

```

<?php
/**
 * ProductAPI Controller class
 * PHP versions 5.1.4
 * @date 6-Feb-2015
 * @Purpose: This controller handles an action to retrieve info from Amazon API
 * @filesource
 * @author Lynn K. Patricia
 * @revision
 * @version 2.4
 */

App::uses('Sanitize', 'Utility');
class ProductAPIController extends AppController
{
    var $name = "ProductAPI";

    /**
     * Specifies helpers classes used in the view pages
     * @access public
     */

    var $helpers = array('Html', 'Form', 'Session', 'General');

    /**
     * Specifies components classes used
     * @access public
     */

    var $components = array('RequestHandler', 'Email', 'Upload', 'AmazonMws.AmazonProductsApi',
        'AmazonMws.AmazonMwsRegistration');

    var $paginate = array();

    var $uses = array(''); // For Default Model

    /***** START FUNCTIONS *****/

    #_____#
    /**
     * @Date: 20-Mar-2014
     * @Method : beforeFilter
     * @Purpose: This function is called before any other function.
     */

    function beforeFilter()
    {
        $this->layout = "front_layout_front";
        // -----
        // write the action n controller name in session for referer url
        // -----
        $this->Session->write("REFER_REDIRECT_URL", array(
            'controller' => $this->params['controller'],

```

```

        'action' => $this->params['action'],
    );

    // -----
    // checkUserSession() fxn called to check whether user is logged in or not.
    // -----
    $this->checkUserSession();
    $this->set('common', $this->Common);
    $this->set('title_for_layout', Configure::read('SITE_NAME'));
    $userSession = $this->Session->read("SESSION_USER");
    $this->KiOuiUserId = $userSession['id'];
    // -----
    // add amazon vendor files to the path to fix includes
    // -----
    $path = "../Vendor/";
    set_include_path(get_include_path() . PATH_SEPARATOR . $path);
    // -----
    // Amazon Mws Config
    // -----
    $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
    $this->MwsAccessKeyId = Configure::read('MWS_ACCESS_KEY_ID');
    $this->MwsSecretAccessKey = Configure::read('MWS_SECRET_ACCESS_KEY');
    $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
    $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');
}

/**
 * @Date: 06-Feb-2015
 * @Method : SalesRankFromASIN
 * @Purpose: This function is to query Amazon's API for Sales Rank and Product Category
 * Author: Lynn K. Patricia
 */

function SalesRankFromASIN()
{
    // -----
    // disable layout for json
    // -----
    $this->autoRender = false;
    // -----
    // grab variable from query string
    // -----
    $ASIN = $this->request->params['pass']['0'];
    $MarketplaceID = $this->request->params['pass']['1'];
    //-----
    // check if salesrank is in the database
    //-----
    App::import('Model', 'AmazonAsinSalesRanks');
    $AmazonAsinSalesRanks = new AmazonAsinSalesRanks();
    $SalesRank = $AmazonAsinSalesRanks->fetchSalesRankFromAsin($ASIN, $MarketplaceID, '0');

    if ($SalesRank === FALSE){

```

```

//Ask Amazon

//-----
// salesrank is not in database or too old so we call Amazon
//-----
App::import('Model', 'AmazonProductApiFunctions');
$AmazonProductApiFunctions = new AmazonProductApiFunctions();
$ResponseArray = $AmazonProductApiFunctions->getMatchingProductFromASIN($ASIN,
$MarketplaceID);
$ProductCategory = $ResponseArray['0']['ProductGroup'];

echo "Asin: $ASIN <br>";

if (isset($ResponseArray['0']['SalesRank'])) {
    $SalesRank = $ResponseArray['0']['SalesRank']['0']['Rank'];
    echo '<div id="salesrank">#'.number_format($SalesRank).' in '.$ProductCategory.
    ' category</div>';
}
//Look for parentASIN from GetMatchingProductForId
} else {
    $ResponseArray = $AmazonProductApiFunctions->getMatchingProductForIdFromASIN(
    $ASIN, $MarketplaceID);
    $parentAsin = $ResponseArray['0']['Relationships']['IdentifiersStripped'];
    $ResponseArrayParent = $AmazonProductApiFunctions->getMatchingProductFromASIN(
    $parentAsin, $MarketplaceID);
    echo "Parent Asin: $parentAsin <br>";
    if (isset($ResponseArrayParent['0'])) {
        $SalesRank = $ResponseArrayParent['0']['SalesRank']['0']['Rank'];
    } else {
        //$SalesRank = "No Sales Rank Available"
    }
    echo '<div id="salesrank">#'.number_format($SalesRank).' in '.$ProductCategory.
    ' category <span style="font-size: small; font-style: italic;">(from Parent
    ASIN: <a target=_Blank href="http://www.amazon.com/gp/product/' . $parentAsin.
    '/?tag=kimagic-20">' . $parentAsin. '</a></span></div>';
}
} else {
    //from database
    $dbASIN = $SalesRank['ParentAsin'];
    $dbProductCategory = $SalesRank['ProductGroup'];
    $dbSalesRank = $SalesRank['SalesRank'];
    if ($SalesRank['ParentAsin'] = $SalesRank['Asin']) {
        echo '<div id="salesrank">#'.number_format($dbSalesRank).' in '.
        $dbProductCategory.' category <span style="font-size: small; font-style:
        italic;">(from Parent ASIN: <a target=_Blank
        href="http://www.amazon.com/gp/product/' . $dbASIN. '/?tag=kimagic-20">' .
        $dbASIN. '</a></span></div>';
    } else {
        echo '<div id="salesrank">#'.number_format($dbSalesRank).' in '.
        $dbProductCategory.' category</div>';
    }
}
}

```

```

}

function CompetitivePricingForASIN() {

    // -----
    // disable layout for json
    // -----
    $this->autoRender = false;
    // -----
    // grab variable from query string
    // -----
    $ASIN = $this->request->params['pass']['0'];
    $MarketplaceID = $this->request->params['pass']['1'];
    App::import('Model', 'AmazonProductApiFunctions');
    $AmazonProductApiFunctions = new AmazonProductApiFunctions();
    $ResponseArray = $AmazonProductApiFunctions->GetCompetitivePricingForASIN($ASIN,
    $MarketplaceID);
    //pr ($ResponseArray);
    //die;
    $LowestLandedPrice = $ResponseArray['0']['CompetitivePricing']['CompetitivePrices']['
    'CompetitivePrice']['0']['Price']['LandedPrice']['Amount'];
    //original intention was to return the lowest landed price.  this variable actually
    returns the current buy box price
    echo '<span id="lowest">'.number_format((float)$LowestLandedPrice, 2, '.', '').'</span>';
    $HaveBuyBox = $ResponseArray['0']['CompetitivePricing']['CompetitivePrices']['
    'CompetitivePrice']['0']['belongsToRequester'];
    echo '<div id="have">'.$HaveBuyBox.'</div>';
    //pr ($LowestLandedPrice); //for testing purposes
    //die();
}

function AsinFees($Asin,$MarketplaceID) {
    // -----
    // disable layout for json
    // -----
    $this->autoRender = false;
    // -----
    // load model
    // -----
    App::import('Model', 'AmazonAsinFeeUtils');
    $AmazonAsinFeeUtils = new AmazonAsinFeeUtils();
    $FeeArray = $AmazonAsinFeeUtils->GetFeesPerAsin($Asin,$MarketplaceID);
    $FeesJson = json_encode($FeeArray);
    echo $FeesJson;
}
}

```