نگاه کلی به تابع class route و چگونگی کارکرد این تابع:

تابع route Class برای بخش بندی قسمت ها و جدا کردن مسیرها از هم میباشد:

<?php

class Route

{

private $\_uri = array();

private $\_method = array();

/\*\*

\* Builds a collection of internal URL's to look for

\* @param type $uri

\*/

public function add($uri, $method = null)

{

$this->\_uri[] = '/' . trim($uri, '/');

if ($method != null) {

$this->\_method[] = $method;

}

}

/\*\*

\* Makes the thing run!

\*/

public function submit()

{

$uriGetParam = isset($\_GET['uri']) ? '/' . $\_GET['uri'] : '/';

foreach ($this->\_uri as $key => $value)

{

if (preg\_match("#^$value$#", $uriGetParam))

{

if (is\_string($this->\_method[$key]))

{

$useMethod = $this->\_method[$key];

new $useMethod();

}

else

{

call\_user\_func($this->\_method[$key]);

}

}

}

}

}

?>

Metod در این تابع ارایه ای است که مقادیر تابع add که در index است را میگیرد.

تابع index:

<?php

/\* Tortoise Automation System

\* Open Source Software

\* github : https://github.com/tortoisePL/tortoise

\*/

include 'route.php';

include 'src/about.php';

include 'src/dashboard.php';

include 'src/contact.php';

include 'src/login.php';

include 'src/logout.php';

include 'src/user.php';

include 'src/profile.php';

$route = new Route();

$route->add('/', 'Dashboard');

$route->add('/dashboard', 'Dashboard');

$route->add('/about', 'About');

$route->add('/contact', 'Contact');

$route->add('/user', 'User');

$route->add('/user/users', 'Users');

$route->add('/profile', 'Profile');

$route->add('/contact/add', 'add');

$route->add('/login', 'Login');

$route->add('/logout', 'Logout');

$route->submit();

?>

در if اول از تابع route متد برسی میشود که خالی نباشد اگر خالی باشد تابع اجرا نمیشود چون باید یکی از مقادیر تابع add در ان باشد تا بتواند وارد صفحه(قسمت) مربوطه شود.

در if دوم از تابع route مقاذیر متد با اندیس ان در تابع add برسی میشود اگر برابر بود وارد if سوم میشود که در این if اگر رشته باشد تابع به درستی اجرا میشود و وارد class یکی از مقادیر add میشود درغیر اینصورت تابع خطا گیری php فعال میشود و خطای مورد نظر را میگوید.

در تست جعبه سفید باید ifهای تابع مورد برسی قرار گیرد:

1.اگر در if اول مقدار متد برار خالی باشد چه اتفاقی میفتد

2. اگر در if اول مقدار متد خالی نباشد تابع چگونه اجرا میشود

3.در if دوم مقادیر متد چگونه باشد تا تابع اجرا شود

4.در if سوم مقداری که وارد میشود عدد است یا کاراکتر یا رشته

4.در if سوم اگر مقدار متد خالی باشد تابع چگونه اجرا میشود