

# Export Sigma Grid Data Programming Tutorial

## Introduction

Sigma grid starts to support export data to file from version 2.2. The following features have been implemented.

- Export to PDF
- Export to XML
- Export to Excel
- Export to CSV

Sigma does much to make exporting data easy. As we know, Sigma grid is a pure client solution, exporting task can't be fulfilled without help of server side. This tutorial will teach you how to write simple code to do exporting.

## Limitation Notes

1. Because of limitation of PDF lib, exported Pdf could look a little different from what you see in browser (IE, Firefox, etc)
2. Because of limitation of PDF lib, Png files with alpha channel.
3. Php code is taken as sample. If you are using some other language, some changes must be made accordingly.

## Backend File Structure

There are file structure of exporting code. All the code are in /sigmagrid/export.

File Location	Description	Required
/export_php/html2pdf/	For exporting grid data to PDF. Please download HTML2PDF library at <a href="http://html2pdf.fr/">http://html2pdf.fr/</a> and unzip all stuff into this directory. Sigma grid package doesn't contain this library because of license issue.	Y
/export_php/ExcelExport.php	For exporting grid data to Excel.	Y
/export/export_php/GridServerHandler.php	Adapter between grid and exporting modules.	Y
/export_php/JSON.php	JSON PHP implementation.	Y
/export/export_php/testDAO.php	Sample code for data feed.	
/export/export_php/testList.php	Sample code for exporting dispatcher.	

The first four items are required for exporting work. Don't change their names or structures. Just copy them into your project. The last two files, say testDAO.php and testList.php, are sample code for demonstration and code templates.

## Coding Step By Step

In this section, I will tell you how to integrate database code with the grid.

### Create Data Table

1. We have created a Sql file for you to follow the tutorial. Create a database, say gridexport, in you local Mysql server.
2. Run /export/export\_php/gridexport.sql in mysql console.

### Write Data Feed File

1. Copy testDAO.php to studentDAO.php in the same directory.
2. With studentDAO.php, developer needs to implement a function, say getStudentData. This function should return an array of JSON objects presenting grid data.

### JavaScript Code

```

1. <?php
2. //Developers need to implement this following function.
3. //Typically, it should connect to the database and return two 2d array.
4. function getStudentData(){
5. //Connect to database
6. mysql_connect("localhost","root","") or
7. die("Could not connect: " . mysql_error());
8. mysql_select_db("gridexport") or
9. die("Could not select database: " . mysql_error());
10. //Execute a sql query.
11. $sql = "select * from student";
12. $handle = mysql_query($sql);
13. //This is a 2d array, every element of which contains an array.
14. $retArray = array();
15. while ($row = mysql_fetch_object($handle)) {
16. $retArray[] = $row;
17. }
18. //return this 2d array
19. return $retArray;
20. }
21. ?>

```

I don't have two much to say about these code. Most of them are for database connection and data retrieving. The only thing I would like to highlight is the function should return a an array of JSON objects. You could use `mysql_fetch_object` in a loop to gain such array.

## Write Export Handler File

Copy `testList.php` to `studentList.php`. Open `studentList.php`, you will see the following lines.

### JavaScript Code

```

1. <?php
2. //This is sigma grid exporting handler
3. require_once('GridServerHandler.php');
4. //This is a php file for data feeding
5. require_once('testDAO.php');
6. //To create grid exporting instant.
7. $gridHandler = new GridServerHandler();
8. $type = getParameter('exportType');
9. if ( $type == 'pdf' ){
10. // to use html2pdf to export pdf
11. // param1 : Orientation. 'P' for Portrait , 'L' for Landscape
12. // param2 : Paper size. Could be A3, A4, A5, LETTER, LEGAL
13. // param3 : Relative picture path to this php file
14. $gridHandler->exportPDF('P','A4','../');
15. }else {
16. //to get the data from data base. //
17. $data1 = getTestData( 25 );
18. if ( $type == 'xml' ){
19. //exporting to xml
20. $gridHandler->exportXML( $data1);
21. }else if ( $type == 'xls' ){
22. //exporting to xls
23. $gridHandler->exportXLS( $data1);
24. }else if ( $type == 'csv' ){
25. //exporting to csv
26. $gridHandler->exportCSV( $data1);
27. }else{

```

```

28. //for grid presentation
29. $gridHandler->setData($data1);
30. $gridHandler->setTotalRowNum(count($data1));
31. $gridHandler->printLoadResponseText();
32. }
33. }
34. ?>

```

Generally, we have two places to change.  
Firstly, we need to include the data feed file we did just now.

#### JavaScript Code

```
1. | require_once('studentDAO.php');
```

Secondly, we need to change line 17 to make sure that we get the right data from database.

#### JavaScript Code

```
1. | $data1 = getStudentData ( );
```

Now, we have all the backend files prepared. It's quit easy, isn't it?

## Write Sigma Grid File

Write Sigma Grid File I suppose you have some background of how to write Javascript for a live grid without server side data. If you don't, you could see [www.simgawidgets.com/forum](http://www.simgawidgets.com/forum) for more client side knowledge. I would like to highlight three properties of the grid. If you open /export/export\_php/index.html, you will see following lines.

#### JavaScript Code

```

1. | var gridConfig={
2. | ...
3. | loadURL : './export_php/testList.php',
4. | exportURL : './export_php/testList.php?export=true',
5. | exportFileName : 'test_export_doc',
6. | }

```

loadURL - The URL for grid to show within the grid.

exportURL - The URL for grid to export data.

exportFileName - The default file name of the exported xml/pdf/cvs/excel. For our student sample, it should go like this

#### JavaScript Code

```

1. | var gridConfig={
2. | ...
3. | loadURL : './export_php/studentList.php',
4. | exportURL : './export_php/studentList.php?export=true',
5. | exportFileName : 'file_name_you_want',
6. | }

```

That's all for the simple tutorial. Thank you for your time.