```php
<?php
/*
    * SystemUser Model class
    * PHP versions 5.3.5
    * @date 5-February-2015
    * @Purpose:This model handles calls to Amazon API
    * @filesource
    * @author  lkpatricia
    * @revision
    * @version 1.0
*/


class AmazonProductApiFunctions extends AppModel
{
    var $name = 'AmazonProductApiFunctions';

    var $validate = array();

        // -------------------------------------------------------
        // GetLowestOfferListingsForASIN
        // -------------------------------------------------------
        function GetLowestOfferListingsForASIN($ASIN, $MarketplaceID, $MerchantID=FALSE)
        {
            // -------------------------------------------------------
            // Amazon Mws Config
            // -------------------------------------------------------
            $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
            $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
            $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');
            // -------------------------------------------------------
            // add amazon vendor files to the path to fix includes
            // -------------------------------------------------------
            $path              = "../Vendor/";
            set_include_path(get_include_path() . PATH_SEPARATOR . $path);
            // -------------------------------------------------------
            // Some Models and Client Needed
            // -------------------------------------------------------
            App::import('Vendor', 'MarketplaceWebServiceProducts/Client');
            App::import('Vendor', 'MarketplaceWebServiceProducts/Exception');
            App::import('Vendor',
            'MarketplaceWebServiceProducts/Model/GetLowestOfferListingsForASINRequest');
            App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
            App::import('Model', 'SystemUtils');
            App::import('Model', 'AmazonDeveloperKeys');
            App::import('Model','AmazonProductServiceUrls');
            App::import('Model','AmazonSkuLists');
            $AmazonSkuLists = new AmazonSkuLists();
            App::import('Model', 'AmazonUsers');
            $AmazonUsers = new AmazonUsers();

            // -------------------------------------------
            // vars we needed for that user
```

```
        // ------------------------------------------
        $MarketplaceId = $MarketplaceID;
        if($MerchantID===FALSE)
        {
            //-------------------------------------------------------------
            // if we don't have a MerchantID we get from logged in user
            //-------------------------------------------------------------
            $MwsUser = $AmazonUsers->getUserFromMarketplaceIdForProductApi($MarketplaceId);
            if($MwsUser===FALSE)
            {
                $RetArr['0']['Error']['Message'] = 'This user does not have access to the
                Product Api.';
                $RetArr['0']['Error']['StatusCode'] = '400';
                return $RetArr;
            }
            else
            {
                        $MwsAuthToken = $MwsUser['MwsAuthToken'];
                        $MerchantId = $MwsUser['MerchantID'];
            }
        }
        else
        {
            //-------------------------------------------------------------
            // we have a MerchantID passed so we use it
            //-------------------------------------------------------------
            $MerchantId = $MerchantID;
            $MwsAuthToken = $AmazonUsers->getMwsAuthTokenFromMerchantIdForProductApi(
            $MerchantID);
            if($MwsAuthToken===FALSE)
            {
                $RetArr['0']['Error']['Message'] = 'MerchantID '.$MerchantID.' does not
                have access to the Product Api.';
                $RetArr['0']['Error']['StatusCode'] = '400';
                return $RetArr;
            }
        }
        $AmazonProductServiceUrls = new AmazonProductServiceUrls();
        $serviceUrl = $AmazonProductServiceUrls->getServiceUrlFromMarketplaceId(
        $MarketplaceId);
        // ------------------------------------------
        // developer keys
        // ------------------------------------------
        $AmazonDeveloperKeys = new AmazonDeveloperKeys();
        $Keys = $AmazonDeveloperKeys->GetKeysFromMarketplaceId($MarketplaceId);
        $this->MwsAccessKeyId = $Keys['AmazonDeveloperKeys']['AWSAccessKeyID'];
        $this->MwsSecretAccessKey = $Keys['AmazonDeveloperKeys']['SecretKey'];
        // ------------------------------------------
        // we load the client for a connection to Amazon
        // ------------------------------------------
        $client = new MarketplaceWebServiceProducts_Client(
                                $this->MwsAccessKeyId,
                                $this->MwsSecretAccessKey,
```

```
                                $this->MwsApplicationName,
                                $this->MwsApplicationVersion,
                                array (
                                        'ServiceURL' => $serviceUrl,
                                        'ProxyHost' => null,
                                        'ProxyPort' => -1,
                                        'ProxyUsername' => null,
                                        'ProxyPassword' => null,
                                        'MaxErrorRetry' => 3,
                                )
        );
        // -------------------------------------------
        // parameters for the call to Amazon
        // -------------------------------------------
        $request = new
        MarketplaceWebServiceProducts_Model_GetLowestOfferListingsForASINRequest();
        $request->setSellerId($MerchantId);
        $request->setMWSAuthToken($MwsAuthToken);
        $request->setMarketplaceId($MarketplaceId);
        $asinList = new MarketplaceWebServiceProducts_Model_ASINListType();
        $asinList->setASIN($ASIN);
        $request->setASINList($asinList);
        try
        {
            // --------------------------------
            // We call Product Api
            // --------------------------------
            $response = $client->GetLowestOfferListingsForASIN($request);
            // --------------------------------
            // We convert response to an array
            // --------------------------------
            App::import('Model', 'AmazonProductApiUtils');
            $AmazonProductApiUtils = new AmazonProductApiUtils();
            $ResponseArray = $AmazonProductApiUtils->LowestOfferListingsResponseToArray(
            $response);
            return $ResponseArray;
        }
        catch (MarketplaceWebServiceProducts_Exception $ex)
        {
            $RetArr['0']['Error']['Exception'] = $ex->getMessage();
            $RetArr['0']['Error']['Message'] = $ex->getMessage();
            $RetArr['0']['Error']['StatusCode'] = $ex->getStatusCode();
            $RetArr['0']['Error']['ErrorCode'] = $ex->getErrorCode();
            $RetArr['0']['Error']['ErrorType'] = $ex->getErrorType();
            $RetArr['0']['Error']['RequestID'] = $ex->getRequestId();
            if ($ex->getStatusCode()=='400')
            {
                // ---------------------------------------
                // user does not have access
                // ---------------------------------------
                App::import('Model', 'AmazonUsers');
                $AmazonUsers = new AmazonUsers();
                $AmazonUsers->DisableProductApiAccessFromMerchantID($MerchantID);
```

```
            }
            return $RetArr;
        }
    }


    // --------------------------------------------------------
    // CountCompetitiveOffersForASIN
    // --------------------------------------------------------
    function CountCompetitiveOffersForASIN($ASIN, $MarketplaceID, $ItemCondition,
    $MerchantID)
    {
    $GetCompetitivePricingForASIN = $this->GetCompetitivePricingForASIN($ASIN,
    $MarketplaceID, $MerchantID);
    if(isset($GetCompetitivePricingForASIN['0']['status'])) {
        if($GetCompetitivePricingForASIN['0']['status']=='Success') {
            foreach($GetCompetitivePricingForASIN['0']['CompetitivePricing'][
            'NumberOfOfferListings']['OfferListingCount'] as $OfferListingCount) {
                if($OfferListingCount['condition']=='Any') $Any=$OfferListingCount['value'];
                if($OfferListingCount['condition']=='New') $New=$OfferListingCount['value'];
            }
            if($ItemCondition=='New') {
                if($New>20) $New=20;
                return $New;
            } else {
                $Used = $Any-$New;
                if ($Used >20) $Used = 20;
                return $Used;
            }
        } else {
            // error returned from AS
            return FALSE;
        }
    } else {
        // error from AZ
        return FALSE;
    }

}

// --------------------------------------------------------
    // GetCompetitivePricingForASIN
    // --------------------------------------------------------
    function GetCompetitivePricingForASIN($ASIN, $MarketplaceID, $MerchantID=FALSE)
    {
    // ==========================================
    // we need to build a cache here
    // ==========================================
        // ------------------------------------------------------
        // Amazon Mws Config
        // ------------------------------------------------------
        $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
        $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
```

```
        $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');
        // --------------------------------------------------------
        // add amazon vendor files to the path to fix includes
        // --------------------------------------------------------
        $path              = "../Vendor/";
        set_include_path(get_include_path() . PATH_SEPARATOR . $path);
        // --------------------------------------------------------
        // Some Models and Client Needed
        // --------------------------------------------------------
        App::import('Vendor', 'MarketplaceWebServiceProducts/Client');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Exception');
        App::import('Vendor',
        'MarketplaceWebServiceProducts/Model/GetCompetitivePricingForASINRequest');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
        App::import('Model', 'SystemUtils');
        App::import('Model', 'AmazonDeveloperKeys');
        App::import('Model','AmazonProductServiceUrls');
        App::import('Model','AmazonSkuLists');
        $AmazonSkuLists = new AmazonSkuLists();
        App::import('Model', 'AmazonUsers');
        $AmazonUsers = new AmazonUsers();

        // ------------------------------------------
        // vars we needed for that user
        // ------------------------------------------
        $MarketplaceId = $MarketplaceID;
        if($MerchantID===FALSE)
        {
            //-------------------------------------------------------------
            // if we don't have a MerchantID we get from logged in user
            //-------------------------------------------------------------
            $MwsUser = $AmazonUsers->getUserFromMarketplaceIdForProductApi($MarketplaceId);
            if($MwsUser===FALSE)
            {
                $RetArr['0']['Error']['Message'] = 'This user does not have access to the
                Product Api.';
                $RetArr['0']['Error']['StatusCode'] = '400';
                return $RetArr;
            }
            else
            {
                        $MwsAuthToken = $MwsUser['MwsAuthToken'];
                        $MerchantId = $MwsUser['MerchantID'];
            }
        }
        else
        {
            //-------------------------------------------------------------
            // we have a MerchantID passed so we use it
            //-------------------------------------------------------------
            $MerchantId = $MerchantID;
            $MwsAuthToken = $AmazonUsers->getMwsAuthTokenFromMerchantIdForProductApi(
            $MerchantID);
```

```
            if($MwsAuthToken===FALSE)
            {
                $RetArr['0']['Error']['Message'] = 'MerchantID '.$MerchantID.' does not
                have access to the Product Api.';
                $RetArr['0']['Error']['StatusCode'] = '400';
                return $RetArr;
            }
        }
        $AmazonProductServiceUrls = new AmazonProductServiceUrls();
        $serviceUrl = $AmazonProductServiceUrls->getServiceUrlFromMarketplaceId(
        $MarketplaceId);
        // ------------------------------------------
        // developer keys
        // ------------------------------------------
        $AmazonDeveloperKeys = new AmazonDeveloperKeys();
        $Keys = $AmazonDeveloperKeys->GetKeysFromMarketplaceId($MarketplaceId);
        $this->MwsAccessKeyId = $Keys['AmazonDeveloperKeys']['AWSAccessKeyID'];
        $this->MwsSecretAccessKey = $Keys['AmazonDeveloperKeys']['SecretKey'];
        // ------------------------------------------
        // we load the client for a connection to Amazon
        // ------------------------------------------
        $client = new MarketplaceWebServiceProducts_Client(
                                $this->MwsAccessKeyId,
                                $this->MwsSecretAccessKey,
                                $this->MwsApplicationName,
                                $this->MwsApplicationVersion,
                                array (
                                        'ServiceURL' => $serviceUrl,
                                        'ProxyHost' => null,
                                        'ProxyPort' => -1,
                                        'ProxyUsername' => null,
                                        'ProxyPassword' => null,
                                        'MaxErrorRetry' => 3,
                                )
        );
        // ------------------------------------------
        // parameters for the call to Amazon
        // ------------------------------------------
        $request = new
        MarketplaceWebServiceProducts_Model_GetCompetitivePricingForASINRequest();
        $request->setSellerId($MerchantId);
        $request->setMWSAuthToken($MwsAuthToken);
        $request->setMarketplaceId($MarketplaceId);
        $asinList = new MarketplaceWebServiceProducts_Model_ASINListType();
        $asinList->setASIN($ASIN);
        $request->setASINList($asinList);
        try
        {
            // -------------------------------
            // We call Product Api
            // -------------------------------
            $response = $client->GetCompetitivePricingForASIN($request);
            // -------------------------------
```

```
            // We convert response to an array
            // --------------------------------
            App::import('Model', 'AmazonProductApiUtils');
            $AmazonProductApiUtils = new AmazonProductApiUtils();
            $ResponseArray = $AmazonProductApiUtils->CompetitivePricingResponseToArray(
            $response);
            return $ResponseArray;
        }
        catch (MarketplaceWebServiceProducts_Exception $ex)
        {
            $RetArr['0']['Error']['Exception'] = $ex->getMessage();
            $RetArr['0']['Error']['Message'] = $ex->getMessage();
            $RetArr['0']['Error']['StatusCode'] = $ex->getStatusCode();
            $RetArr['0']['Error']['ErrorCode'] = $ex->getErrorCode();
            $RetArr['0']['Error']['ErrorType'] = $ex->getErrorType();
            $RetArr['0']['Error']['RequestID'] = $ex->getRequestId();
            if ($ex->getStatusCode()=='400')
            {
                // ---------------------------------------
                // user does not have access
                // ---------------------------------------
                App::import('Model', 'AmazonUsers');
                $AmazonUsers = new AmazonUsers();
                $AmazonUsers->DisableProductApiAccessFromMerchantID($MerchantID);
            }
            return $RetArr;
        }
    }

    // ------------------------------------------------------
    // getMatchingProductFromASIN for logged in user
    // Date: 13-February-2015
    // Author: Lynn K. Patricia
    // ------------------------------------------------------
      function getMatchingProductFromASIN($ASIN, $MarketplaceID, $MerchantID=FALSE)
    {
        // ------------------------------------------------------
        // Amazon Mws Config
        // ------------------------------------------------------
        $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
        $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
        $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');
        // ------------------------------------------------------
        // add amazon vendor files to the path to fix includes
        // ------------------------------------------------------
        $path              = "../Vendor/";
        set_include_path(get_include_path() . PATH_SEPARATOR . $path);
        // ------------------------------------------------------
        // Some Models and Client Needed
        // ------------------------------------------------------
        App::import('Vendor', 'MarketplaceWebServiceProducts/Client');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Exception');
        App::import('Vendor',
```

```
        'MarketplaceWebServiceProducts/Model/GetMatchingProductRequest');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
        App::import('Model', 'SystemUtils');
        App::import('Model', 'AmazonDeveloperKeys');
        App::import('Model','AmazonProductServiceUrls');
        App::import('Model','AmazonSkuLists');
        $AmazonSkuLists = new AmazonSkuLists();
        App::import('Model', 'AmazonUsers');
        $AmazonUsers = new AmazonUsers();

        // ------------------------------------------
        // vars we needed for that user
        // ------------------------------------------
        //$ASIN = $AmazonSkuLists[$i][$asin1];
        //$MarketplaceId = $AmazonSkuLists[$MarketplaceID];
        $MarketplaceId = $MarketplaceID;
        if($MerchantID===FALSE) {
                        $MwsUser = $AmazonUsers->getUserFromMarketplaceId($MarketplaceId);
                } else {
                        $MwsUser = $AmazonUsers->getUserFromMarketplaceId($MarketplaceId,
                        $MerchantID);
                }
        $MwsAuthToken = $MwsUser['MwsAuthToken'];
        $MerchantId = $MwsUser['MerchantID'];
        $AmazonProductServiceUrls = new AmazonProductServiceUrls();
        $serviceUrl = $AmazonProductServiceUrls->getServiceUrlFromMarketplaceId(
        $MarketplaceId);
        // ------------------------------------------
        // developer keys
        // ------------------------------------------
        $AmazonDeveloperKeys = new AmazonDeveloperKeys();
        $Keys = $AmazonDeveloperKeys->GetKeysFromMarketplaceId($MarketplaceId);
        $this->MwsAccessKeyId = $Keys['AmazonDeveloperKeys']['AWSAccessKeyID'];
        $this->MwsSecretAccessKey = $Keys['AmazonDeveloperKeys']['SecretKey'];
        // ------------------------------------------
        // we load the client for a connection to Amazon
        // ------------------------------------------
        $client = new MarketplaceWebServiceProducts_Client(
                    $this->MwsAccessKeyId,
                    $this->MwsSecretAccessKey,
                    $this->MwsApplicationName,
                    $this->MwsApplicationVersion,
                    array (
                        'ServiceURL' => $serviceUrl,
                        'ProxyHost' => null,
                        'ProxyPort' => -1,
                        'ProxyUsername' => null,
                        'ProxyPassword' => null,
                        'MaxErrorRetry' => 3,
                    )
        );
        // ------------------------------------------
        // parameters for the call to Amazon
```

```
        // ------------------------------------------
        $request = new MarketplaceWebServiceProducts_Model_GetMatchingProductRequest();
        $request->setSellerId($MerchantId);
        $request->setMWSAuthToken($MwsAuthToken);
        $request->setMarketplaceId($MarketplaceId);
        $asinList = new MarketplaceWebServiceProducts_Model_ASINListType();
        $asinList->setASIN($ASIN);
        $request->setASINList($asinList);

        try {
            // --------------------------------
            // We call Product Api
            // --------------------------------
            $response = $client->GetMatchingProduct($request);
            // --------------------------------
            // We convert response to an array
            // --------------------------------
            $SystemUtils = new SystemUtils();
            $ResponseArray = $SystemUtils->ProductResponseToArray($response);
            $this->updateOrInsertTablesFromGetMatchingProductFromAsin($ResponseArray);
            return $ResponseArray;
        } catch (MarketplaceWebServiceProducts_Exception $ex){
            echo("Caught Exception: " . $ex->getMessage() . "\n<BR>");
            echo("Response Status Code: " . $ex->getStatusCode() . "\n<BR>");
            echo("Error Code: " . $ex->getErrorCode() . "\n<BR>");
            echo("Error Type: " . $ex->getErrorType() . "\n<BR>");
            echo("Request ID: " . $ex->getRequestId() . "\n<BR>");
            echo("XML: " . $ex->getXML() . "\n<BR>");
            echo("ResponseHeaderMetadata: " . $ex->getResponseHeaderMetadata() . "\n<BR>");
            return $ex;
    }

}

//------------------------------------
//  FUNCTION to update/insert product data from Amazon API call
//------------------------------------
function updateOrInsertTablesFromGetMatchingProductFromAsin($data) {
    foreach($data as $product){
        //-----------------------
        // update amazon_asin_details table
        //-----------------------
        App::import('Model', 'AmazonAsinDetails');
        $AmazonAsinDetails = new AmazonAsinDetails();
        $AmazonAsinDetails->UpdateOrInsertDataFromProductArray($product);
        //-----------------------
        // update or insert amazon_asin_sales_ranks table
        //-----------------------
        App::import('Model', 'AmazonAsinSalesRanks');
        $AmazonAsinSalesRanks = new AmazonAsinSalesRanks();
        $AmazonAsinSalesRanks->UpdateOrInsertDataFromProductArray($product);
    }
}
```

```
    //-------------------------------------
    //  FUNCTION to update/insert parentASIN data from Amazon API call
    //-------------------------------------
    function updateOrInsertTablesFromGetMatchingProductForIdFromAsin($data) {
        foreach($data as $product){
            //-----------------------
            // update amazon_asin_parents table
            //-----------------------
            App::import('Model', 'AmazonAsinParents');
            $AmazonAsinParents = new AmazonAsinParents();
            $AmazonAsinParents->UpdateOrInsertDataFromProductArray($product);
        }
    }


// ---------------------------------------------------------
// get SalesRank of parentASIN: getMatchingProductForId for logged in user
// Date: 13-February-2015
// Author: Lynn K. Patricia
// ---------------------------------------------------------

        function getMatchingProductForIdFromASIN($ASIN, $MarketplaceID, $MerchantID=FALSE)
    {
        // ---------------------------------------------------------
        // Amazon Mws Config
        // ---------------------------------------------------------
        $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
        $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
        $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');
        // ---------------------------------------------------------
        // add amazon vendor files to the path to fix includes
        // ---------------------------------------------------------
        $path              = "../Vendor/";
        set_include_path(get_include_path() . PATH_SEPARATOR . $path);
        // ---------------------------------------------------------
        // Some Models and Client Needed
        // ---------------------------------------------------------
        App::import('Vendor', 'MarketplaceWebServiceProducts/Client');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Exception');
        App::import('Vendor',
        'MarketplaceWebServiceProducts/Model/GetMatchingProductForIdRequest');
        // we do not need ASINListType but now we do need Id List
        //App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Model/IdListType');
        App::import('Vendor',
        'MarketplaceWebServiceProducts/Model/GetMatchingProductForIdResponse');
        App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
        App::import('Model', 'SystemUtils');
        App::import('Model', 'AmazonDeveloperKeys');
        App::import('Model','AmazonProductServiceUrls');
        App::import('Model','AmazonSkuLists');
        $AmazonSkuLists = new AmazonSkuLists();
```

```
        App::import('Model', 'AmazonUsers');
        $AmazonUsers = new AmazonUsers();

        // ------------------------------------------
        // vars we needed for that user
        // ------------------------------------------
        //$ASIN = $AmazonSkuLists[$i][$asin1];
        //$MarketplaceId = $AmazonSkuLists[$MarketplaceID];
        $MarketplaceId = $MarketplaceID;
        if($MerchantID===FALSE) {
            $MwsUser = $AmazonUsers->getUserFromMarketplaceId($MarketplaceId);
        } else {
            $MwsUser = $AmazonUsers->getUserFromMarketplaceId($MarketplaceId,$MerchantID);
        }
        $MwsAuthToken = $MwsUser['MwsAuthToken'];
        $MerchantId = $MwsUser['MerchantID'];
        $AmazonProductServiceUrls = new AmazonProductServiceUrls();
        $serviceUrl = $AmazonProductServiceUrls->getServiceUrlFromMarketplaceId($MarketplaceId);
        // ------------------------------------------
        // developer keys
        // ------------------------------------------
        $AmazonDeveloperKeys = new AmazonDeveloperKeys();
        $Keys = $AmazonDeveloperKeys->GetKeysFromMarketplaceId($MarketplaceId);
        $this->MwsAccessKeyId = $Keys['AmazonDeveloperKeys']['AWSAccessKeyID'];
        $this->MwsSecretAccessKey = $Keys['AmazonDeveloperKeys']['SecretKey'];
        // ------------------------------------------
        // we load the client for a connection to Amazon
        // ------------------------------------------
        $client = new MarketplaceWebServiceProducts_Client(
                    $this->MwsAccessKeyId,
                    $this->MwsSecretAccessKey,
                    $this->MwsApplicationName,
                    $this->MwsApplicationVersion,
                    array (
                        'ServiceURL' => $serviceUrl,
                        'ProxyHost' => null,
                        'ProxyPort' => -1,
                        'ProxyUsername' => null,
                        'ProxyPassword' => null,
                        'MaxErrorRetry' => 3,
                    )
        );
        // ------------------------------------------
        // parameters for the call to Amazon
        // ------------------------------------------
        $request = new MarketplaceWebServiceProducts_Model_GetMatchingProductForIdRequest();
        $request->setSellerId($MerchantId);
        $request->setMWSAuthToken($MwsAuthToken);
        $request->setMarketplaceId($MarketplaceId);
        $asinList = new MarketplaceWebServiceProducts_Model_ASINListType();
        $asinList->setASIN($ASIN);
        //$request->setASINList($asinList);
        $request->setIdType('ASIN');
```

```
        $idlist = new MarketplaceWebServiceProducts_Model_IdListType();
        $idlist->setId($ASIN);
        $request->SetIdList($idlist);

        try
        {
            // ---------------------------------
            // We call Product Api
            // ---------------------------------
            $response = $client->GetMatchingProductForId($request);
            // ---------------------------------
            // We convert response to an array
            // ---------------------------------
            $SystemUtils = new SystemUtils();
            // you need to make anoter function for here
            //$ResponseArray = $SystemUtils->ProductResponseToArray($response);
            $ResponseArray = $SystemUtils->GetMatchingProductForIdResponseToArray($response);
            $this->updateOrInsertTablesFromGetMatchingProductForIdFromAsin($ResponseArray);
            return $ResponseArray;
        }
        catch (MarketplaceWebServiceProducts_Exception $ex)
        {
            echo("Caught Exception: " . $ex->getMessage() . "\n<BR>");
            echo("Response Status Code: " . $ex->getStatusCode() . "\n<BR>");
            echo("Error Code: " . $ex->getErrorCode() . "\n<BR>");
            echo("Error Type: " . $ex->getErrorType() . "\n<BR>");
            echo("Request ID: " . $ex->getRequestId() . "\n<BR>");
            echo("XML: " . $ex->getXML() . "\n<BR>");
            echo("ResponseHeaderMetadata: " . $ex->getResponseHeaderMetadata() . "\n<BR>");
            return $ex;
        }
    }



    // ------------------------------------------------------------------
    // Validate new account registration works
    // ------------------------------------------------------------------
    function ValidateAccountRegistration ($data) {
            try {
                                    $message = $this->ValidateAmazonAccountRegistration(
                                    'Blah', $data['MarketplaceID'], $data['MerchantID'],
                                    $data['MwsAuthToken']);

                                    if($message == 'success'){
                                            return 'success';
                                    }
                                    else{
                                            return "<div class='warning-message flash
                                            notice'>$message</div>";
                }
            }
            catch (Exception $e) {
                $returnArray = array( // creating an array to log msg
```

```
                    'Ack' => 'Error',
                    'ErrorCode' => '30',
                    'Message' => $e->getMessage(),
                    'TraceOfException' => $e->queryString,
                    'RowId' =>'Table:- amazon_condition'
                );
                // we log the error
                App::import('Model', 'SystemErrorLog');
                $SystemErrorLog = new SystemErrorLog();
                $SystemErrorLog->writeLog($returnArray);
                return $returnArray;
            }
    }
                //*********************************************************
            //function for amazon api connection attempt
            //*********************************************************
            function ValidateAmazonAccountRegistration($ASIN, $MarketplaceID, $MerchantID,
            $MwsAuthToken) {
            // ---------------------------------------------------------
    // Amazon Mws Config
    // ---------------------------------------------------------
    $this->MwsDateFormat = Configure::read('MWS_DATE_FORMAT');
    $this->MwsApplicationName = Configure::read('MWS_APPLICATION_NAME');
    $this->MwsApplicationVersion = Configure::read('MWS_APPLICATION_VERSION');

            // ---------------------------------------------------------
    // add amazon vendor files to the path to fix includes
    // ---------------------------------------------------------
    $path              = "../Vendor/";
    set_include_path(get_include_path() . PATH_SEPARATOR . $path);
            // ---------------------------------------------------------
    // Some Models and Client Needed
    // ---------------------------------------------------------
            App::import('Vendor', 'MarketplaceWebServiceProducts/Client');
            App::import('Vendor', 'MarketplaceWebServiceProducts/Exception');
            App::import('Vendor',
            'MarketplaceWebServiceProducts/Model/GetMatchingProductRequest');
            App::import('Vendor', 'MarketplaceWebServiceProducts/Model/ASINListType');
            App::import('Model', 'SystemUtils');
            App::import('Model', 'AmazonDeveloperKeys');
            App::import('Model','AmazonProductServiceUrls');
            App::import('Model', 'AmazonUsers');
            $AmazonUsers = new AmazonUsers();

            // ---------------------------------------------
            // vars we needed for that user
            // ---------------------------------------------
            //$ASIN = $AmazonSkuLists[$i][$asin1];
            //$MarketplaceId = $AmazonSkuLists[$MarketplaceID];
            $MarketplaceId = $MarketplaceID;
            //$MwsAuthToken = $MwsAuthToken;
            $MerchantId = $MerchantID;
            $AmazonProductServiceUrls = new AmazonProductServiceUrls();
```

```
        $serviceUrl = $AmazonProductServiceUrls->getServiceUrlFromMarketplaceId(
        $MarketplaceId);

        // ------------------------------------------
        // developer keys
        // ------------------------------------------
        $AmazonDeveloperKeys = new AmazonDeveloperKeys();
        $Keys = $AmazonDeveloperKeys->GetKeysFromMarketplaceId($MarketplaceId);
        $this->MwsAccessKeyId = $Keys['AmazonDeveloperKeys']['AWSAccessKeyID'];
        $this->MwsSecretAccessKey = $Keys['AmazonDeveloperKeys']['SecretKey'];
        // ------------------------------------------
        // we load the client for a connection to Amazon
        // ------------------------------------------
        $client = new MarketplaceWebServiceProducts_Client(
                                $this->MwsAccessKeyId,
                                $this->MwsSecretAccessKey,
                                $this->MwsApplicationName,
                                $this->MwsApplicationVersion,
                                array (
                                        'ServiceURL' => $serviceUrl,
                                        'ProxyHost' => null,
                                        'ProxyPort' => -1,
                                        'ProxyUsername' => null,
                                        'ProxyPassword' => null,
                                        'MaxErrorRetry' => 3,
                                )
        );
        // ------------------------------------------
        // parameters for the call to Amazon
        // ------------------------------------------
        $request = new MarketplaceWebServiceProducts_Model_GetMatchingProductRequest();
        $request->setSellerId($MerchantId);
        $request->setMWSAuthToken($MwsAuthToken);
        $request->setMarketplaceId($MarketplaceId);
        $asinList = new MarketplaceWebServiceProducts_Model_ASINListType();
        $asinList->setASIN($ASIN);
        $request->setASINList($asinList);

        try {
                // ---------------------------------------
                // We call Product Api and return success
                // ---------------------------------------
                $response = $client->GetMatchingProduct($request);

                $success = 'success';
                return $success;  //amzn.mws.7aa5dc56-a12f-5935-46ec-e0b209ec69ba
                AGGS1A2RJWBZ

        } catch (MarketplaceWebServiceProducts_Exception $ex){
                // ---------------------------------------------------------------------
                // We call Product Api and return error message if it does not connect
                // ---------------------------------------------------------------------
                $errorMessage = $ex->getMessage();
```

```
                        return $errorMessage;
                }

    }
}
?>
```