

SAP Business One Service Layer

Trinidad Martinez / SAP Business One Solutions Architect

May 26-28, 2015

Public



SAP Business One Innovation summit - Barcelona

Agenda

SAP Business One Service Layer Introduction

Connection via standard HTTP/REST Tools

Features / Documentation

Usage / Samples



Introduction

B1 Service Layer

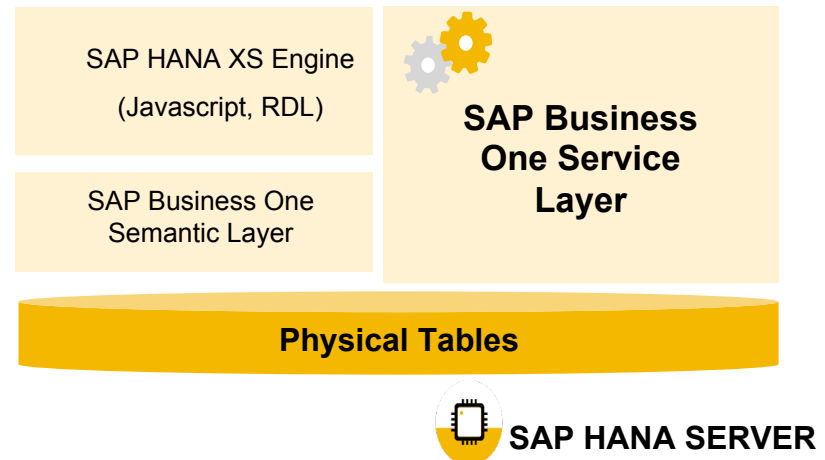
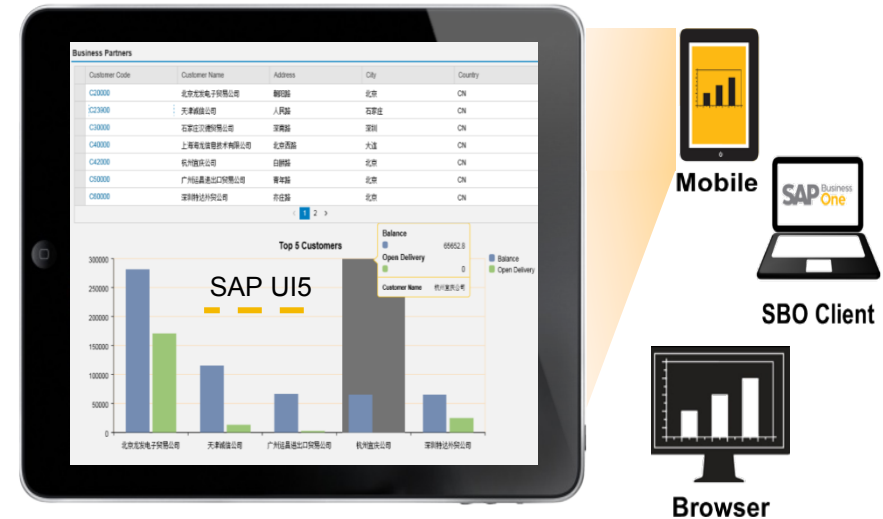
Extensibility for New Breed of Apps

System and Partner business objects exposed as services

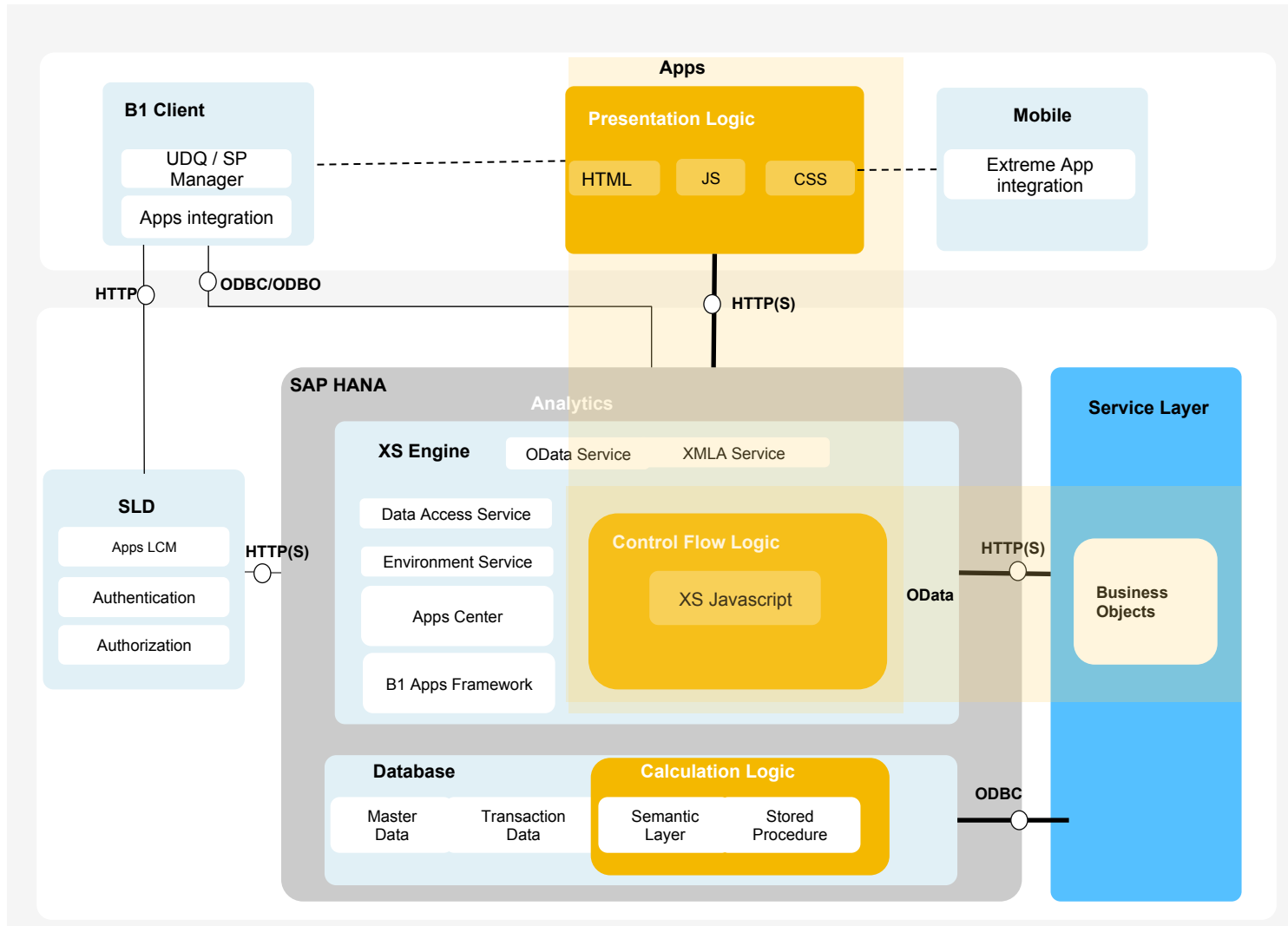
- ❑ Equivalent BO coverage to DI API
- ❑ **oData** protocol (across internet)
- ❑ Support **high volume** and throughput via **multi-threading** as well as **high availability**
- ❑ Available on SAP HANA version from 9.1 version.

Value Proposition

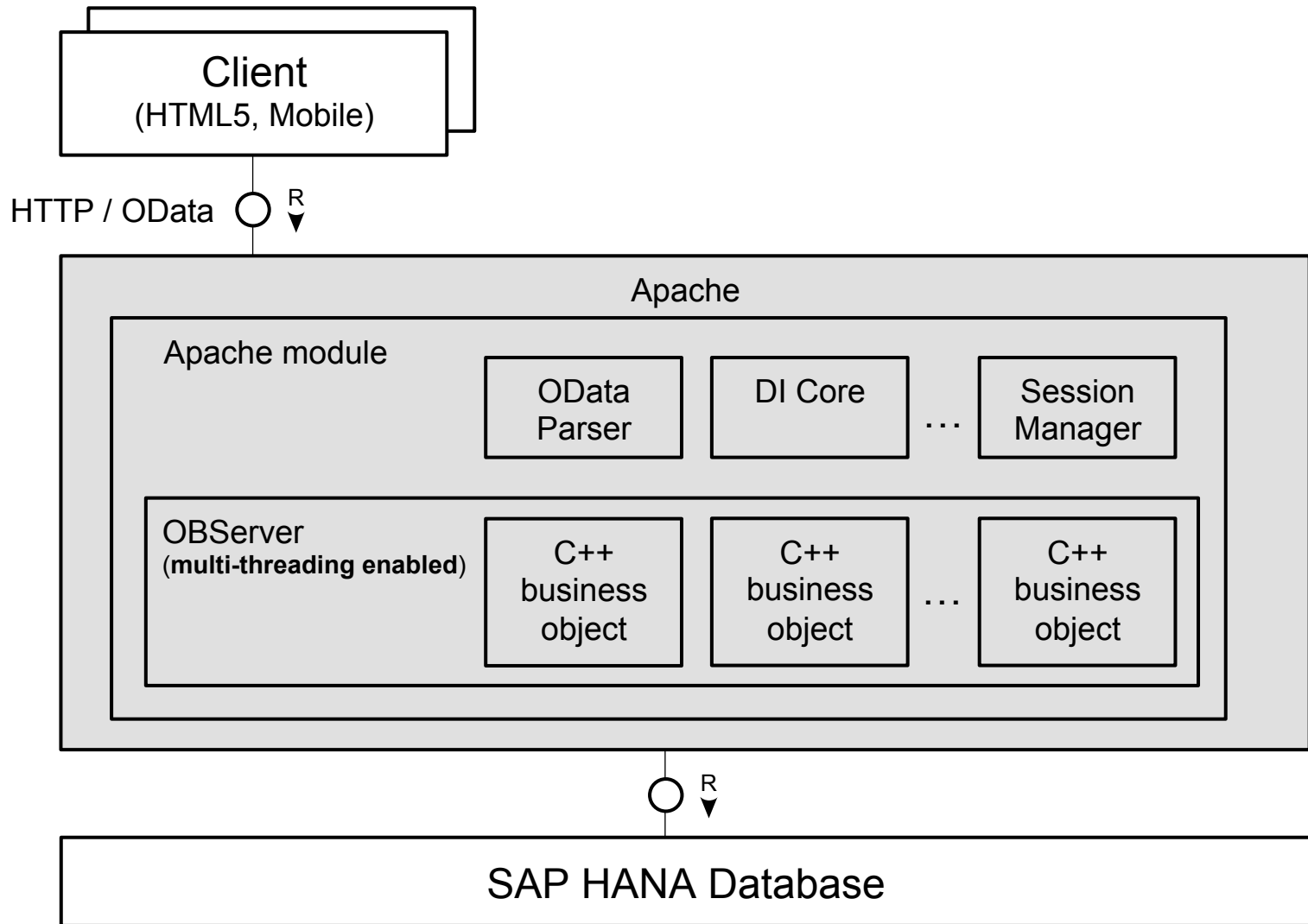
- ❑ Cater customer needs of using mobile and web apps
- ❑ High volume to support new customer segment and industries
- ❑ One Box for OLTP, OLAP and DEV platform



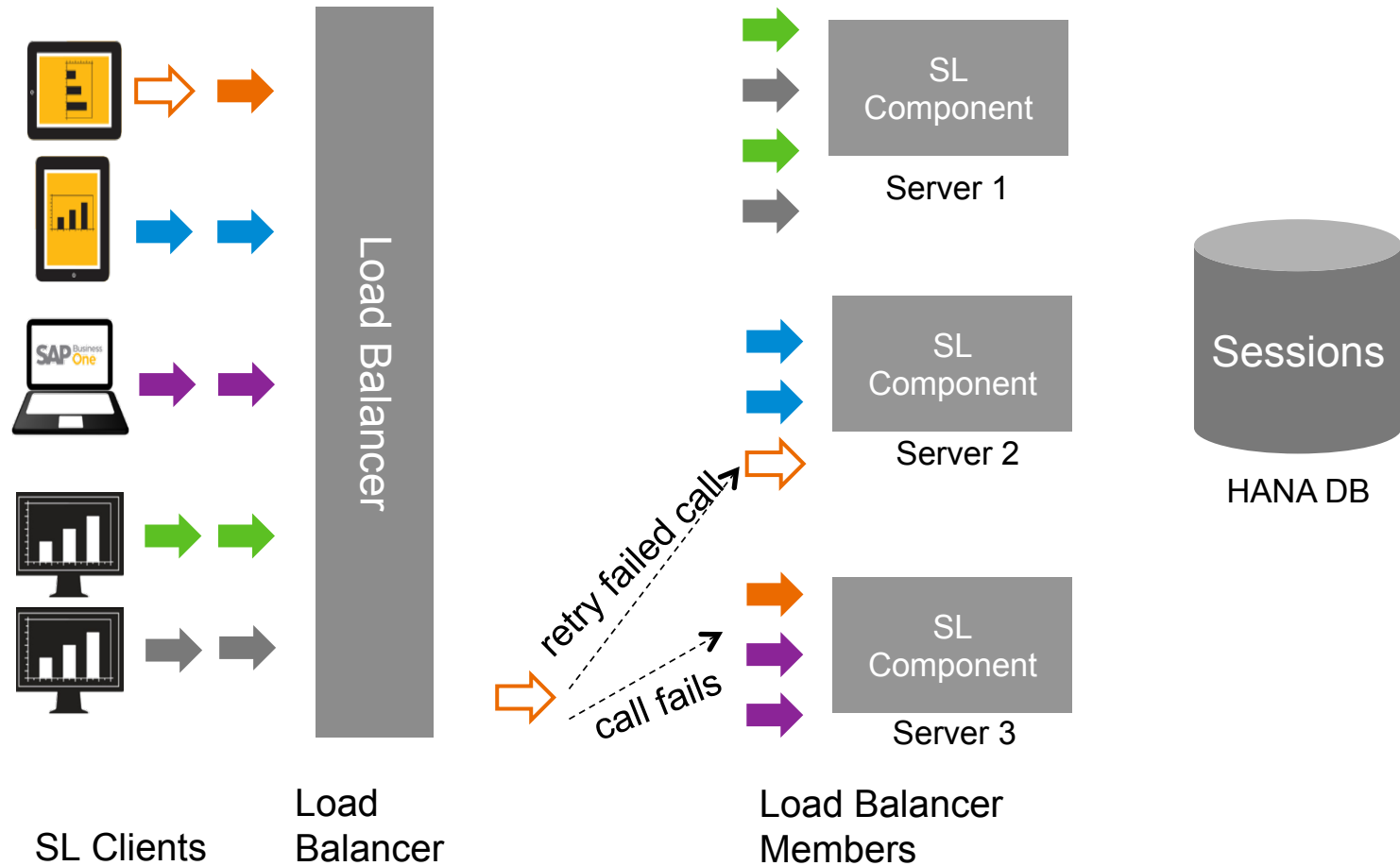
Global Architecture



Service Layer Architecture

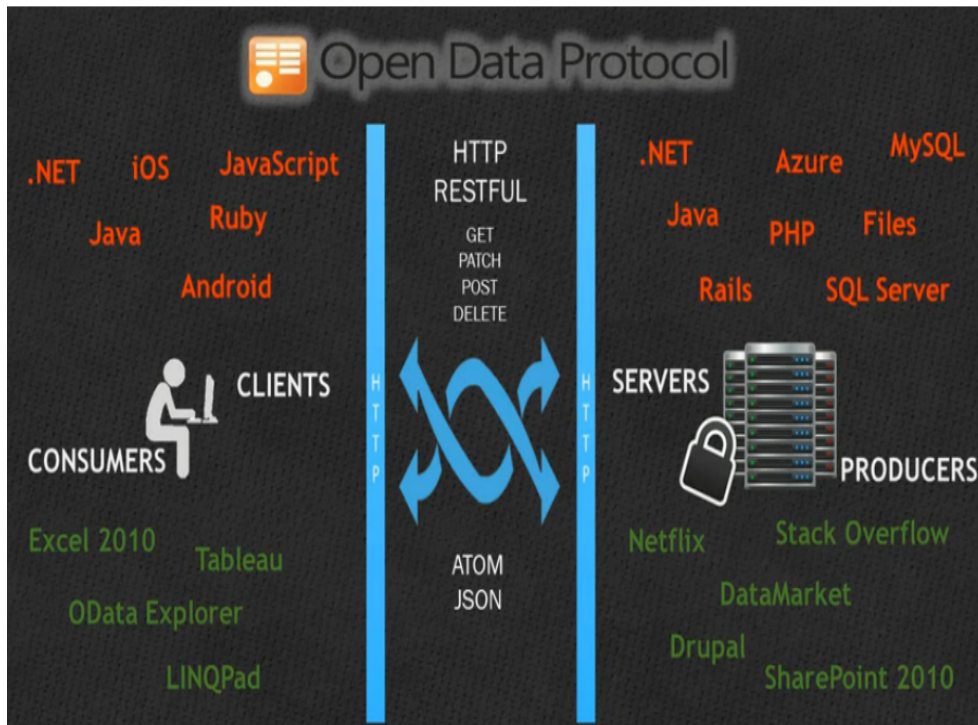


Scalability & High Availability



OData

Key points



Just HTTP

- Data as resources, HTTP methods to act on it

Uniform operations

- GET, POST, PATCH, PUT, DELETE always mean the same

Uniform way of representing structured data

- Atom
- JSON

Uniform URL conventions

- Navigation, filtering, sorting, paging, etc.

Any OData client can access data provided by any OData data source.



Connection via Tools

How to easily test, first contact with Service Layer

Tools: Advanced REST client

<http://chromerestclient.appspot.com/>



Advanced REST client

★★★★★ (2985) | [Developer Tools](#) | from [restforchrome.blogspot.com](#) | 231,366 users

[LAUNCH APP](#) 

OVERVIEWDETAILSREVIEWSRELATED

Advanced Rest Client

GETPOSTPUTPATCHDELETEHEADOPTIONSOther

RawFormHeaders

User-Agent: Chrome-extension2
Cookie: none

Status: 200 OK Loading time: 208 ms

Request headers: User-Agent: Chrome-extension2
Cookie: none

Response headers: Date: Mon, 15 Oct 2012 23:07:59 GMT
Server: GSE
Content-Type: application/json
Expires: Mon, 15 Oct 2012 22:10:34 GMT
Cache-Control: private, max-age=0
Last-Modified: Mon, 08 Oct 2012 07:41:17 GMT
ETag: W/\"C6Dfmsy7f7mWWhJaPKU\"
X-Content-Type-Options: nosniff
X-SS-Protection: 1; mode=block

RawJSONResponse

Copy to clipboard Save as file

```
{
  "version": "1.0",
  "encoding": "UTF-8",
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "xmlns:openSearch": "http://a9.com/-/spec/opensearchrss/1.0/",
    "xmlns:bloggers": "http://schemas.google.com/blogger/2008",
    "xmlns:georss": "http://www.georss.org/georss",
    "xmlns:gd": "http://schemas.google.com/gd/2005",
    "xmlns:slurp": "http://slurp.org/syndication/thread/1.0",
    "id": "tag:blogger.com,1999:blog-6420037820872138426",
    "updated": [
      {
        "id": "2012-10-08T09:41:17-05:00"
      }
    ]
  }
}
```

The web developers helper program to create and test custom HTTP requests.

Advanced REST client for Google Chrome.

Rate it, comment it, share it!

Users reviews:

- "This should be a benchmark for what chrome apps should be like.", Karan Kapoor
- "The best and the simplest client I've ever used. The new GWT look is much better. Thumbs up!!! Keep up the awesome work!", Denis Jajčević
- "very powerful yet simple to use", Diego Guidi

Weekly over 100k developers use this application. So many people can't be wrong!

About background page requirements:
<http://restforchrome.blogspot.com/2013/05/background-page-for-advanced-rest-client.html>

RELATED

REST Console
★★★★★ (291)


Postman - REST Client
★★★★★ (1507)


Dev HTTP Client
★★★★★ (625)


Google Apps Script
★★★★★ (291)


Tools: Postman

<http://www.getpostman.com/>



Postman - REST Client

de www.rickcreation.com

★★★★★ (2632) [Herramientas para desarrolladores](#) 725.237 usuarios

[DESCRIPCIÓN GENERAL](#) [OPINIONES](#) [AYUDA](#) [RELACIONADOS](#)

g+1

1.695

Funciona sin conexión

Postman helps you be more efficient while working with APIs. Postman is a scratch-your-own-itch project. The need for it arose...

Postman helps you be more efficient while working with APIs. Postman is a scratch-your-own-itch project. The need for it arose while one of the developers was creating an API for his project. After looking around for a number of tools, nothing felt just right. The primary features added initially were a history of sent requests and collections.

A number of other features have been

[Sitio web](#)
[Notificar uso inadecuado](#)
Versión: 0.8.4.12
Última actualización: 15 de enero de 2015
Tamaño: 1.25MB
Idioma: English

Postman demo

POSTMAN

History Collections

HTTPbin for testing

Postman demo

HTTPS POST

Normal

Basic Auth

OAuth 1.0

Postman sample

HTTPS POST parameters

With parameters

URL: http://post

POST

URL params

Headers (1)

form-data

x-www-form-urlencoded

raw

JSON

{}
{"something": "test"}

Send

Save

Add to collection

Reset

Body

Preview

STATUS 200 OK

Time 424 ms

Pretty

Raw

Preview

JSON

XML

```
{
  "form": {},
  "headers": {
    "Host": "www.httpbin.org",
    "Accept-Encoding": "gzip, deflate, sdch",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/34.0.1312.52 Safari/537.17",
    "Content-Type": "application/json",
    "Connection": "close"
  }
}
```

0:00 / 4:15

YouTube

Demo: Login service - request

The screenshot shows the Advanced Rest Client application interface. The URL bar displays `chrome-extension://hgmloofddfnphfjgcelkdfbfbjeloo/RestClient.html`. The main panel shows a request configuration for `https://frncel6:50000/b1s/v1/Login` using the `POST` method. The request body is in the `Form` tab, displaying a JSON payload: `{ "UserName": "manager", "Password": "1234", "CompanyDB": "SBODEMOUS" }`. The `Content-Type` header is set to `application/x-www-form-urlencoded`. A large yellow **DEMO** watermark is overlaid on the interface. The status bar at the bottom indicates a `200 OK` response with a loading time of 873 ms. The request details show the `User-Agent` string: `Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36`.

OData

Terminology

Uses the **Entity Data Model** (EDM) to describe the data exposed by an OData service.

Provides an **OData Metadata Document** representing the service's data model exposed for client consumption.

Entity

Entity types are nominal structures with a key.

OData protocol defines standard way to create/retrieve/update/delete entity (CRUD).

Action

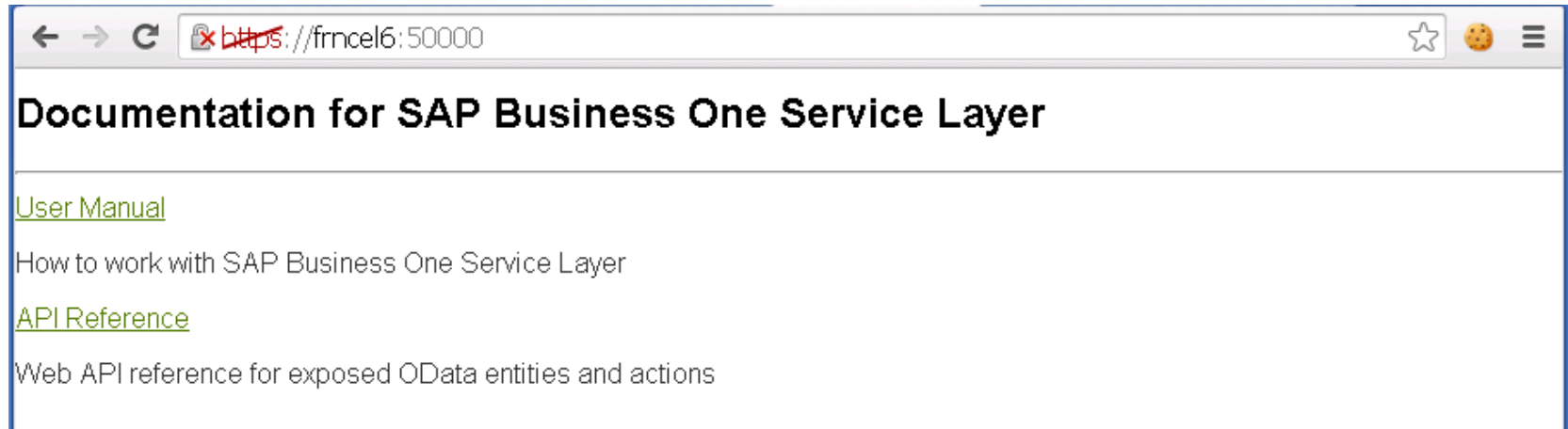
Actions are operations exposed by an OData service.

Actions may return data and are not composable.

Actions are defined in metadata.

Documentation

<https://<server>:<Load Balancer Port>/>



User Manual

Table of Contents	
1.1 About This Document	
This document covers the basic usages of SAP Business One Service Layer and explains the technical details of building a stable, scalable Web service using SAP Business One Service Layer.	
1	Introduction 6
1.1	About This Document 6
1.2	Target Audience 6
1.3	About SAP Business One Service Layer 6
2	Getting Started 7
2.1	System Requirements 7
2.2	Architecture Overview 7
2.3	Installing SAP Business One Service Layer 8
3	Consuming SAP Business One Service Layer 13
3.1	Login and Logout 13
3.2	Metadata Document 15
3.3	Service Document 17
3.4	Create/Retrieve/Update/Delete (CRUD) Operations 18
3.4.1	Creating Entities 18
3.4.2	Retrieving Entities 20
3.4.3	Updating Entities 21
3.4.4	Deleting Entities 22
3.5	Actions 23
3.6	Query Options 26
3.7	Batch Operations 29
3.7.1	Batch Request Method and URI 29
3.7.2	Batch Request Headers 29
3.7.3	Batch Request Body 29
3.7.4	Change Sets 30
3.7.5	Batch Request Sample Codes 31
3.7.6	Batch Response 32

API Reference

 <https://fmccl6:50000/Service%20Layer%20API%20Reference.html>

Service Layer API Reference

This document contains all entities and actions exposed through the Service Layer API. Service Layer API follows [OData 3.0 standard](#). OData defines a uniform REST API to create and consume data. So it is possible for users to leverage [OData libraries](#) to consume data exposed through Service Layer.

Actions:

Actions are used to extend the set of operations that can be performed on or with a service or entities. For example, the standard CRUD (create/read/update/delete) operations cannot be used to cancel a sales order, so one action is added to handle it.

Cancel	Show/Hide List Operations Expand Operations
Close	Show/Hide List Operations Expand Operations
Login	Show/Hide List Operations Expand Operations
Logout	Show/Hide List Operations Expand Operations
CompanyService	Show/Hide List Operations Expand Operations

Service Layer API Reference

BusinessPartners entity (1/2)

BusinessPartners

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

The BusinessPartners object represents the business partners master data in the Business Partners module. You can use this data to record and retrieve business partner (customers, vendors, and leads) information and schedule business partner activities.

GET BusinessPartners

Retrieves business partner data.

Parameters

\$filter	Restrict the set of business objects returned.
\$orderby	Specify the order in which business objects are returned from the service.
\$select	Restrict the service return to only the properties requested by the client.
\$skip	Specify that the result excludes the first n entities.
\$top	Specify that only the first n records should be returned.

POST BusinessPartners

Creates a business partner.

Example

```
POST https://localhost/bls/v1/BusinessPartners
{
  "CardCode": "c001",
  "CardName": "c001",
  "CardType": "C"
}
```

Service Layer API Reference

BusinessPartners entity (2/2)

BusinessPartners('id')

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

The BusinessPartners object represents the business partners master data in the Business Partners module. You can use this data to record and retrieve business partner (customers, vendors, and leads) information and schedule business partner activities.

GET BusinessPartners('id')

Retrieves a business partner by its key.

Parameters

\$select Restrict the service return to only the properties requested by the client.

PUT BusinessPartners('id')

Updates a business partner by replacing all property values with those specified in the request body. Missing properties will be set to their default values.

PATCH BusinessPartners('id')

Updates a business partner by replacing exactly those property values that are specified in the request body. Missing properties will not be altered.

Example

```
PATCH https://localhost/b1s/v1/BusinessPartners('c001')
{
  "CardName": "Updated customer name"
}
```

DELETE BusinessPartners('id')

Removes a business partner.

Service Layer API Reference

oData queries

GET	BusinessPartners
Retrieves business partner data.	
Parameters	
\$filter	Restrict the set of business objects returned.
\$orderby	Specify the order in which business objects are returned from the service.
\$select	Restrict the service return to only the properties requested by the client.
\$skip	Specify that the result excludes the first n entities.
\$top	Specify that only the first n records should be returned.

```
GET /BusinessPartners?$filter=CardCode eq 'I004`
GET /BusinessPartners?$filter=startswith(CardCode,'post')
GET /BusinessPartners?$orderby=CardCode desc
GET /BusinessPartners?$select=CardCode,CardName
GET /BusinessPartners/$count GET /BusinessPartners?$top=2
GET /BusinessPartners?$top=2&$skip=4
```

Service Layer API Reference

Batch Operations

Batch Operations

Service Layer supports executing multiple operations sent in a single HTTP request through the use of batching.

A batch request is submitted as a single HTTP POST request to the batch endpoint of a service, located at the URI \$batch relative to the service root.

```
POST https://servicelayerhost:8443/b1s/v1/$batch
```

Change Set

A change set is an atomic unit of work inside a Batch Operation.

Any failed sub request in a change set will cause the whole change set to be rolled back.

Change sets must not contain any GET requests or other change sets.

```
--<Batch Boundary>  
Content-Type: multipart/mixed;boundary=<Changeset Boundary>
```

```
--<Changeset boundary>  
  
Content-Type:application/http  
Content-Transfer-Encoding:binary  
  
Content-ID:1  
POST /b1s/v1/Items  
<Json format Items Content>
```

```
--<Changeset boundary>  
  
Content-Type:application/http  
Content-Transfer-Encoding:binary  
  
Content-ID:2  
PATCH /b1s/v1/$1  
<Json format Item update content>
```

```
--<Changeset boundary>--  
  
--<Batch Boundary>--
```

Service Layer API Reference

Paging

Implemented via top and skip, allows the data to be fetched chunk by chunk.

Example:

When selecting all orders:

```
GET /Orders
```

Service Layer returns:

```
HTTP/1.1 200 OK
```

```
{
  "value": [
    {"DocEntry": 7, "DocNum": 2, ...},
    {"DocEntry": 8, "DocNum": 3, ...},
    ...
    {"DocEntry": 26, "DocNum": 21, ...}
  ],
  "odata.nextLink": "/b1s/v1/Orders?$skip=20" }
```

Note: Default page size is 20 (0 = no paging).

It can be customized by:

- Configuration option `PageSize` in `conf/b1s.conf`.
- Use OData recommended annotation `odata.maxpagesize` in the request header

Service Layer API Reference

Actions

Provide functions besides basic entity CRUD.

Use HTTP verb POST.

- **bound action** (bound to entity for operations except CRUD)

Ex: POST /Orders(22)/Close

- **global action** (often used to expose B1 services).

Ex: POST /ActivitiesService.AddActivity

```
{
  "Activity": {
    "ActivityCode": 1,
    "CardCode": "c1"
  }
}
```

Service Layer API Reference

Navigation and Associations

Two entities may be associated (independently related).

Example:

Order and customer entities in the metadata are associated and a navigation is defined:

```
GET Orders(1)/BusinessPartner
```

The above request is equal to (if you already knew that the CardCode property of the order is "c1")

```
GET BusinessPartners('c1')
```

You can continue to operate on this entity as on GET BusinessPartners('c1').

For example, to get the foreign name of the customer, the following two requests are also equal:

```
GET Orders(1)/BusinessPartner/ForeignName
```

```
GET BusinessPartners('c1')/ForeignName
```

Service Layer metadata

Metadata describes the capability of the service. It mainly defines types, entities (e.g. B1 objects) and actions (e.g. B1 services).

HTTP request to retrieve metadata:

```
GET /$metadata
```

Please refer to [OData-CSDL](#) (Common Schema Definition Language) for detail on the format of the metadata.

Let's check how the entities and actions are exposed. Take B1 business partners and sales orders as examples. We can find such sections in the metadata:

```
▼<EnumType Name="BoCardTypes">
  <Member Name="cCustomer"/>
  <Member Name="cSupplier"/>
  <Member Name="cLid"/>
</EnumType>

▼<EntityType Name="BusinessPartner">
  ▼<Key>
    <PropertyRef Name="CardCode"/>
  </Key>
  <Property Name="CardCode" Nullable="false" Type="Edm.String"/>
  <Property Name="CardName" Type="Edm.String"/>
  <Property Name="CardType" Type="Edm.String"/>
  <Property Name="GroupCode" Type="Edm.Int32"/>
  <Property Name="Address" Type="Edm.String"/>
  <Property Name="ZipCode" Type="Edm.String"/>
  <Property Name="MailAddress" Type="Edm.String"/>
  <Property Name="MailZipCode" Type="Edm.String"/>
  <Property Name="Phone1" Type="Edm.String"/>
  <Property Name="Phone2" Type="Edm.String"/>
```

- B1 Objects (CRUD operations)
- B1Services
- Enumerations
- UDFs/UDTs/UDOs
- Navigation/Associations





.NET on Windows

Windows Communication Foundation (WCF)

Some theory

Windows Communication Foundation (WCF) is a framework for building service-oriented applications.

WCF Data Services...

... allows you to send data as asynchronous messages from one service endpoint to another

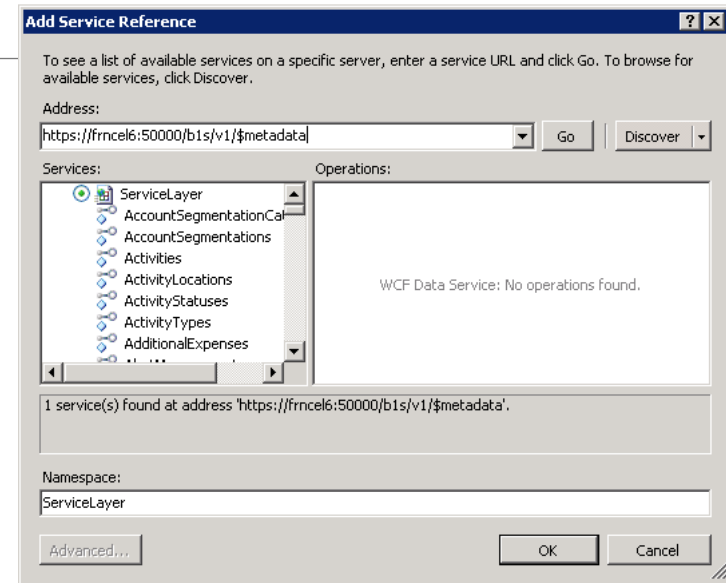
...enables you to create services that use the Open Data Protocol (OData) to expose and consume data over the Web or intranet by using the semantics of representational state transfer (REST) for addressing and updating resources.

... includes a set of client libraries, one for general .NET Framework client applications.

...its client libraries provide an object-based programming model when you access an OData feed from environments such as the .NET Framework.

Windows Communication Foundation (WCF) Sample

Add a reference to the Service Layer Open Data Protocol (OData) metadata



Access the OData feed from the application by using the generated client data service classes and WCF Data Services client library.

```
Document order = new ServiceLayer.SAPB1.Document();
order.CardCode = strCurrentBusinessPartnerCardCode;
order.DocDueDate = DateTime.Today;
order.DocDate = DateTime.Today;
order.DocType = "dDocument_Items";
order.DownPaymentType = "tYES";
```

```
//Add lines
DocumentLine newLine = GetNewDocumentLine();
order.DocumentLines.Add(newLine);
```

```
currentServiceContainer.AddToOrders(order);
DataServiceResponse response =
currentServiceContainer.SaveChanges();
```

Windows Communication Foundation (WCF) Demo

The screenshot displays a Windows Forms application titled "Windows Forms Demo App - Consume Service Layer Odata Service by WCF". The interface is divided into several sections. At the top, a "Connection Management" panel includes fields for "Service URL" (https://frncel6:50000/b1s/v1/), "CompanyDB" (SBODEMOUS), "User Name" (manager), and "Password" (xxxx), along with a green "Logout" button. Below this is an "Insert Order" panel with fields for "DocEntry", "Customer" (C20000), and "Item Number" (CAR), and buttons for "Add New Order", "Update Order [Update Comments]", "Update Order [Add Doc Line]", "Delete Order", and "Close Order [Bound Action]". The main area shows the results of a POST request to the OData service, including the URI, headers, and body. A large yellow "DEMO" watermark is overlaid on the right side of the image.



Client side Java Script

SAP Business One Service Layer JavaScript - jQuery

```
<script type="text/javascript" src="thirdparty/jquery.min.js"></script>

$.ajax({
  type: "POST",

  url: "https://hanaserver:50000/bls/v1/Login",

  xhrFields: { withCredentials: true },

  data: '{"UserName": "' + user + '", "Password": "' + pwd + '", "CompanyDB": "' + db + '"}',

  dataType: "json",

  error: function (err) {
    window.alert('Error:' + err.responseJSON.error.message.value);
    connected = false;
  },

  success: function () {
    window.alert('Successfully connected!');
    connected = true;
  }
});
```

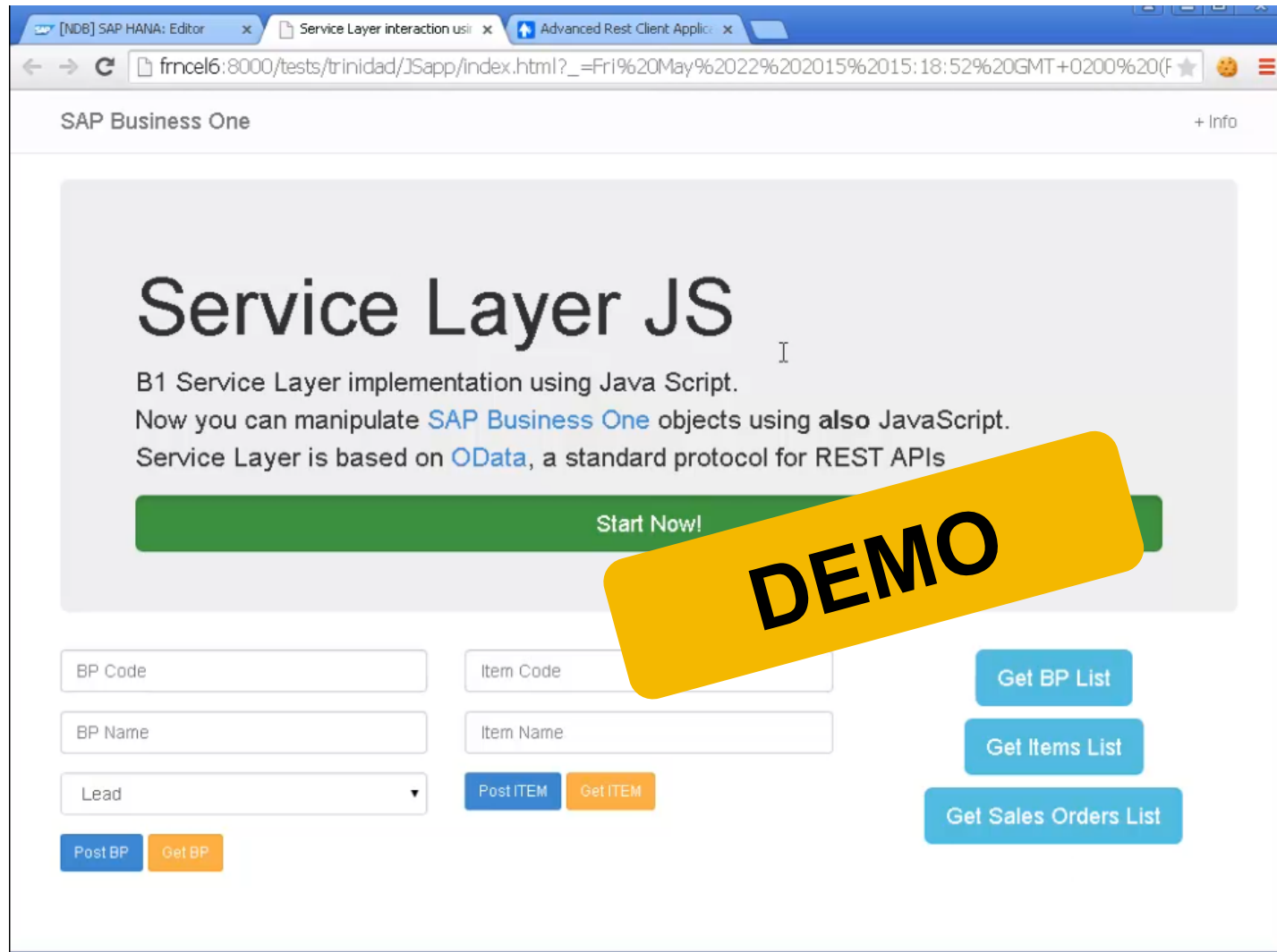
Note1: Browsers might block cross site requests.

Note2: Service Layer CORS support not yet out of the box, can be manually configured.

Sample available at SCN blog:

B1 Service Layer with JavaScript - Handling B1 Objects without .Net (nor DI API)

Client side Java Script Demo





Server Side Java Script

SAP HANA Extended Application Services (XS) Technologies

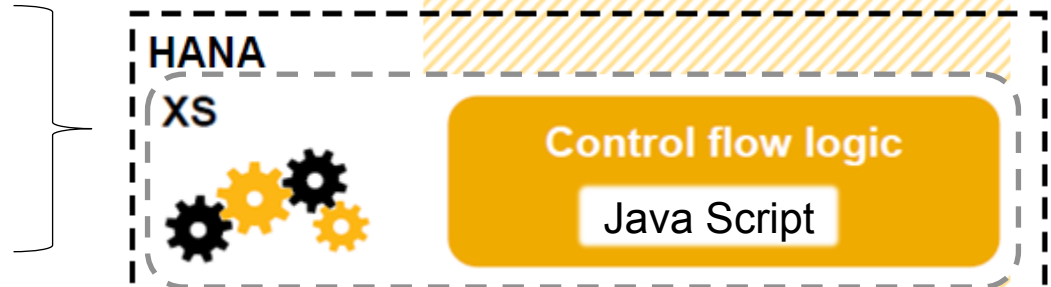
Front-end Technologies

- http/s
- HTML5 / SAPUI5
- Client-Side JavaScript



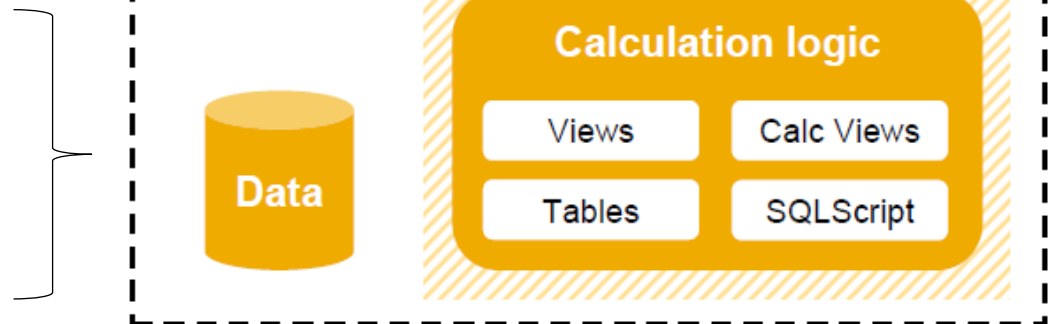
Control Flow Technologies

- OData
- Server-Side JavaScript(XSJS)
- XML for Analytics(XMLA)



Data Processing Technologies

- SQL / SQLScript
- Calculation Engine Functions
- Application Function Library (AFL)



SAP Business One Service Layer

Server Side JavaScript (xsjs)

```
var client = new $.net.http.Client();
var req = new $.web.WebRequest($.net.http.POST, "");

// Session ID stored after login
req.cookies.set("B1SESSION", sessionId);

// Body contains BusinessPartner details
req.setBody(body);

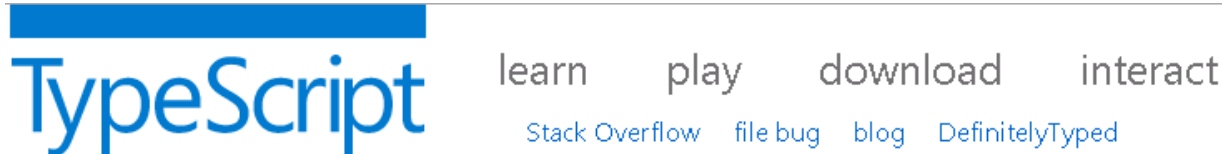
client.request(req, "https://server:50000/b1s/v1/BusinessPartners");

var response = client.getResponse();
```

SAP Business One Service Layer

Java Script - TypeScript

<http://www.typescriptlang.org/>



TypeScript lets you write JavaScript the way you really want to.
TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.
Any browser. Any host. Any OS. Open Source.

Types enables static checking, symbol-based navigation, statement completion and code refactoring.
TypeScript compiles to JavaScript code which runs on any browser, in Node.js, or in any other ES3-compatible environment.

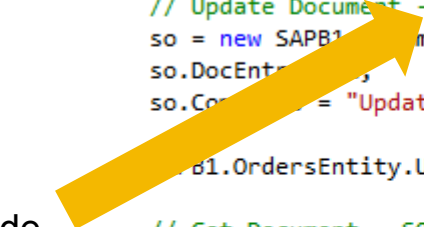
SAP Business One Service Layer TypeScript for B1 Service Layer

Autogenerated TypeScript classes from SAP Business One Service Layer metadata.

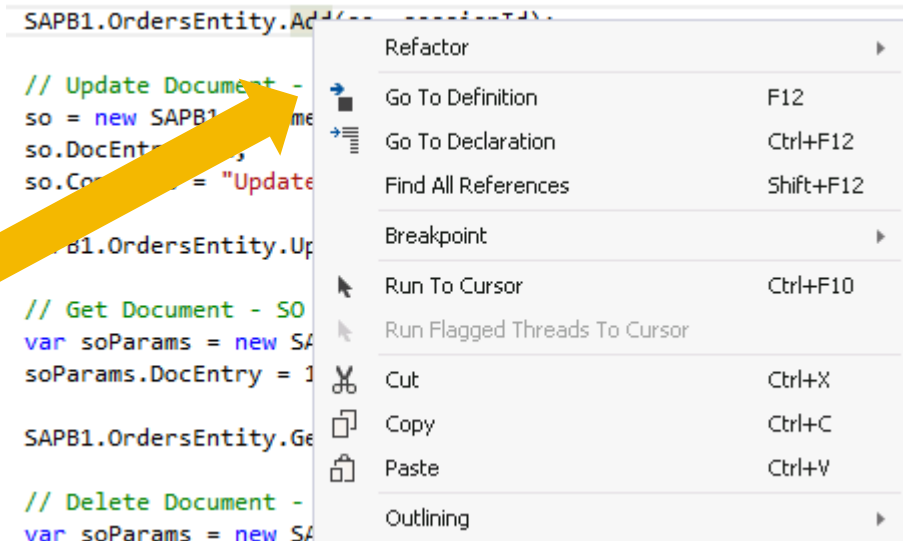
```
var so = new SAPB1.Document();
so.DocType = SAPB1.BoDocumentTypes.dDocument_Items;
so.CardCode = "C20000";
so.DocDueDate = new Date(Date.now());
so.
so. AuthorizationStatus
so. BaseAmount
so. BaseAmountFC
so. BaseAmountSC
so. BillOfExchangeReserved
so. BlanketAgreementNumber
so. BlockDunning
so. Box1099
so. BPChannelCode
so.DocumentLines[1].Quantity = 3;
so.DocumentLines[1].Price = 30;
```



Code auto completion and documentation in TypeScript language



Navigation through compiled code



SAP Business One Service Layer

Generate JavaScript out of TypeScript

B1 TypeScript Classes Generator Tool

Provides TypeScript classes representing all B1 Objects and Services (including UDTs/UDFs/UDOs).

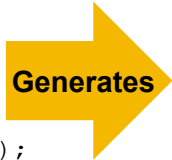
TypeScript compiled code generates JavaScript code that can be directly run by SAP HANA XS Engine.

TypeScript

```
// Add Document
var so = new SAPB1.Document();
so.DocType = SAPB1.BoDocumentTypes.dDocument_Items;
so.CardCode = "C20000";
so.DocDueDate = new Date(Date.now());
so.DocDate = new Date(Date.now());
so.DocTime = new Date(Date.now());
```

```
so.DocumentLines = new Array<SAPB1.DocumentLine>();
so.DocumentLines[0] = new SAPB1.DocumentLine();
so.DocumentLines[0].ItemCode = "A00001";
so.DocumentLines[0].Quantity = 2;
so.DocumentLines[0].Price = 20;
so.DocumentLines[1] = new SAPB1.DocumentLine();
so.DocumentLines[1].ItemCode = "A00002";
so.DocumentLines[1].Quantity = 3;
so.DocumentLines[1].Price = 30;
```

```
SAPB1.OrdersEntity.Add(so, sessionId);
```



Generates

JavaScript

```
// Add Document
var so = new SAPB1.Document();
so.DocType = SAPB1.BoDocumentTypes.dDocument_Items;
so.CardCode = "C20000";
so.DocDueDate = new Date(Date.now());
so.DocDate = new Date(Date.now());
so.DocTime = new Date(Date.now());
```

```
so.DocumentLines = new Array();
so.DocumentLines[0] = new SAPB1.DocumentLine();
so.DocumentLines[0].ItemCode = "A00001";
so.DocumentLines[0].Quantity = 2;
so.DocumentLines[0].Price = 20;
so.DocumentLines[1] = new SAPB1.DocumentLine();
so.DocumentLines[1].ItemCode = "A00002";
so.DocumentLines[1].Quantity = 3;
so.DocumentLines[1].Price = 30;
```

```
SAPB1.OrdersEntity.Add(so, sessionID);
```

SAP Business One Service Layer HANA XS Demo

Extreme App Platform Portal x

frncel6:8000/tests/trinidad/TestTSSL/SOOperations.html?sessionID=vZki1WRA-dql ☆ 🍪 ☰

Service Layer operations

Sales Orders operations

DocEntry

CardCode

Line 1

ItemCode

Quantity

Price

Line 2

Line2 ItemCode

Quantity

Price

 © SAP AG. 2013

DEMO



Links

Service Layer

[Working with SAP Business One Service Layer in SAP Business One 9.1, version for SAP HANA – partneredge](#)

[SAP Business One, version for SAP HANA Extensibility topics – SCN](#)

SAP HANA

[openSAP online courses](#)

- [Introduction to Software Development on SAP HANA](#)
- [Next Steps in Software Development on SAP HANA](#)

<http://developers.sap.com/hana>

http://help.sap.com/hana_appliance

<http://www.saphana.com/community/implement/hana-academy>

oDATA

<http://www.odata.org>



Thank you

Contact information:

Trinidad Martinez
SAP Business One Solution Architects
[@TrinidadAtSAP](#)

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

© 2015 SAP SE oder ein SAP-Konzernunternehmen. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP SE oder ein SAP-Konzernunternehmen nicht gestattet.

SAP und andere in diesem Dokument erwähnte Produkte und Dienstleistungen von SAP sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP SE (oder von einem SAP-Konzernunternehmen) in Deutschland und verschiedenen anderen Ländern weltweit. Weitere Hinweise und Informationen zum Markenrecht finden Sie unter <http://global.sap.com/corporate-de/legal/copyright/index.epx>.

Die von SAP SE oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.

Produkte können länderspezifische Unterschiede aufweisen.

Die vorliegenden Unterlagen werden von der SAP SE oder einem SAP-Konzernunternehmen bereitgestellt und dienen ausschließlich zu Informationszwecken. Die SAP SE oder ihre Konzernunternehmen übernehmen keinerlei Haftung oder Gewährleistung für Fehler oder Unvollständigkeiten in dieser Publikation. Die SAP SE oder ein SAP-Konzernunternehmen steht lediglich für Produkte und Dienstleistungen nach der Maßgabe ein, die in der Vereinbarung über die jeweiligen Produkte und Dienstleistungen ausdrücklich geregelt ist. Keine der hierin enthaltenen Informationen ist als zusätzliche Garantie zu interpretieren.

Insbesondere sind die SAP SE oder ihre Konzernunternehmen in keiner Weise verpflichtet, in dieser Publikation oder einer zugehörigen Präsentation dargestellte Geschäftsabläufe zu verfolgen oder hierin wiedergegebene Funktionen zu entwickeln oder zu veröffentlichen. Diese Publikation oder eine zugehörige Präsentation, die Strategie und etwaige künftige Entwicklungen, Produkte und/oder Plattformen der SAP SE oder ihrer Konzernunternehmen können von der SAP SE oder ihren Konzernunternehmen jederzeit und ohne Angabe von Gründen unangekündigt geändert werden. Die in dieser Publikation enthaltenen Informationen stellen keine Zusage, kein Versprechen und keine rechtliche Verpflichtung zur Lieferung von Material, Code oder Funktionen dar. Sämtliche vorausschauenden Aussagen unterliegen unterschiedlichen Risiken und Unsicherheiten, durch die die tatsächlichen Ergebnisse von den Erwartungen abweichen können. Die vorausschauenden Aussagen geben die Sicht zu dem Zeitpunkt wieder, zu dem sie getätigt wurden. Dem Leser wird empfohlen, diesen Aussagen kein übertriebenes Vertrauen zu schenken und sich bei Kaufentscheidungen nicht auf sie zu stützen.

The Grid

Headline area

White space

Drawing area