OITM"."ItemName" as ItemName
is_where	"OITM"."ItemCode" = '0815'
is_orderby	"OITM"."ItemCode"

Column information

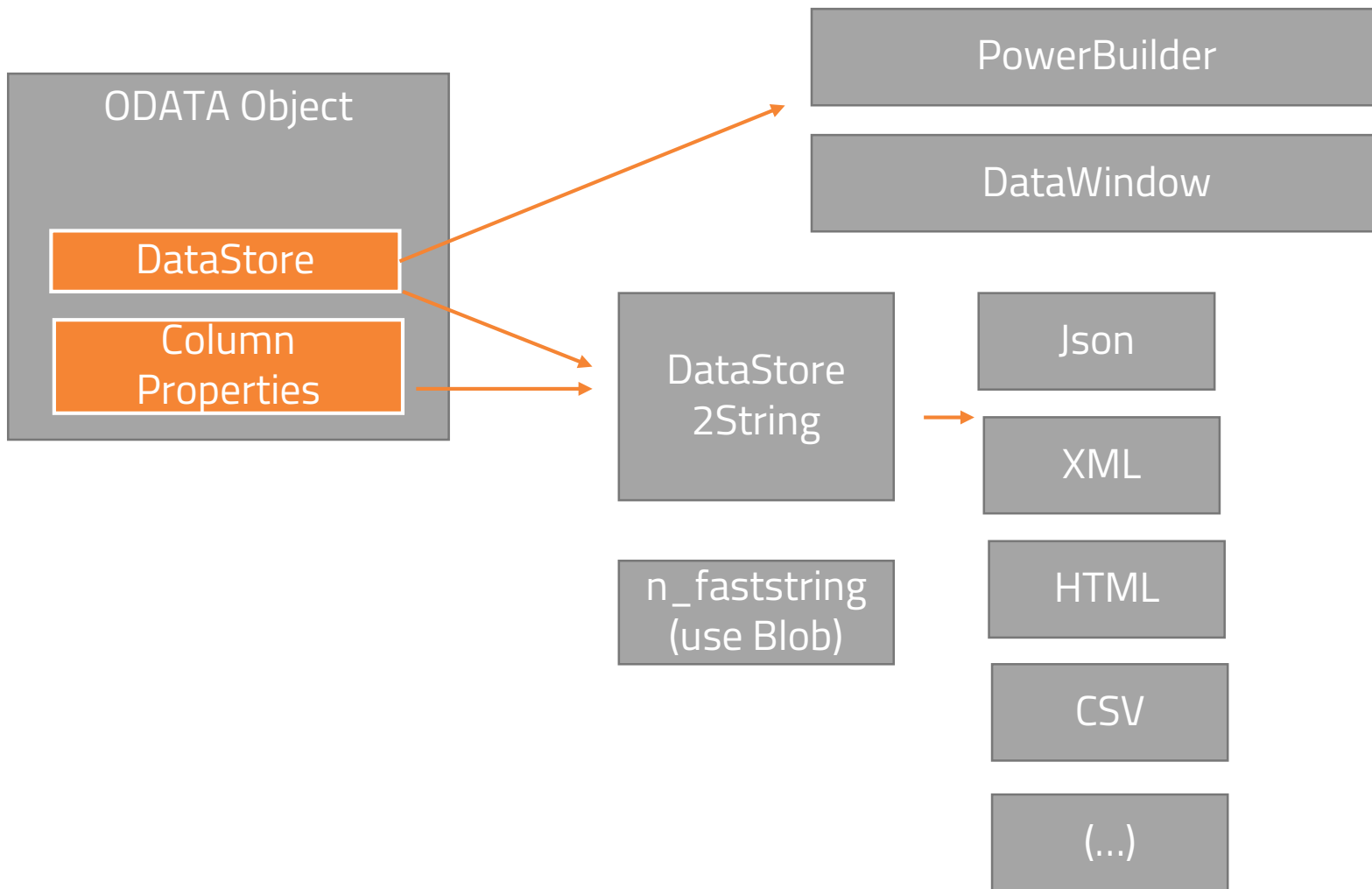
Field	Type	Description
ItemCode	Edm.String	Item Code
ItemName	Edm.String	Item Name

Typ	Value
is_fromtable	OITM
is_select	"OITM"."ItemCode" as "ItemCode","OITM"."ItemName" as "ItemName"
is_where	"OITM"."ItemCode" = '0815'
is_orderby	"OITM"."ItemCode"

Create Datastore with select command

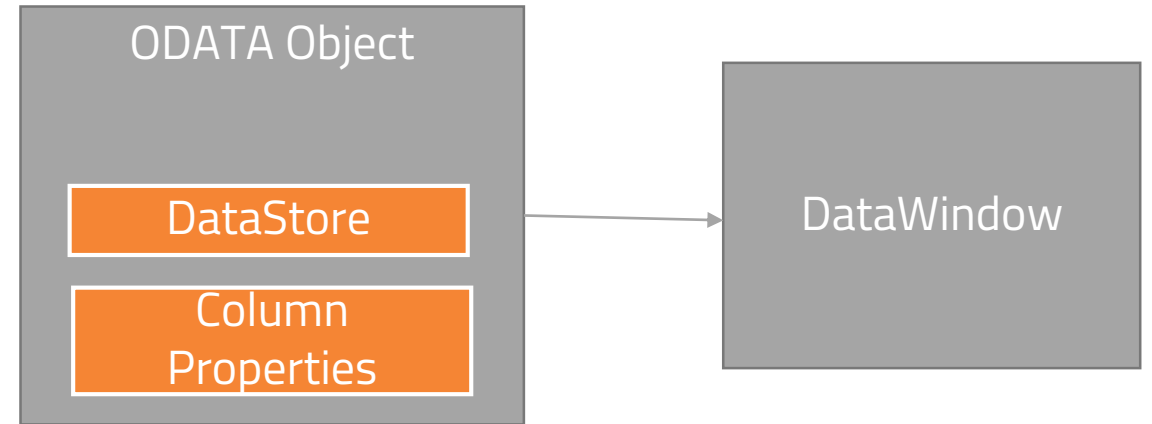
```
Select "OITM"."ItemCode" as "ItemCode","OITM"."ItemName" as "ItemName" from "OITM" where  
lower("OITM"."ItemCode")=lower('0815') order by "OITM"."ItemCode"
```

&format=[format]



Copy DataStore to DataWindow

- different orders of fieldnames
- fast



```
ll_column_count = LONG(adw.Describe("Datawindow.Column.Count"))
FOR i = 1 TO ll_column_count
  /** Get the Column Name. */
  ls_col_nm = adw.Describe("#" + STRING(i) + ".Name")

  /** Get the column ID from bsl.ids. */
  ll_dwa_col_id = LONG(bsl.ids.Describe(ls_col_nm + ".ID"))

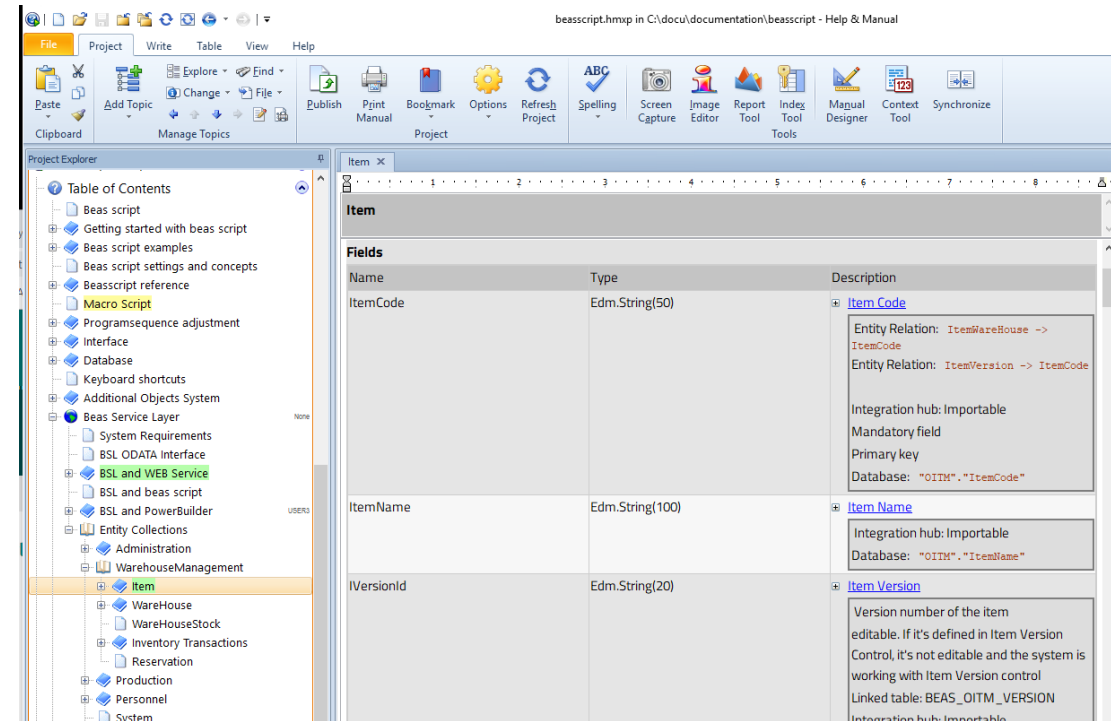
  /** Copy this column's values from the bsl.ids to adw. */
  try
    adw.Object.Data[1, i, ll_row_count, i] = &
      bsl.ids.Object.Data[1, ll_dwa_col_id, ll_row_count, ll_dwa_col_id]
  catch (runtimeerror re)
    gapp.of_meldung("error", "error in retrieve. column: "+ls_col_nm+": "+re.i)
  end try
NEXT
```

Documentation

\$metadata

```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="1.0" xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx">
  <edmx:DataServices m:DataServiceVersion="1.0" m:MaxDataServiceVersion="3.0" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
    <Schema Namespace="bsl" xmlns="http://schemas.microsoft.com/ado/2008/09/edm">
      <EntityType Name="Warehouse">
        <key>
          <PropertyRef Name="WhsCode" />
        </key>
        <Property Name="WhsCode" Type="Edm.String" Nullable="false" MaxLength="20"
          Unicode="true" />
        <Property Name="WhsName" Type="Edm.String" Nullable="false" MaxLength="100"
          Unicode="true" />
        <Property Name="WhsType" Type="Edm.String" Nullable="false" MaxLength="50"
          Unicode="true" />
        <Property Name="WhsBinType" Type="Edm.Int32" />
        <Property Name="WhsBinRFID" Type="Edm.Boolean" />
        <Property Name="Visualization" Type="Edm.Boolean" />
        <Property Name="WhsColor" Type="Edm.Int32" />
        <Property Name="WhsBitmap" Type="Edm.String" Nullable="false" MaxLength="50"
          Unicode="true" />
        <Property Name="OnHand" Type="Edm.Decimal" />
        <Property Name="Active" Type="Edm.Boolean" />
        <Property Name="BranchId" Type="Edm.Int32" />
        <Property Name="DefaultVariant" Type="Edm.String" Nullable="false" MaxLength="1"
          Unicode="true" />
        <Property Name="Length" Type="Edm.Decimal" />
        <Property Name="Width" Type="Edm.Decimal" />
      </EntityType>
      <EntityType Name="Item">
        <key>
          <PropertyRef Name="ItemCode" />
        </key>
```

Extended documentation



Join Tables - \$expand command

odata4/Item("1111")?\$select=Item/ItemCode,Item/ItemName,
ItemWareHouse&\$expand=ItemWareHouse

```
ue_EntityCollection.of_addjoin(  
    "Item","ItemWareHouse",  
    " Left outer join ~"BeasWareHouse~" on .... "  
)
```

```
{  
  "value": {  
    "ItemCode": "1111",  
    "ItemName": "Simple Version Admin Item 1",  
    "ItemWareHouse": [  
      {  
        "ItemCode": "1111",  
        "WhsCode": "01",  
        "OnHand": 579,  
        "UoMCode": "Pcs",  
        "Locked": false,  
        "Active": true  
      },  
      {  
        "ItemCode": "1111",  
        "WhsCode": "01-locke",  
        "OnHand": 0,  
        "UoMCode": "Pcs",  
        "Locked": false,  
        "Active": true  
      },  
      {  
        "ItemCode": "1111",  
        "WhsCode": "02",  
        "OnHand": 15,  
        "UoMCode": "Pcs",  
        "Locked": false,  
        "Active": true  
      },  
      {  
        "ItemCode": "1111",  
        "WhsCode": "03",  
        "OnHand": 1,  
        "UoMCode": "Pcs",  
        "Locked": false,  
        "Active": true  
      }  
    ]  
  }  
}
```

Join Tables - \$expand command

odata4/Item("1111")?\$select=Item/ItemCode,Item/ItemName,
ItemWareHouse

```
{
  "value": [
    {
      "ItemCode": "1111",
      "ItemName": "Simple Version Admin Item 1",
      "WhsCode": "01",
      "OnHand": 579,
      "UoMCode": "Pcs",
      "Locked": false,
      "Active": true
    },
    {
      "ItemCode": "1111",
      "ItemName": "Simple Version Admin Item 1",
      "WhsCode": "01-locke",
      "OnHand": 0,
      "UoMCode": "Pcs",
      "Locked": false,
      "Active": true
    },
    {
      "ItemCode": "1111",
      "ItemName": "Simple Version Admin Item 1",
      "WhsCode": "02",
      "OnHand": 15,
      "UoMCode": "Pcs",
      "Locked": false,
      "Active": true
    }
  ]
}
```


Export \$format

JSON, JSONArray, XML, CSV, HTML

DataWindow and more

```

1 <?xml version="1.0"?>
2 <row>
3   <ItemCode>1111</ItemCode>
4   <ItemName>Simple Version Admin Item 1</ItemName>
5   <ItemCode>1111</ItemCode>
6   <WhsCode>01</WhsCode>
7   <OnHand>579</OnHand>
8   <UoMCode>Pcs</UoMCode>
9   <Locked>>false</Locked>
10  <Active>>true</Active>
11 </row>
12 <row>
13   <ItemCode>1111</ItemCode>
14   <ItemName>Simple Version Admin Item 1</ItemName>
15   <ItemCode>1111</ItemCode>
16   <WhsCode>01-locke</WhsCode>
17   <OnHand>0</OnHand>

```

i	1	itemcode	itemname	itemcode	whscode	onhand	uomcode	locked	active
2	1111	Simple Version Admin Item 1	1111	01	579.000000	Pcs	N	1	
3	1111	Simple Version Admin Item 1	1111	01-locke	0	Pcs	N	1	
4	1111	Simple Version Admin Item 1	1111	02	15.000000	Pcs	N	1	
5	1111	Simple Version Admin Item 1	1111	03	1.000000	Pcs	N	1	
6	1111	Simple Version Admin Item 1	1111	06	0	Pcs	N	1	
7	1111	Simple Version Admin Item 1	1111	09	5.000000	Pcs	N	1	
8	1111	Simple Version Admin Item 1	1111	11	0	Pcs	N	1	
9	1111	Simple Version Admin Item 1	1111	12	0	Pcs	N	1	
10	1111	Simple Version Admin Item 1	1111	1410	0	Pcs	N	1	
11	1111	Simple Version Admin Item 1	1111	1450	0	Pcs	N	1	
12	1111	Simple Version Admin Item 1	1111	2000-a	4.000000	Pcs	N	1	
13	1111	Simple Version Admin Item 1	1111	2000-b	0	Pcs	N	1	

```

1 {
2   "value": [
3     {
4       "ItemCode": "1111",
5       "ItemName": "Simple Version Admin Item 1",
6       "WhsCode": "01",
7       "OnHand": 579,
8       "UoMCode": "Pcs",
9       "Locked": false,
10      "Active": true
11     },
12    {
13      "ItemCode": "1111",
14      "ItemName": "Simple Version Admin Item 1",
15      "WhsCode": "01-locke",
16      "OnHand": 0,
17      "UoMCode": "Pcs",
18      "Locked": false,

```

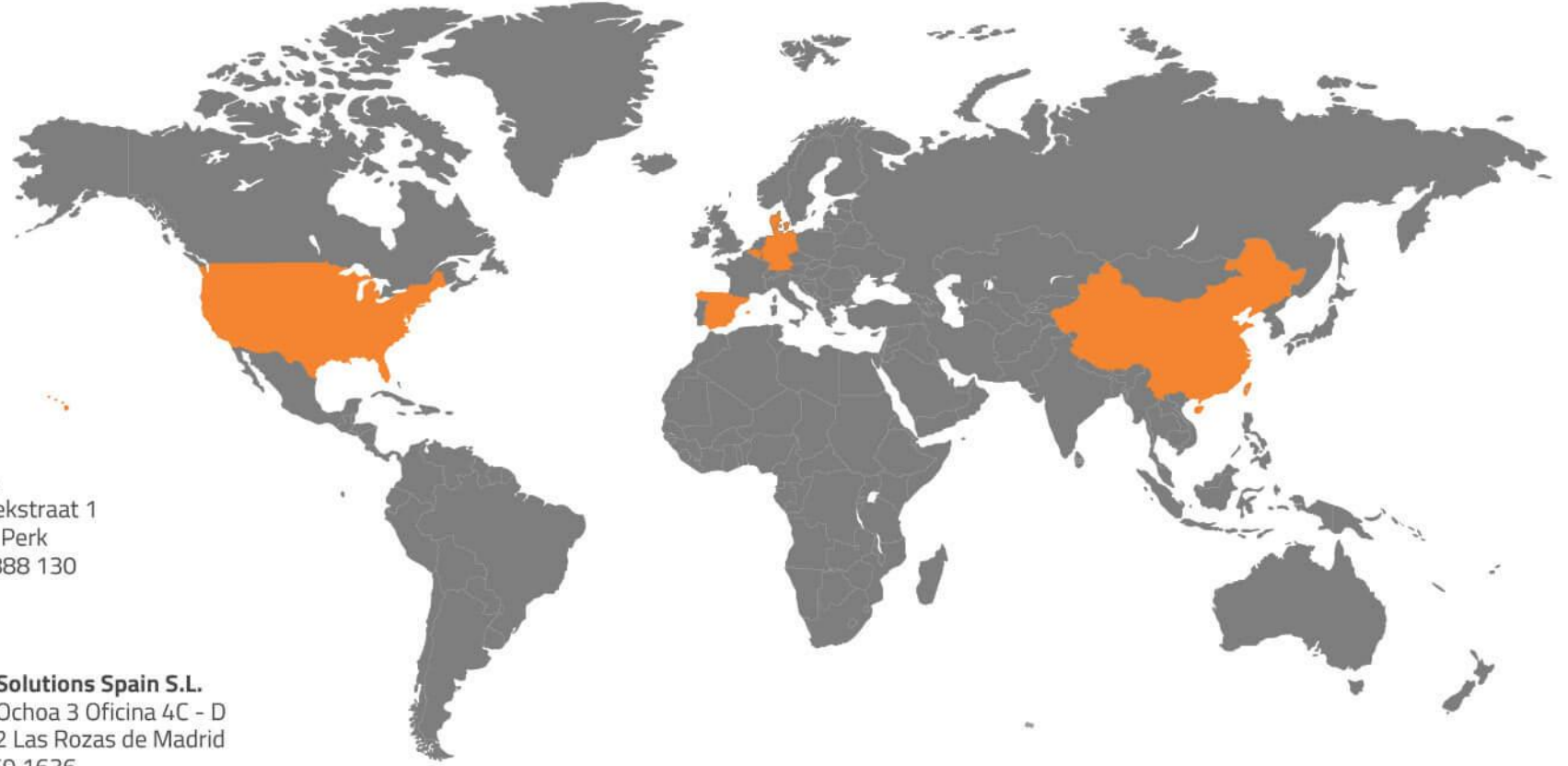
```

1 {
2   "value": [
3     [
4       "1111",
5       "Simple Version Admin Item 1",
6       "1111",
7       "01",
8       579,
9       "Pcs",
10      false,
11      true
12     ],
13    [
14      "1111",
15      "Simple Version Admin Item 1",
16      "1111",
17      "01-locke",
18      0,
19      "Pcs",
20      false,
21      true

```

itemcode	itemname	itemcode	whscode	onhand	uomcode	locked	active
1111	Simple Version Admin Item 1	1111	01	579.000000	Pcs	N	1
1111	Simple Version Admin Item 1	1111	01-locke	0	Pcs	N	1
1111	Simple Version Admin Item 1	1111	02	15.000000	Pcs	N	1
1111	Simple Version Admin Item 1	1111	03	1.000000	Pcs	N	1

Description		Simple Version Admin Item 1		
Warehouse		bw		
Bin Location				
	Bin Location	Warehouse	Item	First Item Inventory
	001001	A	3117159	12.00
Quantity	001002	A	25112150003_C	1.00
	001003	A	1111	4.00
	001004	A	2511216	7.00
Entry reason	001005	A	1112	1.00
	001006	A	0815	1.00
	001007	A	1111	1.00
Pricing	001008	A	po_serauto	1.00
	001009	A	A11896	9.00
	001010	A	9348	3.00



DENMARK (Headquarters)

Boyum IT Solutions A/S
 Sintrupvej 71b, 1th
 DK - 8220 Brabrand
 + 45 87 32 90 00

GERMANY

Boyum IT Solutions GmbH
 Borsigstraße 20
 DE - 65205 Wiesbaden
 + 49 6122 170 90 90

USA

Boyum IT Solutions Inc
 175 SW 7th St - Suite 1503
 Miami - FL 33130, USA
 + 1 727 773 2900

BELGIUM

Produmex
 Kasteelhoekstraat 1
 BE - 1820 Perk
 + 32 477 888 130

SPAIN

Boyum IT Solutions Spain S.L.
 C/ Severo Ochoa 3 Oficina 4C - D
 ES - 28232 Las Rozas de Madrid
 + 34 91 069 1636

HUNGARY

Budapest Development Center
 Baross u. 52
 HU - 1085 Budapest