International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 01, January 2017
www.ijates.com
ijates
ISSN 2348 - 7550

ACCESSING REMOTE DATABASE IN IOS APPLICATION USING JSON PARSING WITH OBJECTIVE-C

Ishwarjit Kaur[1], Sharanpreet Kaur[2], Gurinder Kaur[3]

[1,3]*Assistant Professor, Mata Gujri College, Fatehgarh Sahib.*

[2]*Research Scholar Punjab Technical Universty Jalandhar*

ABSTRACT

Mobile applications are in very much use these days. Mobile Applications are commonly used for entertainment, education, social networking etc. Database is the most important part of a mobile application. When the database is on the same device on which application is installed it is known as local database. Database for a mobile application can also be placed on some remote machine or computer. Remote servers are preferred when there is need to store very large amount of data and same data is to be shared between the mobile applications installed on different phones and applications that can be accessed from computers as well. But data cannot be accessed straight forward from remote servers. In our work we will create a Web Service that will fetch data from Remote database and returns a JSON format.This JSON format will be read in iOS application. Web Service is created in PHP language. WAMP platform is usedto execute this Service. iOS applications can connect with this Web Service to read the data returned by the Service.iOS application is created and executed using Xcode installed on Mac system.

Keywords: Introduction, Development Tools, Steps to Create a Database, Steps to Create a Table, PHP Web Service to fetch the table and convert it into JSON Format, Output of Web Service in JSON Format, iOS Application for accessing data from MySql server on remote machine.

I. INTRODUCTION

Android and iOS are the most commonly used operating systems in mobile phones nowadays. The applications build on these operating systems commonly exchange data with web applications. In our work we have created a simple database in MySQL of WAMP. The data is stored in a table in MySQL. We have created a simple Web Service in PHP to fetch data from this table. The Web Service is run by Apache Web Server of WAMP. The output of our service will be in JSON format. The way that data is expressed may vary but **JSON** format is preferred.

JSON stands for JavaScript Object Notation. It is based on the JavaScript's object notation. JSON is a simple and lightweight data-interchange script notation. Humans can easily read and write JSON. Machines can easily

generate and parse JSON. JSON format is not dependent on any language. It is a simple text format that does not need JavaScript to read or write. Basic Elements contained in JSON are:

Objects: Objects begin and end with curly braces ({}).

Object Members: Members consist of keys and values. Keys and values are separated by colon (:). Object Members are separated by commas.

Arrays: Arrays contain values and begin and end with braces.

In our work we have created JSON. Write the IP of computer which host the PHP Web Service program. The JSON as output will be shown in the browser. We will read this JSON in our iOS application created in Xcode on Mac system. The same JSON can also be accessed in Android application.

II. DEVELOPMENT TOOLS

The following tools have been used in this appliction

1.1 **Xcode-** Xcode is developed by Apple. It is an IDE(Integrated Development Environment) used to develop and execute iOS applications.It contains a suite of software development tools. It is used for developing iOS, macOS software. It was released in 2003. Latest version is available through the Mac App Store. It is available free of cost. Prior Versions are available on Apple Developer website for registered users only.

1.2 **WAMP**- WAMP is a Web Development environment. It works on Windows platform only. It is used to develop Websites and Web Applications. The components of WAMP are:

W-Windows

Windows is an operating system platform on which Servers are installed.

A- Apache

Apache is a web server that is used to run PHP script.

M- MySQL

MySQL is a server to host the database. It is an open source RDBMS with GUI. It enables user to create database and tables graphically.

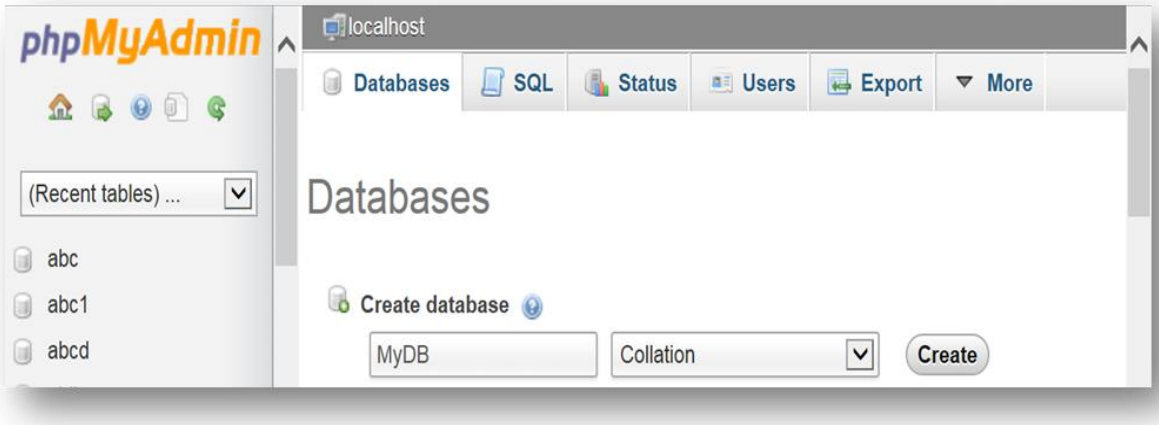
P-PHP

PHP is a server side scripting language that is used to develop dynamic web pages. PHP can be embedded in HTML pages. It can run on MAC, Linux and Windows. PHP can give output in JSON Notation. In our work we have run select query which retrieves data from MySQL. We have converted this output into JSON format using json_encode.

III. STEPS TO CREATE A DATABASE

I. Run WAMP> Select phpmyadmin> Select Databases.

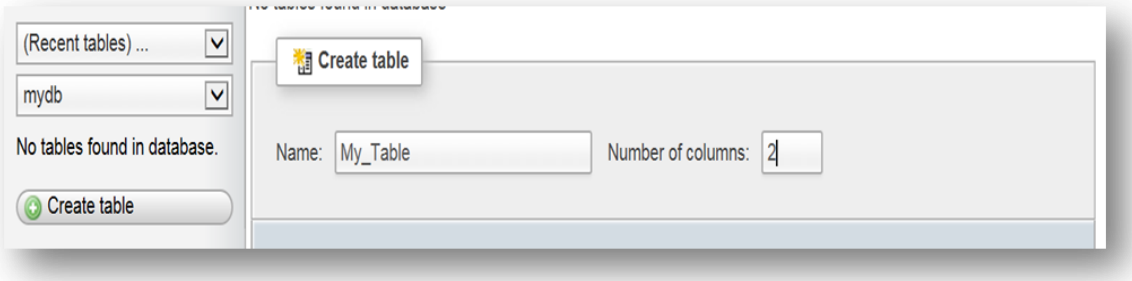
II. Write the name of the database in Create database Text Box. Eg. MyDB. Click Create.



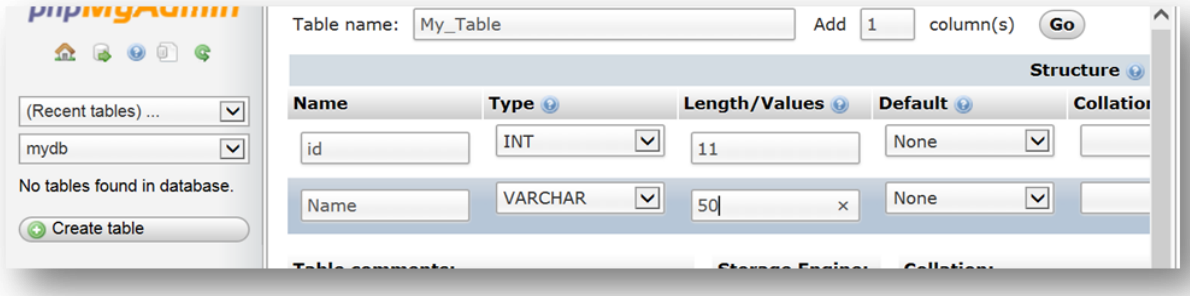
IV. STEPS TO CREATE A TABLE

I. Double click on the name of the database.

II. Write the name of the table under Create table. Eg. My_Table. Also write the number of columns. Click Go.

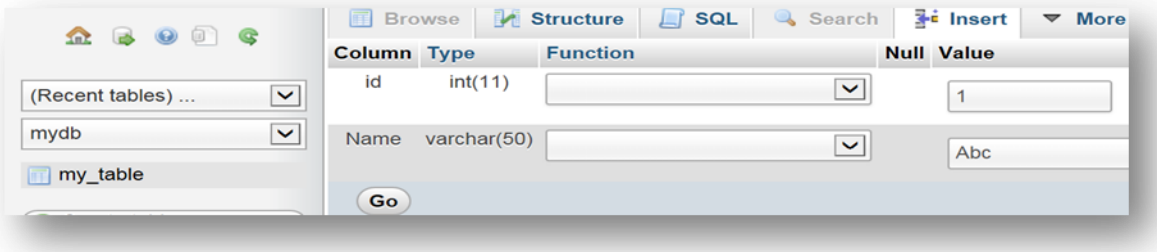


III. Write the names, Data type and length of Columns. Click Save.

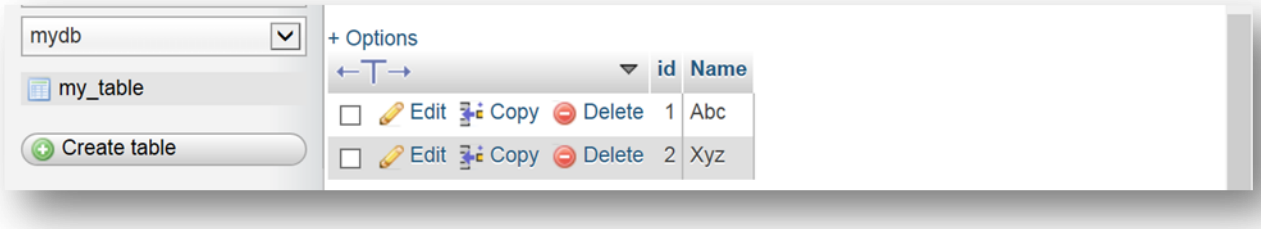


IV. Double click on the name of the table. Click Insert.

V. Write the values you want to insert into the table and click Go.



VI. Double Click on the name of the table. Inserted values will be shown as:

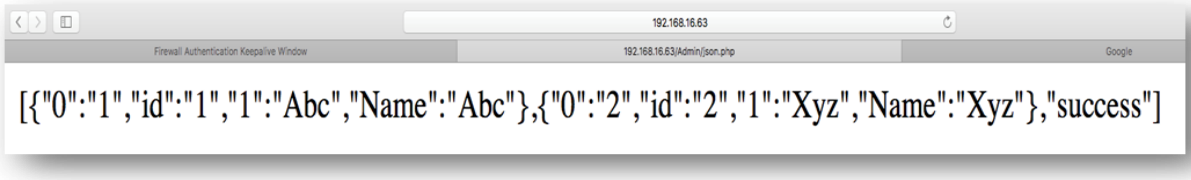


V. PHP WEB SERVICE TO FETCH THE TABLE AND CONVERT IT INTO JSON FORMAT

To fetch the data inserted in table we have used select query. The accessed data is converted into JSON using json_encode() method. Web Service for this is as follows:

```php
<?php
$link = mysql_connect('localhost','root','');
$db = mysql_select_db('mydb',$link);
$i=mysql_query('select * from My_Table');
$num_rows = mysql_num_rows($i);
while($row = mysql_fetch_array($i))
{
    $r[]=$row;
    $check=$row['id'];
}
if($check==NULL)
{
    $r[$num_rows]="Record is not available";
    print(json_encode($r));
}
else
{
    $r[$num_rows]="success";
    print(json_encode($r));
}
mysql_close($link);
?>
```

5.1 **Output of Web Service in JSON Format**



To fetch this JSON, put WAMP online. Write the IP of the Server/Folder name/ File name in browser. Eg. 192.168.16.63/Admin/json.php

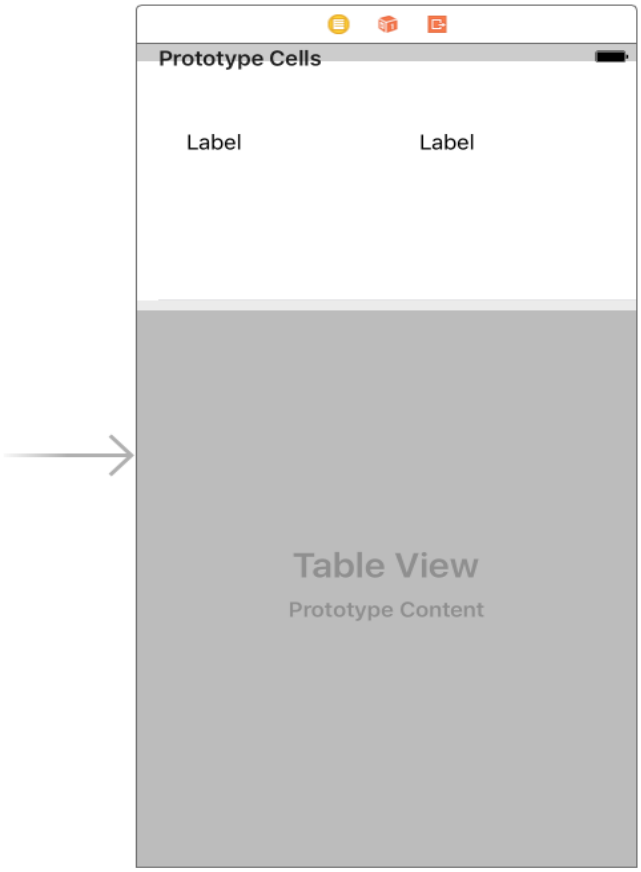
Where 192.168.16.63 is the IP address of server on which Service is hosted.

Admin is the name of folder in which Service is stored on Server.

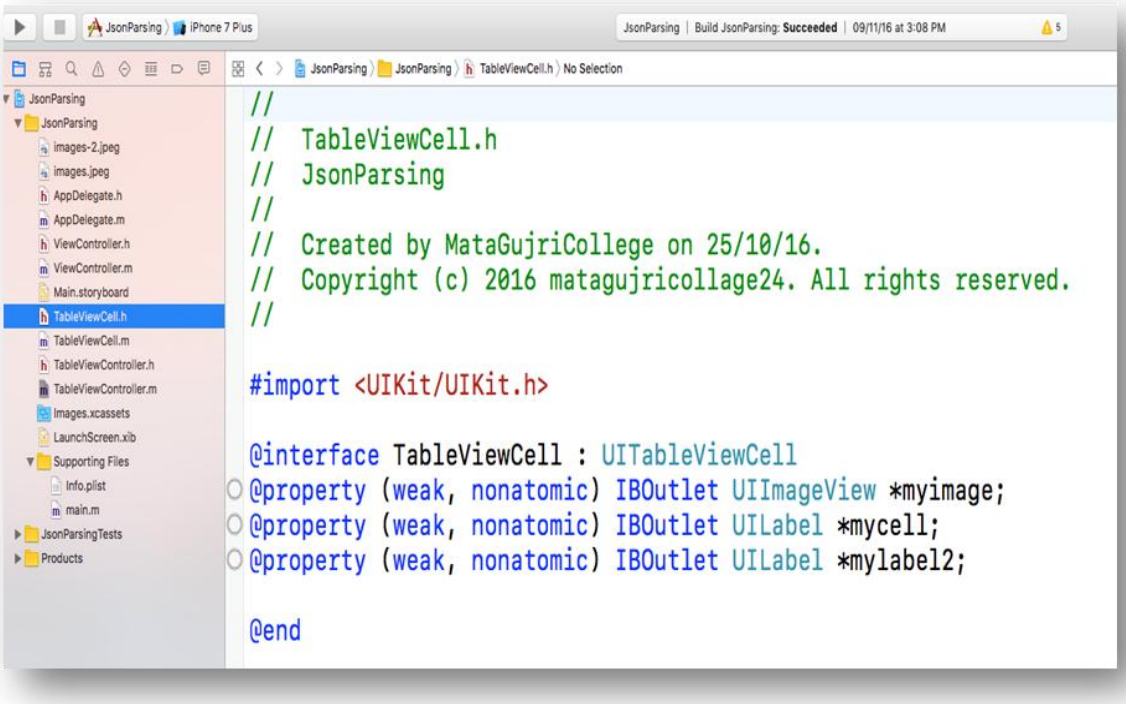
Json.php is name of the file in which php code to create JSON is written.

VI. IOS APPLICATION FOR ACCESSING DATA FROM MYSQL SERVER ON REMOTE MACHINE

In design a TableViewController has been inserted which is linked to a class name "TableViewController".Place two labels inside the cell and create referencing outlets for the labels. In Property Inspector this cell is reffered to as "***cell***".



The cell in TableViewController is linked to a custom class named"TableViewCell" which is as follows:



In this class mycell and mylabel2 are referencing outlets for labels respectively.

6.1 Coding of TableViewController.m class is as follows:

```
#import "TableViewController.h"
#import "TableViewCell.h"
@interface TableViewController ()
{
    NSMutableArray *json;
    NSMutableDictionary* dic;
    NSString* str;
    NSURL * URL;
    NSMutableArray * jsonsearch;

}
@property (strong, nonatomic) IBOutlet UITableView *tableview;
@end

@implementation TableViewController
- (void)viewDidLoad
{
    [super viewDidLoad];
    jsonsearch=[NSMutableArray  new];
    self.tableview.delegate=self;
}
```

```
-(void)viewWillAppear:(BOOL)animated
{
   [super viewWillAppear:animated];
   [self methodForGetServiceResponse];
}
-(void)methodForGetServiceResponse
{

  str=@"http://192.168.16.63/Admin/json.php";
   NSURLRequest * request = [NSURLRequest requestWithURL:URL];

   [NSURLConnection sendAsynchronousRequest:request queue:[NSOperationQueue mainQueue] completionHandler:^
       (NSURLResponse *response, NSData *data, NSError *connectionError)
    {
       if([data length]>0 && connectionError==nil)
       {

           json =[NSJSONSerialization JSONObjectWithData:data options:kNilOptions error:nil];
           [jsonsearch addObjectsFromArray:json];
           [self.tableview reloadData];
```

```
                if([json isKindOfClass:[NSDictionary class]])
                {
                    NSDictionary *a=dic;
                    NSLog(@"dictionary is %@",a);
                }

            }

            else if(connectionError!=nil)
            {
                NSLog(@"%@",connectionError);
            }
               NSLog(@"%@",jsonsearch);

        }];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
#pragma mark - Table view data source
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {

    // Return the number of sections.
    return 1;

}
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {

    // Return the number of rows in the section.
    return [jsonsearch count];
    }

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    TableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"cell" forIndexPath:indexPath];

    dic=[jsonsearch objectAtIndex:indexPath.row];
     NSLog(@"Dictionary is %@",dic);
    cell.mylabel2.text=[[jsonsearch objectAtIndex:indexPath.row] valueForKey:@"id"];
     cell.mycell.text=[[jsonsearch objectAtIndex:indexPath.row] valueForKey:@"Name"];
    cell.accessoryType=UITableViewCellAccessoryDetailButton;
    return cell;
}
@end
```

In above written code . 192.168.16.63/Admin/json.php is Url of Web Service(JSON output).*NSJSONSerialization*is a class for managing JSON data.JSON data can be converterd into NSArray or NSDictionary with the help of this class. JSON has to be a NSData object from which it is converted into

foundation object(NSArray, NSDictionary).Before performing any conversion*NSJSONSerialization* class also provides a mechanism to determine whether JSON is valid or not.

In cellForRowAtIndexPathmethod data is extracted from jsonsearch array and is displayed on labels in cell prototype. The output of the above code is as follows:

1	Abc
2	Xyz

VII. CONCLUSION

Data can be exchanged between mobile applications and remote servers. But this is not very simple process. Data from server database is converted into JSON format using some server side language like PHP and java. This JSON format can be read in mobile applications. Data can be expressed in many formats but JSON is preferred because it is simple and easy to read and write.

REFERENCES

[1] AnirudhNagesh, KeshavKhandelwal, Carlos E. Caicedo, "**Accessing External Databases from Mobile Applications**".

[2] Daniele Teti, "**Mobile Application Development Connecting with PHP REST Servers from Android**".

[3] Kiran Dhokale, NamdeoBange, ShelkePradip, SachinMalave, "**IMPLEMENTATION OF SQL SERVER BASED ON SQLITE ENGINE ON ANDROID PLATFORM**".

[4] Ahmed Hasan Oudah Al-Ghanimi, "**Attendances System for College Students".**

[5] https://www.safaribooksonline.com/blog/2013/03/06/parsing-json-in-objective-c-using-nsjsonserialization/

[6] http://www.appcoda.com/fetch-parse-json-ios-programming-tutorial/

[7] https://en.wikipedia.org/wiki/Xcode

[8] http://www.json.org/

[9] http://json-schema.org/example1.html