EHS Web Services

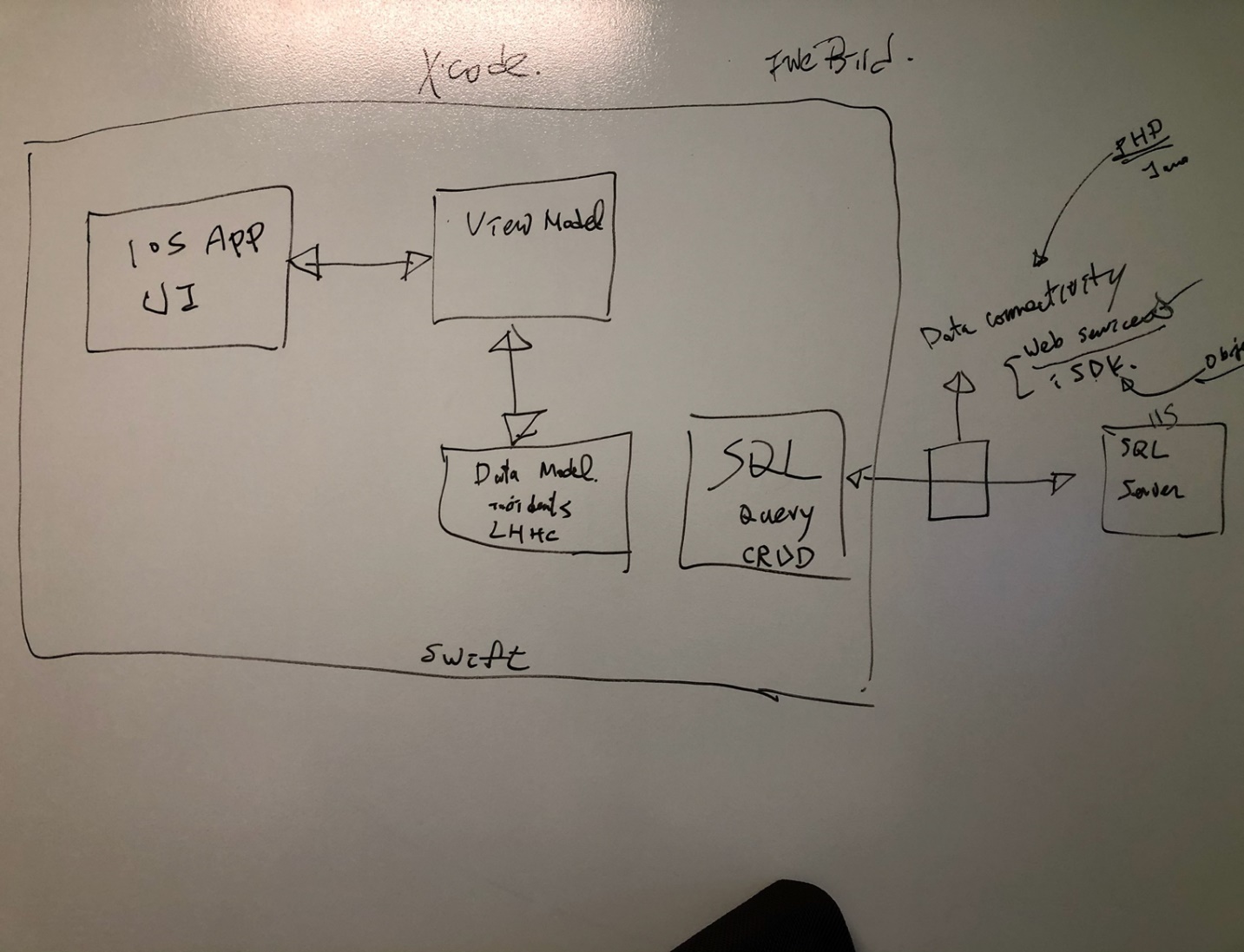
*Traditional web applications using data connection (like JDBC) that serves data to single point, server-based access, with web browser as the user interfaces. While Web services serve data in any format (XML/JSON etc.) through HTTP, called API (Application Programming Interface). REST and SOAP are two different ways to design the API.*

*Applications or clients of any kind (IOS app, Android app, web, desktop app etc.) using this type of APIs to access legacy data for multiple applications. In FACEBOOK EHS, this database should be the MS SQL Server that we are familiar with. Our IOS app development fits into this open web service approach. Hence, we will be adopting Representational State Transfer (REST) as the main architecture for well-behaved Web services.*

*Web service has no user interfaces but provide a set of standard APIs to serve multiple clients with intended services. Those services include GET, POST, PUT, and DELETE operations to system/database objects through HTTP protocols.*

***Web Services vs Web Applications***

* *Web Services can be used to transfer data between any type of HTTP clients including but not limited to Web Applications.*
* *Web Services can be accessed from any languages or platform.*
* *A Web Application is meant for humans interface; while a Web Service is meant for computers to interact with.*
* *Web Application is a complete Application with a Graphical User Interface (GUI), however, web services do not necessarily have a user interface since it is used as a component to serve any application interested in its data.*
* *Web Application can be only access through browsers.*





EHS IOS application diagram

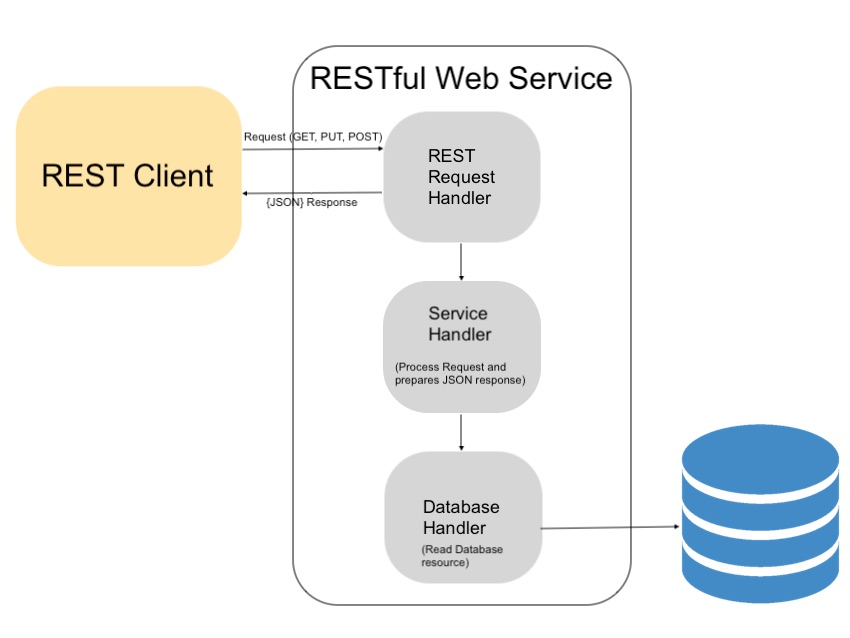
For this document we focus on the data connectivity – Web Services portion of the diagram

## RESTful web services API architecture

The following diagram shows a RESTful web service architecture. In this diagram, the request-response flow among the client-server is represented.

In this diagram, the database is shown as a resource. Based on the web service the resource can be XML feed, JSON data extracted from the file system or any.





**Uses of RESTful API**

RESTful API provides services to access resources from external applications or REST clients. Some of the predominant uses of the RESTful API is listed below.

* As an interface with multi-platform support which is used to access resources from outside applications coded in various programming languages like PHP, JAVA, for different clients and resources, like IOS, and Android devices, and more.
* REST is the simple architectural style for transmitting data over HTTP.
* The REST API is the most suitable resource provider for an AJAX-based application interface which requires data to update UI without page reload.
* By meeting more the REST constraints, the web applications or services can support a wide range of clients.

**What technologies are available to develop EHS RESTful web services?**

* Server Platforms: Apache, IIS etc.
* Languages: Java, PHP, or Python etc.
* Why we are using PHP on IIS over the others?
  + Support and reuse our current available EHS SQL server database over IIS
  + Simple, streamlined, and smaller scaled, and with OOP support
  + Solid industrial proven
  + Java and Python can support large scale of web services, but are much heavier and require more resources to develop and maintain
* How do we connect to data? Through PHP SQL server driver

**EHS RESTful web service design:**

In the PHP RESTful web service example, the following domain class contains the resource data array and service handlers. These handlers are called based on the request sent by the REST client or external apps.

1. REST API services controller
2. API services handlers
3. Clients send the service requests via URL
4. URL format (example):

Web site: <http://localhost:8081/>, or ash server

Path: ehswebservices/RestController.php

Parameters: api=getEhsTechUsers

Whole url: <http://localhost:8081/ehswebservices/RestController.php?api=getEhsTechUsers>

1. Service handlers return response as JSON data with status code
2. User security – Facebook only Single sign on (SSO)

## EHS RESTful services URI mapping (a better approach with URL rewrite)

Every resource is identified via a URI (Uniform Resource Identifier).

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.

RestController.php shown in the above file structure is the PHP endpoint to which the request is to be forwarded.

In this example, there are two URIs for accessing this web service from external applications or REST client. One URI will be used to get the complete array of mobile names in a JSON format and the other is to get a particular mobile name based on the ident passed via the request URI.

|  |  |  |  |
| --- | --- | --- | --- |
| **URI** | **Method** | **Type** | **Description** |
| http://localhost/ehswebservices/FBUsers/EHSTech/ | GET | JSON | To get the list of EHS tech User names in an JSON array. |
| http://localhost/ehswebservices/FBUsers /EHSAll/ | GET | JSON | To get the EHS all user data array by in JSON. |

The following URIs are mapped to the real file via the .htaccess or a PHP file to re-route the URL for URL re-writing.

**URI to get the list of all FB Users:**

http://localhost/ehswebservices/FBUsers/list/

**URI to get a particular user’s detail using its id:**

In the below URI the number ‘2’ is the id of a mobile. The resource domain class can get the particular data with the reference of this id parameter.

http://localhost/ehswebservices/FBUsers/list/2/

## PHP RESTful services implementation, code and examples

## PHP with Eclipse

## PHP data objects, ex: FbUser.php

**Class** FbUser {

// Properties

**private** $id;

**private** $EmpName;

**private** $LoginName;

**private** $LastLogin;

**private** $FBFunc;

// Methods

**function** setName($name) {

$this->EmpName = $name;

}

**function** setId($id) {

$this->id = $id;

}

**function** setLoginName($name) {

$this->LoginName = $name;

}

**function** setLastLoginDate($date) {

$this->LastLogin = $date;

}

**function** setFBFunc($func) {

$this->FBFunc = $func;

}

**function** get\_LoginName() {

**return** $this->LoginName;

}

**function** get\_EmpName() {

**return** $this->EmpName;

}

**function** get\_FbFunc() {

**return** $this->FBFunc;

}

}

## REST controller, RestController.php

switch($api){

case "getEhsUsers":

// to handle REST Url /FBUser/list/GetEhsUsers

$EHSUserRestHandler = new EHSUserRestHandler();

$EHSUserRestHandler->getEHSUsers();

break;

case "getEhsTechUsers":

// to handle REST Url /FBUser/list/ GetEhsTechUsers

$EHSUserRestHandler = new EHSUserRestHandler();

$EHSUserRestHandler->getEHSTechUsers();

break;

case "single":

echo "single ";

// to handle REST Url /FBUser/show/<id>/

$EHSUsersRestHandler = new EHSUserRestHandler();

$EHSUsersRestHandler->getUser($\_GET["id"]);

break;

## REST service handlers, to perform the service detail, EHSUserRestHandler.php

## Data and data access operation – FbUserDao.php

## PHP RESTful services debugging and testing

## Traditional debugging with break points and walk thru the code

## Advanced REST Client - DEMO

## Postman

## Security of EHS RESTful web services:

1. User authentication: User authentication is followed with embedded AUTH token in the API URL call to authenticate and identify the valid user to access the resource
   1. Need extra time to architect and design this
2. Single sign on (SSO):
   1. Through SSO we filter out all the un-authorized accesses, and only allow the EHS users (thru login id) to access the app.
   2. Currently defined 4 different user type. As potential clients call the web services with the following pre-defined user type, web services will grand the access to EHS database.
      1. FB\_EHS\_USER\_IOS
      2. FB\_EHS\_USER\_ANDROID
      3. FB\_EHS\_USER\_WEB
      4. FB\_EHS\_USER\_OTHER

## After we got EHS web services, then, what is the next step?

## Put things together and deploy the services in a server environment like ash server machine, test the deployment with client API calls.

## We can start working on the client-side development that utilize the web services

## SSO on IOS app to use the GetEhsUser/GetEhsTechUser api calls for verification of the user and allow the use of app.

## New features

**Things needed to improve on EHS web services**

1. URL Rewrite
2. JSON response
3. How Facebook do IOS app?
   1. Facebook IOS user group
   2. Facebook mobile app development process and rules to follow
   3. How to upload app to store and make it available to only a group of valid people to

download

## Platforms and feature functionalities



## Web Services

## IOS APP

## iPad – Larger real estate with features

## iPhone

## iWatch

## Android

## Road map

## Web services

## Legacy features

## SSO

## Incidents

## Head count and labor hours

## New features that can fully utilize the power of mobile devices like camera, map, GPS etc.

## Inspections – to integrate camera features into the app

**Data operation on the EHS portal SQL server database:**

* For the first phase, we restrict database access to existing EHS db tables (FbUser, Incidents etc.) to READ ONLY access
* We will create new tables on the same database for any new IOS features and functionalities.
* Any update/delete operations to the current existing table will need to be discussed and pre-approved by EHS technical team.