



SinglePageCubiApp

Write Cubi Single Page App with Angularjs

Updated May 17, 2014 by [rockys...@gmail.com](#)

Write Cubi Single Page App with Angularjs

After the initial Angularjs rewrite RewriteCubiUIWithAngular, it is time to apply singlepage application (SPA) model on Cubi.

Single Page Application (SPA)

What is SPA? A single-page application (SPA), also known as single-page interface (SPI), is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application. This definition is copied from Wikipedia http://en.wikipedia.org/wiki/Single-page_application.

Single page applications (SPA) are more capable of decreasing load time of pages by storing the functionality once it is loaded the first time, allowing easier data transfer between pages and a more complex user interface instead of trying to control so much from the server. This allows for more interference from an end user.

Write Cubi as SPA

Angularjs routeprovider is the main tool to build SPA. [https://docs.angularjs.org/api/ngRoute/provider/\\$routeProvider](https://docs.angularjs.org/api/ngRoute/provider/$routeProvider)

In Cubi view template, we define additional section for routing.

```
<body ng-app="cubiviewApp">

<script>
var APP_INDEX = '{$app_index}';
{literal}
var cubiviewApp = angular.module('cubiviewApp', ['ngResource', 'ngRoute']);
cubiviewApp.config(['$routeProvider', '$locationProvider',
    function($routeProvider, $locationProvider) {
        $routeProvider
            .when(APP_INDEX+'/:module/:view/:id', {
                templateUrl: function(params){ return APP_INDEX+'/'+params.module+'/'+params.view+'/'+params.id+'?parti
            })
            .when(APP_INDEX+'/:module/:view', {
                templateUrl: function(params){ return APP_INDEX+'/'+params.module+'/'+params.view+'/?partial=1'; }
            });

        // configure html5 to get links working on jsfiddle
        $locationProvider.html5Mode(true);
    }]);
{literal}
</script>
```

The routing config tells

- url like /system/user_list loads page from http://host/cubi/index.php/system/user_list/?partial=1
- url like /system/user_detail/347 loads page from http://host/cubi/index.php/system/user_detail/347?partial=1

Here we add a partial parameter to render a view with content section (without render header, footer and left menu). It also means no download for resource like css, js and etc.

In the view template, we set

```
<div ng-view></div>
```

as the placeholder for all content view.

Cubi Index File Change

Cubi used to have *forward.php* to route request url to specific view. Now that we have Slim, we rewrite the routing logic with Slim in *index.php*

```
include_once 'bin/app_init.php';
include_once OPENBIZ_HOME."/bin/ErrorHandler.php";

require 'bin/Slim/Slim.php';
\Slim\Slim::registerAutoloader();

$app = new \Slim\Slim();

// start session context object
BizSystem::sessionContext();

// GET view
```

```
$app->get('/:module/:view(/:id)(?::querystring+)', function ($module,$view,$id=null,$querystring=null) {  
    $app = \Slim\Slim::getInstance();  
    // get real view name module.view.viewnameview  
    $realviewName = getViewName($view, $module);  
    // render view  
    $viewObj = BizSystem::getObject($realviewName);  
    if (!$viewObj) {  
        // render NOTFOUND_VIEW  
        return;  
    }  
    if ($id) $viewObj->setRequestId($id);  
    $viewObj->render();  
});  
  
$app->run();
```

► [Sign in](#) to add a comment

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)