Database Concepts

Tutorial:
Mobile Interface to MySQL
CSE 30246 Fall 2014

S. Aguinaga

Objective:

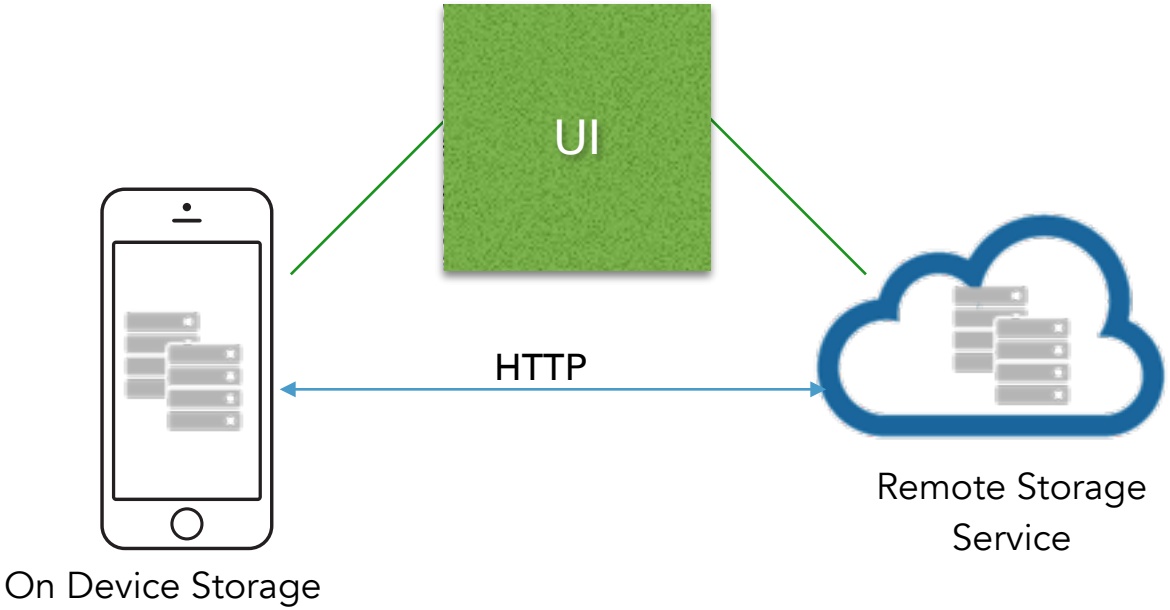
This tutorial outlines a simple example of how to interface to a MySQL engine running on DSG1 for the iOS platform

Goals:

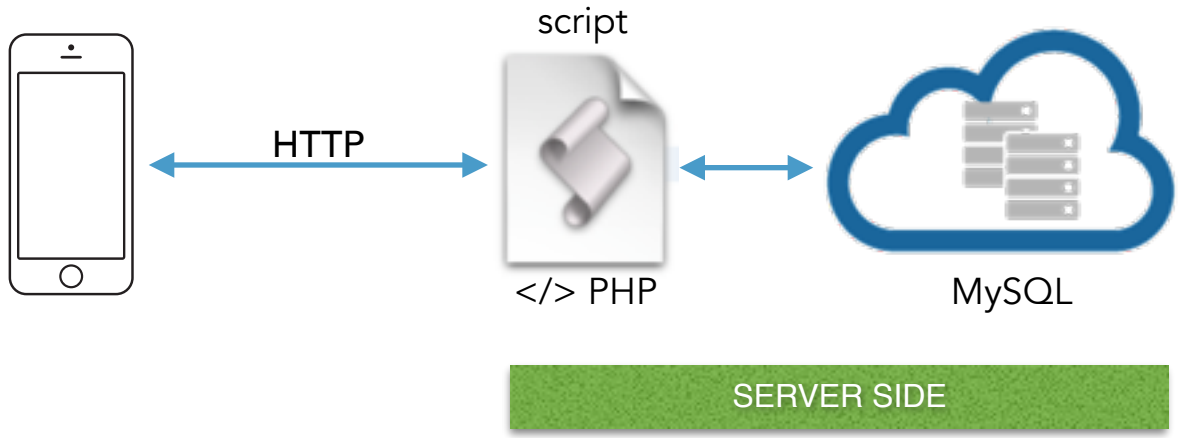
- Learn the basic requirements to successfully access a remote database server
- Learn the basic hooks required by an iOS project
- Learn what server-side objects are needed to interface to the MySQL engine
- Walk through an iOS Xcode example

Overview

1



2



S. Aguinaga

iOS Example: Login or Register Users to Your Database

Brief Description:

- This is a simple Login/Register app that interfaces to a MySQL database over HTTP
  - Select to either **login** or **register**.
  - Login: Send a name and password, if the user exists and matches both username and password … you login.
  - Register: Send the name and password of a new user, if the communication request works and the new record is created you are notified of the record id created.

What you need:

- Xcode (ObjectiveC) project … Build your UI
- On the server-side: create a new database (any name) on DSG1 and add a table named 'users' with at least 3 fields/columns
  - These 3 fields are: **uid** INT primary key auto_increment, **username** VARCHAR, **plain_password** VARCHAR
  - create the PHP scripts in DSG1:/home/netid/public_html

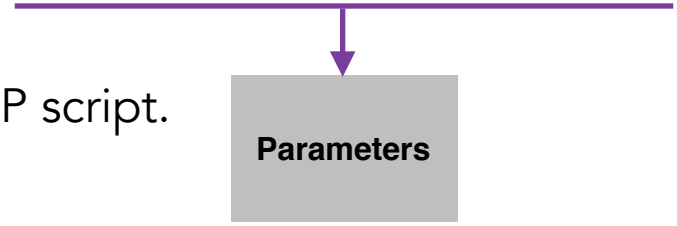
iOS Example: Login or Register Users to Your Database

1

In your Xcode Project:
You will build your UI and use your ViewController class to define the actions that will trigger communication with the remote server using HTTP

Your goal is to build a URL will be send to the remote server and a response will be received that you will parse locally and then take appropriate action.

To build the URL you will need the server address +
parameters you
will pass to the PHP script.




```
- (void)viewDidLoad
{
     [super viewDidLoad];
      …
      username = [[UITextField alloc] init];
      username.placeholder = @“Username";
      …
      loginButton = [[UIButton alloc] init];
      loginButton.frame = buttonFrame;
      …
      [loginButton setTitle:@"Login"
forState:UICont…];
      [loginButton addTarget:self

action:@selector(submitLogin:)
```

2

In the iOS example' main view controller I subclass uiwebview to save time and create two methods for Login and Register

Read/get/fetch user information from MySQL database to authenticate User

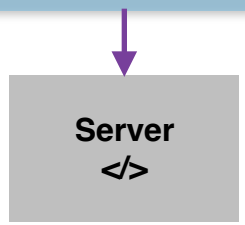
Add new user to the MySQL database

Below is how I construct the call to HTTP

```
NSURL *myUrl = [NSURL URLWithString:[NSString stringWithFormat:@"%@"
                                     @"/login_pass_all_svc.php?username=%@"
                                     @"&password=%@",
                                     requestParameters[@"authenticatingServerBaseUrl"],
                                     requestParameters[@"username"],
                                     requestParameters[@"password"]]
                       ];
    NSMutableURLRequest *myRequest = [NSMutableURLRequest requestWithURL:myUrl
                                                             cachePolicy:NSURLCacheStorageAllowed
                                                         timeoutInterval:60.0];

    [myRequest setHTTPShouldHandleCookies:YES];

    [self loadRequest:myRequest];
```



S. Aguinaga

Server-Side Requirements

3

Server
</>

PHP script running on the DSG1 server

```
<?php
ini_set('display_startup_errors',1);
ini_set('display_errors',1);
error_reporting(-1);
…
```

Here you receive from the server a JSON (similar to XML) response that you will parse locally