

API Integration in Web Application

Dr. Chetan R. Dudhagara

Assistant Professor
Computer Science Department
N. V. Patel College of Pure and Applied Sciences
Vallabh Vidyanagar, Gujarat

Mr. Ashish Joshi

Assistant Professor
BCA Department
V.P. & R.P.T.P. Science College
Vallabh Vidyanagar, Gujarat

Abstract - Application Program Interface (API) widely used for the programmers. It helps programmers to focus on their programming rather than the other features. API provides the facility to programmer for use the "Predefined Code". Among them one type of API provides by the google that is chart API. Chart API is used to generate the chart from the data which are available on our database. Sometimes it would be very hard to develop such type of another system. Rather than we can use API and save the time and effort that can be use at proper place. This paper will describe how we can integrate the google chart API in the php website.

Keywords : API, PHP, Mysql, JSON, Payroll, Interface

I. INTRODUCTION

It is same as headache to develop another system for the basic required system for the web developers. For examples company requires to establish a chart in web application for monthly selling. The main application is the company's web site but you have to develop another application through that you can create the chart. So developer requires more time and efforts. Basically this facility is available that is developed by another developer so why we should spend time to re build it. Simply we can integrate it and use it. Generally user use the google chart API tool to generate the chart from given data base. In this paper we describe how to integrate google chart API with php.

II. METHODOLOGY

We are using one chart API that provides by google through that we can use the predefined code and use it in our web application. Basically this code can be integrated with any server side

scripting language. Here I am taking the example of php. In this example we discuss to integrated the google chart API with php file. The following steps are use for it.

1. Create database as payroll and create table as sell as per following

	id	month	sell
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	2000
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	3000
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	500
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	5000

Fig. 1 : database structure

As you can see in above Fig. 1, there are three columns in this table as id, month and sell that describe the data about selling of that month.

2. Create blank JSON (Java Script Object Notation) file as *json_data.json*. It is very important because the API can read this file only. JSON is the technology through that you can pass the data from one application to another. The file appears after writing data by script looks like below.

```
{
  "cols": [
    { "id": "", "label": "month", "pattern": "", "type": "number" },
    { "id": "", "label": "selling", "pattern": "", "type": "number" }
  ],
  "rows": [
    { "c": [ { "v": "1", "f": null }, { "v": 2000, "f": null } ] },
    { "c": [ { "v": "2", "f": null }, { "v": 3000, "f": null } ] },
    { "c": [ { "v": "3", "f": null }, { "v": 500, "f": null } ] },
    { "c": [ { "v": "4", "f": null }, { "v": 5000, "f": null } ] }
  ]
}
```

3. Create PHP file as *get_data.php*. This file will fetch the data from the data base and write the JSON file. The code is as per following:

```
<?php

    $t='{ "cols": [{"id":"","label":"month","pattern":"","type":
    : "number"}],
    {"id":"","label":"selling","pattern":"","type":"number"}
    ],"rows": [';    file_put_contents("json_data.json",
    $t);

    $con=mysql_connect("localhost","root","");

    mysql_select_db("payroll",$con);

    $res = mysql_query("SELECT month,sell FROM sell");

    while($row=mysql_fetch_array ($res)) {

        $a=$row['month'];

        $b=$row['sell'];

        $t1='{ "c": [{"v":';

        $t1 .= $a;

        $t1 .= ", "f":null}, {"v":';

        $t1 .= $b;

        $t1 .= ", "f":null}]}';

    file_put_contents("json_data.json",$t1,
    FILE_APPEND);}

    file_put_contents("json_data.json","}]",
    FILE_APPEND);

    $string = file_get_contents ("json_data.json");

    echo $string;

?>
```

4. Create another file as *sell_analysis.php*. It will call the *get_data.php* file and send the JSON file to the API server. API

server will read it properly and generate the chart related to the data and respond to the application. The code is as per following. This code generated by API server, you have to do some changes as per your requirement.

```
<html>

<head>

<style>

.head

{    font-family:verdana;

    font-size:14px;

    background-color:#000066;

    color:#FFFFFF; }

.head1

{    font-family:verdana;

    font-size:24px;

    color:blue;    }

.data

{    font-family:verdana;

    font-size:12px;

    font-weight:bold;    }

</style>

<!-- Note : This code is available on API provider.-->

<!--Load the AJAX API-->

<script type="text/javascript"

src="https://www.google.com/jsapi"></script>

<script    type="text/javascript"    src="//ajax.googleapis.

com/ajax/libs/jquery/1.10.2/jquery.min.js"></scrip>

<script type="text/javascript">

google.load('visualization','1', {'packages':['corechart']});
```

```
google.setOnLoadCallback(drawChart);
```

```
<div id="chart_div" height="400" width="600" > </div>
```

```
function drawChart() {
```

```
</center>
```

```
var jsonData = $.ajax({
```

```
</body>
```

```
    url: "get_data.php",
```

```
</html>
```

```
    dataType: "json", async: false
```

```
}).responseText;
```

5. Finally this code will generate the chart.

```
// Create our data table out of JSON data loaded from server.
```

```
var data = new
```

```
google.visualization.DataTable(jsonData);
```

```
var options = {
```

```
    title : 'Monthwise Selling Analysis',
```

```
    vAxis: {title: "Selling Growth"},
```

```
    hAxis: {title: "Month"},
```

```
    seriesType: "bars",
```

```
    series: {1: {type: "line"}},
```

```
    series: {2: {type: "line"}}, 'hieght':400, 'width':920 };
```

```
// Instantiate and draw our chart, passing in some options.
```

```
var chart = new google.visualization.ComboChart(
document.getElementById('chart_div'));
```

```
chart.draw(data,options);
```

```
}
```

```
</script> </head>
```

```
<body>
```

```
<!--Div that will hold the pie chart-->
```

```
<br><br><br><br><br><br><br><br><br>
```

```
<center>
```

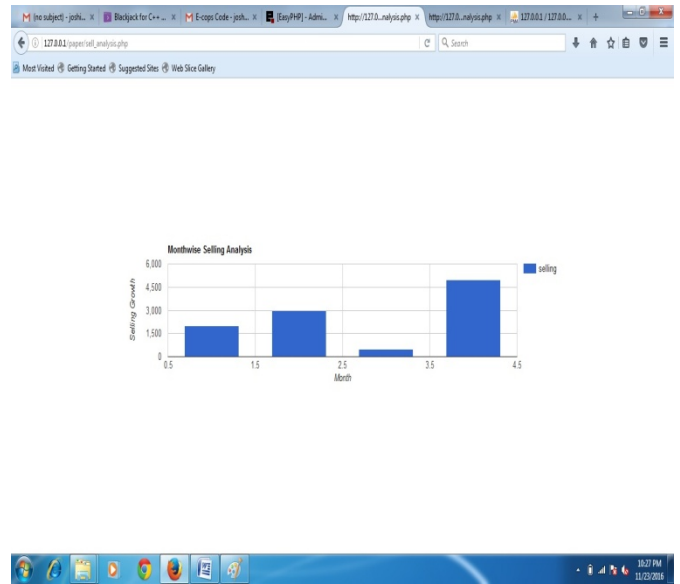


Fig. 2 : chart

As per overall process, you have to run only sell_analysis.php file, it will automatically call the get_data.php file as per written script in it.

III. FUTURE WORK

It is important use the google chart API to generate the charts from database. This paper concludes that how can implement the chart API in php (server side scripting language). It can also be use with different server side scripting language like asp.net, jsp etc. Thus google have many more API so I will try to implement another API which can be also helpful to reduce the extra work and concentrating on main work.

IV. CONCLUSION

This paper concludes that google API provides facility to generate the chart from the XML or JSON files, but how we can create and integrate JSON file. The chart API is worth for time and effort saving. By using this facility we can generate various types of charts such as bar chart, pie chart, line charts, etc. The authors have contributed here to use the server side scripting language for google chart API. Here we use php as a server side scripting language.

V. REFERENCES

- [1] Beginning of php By Elizabeth Naramore
- [2] High Performance MySQL By Jeremy D. Zawodny, Derek J. Balling
- [3] Javascript & AJAX By Tom Negrino and Dori Smith
- [4] Full Web Building Tutorials www.w3schools.com
- [5] PHP Manual.chm <http://www.php.net/docs.php>
- [6] www.google.com