UploadFom.php

$this->setAttribute(‘enctype’, ‘multipart/form-data’)

$this->add(

Array(

‘type’ => ‘file’,

‘name’ => ‘image’,

‘attributes’ => array(‘class’ => ‘form-control’)

‘options’ => array(‘label’ => ‘Image’)

$input=new InputFilter;

$this->setInputFilter($input);

$input->add(

‘name’ => ‘image’,

‘type’ => ‘Zend\InputFilter\FileInput’,

‘required’ => true,

‘filter’ => array(

Array( ‘name’ => ‘FileRenameUpload’,

‘options’ => array(

‘target’ => ‘./data’,

‘useUploadName’ => true,

‘useUploadExtension’ => true,

‘overwrite’ => true,

‘randomsize’ => false,

‘validators’ => array(

Array( ‘name’ => ‘FileUploadFile’),

Array(‘name’ => ‘FileisImage’),

Array(‘name’ => ‘FileImageSize’,

‘options’ => array(

‘minWidth’ => 100,

‘minHeight’ => 100,

Controller:

//$data=$this->getRequest()->getFiles()->toArray();

//$data=$this->params()->fromFiles();

//$data=$this->params()->fromFile(‘image’);

$form = new UploadForm;

If($this->getRequest()->isPost()){

$request = $this->getRequest();

$post = $request->getPost()->toArray(); // Lấy dữ liệu từ post

$upload= $request->getFiles()->toArray(); // Lấy dữ liệu từ upload file

If($upload[‘image’][‘name’] != “”){

$data=array\_merge\_recursive($post,$upload); // Gộp mảng

}else{

$data= $post;

$form->setValidationGroup(‘name’); // Thiết lập Validator nào active

}

$form->setData($data);

If($form->isValid()){

$dataUpload = $form->getData();

---Multi Upload

$data = range(1,100) // Tạo mảng giá trị từ 1 -> 100 (Key : 0 - 99)

array\_combine(arr\_key , arr\_val) // array\_combine: gộp 2 mảng với khóa là mảng arr\_key và giá trị là arr\_val

Controller: Input

$form = new MultiFileForm;

$num = $this->getRequest()->getPost(‘number’);

$form->get(‘hidden’)->setValue($num); // Truyền giá trị

for($i = 1; $i <= $num; $i++){

$form->add(array(

‘name’ => ‘img’.$i,

‘type’ => ‘file’,

‘attributes’ => array(‘class’ => ‘form-control’),

‘options’ => array(‘label’ => ‘File .’$i)

View

Echo $this->formElement($form->get(‘hidden’));

For($i = 1; $i <= $this->num; $i++){

Echo $this->formLabel($form->get(‘img’.$i));

Echo $this->formElement($form->get(‘img’.$i));

Echo $this->formElementErrors($form->get(‘img’.$i));

Controller: Process

$request = $this->getRequest();

$num = $request->getPost(‘hidden’);

for($i = 1; $i <= $num; $i++){

$form->add(array(

‘name’ => ‘img’.$i,

‘type’ => ‘file’,

‘attributes’ => array(‘class’ => ‘form-control’),

‘options’ => array(‘label’ => ‘File .’$i)

$files = $request->getFiles()->toArray();

Foreach($files as $k=>$file){

If($file[‘name] == ‘’){

Unset($files[$k]); // Loại bỏ phần tử rỗng

$form = new MultiFileForm;

$form->addInputFilter($files)

$form->setData($files);

If($form->isValid()){

$data = $form->getData();

}else{

// Error

$form->get(‘hidden’)->setValue($num)

$view = new ViewModel(array(‘form’=>$form, ‘num’=>$num))

$view->setTemplate(‘train/multi-upload/select’)

Form:

Public function addInputFilter($files){

$input = new InputFilter;

$this->setInputFilter($input);

Foreach($files as $name=>$file){

$input->add(array(

‘name’ => $name,

‘type’ => ‘Zend\InputFilter\InputFilter’,

‘required’ => true,

‘filters’ => array(),

‘validators’ => array()