The We Love Programming Facebook Forum API.

Written By Devon Hazelett (XDEV)

TABLE OF CONTENTS

1. There is Always Something TODO...
2. Everybody is Challenged
3. What's Done is Done
4. OOPHP, Functions and Classes
5. Database Schema
6. About the Highlighter
7. Server Requirements
8. Thanks

# Chapter 1: There is Always Something TODO...

Create an ACTUAL framework and get a good structure.. This is where I went worng the first try was no structure and did ALL the work first when it should be PLAN, PLAN, PLAN then work.

Convert Functions->Classes

Create a DB setup page, so there is no need to modify the PHP pages.

Who's Online

The Top Coder, and or The top Code.

Announcement Section or popup div/span

# Chapter 2: Everybody is Challenged

//

# Chapter 3: What's Done is Done

//

# Chapter 4: OOPHP, Functions && Classes

//

# Chapter : The AJAX System.

This is one of the first forums ever based solely off of PHP and AJAX, In that light makes it the most dynamic, fastest, and one of a kind forum to date.

If you ever think about when you are in a chat room, you don't have to reload the page you just sit there and wait for a reply, well the same thing happens in this forum! You don't have to push F5 a million times waiting for a reply OR sit at your email and wait for that "Instant email notification" to come.   
No I believe that that is just wrong and there is no reason it should not just load in front of your face...  
With this Framework everything I'm discussing is possible.

# Chapter : Database Schema and Object Models

KEYLIST:  
[] = notes/idea/thought/reminders; () = TODO/implementing; ([]) =inside of the () is an e.g.;  
Need to make IS\_PUBLIC a class for visibility options. [none class style shown below]

Registration

::Form{

->id [value is NULL]

->REG\_TIME\_DATE

->first

->last

->password (sha1)

->confirm

->email

->reCapcha {

->Public Key: 6LcNF8YSAAAAAPlwUBgM3joLgGzxiLK2YQEUmttk

->Private Key: 6LcNF8YSAAAAAFYMzdzdPqIdgSEdCGF0J4YqmHOE

}

}

Users

::Profile{

->Facebook->(userid)

->&DB->id(auto inc[++]) (set to NULL on registrations)

}

->Name{

->First{IS\_PUBLIC Y?N ((SHOW\_FN == TRUE) : (SHOW\_FN == FALSE))}

->Last{IS\_PUBLIC Y?N ((SHOW\_LN == TRUE) : (SHOW\_LN == FALSE))}

}

->email {

->IS\_PUBLIC Y?N ((SHOW\_EMAIL == TRUE) : (SHOW\_EMAIL == FALSE))

}

->Signature{

->HAS\_SIGNATURE Y?N ((SHOW\_SIG == TRUE) : (SHOW\_SIG == FALSE))

}

->Description{

->HAS\_UDESC Y?N ((SHOW\_DESC == TRUE) : (SHOW\_DESC == FALSE))

}

->Fav\_PLang [dropdown\_list->SELECT langs FROM favs]

->Fav\_OS [dropdown\_list->SELECT opsys FROM favs]

->Fav\_IDE [dropdown\_list->SELECT ide FROM favs]

->Reputation::{

->Level(if HREP == TRUE {SET\_REP\_LVL = $rep\_leve})

->Given(GAVE\_REP Y?N (GREP = count[times\_repd], TRUE) : (GREP = 0, FALSE))

->Received(HAS\_REP Y?N ((HREP = count[num\_repd], TRUE) : (REP = 0, FALSE))

->Rep\_Tag(if (HREP = TRUE; SET\_REP\_TAG->Flow)

->Flow{ [as Ratio -0 : +0] if (Flow > 0 Y?N (&Positive : &Negative))

->Positive{

->Flow >= 1 {"Just Starting"}

->Flow >= 10 {"Average"}

->Flow >= 20 {"Undergrad"}

->Flow >= 35 {"Associate's Degree"}

->Flow >= 50 {"Bachelor's Degree"}

->Flow >= 100 {"Master's Degree"}

->Flow >= 200 {"Created Theory of Relativity."}

}

->Negative{

->Flow <= -1 {"Made a Mistake"}

->Flow >= -10 {"TROLLING"}

->Flow >= -20 {"STRIKE ONE"}

->Flow >= -35 {"STRIKE TWO"}

->Flow >= -50 {"ON THE ROAD TO A BAN"}

->Flow >= -100 {"ON LAST STRIKE (3)"}

->Flow >= -200 {"BANNED"}

}

}

}

->Repositories::HAS\_REPO {

->&Languages

->

->

}

->Donator::is\_donor->Level {Y?N ((DONOR\_ICON == TRUE):(DONOR\_ICON == FALSE))}

->User\_Level (Admin, Moderator, Registered, Developer, Guest)

}

::Posts{

->Users::Profile->id

->Users::Profile->ProPic

->Users::Profile->name

->Threads::Categories->

->Threads::Categories->post\_icon

->Threads::Categories->id

->Threads::Categories->title

->Threads::Categories->HAS\_REP Y?N ((POST\_REP = $prep) : (POST\_REP = NULL))

->Threads::Categories->IS\_TOP Y?N ((IS\_TOP = SET\_TOP\_POST ICON):(IS\_TOP = FALSE))

}

Threads

::Categories{

->&Language

->id

->has\_parent || is\_parent

->post\_title[SELECT ptitle FROM posts]

->posted\_by == Users::Posts->Name [actual will be SQL]

->post\_icon = \_\_THREAD\_TYPE\_ICON\_\_::(Question, Article, Tutorial);

->is\_sticky ? Y:N ((POST\_TO\_INDEX == TRUE) : (POST\_TO\_INDEX == FALSE))

}

Reputation

&Users{

Level->(if HREP == TRUE {SET\_REP\_LVL = $rep\_leve})

Given->(GAVE\_REP Y?N (GREP = count[times\_repd], TRUE) : (GREP = 0, FALSE))

Received->(HAS\_REP Y?N ((HREP = count[num\_repd], TRUE) : (REP = 0, FALSE))

}

Repositories

::Languages{

TYPES(C->CPP->CS->JAVA->JS->X/HTML->XML->CSS->VB...)

DISPLAY\_USER\_INFO(&User)

IS\_ TOP Y?N ((IS\_TOP = SET\_TOP\_POST ICON):(IS\_TOP = FALSE))

DISPLAY\_CODE(&TYPES->$code)

}

}

# Chapter : About the Highlighter

//

# Chapter : Server Requirements

//

# --Thanks