BitWasp Architecture Analysis

ampedup [amphetamine@tormail.org]

Contents

1   Introduction

This document aims to provide a clear, well specified summary of the BItWasp project and very clear indiciations of its progress so far based solely on the existing source for BitWasp.

1.1   Rationale

Properly specifying BitWasp and evaluating its progress is a key step before performing any modifications to the software. This information will be invaluable to further project decision making and will also greatly benefit the original BitWasp project.

1.2   Methodology

The structure analysis of the software has been done using Visual Paradigm to extract class data organized by directory. Dependencies and coding conventions are discovered through both the UML class design and manual code auditing.

2 Coding Conventions

2.1 Naming Conventions

2.1.1 Classes

2.1.2 Methods

2.1.3 Attributes

2.1.4 Instance Variables

2.2 Syntax Conventions

2.2.1 PHP Tags

2.2.2 Classes

2.3 Layout Conventions

3 Dependencies

3.1 Frameworks

3.1.1 CodeIgniter

3.2 PHP Libraries

3.2.1 php-gd

3.3 External Software

3.3.1 GPG

3.3.2 bitcoind

4 Path Structure

4.1 Application (application/)

config

controllers

errors

language

libraries

models

views

4.2 System (system/)

core

database

helpers

language

libraries

4.3 Root directory (/)

5 Class Structure

5.1 Controllers

5.1.1 Account

Description: The class handles the Account management interface. Extends CI_Controller.

Attributes: None.

Methods:

__construct()
Constructor for the Account object. Calls the parent constuctor and loads required libraries and models.

index()
Loads data for the account/index page ("My Account") and loads the layout library.

edit()
Loads data for the account/edit page ("Edit Account") and loads the layout library.

replacePGP()
Loads data for the account/replacePGP page ("Replace PGP key"), handles the updating of PGP keys and loads the layout library.

update()
Handles updating of the account information.

deletePubKey()
Delete the stored pubKey for the user, and disable two-step authentication if necessary.

5.1.2 Admin

Description: The class handles the Admin management interface. Extends CI_Controller.

Attributes: None.

Methods:

__construct()
Constructor for the Account object. Calls the parent constuctor and loads required the model.

index()
Loads data for the admin/siteConfig page ("Admin Panel") and loads the layout library.

users($userHash = NULL)

Loads data for the admin/users page ("Users") and loads the layout library.

editConfig()

Loads data for the admin/editConfig page ("Edit Configuration") and loads the layout library.

updateConfig()

Handles updating of the site configuration.

fixOrphans()

Handles the removal of orphan categories.

removeCategory()

Loads data for the admin/removeCategory page ("Remove Category"), handles category removal and loads the layout library.

addCategory()

Loads data for the admin/addCategory page ("Add Category"), handles category addition and loads the layout library.

check_parentID_exists($parentID)

Method does not follow naming conventions! Checks that the category identified by $parentID has a parent that exists.

check_category_exists($id)

Method does not follow naming conventions! Checks that the category identified by $id exists.

5.1.3 Bitcoin

Description: The class handles the Bitcoin management interface. It is in its very early stages. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

Constructor for the Account object. Calls the parent constuctor and loads required the model.

index()

Loads some test data using a JSON interface to a hardcoded bitcoin server using hardcoded credentials and loads the layout library.

5.1.4 Error

Description: The class handles displaying of errors. Extends CI_Controller.

Attributes:

8

$params

    Array instance variable not actually used in the code.

Methods:

__construct()

    Empty constructor.

index($params)

    Loads data pertaining to the error and displays the error.

5.1.5 Home

Description: The class handles display of the home page. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

    Constructor for the Home object. Calls the parent constuctor.

index()

    Loads data for the home/index page ("Home") and loads the layout library.

5.1.6 Items

Description: The class handles the display of items in the marketplace. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

    Constructor for the Items object. Calls the parent constuctor and loads required the models and libraries.

tmp_items_per_page()

    Method does not follow naming conventions! Handles the changing of items shown per page for the session.

index()

    Loads data for the items/index page ("Items"), "paginates" the lastest items and loads the layout library.

view($itemHash)

    Loads data specified item and displays it if it exists, otherwise it "paginates" the lastest items loads the layout library. Display of user feedback is commented out.

cat($catID = FALSE)

Loads items for specified category if it exists and "paginates" the items, otherwise it "paginates" the latest items and loads the layout library.

check_category_exists($id)

Method does not follow naming conventions! Checks whether or not the specified category exists.

5.1.7   Listings

Description: The class handles the management of listings in the marketplace. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

Constructor for the Listings object. Calls the parent constuctor and loads required the models and libraries.

edit($itemHash)

Loads data for the listings/editConfig page ("Edit Configuration") and loads the layout library.

manage()

Loads data for the listings/manage page ("Manage Listings") and loads the layout library.

images($itemHash)

Loads data for the listings/images page ("Item Images") and loads the layout library.

mainImage($imageHash)

Handles the selection of the main image for listings, loads the data for the result and loads the layout library.

imageRemove($imageHash)

Handles the removal of images for listings, loads the data for the result and loads the layout library.

imageUpload($itemHash)

Handles the upload of images for listings, loads the data for the result and loads the layout library.

remove($itemHash)

Handles the removal of items from vendor listings, loads the data for the result and loads the layout library.

create()

Loads data for the listings/addItem page ("Add Item") and loads the layout library.

5.1.8 Messages

Description: The class handles the management of user messages. Extends CI_Controller.

Attributes: None.

Methods:

__construct()
Constructor for the Messages object. Calls the parent constuctor and loads required the models and libraries.

inbox()
Loads data for the messages/inbox page ("Inbox") and loads the layout library.

delete($messageHash)
Handles the deletion of messages, loads the data for the result and loads the layout library.

read($messageHash)
Loads data for the specified message for the messages/read page and loads the layout library.

remove($itemHash)
Handles the removal of items from vendor listings, loads the data for the result and loads the layout library.

send($toHash = NULL)
Loads data for the messages/send page ("Send Message"), handles message sending and loads the layout library.

5.1.9 Orders

Description: The class handles the management of orders. Extends CI_Controller.

Attributes: None.

Methods:

__construct()
Constructor for the Orders object. Calls the parent constuctor and loads required the models and libraries.

index()
Loads data for the orders/index page ("Orders") and loads the layout library.

recount()
Handles the update of order quantities, loads the data for the result and loads the layout library.

orderItem($itemHash)

Handles the creation of a new order for the specified item, loads the data for the result and loads the layout library.

place($sellerHash)

Handles the placement of orders with vendors, loads the data for the result and loads the layout library.

review($id)

Handles the placement of feedback on orders, loads the data for the result and loads the layout library.

confirmDispatch($buyerHash)

Handles the update of dispatch status on orders, loads the data for the result and loads the layout library.

confirmPayment($buyerHash)

Handles the payment confirmation on orders, loads the data for the result and loads the layout library.

purchases()

Loads data for the orders/purchases page ("Purchases") and loads the layout library.

callback_check_rating($param)

Method does not follow naming conventions! Checks whether the feedback value provided is within the specified range (1-5 inclusive).

5.1.10 Pages

Description: The class handles display of the custom pages. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

Constructor for the Pages object. Calls the parent constuctor.

view($page = 'home')

Loads data for the specified page and loads the layout library.

5.1.11 Users

Description: The class handles the user management interface. Extends CI_Controller.

Attributes: None.

Methods:

__construct()

Constructor for the Users object. Calls the parent constuctor and loads required the models and libraries.

registerPGP()

Loads data for the users/registerPGP page ("Setup PGP"), handles PGP key registration and loads the layout library.

twoStep()

Loads data for the users/twoStep page ("Two Step Authentication"), handles verification of the two step authentication and loads the layout library.

logoutInactivity()

Loads data for the users/login page ("Login"), notifies the user that the previous session has timed out, and loads the layout library.

view($userHash)

Loads data for the users/individual page for the specified user and loads the layout library.

login()

Loads data for the users/login page ("Login"), handles user credential and CAPTCHA authentication and loads the layout library.

logout()

Logs out the user and redirects them to the users/login page.

register()

Loads data for the users/register page ("Register"), handles user registration and CAPTCHA authentication and loads the layout library.

find_role($roleId)

Method does not follow naming conventions! Resolves the role ID to a string "Vendor" or "Buyer".

check_captcha($string)

Method does not follow naming conventions! Checks the validity of the CAPTCHA answer.

register_check_role($role)

Method does not follow naming conventions! Checks whether the role value provided is within the specified range (1-2 inclusive).

5.2 Libraries

5.2.1 General

5.2.2 jsonRPCClient

5.2.3 Layout

5.2.4 My_captcha

5.2.5 My_config

5.2.6 My_image

5.2.7 My_session

5.3 Models

5.3.1 Accounts_model

5.3.2 Admin

5.3.3 Bitcoin

5.3.4 Error

5.3.5 Home

5.3.6 Items

5.3.7 Account

5.3.8 Admin

5.3.9 Bitcoin

5.3.10 Error

5.3.11 Home

5.3.12 Items