

```
<?php

/**
 * Controller for /purchase
 *
 */
class StaticPurchaseController extends PageController
{
    /**
     * @var PaymentGateway
     */
    var $paymentGateway = false;
    var $baseUrl;
    var $moduleName;
    var $dataObjectClassName;
    var $item;
    var $plural;
    var $itemLowerCase;
    var $pluralLowerCase;

    /**
     * Constructor
     *
     * @return StaticPurchaseController
     */
    function StaticPurchaseController($baseUrl, $moduleName, $dataObjectClassName, $plural = false)
    {
        $this->title = "Order - ". SITE_NAME;

        $this->baseUrl = $baseUrl;

        $this->moduleName = $moduleName;

        $this->dataObjectClassName = $dataObjectClassName;

        // defaults
        $this->item = $dataObjectClassName;
        $this->plural = $plural ? $plural : $dataObjectClassName .'s';

        $this->itemLowerCase = strtolower($this->item);
        $this->pluralLowerCase = strtolower($this->plural);

        if (defined('STATICPURCHASE_ENABLE_PAYMENT_GATEWAY') && STATICPURCHASE_ENABLE_PAYMENT_GATEWAY)
            $this->paymentGateway = PaymentGateway::create(PAYMENT_GATEWAY, array());
    }

    /**
     * Handler for /order
     *
     */
    function defaultAction()
    {
        if (!$this->baseUrl)
            trigger_error('$this->baseUrl not set. Sub class must call
StaticPurchaseController::StaticPurchaseController() with at least the one param.');
```

```
/**
 * Handler for /purchase.new
 *
 */
function newAction()
{
    $this->showPurchase(false);
}

/**
 * Handler for /purchase.retry
 *
 */
function retryAction()
{
    $formClassName = $this->purchaseFormDataClass();
    $purchaseClassName = $this->dataObjectClassName;

    $purchaseObj = new $purchaseClassName();
    $purchaseObj->create($purchaseClassName);
    $form = new $formClassName();

    if(!$form->loadFromSession())
        $this->redirect($this->baseUrl,$this->id());

    $purchaseObj->setFields($form->getRawValues());

    $this->showPurchase($purchaseObj);
}

function beforePurchaseForm(&$item)
{
    // Subclasses can extend
}

function showPurchase($purchaseObj=false)
{
    $idpurchase = $purchaseObj ? $purchaseObj->idpurchase : $this->id();

    if(!$purchaseObj){
        $purchaseObj = DataObject::create($this->item);
        $purchaseObj->name = retrieve('name','all');
        $purchaseObj->purchase_total = retrieve('price','all');
        $this->beforePurchaseForm(&$purchaseObj);
        $this->assignTplVar('purchase', $purchaseObj);
    }
    else
    {
        $this->assignTplVar('purchase', $purchaseObj);
    }

    if($this->paymentGateway)
        PaymentDefaultForm::assignTplVars($this, $this->paymentGateway);

    StaticPurchaseForm::assignTplVars($this);
    $this->assignTplVar('form_action', $this->makeUrl($this->baseUrl, 'post'));
    $this->assignTplVar('form_name', $this->itemLowerCase);
}
```

```
$this->addModuleCss(STATICPURCHASE_TEMPLATE_MODULES, 'main.css');
$this->addTpl($this->purchaseFormTplPath());
}

/**
 * Returns the template path for the purchase form.
 *
 * @return string
 */
function purchaseFormTplPath()
{
    return Module::pathForTpl($this->moduleName, "{$this->itemLowerCase}-form.tpl");
}

function purchaseFormDataClass()
{
    return $this->dataObjectClassName.'PurchaseForm';
}

function purchaseMailClass()
{
    return 'StaticPurchaseMail';
}

function sendCustomerPurchaseCompleteEmail(&$p,&$data)
{
    StaticPurchaseMail::sendCustomerPurchaseCompleteEmail($purchaseObj,$data);
}

function sendClientPurchaseCompleteEmail(&$p,&$data)
{
    StaticPurchaseMail::sendClientPurchaseCompleteEmail($p,$data);
}

/**
 * Handler for /purchase.post
 *
 */
function postAction()
{
    $idevent = retrieve('idevent','all',false);

    $formClassName = $this->purchaseFormDataClass();
    $purchaseClassName = $this->dataObjectClassName;
    $mailClassName = $this->purchaseMailClass();

    $form = new $formClassName();

    $data = $this->retrieveForm($form, $this->makeUrl($this->baseUrl, 'retry'));

    $purchaseObj = new $purchaseClassName();

    $this->purchaseProcess = new staticPurchaseProcess($purchaseObj);
    if (!$this->purchaseProcess->setData($data, $errorMessage))
    {
        $this->addMessage($errorMessage);
        $this->redirect($this->baseUrl, $purchaseObj->primaryKey());
    }
}
```

```

    if ($this->purchaseProcess->payment($data))
    {
        $redirectAction = 'complete';

        // Send Notification Email to Customer
        if(STATICPURCHASE_SEND_CUSTOMER_PURCHASE_COMPLETE_EMAIL)
            $this->sendCustomerPurchaseCompleteEmail(&$purchaseObj,&$data);
    }
    else
    {
        $redirectAction = 'retryPayment';

        $errorMsg = <<<EOT
<strong>Sorry, there was a problem processing your payment. The bank said: </strong>
<br />
<br />
{$purchaseObj->payment->gateway_message}
<br />
<br />
<strong>You may re-attempt your payment below.</strong>

EOT;

        $this->addErrorMessage($errorMsg);
    }

    //Send Notification Email to Client.
    if(STATICPURCHASE_SEND_CLIENT_PURCHASE_COMPLETE_EMAIL)
        $this->sendClientPurchaseCompleteEmail(&$purchaseObj,&$data);

    $this->redirect($this->baseUrl.".$redirectAction", $eventBooking->idevent_booking);
}

/**
 * Handler for /purchase.retrypayment
 *
 */
function retryPaymentAction()
{
    $staticPurchase =& $this->staticPurchase();

    $this->showPurchase($staticPurchase);
}

/**
 * Handler for /purchase.complete
 *
 */
function completeAction()
{
    $this->addModuleTpl(STATICPURCHASE_TEMPLATE_MODULES, 'purchase-complete.tpl');
}

/**
 * Returns the purchase object with the request id.
 *
 */
function &staticPurchase()

```

```
{
    $idpurchase = (int) $this->id();
    $purchaseClassName = $this->dataObjectClassName;

    $staticPurchase = DataObject::byId('cazalyspurchase', $idpurchase);
    var_dump($staticPurchase);
    echo "staticPurchase = DataObject::byId($this->dataObjectClassName, $idpurchase);";
    // fetch all associated data
    $staticPurchase->fetchAllRelatives();
    $this->staticPurchaseProcess = new staticPurchaseProcess($staticPurchase);

    return $staticPurchase;
}

}

?>
```