

How to Create Facebook Applications using SAP Gateway Services



Applies to:

SAP Netweaver Gateway 2.0 sp02 and Facebook Integration.

Summary

How to guide aims to explain the process of consuming SAP Netweaver Gateway services and build a Facebook application.

Author: Ravi Sharma

Company: C2C Consulting

Created on: 10 January 2012

Author Bio



Ravi Sharma is a Development Architect and solution manager working for C2C consulting in Germany. His current Tasks: Duet Enterprise For Microsoft SharePoint and SAP, SAP NetWeaver Gateway and iPhone / Android Apps, SAP ESOA Tools & Utilities SAP IDM & GRC Integration. Knowledge of SAP ABAP, Cross Applications, ALE/IDOC, RFC's qRFC, tRFC, BADI, OOP's Concepts, CCMS and Solution Manager, SAP RFID, SAP Dynpro, SAP ABAP Webdynpro, SAP Net weaver, DUET 1.5, ABAP and Java Web services.

Table of Contents

Conventions:	3
Prerequisites:	3
How to- Guide:	3
Scenario description:	4
Solution:	5
Login to the facebook account	5
Go to developer section	5
Create an application	5
Fill the all data required	5
Extract the OData SDK	6
Installation of framework	7
Get the OData service URL	8
Create Proxy Class for Web Service using OData SDK	8
Start Apache	10
Create an Application	10
Netbeans and application formation	11
Design the application	11
Buy a Server	12
Run the first draft on localhost	12
Create facebook API instance	12
HTML Formatting	12
Run the first draft on facebook	13
Go deep into code	13
Ready to run	16
OData Concepts	16
Data Service Query Expressions	16
Running Query Expressions	16
DataServiceQuery Class	16
QueryOperationResponse Class	17
OData Exception Classes	19
DataServiceRequestException class	19
ODataServiceException class	19
InvalidOperation class	20
ACStilException class	20
Related Content	21
Copyright	22

Conventions:

- All the examples of this tutorial are based on Windows OS and Xampp
- We will use Netbeans IDE for PHP for Development purpose.
- Codes snippets are written like
`C:\PHPLib\ odataphp>php PHPDataSvcUtil.php /config=D:/myApp/settings.ini`
- Links are as
<http://www.facebook.com/developers/>

Prerequisites:

- PHP Server with cURL and XSL enabled (This tutorials follows Xampp)
- Netbeans IDE
- OData-SDK for PHP
- Facebook SDK for PHP
- A facebook account

How to- Guide:

- **How to Install PHP Server?**
 - Download XAMPP form
<http://sourceforge.net/projects/xampp/files/>
 - XAMPP is a very easy to install Apache Distribution for Linux, Solaris, Windows and Mac OS X. The package includes the Apache web server, MySQL, PHP, Perl, a FTP server and phpMyAdmin.
- **How to enable cURL and XSL?**
 - Go to <xampp installation folder> → Php → php.ini
 - Look for
`;extension=php_curl.dll`
and remove the comment as
`extension=php_curl.dll`
 - By default XSL is enabled in xampp for windows
- **How to get OData-SDK for PHP?**
 - Download it from <http://odataphp.codeplex.com/>
- **How to get Facebook SDK for PHP?**
 - Download it from <https://github.com/facebook/php-sdk/zipball/master>
- **How to create Facebook Account?**
 - Although I do not need to add this in How to guide. C'mon it is Facebook. ☺
 - <http://facebook.com> is enough

Scenario description:

Open facebook application



SAP Data in tabular format

S.No.	Name	Value	Scheme ID	Scheme Agency ID
1.	Tools	10_0000000001	ZTCUST5	USD_004
2.	Targo Ltd.	10_0000000002	ZTCUST5	USD_004
3.	Cell Computers	10_0000000006	ZTCUST5	USD_004
4.	Heavy Tool Pit	10_0000000011	ZTCUST5	USD_004
5.	Software Corner	10_0000000016	ZTCUST5	USD_004
6.	Chem Bros	10_0000000026	ZTCUST5	USD_004
7.	CGR Industries	10_0000000031	ZTCUST5	USD_004
8.	BBB Industries	10_0000000032	ZTCUST5	USD_004
9.	Advance Product GmbH	10_0000000036	ZTCUST5	USD_004
10.	Andres Consulting AG	10_0000000041	ZTCUST5	USD_004



Show details in pop up for each row



With additional facebook functionalities

Tell us what do you think of SAP Service for Materials?

Like One person likes this. Be the first of your friends.

Add a comment...

 **Sap Sisoft** 3:32 pm
Hello!!
Message · Report

Facebook social plugin Displaying the only post.

Solution:

Now after understating the scenario, here in this section we have the complete solution of it.

Login to the facebook account

- www.facebook.com → Login with your Email and Password



The image shows the Facebook login interface. It features the Facebook logo on the left. On the right, there are two input fields: 'Email' and 'Password'. Below the 'Email' field is a checkbox labeled 'Keep me logged in'. To the right of the 'Password' field is a link for 'Forgotten your password?'. A 'Log in' button is positioned to the right of the 'Password' field.

Go to developer section

- You can use this link <http://www.facebook.com/developers/>

Create an application



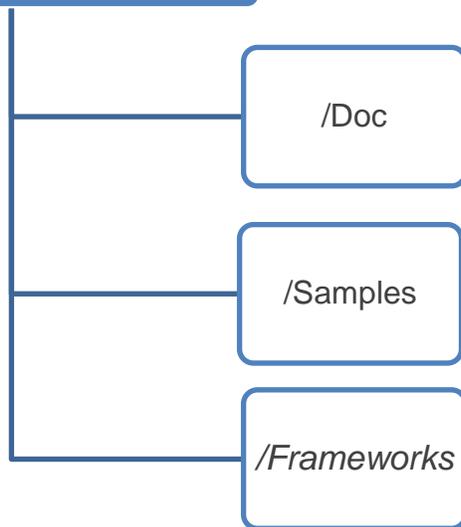
The image shows the Facebook Developer page. At the top, there is a navigation bar with the Facebook logo, a search bar, and links for 'Home' and 'Profile'. Below the navigation bar, there are links for 'Documentation', 'Forum', 'Policy', and 'Bookmark Developer App', along with a 'Help' link. The main content area features a 'Developer' header with a '+ Set Up New Application' button.

- Click on "Set up New Application"

Fill the all data required

- Application Name (Name of your choice)
- Facebook Integration Tab
 - Canvas URL: where your application is hosted: www.example.com/Sap_Demo/
 - Canvas Page: URL to access your application: www.apps.facebook.com/sap_demo/
 - Canvas Type: iFrame/FBML?

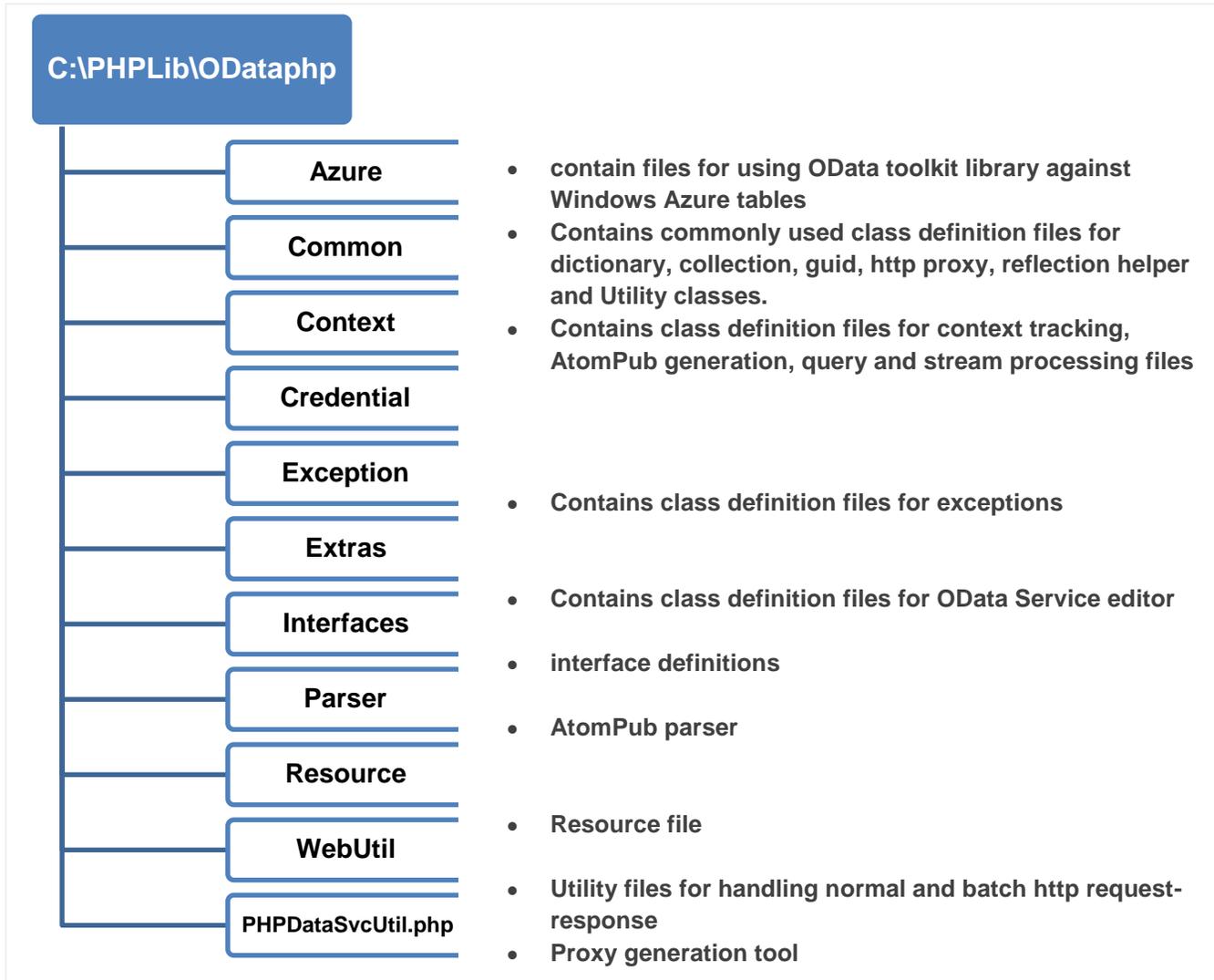
Extract the OData SDK

Package Contents

- It contains Toolkit User guide
- It contains the sample code that's explains the usages of OData SDK
- Contains the source code for the tools used to generate the proxy class and to communicate with data service.

Installation of framework

- Create a folder named 'odataphp' eg: C:\PHPLib\odataphp
- Copy the files and folders in the framework folder to the folder created above.
- Now your 'ODataphp' look like



- Search for 'Paths and Directories'
Add the path of the folder created above to the 'include_path' directive in php.ini `include_path =`
`";C:\PHPLib\ODataphp"`
- Just below the definition of 'include_path' directive, add the following two lines:
`; OData toolkit for PHP Library Pata`
`ODataphp_path = "C:\PHPLib\ODataphp"`

Get the OData service URL

- OData Services has file extension “.svc”. Example are below:
- Customer Services: <http://155.56.49.85:50000/sap/opu/sdata/ZTCUST5.svc/>
- Material Services: <http://155.56.49.85:50000/sap/opu/sdata/ZTMAT5.svc/>
- Contact Services: <http://155.56.49.85:50000/sap/opu/sdata/ZTCONT5.svc/>
- Check it in the URL whether it is accessible or not.
- You can browse the data of services using Silverlight tool. You can download it from <http://www.silverlight.net/content/samples/odataexplorer/>
- In case of Authentication, it only supports windows and Azure based authentication.

Create Proxy Class for Web Service using OData SDK

- Open cmd
- Navigate to xampp → Php folder
- Type

```
php <path of Client Library>PHPDataSvcUtil.php /uri=<data service Uri> | /metadata=<service
metadata file> [/out=<output file path>] [/auth=windows|acs /u=username /p=password
[/sn=servicenamespace /at=applies_to] ] [/ph=proxy-host /pp=proxy-port [/pu=proxy-user
/ppwd=proxy-password] [/ups=yes|no] ]
```

→ Where <Path of Toolkit Library> is the path where the toolkit library files are installed

Example usage:

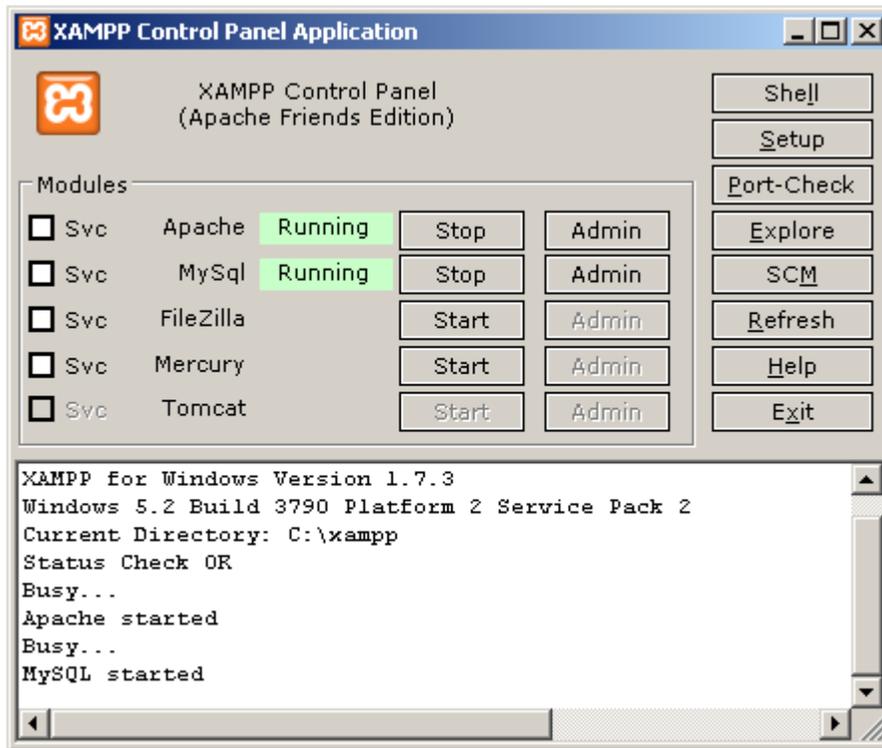
```
→ C:\PHPLib\odataphp>php PHPDataSvcUtil.php
/uri=http://localhost:13986/NorthWindDataServices.svc /out=D:
\samples\SimpleApplication
→ C:\PHPLib\odataphp>php PHPDataSvcUtil.php
/config=C:/xampp/htdocs/myApp/settings.ini
```

- Say file created is **DemoProxy.php**
- **Command-line parameters**

Option	Definition	Note
/config	Path to the configuration file.	The <i>PHPDataSvcUtil.php</i> can be used in two ways, either you can specify required options (described below) through command-line or you can define the same options in a configuration file and use /config option to specify path to the config file.
/uri	OData Service Uri	The <i>PHPDataSvcUtil.php</i> uses this Uri to retrieve the service metadata. e.g. http://155.56.49.85:50000/sap/opu/sdata/ZTCUST5.svc/
/metadata	Path to service meta data file.	If you already have the service metadata file saved on your local machine, you can use this option to specify the path to this metadata file.
/out	Output directory or file path	The output path where the generated proxy file is to be saved. If file name is not specified, then the Entity Container name (defined in the WCF Data Service) is used as filename. If this parameter is absent then proxy file will be generated in the current directory.
/auth	Type of authentication. Possible values are: windows / acs	This parameter is needed if access to the service requires authentication. Currently <i>PHPDataSvcUtil</i> supports two type of authentication. 1. Windows Authentication 2. Azure Access Control Service Authentication.
/u	Windows username or acs scope	If /auth is of type 'windows', then /u will be the windows username in the form domain/username. If /auth is of type 'acs', then /u will be the acs scope name.
/p	Windows password or acs issuer key	If /auth is of type 'windows, then /u will be the windows password. If /auth is of type 'acs', then /u will be the acs issuer key
/sn	acs service namespace	If /auth is of type 'acs', then /sn will be the acs service namespace.
/at	acs applies to	If /auth is of type 'acs', then /at will be the acs 'applies to' value
/ph	Http Proxy Host	If you are running behind a proxy, then these parameters are required. If your proxy requires authentication.
pp	Http Proxy port	
/pu	Http Proxy username	
/ppwd	Http Proxy password	
/ups	Use proxy settings for service request. Possible values are: Yes/no	By default the user specified proxy settings will be used while requesting metadata from OData Service. If you are using ACS auth and access to your service not require any proxy settings. (e.g. service running locally) then set this flag to no, /ups=no

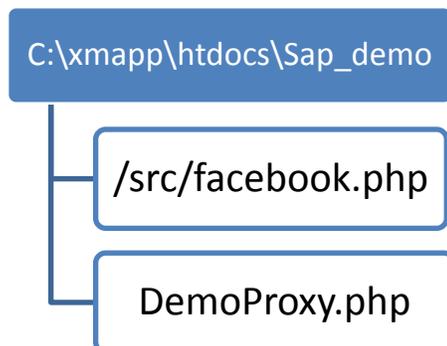
Start Apache

- To start apache Go to <xampp installation folder> → click xampp-control.exe
- You will get the following Screen, you have to just start the apache and mysql



Create an Application

- For creating an application you have to create a folder in <xampp installation folder> → htdocs
Say Sap_Demo.
- Add Facebook API folder and DemoProxy.php file created in Step 8.
- At this stage your directory structure will look like



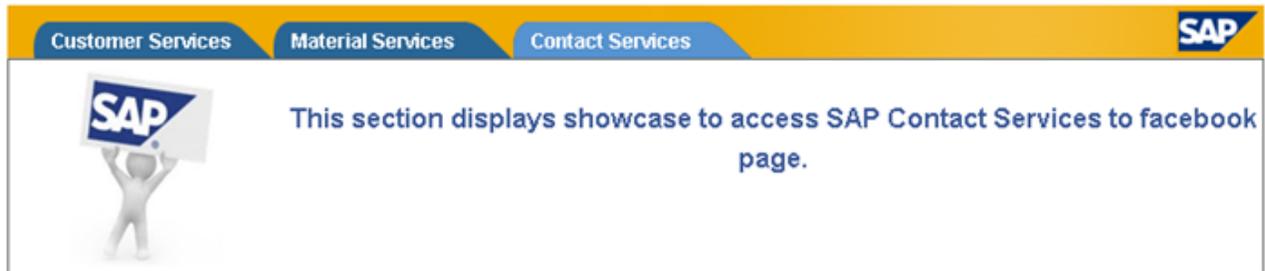
Netbeans and application formation

- Start Netbeans
- Once you have started navigate as follows:
- File → New Project → Select 'PHP' from categories → PHP Application with Existing Source → Browse to Source box → Navigate to htdocs → Select Sap_Demo → Finish
- Create a new PHP file name 'index.php' in it.
- Open 'index.php'
- Now include the facebook library and Proxy file into it as

```
<?php
require_once 'facebook.php';
require_once 'DemoProxy.php';
?>
```

Design the application

- Get a design for your application, in this tutorial it will look like



S.No.	Name	Tel No.	Email ID	Fax No.
1.	Dieter Berger	0308853120		
2.	Dieter Berger	0308853120		
3.	Mr. Christian Lehmann	02119986115	chris.leh@carbor.de	02119987300
4.	Mr. Markus Schmidt	030982699215		030923899800
5.	Frau Bettina Hoffmann			
6.	Hr. Robert Wolf			

- Get its HTML, CSS should be inline
- Save all images in 'image' folder

Buy a Server

- In order to run the application on facebook you have to host your application on the server OR make your local system a server by assigning a static IP to it. If you have already then you are ready to Go OR buy it first and configure it as described.

Run the first draft on localhost

- Open HTML file of design in WordPad and copy all content and paste it to the index file of the application below closing tag of PHP
- Copy image folder to the application directory
- Now run the application as
Open Browser → type 'localhost/Sap_Demo' in address bar (Considering your local apache is running on port 80) →Enter
- Now your static design will appear

Create facebook API instance

- Add the below code in index.php below include inside `<?php ?>`
- ```
$facebook = new Facebook(array(
 'appId' => '10205076*****', // Replace it with yours
 'secret' => '**1e3fadf4*****7ad4c2c0fa39*****', // Replace it
 with yours
 'cookie' => true,
));
```

## HTML Formatting

- Replace the default header of HTML with header described below:
- ```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:fb="http://www.facebook.com/2008/fbml"
xmlns:og="http://opengraphprotocol.org/schema/" >
```
- ```
<div id="fb-root"></div>
```

 Add this just below the closing of head tag.

## Run the first draft on facebook

- Copy your application to the htdocs folder of server where you have to host the application
- Replace all the relative URL of images with the absolute URL.  
E.g.: [www.example.com/Sap\\_Demo/image/example.png](http://www.example.com/Sap_Demo/image/example.png)
- Now open [www.example.com/Sap\\_Demo/](http://www.example.com/Sap_Demo/). If it will be running fine then go for next.
- Run [www.apps.facebook.com/sap\\_demo/](http://www.apps.facebook.com/sap_demo/) in browser and you will see your draft application on facebook.
- Hey!! Cheers!!

## Go deep into code

- Now it's time to get the data from service and display its.
- All the additional data will be displayed in facebook pop up
- Change the table header according to what do you want to display as below.

```
<td width="4%" style="text-align:left; padding-left: 1px;">S.No.</td>
<td width="30%" style="text-align:left; padding-left: 5px;">Name</td>
<td width="30%" style="text-align:center">Value</td>
<td width="20%" style="text-align:center">Scheme ID</td>
<td width="16%" style="text-align:right;">Email</td>
```

- Now start the code for getting the above values from the Customer web services.

```
<?php
try{
//##### Getting Data form Customer Services web service #####
```

- Initialized the customer proxy class and call the required collection function and run the OData query over it. Finally execute the query as described below.

```
$svcCustomer= new ZTCUST(CUSTOMER_SERVICE_URL); $queryCustomer
= $svcCustomer->Ztcust5Collection()->filter("");
$customerResponse = $queryCustomer->Execute();
```

- Now initialized the token which will used to calculate the number of rows in the Collection.

```
$nextCustomerToken = null;
$n = 1;
```

- Execute the query for each row of Collection and print it in the application
- The query will run until all the rows has executed

```
do
if($nextCustomerToken != null)
{
$customerResponse = $svcCustomer->Execute($nextCustomerToken);
}
}
```

- Every time we get the customer response we will print that value in our table.

```
foreach($customerResponse->Result as $customer)
{ ?>
```

- Following LOC will change the color of rows alternatively

```
<tr <?php if($n%2==0){?>bgcolor="#c4d8ef"<?php } if($n%2==1){
?>bgcolor="#a5c3e7"<?php } ?>>
```

- Display Serial Number

```
<td height="29px" width="4%"><?php echo $n; ?>.</td>
```

- Display Name of Customer. We will assign a ID to it which we will further use it to show pop up for details of customer in facebook.

```
<td width="30%">
<a href="#" clicktoshowdialog="Code<?php echo $n; ?>">
<?php echo $customer->name ?>

</td>
```

- Display Value and Scheme Id

```
<td width="16%" style="text-align:right; padding-right: 5px;"> <?php echo
$customer->value ?></td>
<td width="20%" style="text-align:center"><?php echo $customer->scheme_id
?></td>
```

- If this collection is associated with other collection then in order to get the data from that collection for particular row we will use the LoadProperty.

- Here 'Companydata' is the name of other collection and We will fetch the email from that as below:

```
<td width="30%" style="text-align:center">
<?php
$nextOrderToken = null;
do
{
$ordersResponse = $svcCustomer->LoadProperty($customer, 'Companydata',
$nextOrderToken);
foreach($customer->Companydata as $order)
{
echo $order->e_mail;
}
}while(($nextOrderToken = $ordersResponse->GetContinuation()) != null);
?>
</td>
</tr>
```

- For Popup to show the details on click of name, we will first match the ID and fetch the data of that customer matches with that ID.

```
<fb:dialog id="Code<?php echo $n; ?>" width="400" height="500">
<fb:dialog-title>
<?php echo $customer->name ?>
```

```

</fb:dialog-title>
<fb:dialog-content>
<div>The details of customer will come here.</div>
</fb:dialog-content>
<fb:dialog-button type="button" value="Close" close_dialog="1" />
</fb:dialog>
<?php
$n++;
}
while(($customerResponse = $customerResponse->GetContinuation()) != null);

```

- Catch Blocks for exception handling

```

} catch (DataServiceRequestException $ex){
echo 'Error: while running the query ' . $ex->Response->getQuery();
echo "
";
echo $ex->Response->getError();
} catch (ODataServiceException $e){
echo "Error:" . $e->getError() . "
" . "Detailed Error:" . $e-
>getDetailedError();
}??>

```

## Ready to run

- Now you are ready to run the application, just load the updated app on server and hit the url of your browser which is already open probably.

## OData Concepts

### Data Service Query Expressions

The data service query expressions are conditions that determine what data is retrieved from the WCF Data Service. Using query expressions we can perform traditional query operations against resources (entity sets), such as filtering, sorting, and paging. Query expressions are composed of query options, query operators, and query function. There are nine Data Service Query options supported by the OData SDK for PHP.

- expand
- filter
- orderby
- skip
- top
- inlinecount
- count
- skiptoken
- select

### Running Query Expressions

The OData SDK for PHP exposes two APIs which can be used for running query expressions against WCF Data Service servers.

- `ObjectContext::Execute`
- `DataServiceQuery::Execute`

The `ObjectContext::Execute` accepts query expression URI as parameter, this method will use HTTP GET request to retrieve the entity set (in Atom format) addressed by the query expression URI, create a collection of entity objects by parsing the atom response and returns `DataServiceResponse` object which includes this collection.

An instance of `DataServiceQuery` class represents single query request to a data service, this class exposes few methods which helps to reduce the difficulty in building the query expression URI.

### DataServiceQuery Class

OData SDK for PHP includes a class that represents a single query request to a data service.

The `DataServiceQuery` class exposes the following member function: the 'Query Options in Detail' section contains examples on how to use the `DataServiceQuery` functions

API Prototype	Parameters	Description
<u>AddQueryOption</u> (\$name, \$value)	\$name: The string value that contains the name of the query string option (filter, orderby, skip, top, select, expand) to add.  \$value: The string that contains the value of the query string option.	The query options are added to the resultant query expression URI using <i>?name=value&amp;name2=value2...</i> syntax where the name maps directly to the \$name parameter and the value is from \$value parameter. .
<u>Expand</u> (\$relatedEntities)	\$relatedEntities: The related entities separated by comma, which is to be embedded in the result.	Add the \$expand option in the resultant query expression URI.
Filter(\$expression)	\$expression: The expression to be applied to restrict the entities to be embedded in the result.	Add the \$filter option in the resultant query expression URI.
Select(\$properties)	\$properties: Properties of the entity separated by comma.	Add the \$select option in the resultant query expression URI.
Top(\$count)	\$count: The maximum number of entities to be embedded in the result.	Add the \$top option in the resultant query expression URI.
Skip(count)	\$count: The number of rows to be skipped when returning results	Add the \$skip option in the resultant query expression URI.
IncludeTotalCount()	None.	Requests to include the number of entities returned by the query expression URI along with the query result.
Count()	None.	Returns only count of entities satisfying the resultant query URI expression.
RequestUri()	None.	Returns the resultant query URI expression.
Execute()	None.	Executes the query expression and returns QueryOperationResponse object which holds the results as a collection.

If any error occurred while building the query expression URI (ex: you cannot use IncludeTotalCount() and Count() together) or running the query (ex: Unauthorized) then this class will throw exception of type *DataServiceRequestException*.

### QueryOperationResponse Class

OData SDK for PHP includes a class that represents the return type of ObjectContext::Execute and DataServiceQuery::Execute APIs.

The QueryOperationResponse class exposes the following member functions:

API Prototype	Parameters	Description
<u>TotalCount()</u>	None.	Returns total number of entities in the entity set, if user asked for row count (using <code>DataServiceQuery::IncludeInlineCount</code> or <code>inlinecount=allpages</code> option in the query expression).
<code>getQuery()</code>	None.	Returns the <u>query</u> expression URI that generates the <u>QueryOperationResponse</u> item.
<code>getStatusCode()</code>	None.	Returns the HTTP response code associated with HTTP response for the query expression URI.
<code>getHeaders()</code>	None.	Returns an associative array which contains the HTTP response headers associated with HTTP response for the query expression URI.
<code>getError()</code>	None.	Gets error thrown by the data service query operation.
<code>GetContinuation(\$collection)</code>	\$collection: The collection of related objects being loaded	Returns <u>DataServiceQueryContinuation</u> object that contains the URI that is used to retrieve the next page of related entities in the specified collection. See Server Side Paging Section for more details.

This class includes the following member variable:

Member variable	Description
Result	The result of execution of 'query expression URI' as a collection.

## OData Exception Classes

OData SDK for PHP supports 4 types of exception.

### [DataServiceRequestException class](#)

Exception class representing any exception that can occur, while building the query expression URI (using DataServiceQuery member APIs) or running the query.

Member variable	Description
Response	Type: QueryOperationResponse  The Response::getError(), Response::getStatusCode(), and Response::getHeaders can be used for getting error details.

This exception class includes one member variable of type QueryOperationResponse, Response. Please see the samples in *Query Options in Details* section to understand how to handle this exception.

### [ODataServiceException class](#)

Exception class representing any error returned from the WCF Data services, while processing the requests (except query request) from the client.

The ODataServiceException class exposes following functions:

API Prototype	Parameters	Description
getError()	None.	Returns short description of error.
getDetailedError()	None.	Returns detailed description of error.
getStatusCode()	None.	Returns the HTTP response code associated with HTTP response representing the error
getHeaders()	None.	Returns an associative array which contains the HTTP response headers associated with HTTP response representing the error.

## InvalidOperation class

Exception class representing any exception that can occur, due to the invalid usage of the APIs by the client application. For example if client application try to add same entity instance twice, then the context tracking logic will throw this exception with message 'The context is already tracking the entity'.

The InvalidOperation class exposes following functions

API Prototype	Parameters	Description
getError()	None.	Returns short description of error.
getDetailedError()	None.	Returns detailed description of error.

## ACSUtilException class

Exception class representing any error that occurred while trying to retrieve ACS token from Azure. This error can occur while using ACSUtil class or ACSCredential class.

API Prototype	Parameters	Description
getError()	None.	Returns description of error from ACS.
getStatusCode()	None.	Returns the HTTP response code associated with HTTP response representing the error
getHeaders()	None.	Returns an associative array which contains the HTTP response headers associated with HTTP response representing the error.

## Related Content

[Reference 1](#)

[Reference 2](#)

<https://developers.facebook.com/>

## Copyright

© Copyright 2012 SAP AG. 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 AG. The information contained herein may be changed without prior notice.

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

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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.