<?php

class IndexController extends SMCore\_Controller\_Action

{

public function indexAction()

{

$cookie = SMCore\_Auth\_CookieGCE::getInstance();

$model\_user = new Model\_User();

$mail\_exist = (count($model\_user->search('user\_mail', trim($cookie->mail))) > 0)?true:false;

if($cookie->mail == ''){

header("Location: /homepage/index/emailform/");

exit();

}else if(!$mail\_exist){

$tooken = base64\_encode($cookie->mail);

header("Location: /homepage/index/gettoken/token/$tooken/");

}

$params = $this->\_request->getParams();

$profile = $this->view->userProfile;

$objIdea = new Model\_Idea();

//get data for block1

$objIdeaStatus = new Model\_IdeaStatus();

$userId = $profile['user\_id'];

$arrCountStatus = $this->\_countIdeaByStatus($userId);

// echo "<pre>";

// print\_r($arrCountStatus);

// echo "</pre>";

$arrLastestUpdatedStatusIdea = $objIdea->getLastestUpdatedStatusIdeas($userId, $limit = 1);

// get my fev idea to display button

$wheres = array();

$wheres[] = "ideafav\_fk\_user\_id = " . $profile['user\_id'];

$numMyFavIdeas = $objIdea->getTotalMyFavIdeas($wheres);

// get my idea to display button

$wheres = array();

$wheres[] = "idea\_fk\_user\_id = {$profile['user\_id']} OR idea\_fk\_porteur\_id = {$profile['user\_id']}";

$numMyIdeas = $objIdea->getTotalIdeas($wheres);

// get my contribution idea to display button

$wheres = array();

$wheres[] = "idea\_comment\_user\_id = " . $profile['user\_id'];

$numMyContributionIdeas = $objIdea->getTotalMyContributionIdeas($wheres);

//get data for block2

$objNews = new Model\_IdeaNews();

$arrIdeaNews = $objNews->getArray(0, 0, array(), array('ideanews\_id DESC'));

//get data for block3 & block4

$rowsDernieres = $objIdea->getLastestIdeas(null, 4, array('idea\_fk\_status\_id <> 6'));

// get Idea Appréciées

$modelR = new Model\_IdeaRecommend();

$rowsAppreciees = $objIdea->getMostIdeaLike(10, array('idea\_fk\_status\_id <> 6'));

$temp = $rowsAppreciees;

for ($i = 0; $i < count($temp); $i++) {

$rowsAppreciees[$i]['numRecomment'] = $modelR->getNumRecommendByIdeaId($temp[$i]['idea\_id']);

}

$column = array();

//sort by num of the like (Recomment)

foreach ($rowsAppreciees as $key => $value) {

if (isset($value['numRecomment'])) {

$column[] = $value['numRecomment'];

} else {

$column[] = 0;

}

}

array\_multisort($column, SORT\_DESC, $rowsAppreciees);

//get most Idea Commentees

$rowsCommentees = $objIdea->getMostCommentIdeas(10);

$this->view->rowsDernieres = $rowsDernieres;

$this->view->rowsAppreciees = $rowsAppreciees;

$this->view->rowsCommentees = $rowsCommentees;

$this->view->arrCountStatus = $arrCountStatus;

$this->view->arrIdeaNews = $arrIdeaNews;

$this->view->arrLastestUpdatedStatusIdea = $arrLastestUpdatedStatusIdea[0];

$this->view->assign(

'numIdeas',

array(

'numMyIdeas' => $numMyIdeas,

'numMyFavIdeas' => $numMyFavIdeas,

'numMyContributionIdeas' => $numMyContributionIdeas

)

);

}

public function annuaireAction() {

$config = Zend\_Registry::get('configuration');

$this->view->assign(

'config',

$config

);

$layout = Zend\_Layout::getMvcInstance();

$layout->setLayout('annuaire');

}

private function \_countIdeaByStatus($userId)

{

$objIdea = new Model\_Idea();

$objIdeaStatus = new Model\_IdeaStatus();

//get data all status

//$wheres = array("status\_id IN (0, 7, 2, 6)");

$result = $objIdeaStatus->getAllStatus();

// get total Idées Brouillon:

$numBrouillon = $objIdea->getNumStatusIdea(0, $userId);

$result[0]['num'] = $numBrouillon;

// get total idées dans le lab :

$numLab = $objIdea->getNumStatusIdea(7, $userId);

$result[7]['num'] = $numLab;

//get total idées en cours :

$numCours = $objIdea->getNumStatusIdea(array(1,3), $userId);

$result[2]['num'] = $numCours;

//get total idées refusée(s) :

$numRefusee = $objIdea->getNumStatusIdea(6, $userId);

$result[6]['num'] = $numRefusee;

return $result;

}

public function changelogAction()

{

$this->getHelper('layout')->setLayout('changelog');

}

public function nopermissionAction()

{

$this->\_helper->layout()->setLayout('nopermission');

$this->\_helper->viewRenderer->setNoRender(true);

}

public function gettokenAction(){

$params = $this->\_request->getParams();

if($params['token']!=''){

$email = base64\_decode($params['token']);

$cookieVar = "uid=%40it-ce&lastname=&firstname=&uidregion=&uidprod=55001886&localisation=CELidentifiant%3dRILLIEUX+LA+PAPE%2cOU%3dlocalisations%2cOU%3dIT-CE%2cO%3dcaisse-epargne&codemetiern=759101&codemetierr=759101&role=MA&esrattachement=&eshierarchique=&mail=" . $email . "&user\_mail=$email";

$current\_domain = $\_SERVER['HTTP\_HOST'];

$chaine\_domaine = explode(".", $current\_domain);

$final\_domain = $chaine\_domaine[count($chaine\_domaine) - 2] . "." . $chaine\_domaine[count($chaine\_domaine) - 1];

$domain = preg\_replace("/^(.\*\.)?([^.]\*\..\*)$/", "$2", $final\_domain);

setCookie('CIDF', '', (time() - 3600), '/');

setcookie('CIDF', $cookieVar, time() + 365 \* 86400, '/', $domain);

}else{

header("Location: /");

exit();

}

$this->\_helper->layout()->email = $email;

$this->\_helper->layout()->setLayout('register\_form');

$this->\_helper->viewRenderer->setNoRender(true);

}

public function emailformAction(){

$this->\_helper->layout()->setLayout('email\_form');

$this->\_helper->viewRenderer->setNoRender(true);

}

public function checkemailAction(){

$validator = new Zend\_Validate\_EmailAddress();

if ($validator->isValid($\_POST['email'])) {

$config = Zend\_Registry::get('configuration');

$myEmail = $\_POST['email'];

$arr = explode("@", $myEmail);

$myExtension = $arr[1];

$emailExtension = $config->site->email->extension;

$emailExtension = str\_replace(" ", "", $emailExtension);

$extensionArr = explode(",",$emailExtension);

if(in\_array($myExtension, $extensionArr)){

$url = "http://{$\_SERVER['HTTP\_HOST']}/homepage/index/gettoken/token/". base64\_encode($\_POST['email']."/");

$m = new Zend\_Mail();

$m->setBodyText("Please click the link to access <a href='$url'>$url</a>");

$m->setFrom('noreply@sutrixmedia.com', 'Sutrix Media');

$m->addTo($\_POST['email'], 'Guest');

$m->setSubject('Login Email');

$rs = $m->send();

if($rs!==false)

$msg = "Lien Admin est envoyé à votre adresse email";

else

$errorMsg = "Envoyez un mail a échoué. S'il vous plaît vérifier votre configuration";

}else{

$errorMsg = "Email Extension n'est pas de laisser";

}

} else {

$errorMsg = "Votre adresse email '{$\_POST['email']}' pas valide. Merci de saisir votre adresse complète de type prenom.nom@mecaisse.caisse-epargne.fr";

}

$this->\_helper->layout()->setLayout('email\_form');

$this->\_helper->viewRenderer->setNoRender(true);

$this->\_helper->layout()->msg = $msg;

$this->\_helper->layout()->errorMsg = $errorMsg;

}

public function userprofileAction(){

if($\_POST){

$cookie = SMCore\_Auth\_CookieGCE::getInstance();

print\_r($cookie);

$\_POST['user\_mail'] = ($cookie->user\_mail!='')?$cookie->user\_mail:$cookie->mail;

//$\_POST['user\_mail'] = $\_

//$dbAdapter = Zend\_Db\_Table::getDefaultAdapter();

//$authAdapter = new Zend\_Auth\_Adapter\_DbTable($dbAdapter);

//$query = "select \* from user";

//print\_r($authAdapter);

//$u = new Module\_User();

$objUser= new Model\_User();

//$is\_exist = $objUser->search('user\_mail', $\_POST['user\_mail']);

$objUser->setOptions($\_POST);

$objUser->save();

// if(){

// print "ok";

// }else{

// print "error";

// }

//$dbAdapter = Zend\_Registry::get("db");

}

exit();

}

}

**Model\_User** extends SMCore\_Model\_User

{

}

class SMCore\_Model\_User {

\_mapper => SMCore\_Model\_Mapper\_UserMapper

save(Array $filter\_array = array()){

return $this->getMapper ()->save ( $this, $filter\_array, true, 'user\_id' );

}

class SMCore\_Model\_DbTable\_User extends Zend\_Db\_Table\_Abstract{

}

class SMCore\_Model\_Mapper\_UserMapper

{

public function getDbTable()

{

if (null === $this->\_dbTable){

$this->setDbTable ( 'SMCore\_Model\_DbTable\_User' );

}

return $this->\_dbTable;

}

public function setDbTable(**$dbTable** = null)

{

if (is\_string ( $dbTable )){

$dbTable = new **$dbTable** ( );

}if (! $dbTable instanceof Zend\_Db\_Table\_Abstract){

throw new Zend\_Exception ( 'Invalid table data gateway provided' );

}

$this->\_dbTable = $dbTable;

return $this;

}

public function save(SMCore\_Model\_User $object, Array $filter\_array = array(), $ignoreEmptyValuesOnUpdate = true, $pkey = 'id'){

…...............

//thiết lập dữ liệu, column

if ($ignoreEmptyValuesOnUpdate){

foreach ( $data as $key => $value )

if (is\_null ( $value ) or $value == '')

unset ( $data [$key] );

}

if (null === ($id = $object->getUserId ())){ <== **check id tồn tại hay không**

unset ( $data ['user\_id'] );

$this->getDbTable ()->insert ( $data );

return $this->getDbTable ()->getDefaultAdapter ()->lastInsertId ();

} else {

if ($this->getDbTable ()->update ( $data, array ('user\_id = ?' => $id ) ) < 0){

return false;

} else {

return $id;

}

}

}

}