

MN Open API Part Three: Sample Code

October 2014 [2014-09-30]

Table of Contents

Audiences for this Document.....	2
Developer's Roadmap	2
Integration Environment (IE) vs. Production Environment (PROD).....	2
Sample Code	3
1. DOT NET: Generating a Signature with Your Shared Secret.....	3
2. Java: Generate a Signature with Your Shared Secret.....	5
3. PHP: Search for an Artist's Albums and Tracks	7
4. PHP: Add a First Item to a New Cart	12
5. PHP: Add an Item to a New Cart and Perform a Purchase	17
6. PHP: Create a User	23
7. PHP: Purchase Using a Credit Card for a Specific User	25
8. JavaScript (Internet Explorer): Search for an Artist's Albums and Tracks.....	28
9. PHP/jQuery: Search for an Artist's Albums and Tracks.....	31
10. PERL: Search for an Artist's Albums and Tracks	35
11. PHP: Add an Item to a New Cart and Perform a Purchase	40
12. Java: Searching for Streaming Tracks.....	45
13. DOT NET: Search for an Artist's Tracks	49
14. DOT NET: Add an Item to a Cart (XML Format).....	50
15. DOT NET: Add an Item to a Cart (JSON Format).....	52

Audiences for this Document

The MN Open API Implementation Guide is composed of three parts:

1. The MN Open API Implementation Guide Part One: Features and Business Rules, which provides an overview of features and the options for engagement.
2. The MN Open API Implementation Guide Part Two: Technical Details of the MN Open API, which includes API references and use cases.
3. The MN Open API Implementation Guide Part Three: Sample Code, which includes API sample code samples (this document) to help jumpstart your development.

The MN Open API Implementation Guide is written for two Customer audiences: business decision makers and technical implementers. Part One is applicable to both groups, while Part Two contains the technical details required for a successful implementation, and is intended for a technical audience. As such, the second section assumes a certain level of proficiency with web technologies to include HTTP, HTML, and JavaScript, as well as with at least one programming language; e.g., PHP, Java, ASP, JQuery, etc. Part Three contains sample code.

Developer's Roadmap

There are two basic tasks performed in most of the code samples for a successful API call: An HTTP request and parsing of the response. The simpler search APIs use HTTP GET. The more complex API calls use HTTP POST.

If you are completely new to the API, you may want to:

1. Review the *Use Cases* section of this Guide (Page 69, Part II) and decide on one or more use cases that match your model.
2. Start with an HTTP GET, perhaps a search for tracks API call (URL and Query), and get it to perform in your browser, so you can see the interaction with the search API and the response returned. Copy and paste this HTTP API call into your browser and substitute your ApiKey:
<http://ie-api.mndigital.com?method=Search.GetArtists&format=xml&name=&keyword=&page=1&pageSize=10&apiKey=Yourapikey>
3. Review the relevant code sample in this section, in your programming language of choice, and write a routing to programmatically execute the GET and parse the XML (or JSON) response from the HTTP GET.
4. Write a helper routine to generate signatures for the API calls, based on your shared secret. The very first code sample, and please also review the heading "*Generating Signatures with your Shared Secret*" in MN Open API Implementation Guide Part Two, "Technical Discussion of the MN Open APIs". Also see the Signature Generator Demo on our Customer Portal > API Menu.
5. Expand your HTTP GET routine to HTTP POST and provide the post data and signature for the more complex API calls, such as adding an item to the cart.
6. Review `Purchase.UseBalance` and `Purchase.UseCreditCard` APIs, extend your HTTP POST capabilities to include one or both of these depending on your use cases.
7. Next focus on `Download Locations` returned, and review `Report.GetOrderHistory`.

Integration Environment (IE) vs. Production Environment (PROD)

There are two environments:

- An Integration Environment (IE)
- A Production Environment (PROD).

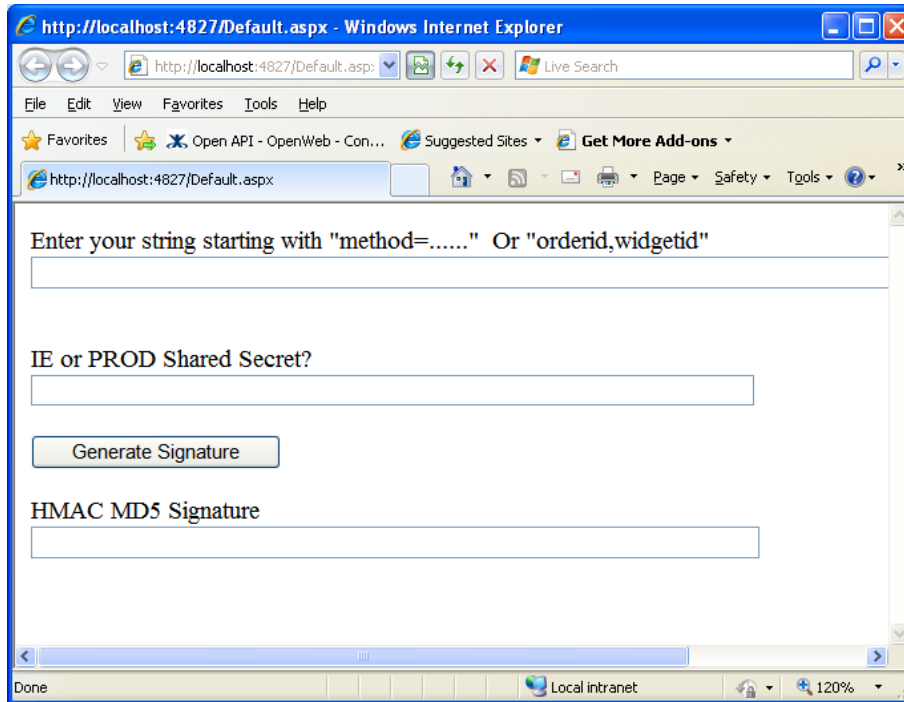
The Integration Environment (IE) is used for development and testing; i.e., so that you need not execute real transactions, such as a real credit card purchase of an album or a track in Production. Once your development and testing phases are completed, in order to switch from Integration to Production, in the examples below, simply modify `http://ie-api.mndigital.com` to `http://api.mndigital.com`. Be sure to also swap in your Production API keys.

Sample Code

1. DOT NET: Generating a Signature with Your Shared Secret

Use Case:

The following code uses standard ASP.Net library calls to encode a string into a signature using your shared secret.



Code Listing ASPX:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="GenerateSignature._Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

      <asp:Label ID="Label1" runat="server"
        Text="Enter your string starting with &quot;method=.....&quot;"></asp:Label>

      &nbsp;Or &quot;orderid,widgetid&quot;</div>
      <asp:TextBox ID="TextBox1" runat="server" ontextchanged="TextBox1_TextChanged"
        Width="682px"></asp:TextBox>
      <br />
      <br />
      <asp:Label ID="Label3" runat="server" Text="IE or PROD Shared Secret?"></asp:Label>
      <br />
    </div>
  </form>
</body>
</html>
```

```
<asp:TextBox ID="TextBox3" runat="server" Width="449px"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
    Text="Generate Signature" />
<br />
<br />
<asp:Label ID="Label2" runat="server" Text="HMAC MD5 Signature"></asp:Label>
<br />
<asp:TextBox ID="TextBox2" runat="server" Width="452px"></asp:TextBox>
</form>
</body>
</html>
```

Code Listing ASPX.CS:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Configuration;
using System.Web.Security;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Collections;
using System.Security.Cryptography;

namespace GenerateSignature
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string message;
            string key;

            key = this.TextBox3.Text;
            message = this.TextBox1.Text;

            System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();

            byte[] keyByte = encoding.GetBytes(key);

            HMACMD5 hmacmd5 = new HMACMD5(keyByte);

            byte[] messageBytes = encoding.GetBytes(message);

            byte[] hashmessage = hmacmd5.ComputeHash(messageBytes);

            this.TextBox2.Text = ByteToString(hashmessage);
        }

        protected string ByteToString(byte[] buff)
        {
            string sbinary = "";

            for (int i = 0; i < buff.Length; i++)
            {
                sbinary += buff[i].ToString("X2");
            }
        }
    }
}
```

```
        return (sbinary);  
    }  
}
```

2. Java: Generate a Signature with Your Shared Secret

Use Case:

The following code uses standard Java and library calls to encode a string into a signature using your shared secret.

Code Listing Java:

```
public static String getSecretKey(String urlWithApiKey, String secKey)  
    throws IllegalStateException, UnsupportedEncodingException, NoSuchAlgorithmException {  
    String ret = "";  
    MNSecretKey secret = new MNSecretKey(secKey);  
  
    Mac mac = Mac.getInstance("HmacMD5");  
  
    try {  
        mac.init(secret);  
    } catch (InvalidKeyException e) {  
        e.printStackTrace();  
    }  
    byte[] digest = mac.doFinal(urlWithApiKey.getBytes("utf8"));  
    ret = new String(ByteToString(digest));  
  
    return ret;  
}  
  
protected static String ByteToString(byte[] buff) {  
    StringBuffer hexString = new StringBuffer();  
    String hex;  
    for (int i=0; i<buff.length; i++) {  
        hex = Integer.toHexString(0xFF & buff[i]);  
        if (hex.length() == 0)  
            hex = "00";  
        else if (hex.length() == 1)  
            hex = "0" + hex;  
        hexString.append(hex);  
    }  
    return (hexString.toString());  
};
```

Code Listing MNSecretKey Class:

```
package com.mnd.medianetlib;  
import java.io.UnsupportedEncodingException;  
import javax.crypto.SecretKey;  
  
public class MNSecretKey implements SecretKey {  
    private static final long serialVersionUID = 0xafcd1b4d71bd1b4dL;
```

```
private String key;

public MNSecretKey(String key) {
    this.key = key;
}
public void setSecretKey(String key) {
    this.key = key;
}
public String getStringKey() {
    return key;
}
public String getAlgorithm() {
    return "HmacMD5";
}
public byte[] getEncoded() {
    byte[] ret=null;
    try {
        ret= key.getBytes("UTF-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    return ret;
}
public String getFormat() {
    return "RAW";
}
public boolean equals(Object o) {
    if(o == this)
        return true;
    if(!(o instanceof MNSecretKey)) {
        return false;
    } else {
        MNSecretKey k = (MNSecretKey)o;
        return k.getStringKey().equals(key);
    }
}
public int hashCode() {
    return key.hashCode();
}
}
```


Output

Searching for artist "sting" for all Albums and all tracks.

Album: The Very Best Of Sting And The Police (2002 Brits Version)

Track: Message In A Bottle
Track: Can't Stand Losing You
Track: An Englishman In New York
Track: Every Breath You Take
Track: Seven Days
Track: Walking On The Moon
Track: Fields Of Gold
Track: Fragile
Track: Every Little Thing She Does Is Magic
Track: De Do Do Do, De Da Da Da
Track: If You Love Somebody Set Them Free
Track: Brand New Day
Track: Desert Rose
Track: If I Ever Lose My Faith In You
Track: When We Dance (Edit)
Track: Don't Stand So Close To Me (Original Album Version)
Track: Roxanne
Track: So Lonely

Album: Fields Of Gold: The Best Of Sting 1984-1994

Track: When We Dance
Track: If You Love Somebody Set Them Free
Track: Fields Of Gold
Track: All This Time
Track: Fortress Around Your Heart
Track: Be Still My Beating Heart
Track: They Dance Alone
Track: If I Ever Lose My Faith In You
Track: Fragile
Track: Why Should I Cry For You
Track: An Englishman In New York
Track: We'll Be Together (Previously Unreleased Version)
Track: Russians
Track: This Cowboy Song

Album: Ten Summoner's Tales

Track: If I Ever Lose My Faith In You
Track: Love Is Stronger Than Justice (The Munificent Seven)
Track: Fields Of Gold
Track: Heavy Cloud No Rain
Track: She's Too Good For Me
Track: Seven Days
Track: Saint Augustine In Hell
Track: It's Probably Me
Track: Shape Of My Heart
Track: Something The Boy Said
Track: Epilogue (Nothing 'Bout Me)

Album: ...Nada Como El Sol - Selecciones Especiales En Espanol Y Portugues

Track: Mariposa Libre
Track: Frágil (Portuguese)
Track: Si Estamos Juntos
Track: Ellas Danzan Solas (Cueca Solo)
Track: Fragilidad

Album: Bring On The Night

Track: Bring On The Night/When The World Is Running Down You Make The Best Of What's (Still Around)
Track: Consider Me Gone
Track: Low Life
Track: We Work The Black Seam
Track: Driven To Tears

Track: The Dream Of The Blue Turtles/Demolition Man
Track: One World (Not Three)/Love Is The Seventh Wave
Track: Moon Over Bourbon Street
Track: I Burn For You
Track: Another Day
Track: Children's Crusade
Track: Down So Long
Track: Tea In The Sahara

Album: ...Nothing Like The Sun
Track: The Lazarus Heart
Track: Be Still My Beating Heart
Track: An Englishman In New York
Track: History Will Teach Us Nothing
Track: They Dance Alone
Track: Fragile
Track: We'll Be Together
Track: Straight To My Heart
Track: Rock Steady
Track: Sister Moon
Track: Little Wing
Track: The Secret Marriage

Album: ...All This Time
Track: Fragile
Track: A Thousand Years
Track: Perfect Love...Gone Wrong
Track: All This Time
Track: The Hounds Of Winter
Track: Don't Stand So Close To Me
Track: When We Dance
Track: Dienda
Track: Roxanne
Track: If You Love Somebody Set Them Free
Track: Brand New Day
Track: Fields Of Gold
Track: Moon Over Bourbon Street
Track: If I Ever Lose My Faith In You
Track: Every Breath You Take

Album: Absolute Sting: The Unauthorised Interview
Track: Absolute Sting: The Unauthorised Interview

Album: Brand New Day
Track: A Thousand Years
Track: Desert Rose
Track: Big Lie Small World
Track: After The Rain
Track: Perfect Love... Gone Wrong
Track: Tomorrow We'll See
Track: Prelude To The End Of The Game
Track: Fill Her Up
Track: Ghost Story
Track: Brand New Day

Album: Bring On The Night
Track: Bring On The Night/When The World Is Running Down You Make The Best Of What's (Still Around) (Live In Paris)
Track: Consider Me Gone (Live In Paris)
Track: Low Life (Live In Rome)
Track: We Work The Black Seam (Live In Paris)
Track: Driven To Tears (Live In Paris)
Track: The Dream Of The Blue Turtles/Demolition Man (Live In Paris)
Track: One World (Not Three)/Love Is The Seventh Wave (Live In Paris)
Track: Moon Over Bourbon Street (Live In Paris)
Track: I Burn For You (Live In Arnhem)
Track: Another Day (Live In Rome)
Track: Children's Crusade (Live In Arnhem)
Track: Down So Long (Live In Arnhem)
Track: Tea In The Sahara (Live In Arnhem)

Response:

The xml returned from curl_exec (\$curl)

```

- <AlbumSearchResults xmlns="http://api.mndigital.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  - <Debug>
    <Version i:nil="true" />
    <QueryString i:nil="true" />
    <Trace>Time: 5:27:51 PM 0.0000000 0.0000000 Timer Initialized 0.0000000 0.0000000 Volt.OpenApi.Packages.SearchPackage
  </Debug>
  GetAlbums</Trace>
  <Success>true</Success>
  <TotalResults>42</TotalResults>
  - <Albums>
    - <Album>
      <MnetId>673181</MnetId>
      <Title>The Very Best Of Sting And The Police (2002 Brits Version)</Title>
      - <Artist>
        <MnetId>31068</MnetId>
        <Name>Sting</Name>
        - <Images>
          <Artist180x80>http://images.musicnet.com/artists/000/031/068/b.jpeg</Artist180x80>
          <Artist190x230>http://images.musicnet.com/artists/000/031/068/c.jpeg</Artist190x230>
          <Artist375x250>http://images.musicnet.com/artists/000/031/068/a.jpeg</Artist375x250>
          <Artist200x>http://images.musicnet.com/artists/000/031/068/f.jpeg</Artist200x>
        </Images>
      </Artist>
      <Genre>Rock</Genre>
      <ExplicitLyrics>>false</ExplicitLyrics>
      <Label>A&M</Label>
      <Duration>78:14</Duration>
      <ReleaseDate>01-01-2002</ReleaseDate>
      <NumberOfTracks>18</NumberOfTracks>
      - <Images>
        <Album75x75>http://images.musicnet.com/albums/000/673/181/s.jpeg</Album75x75>
        <Album150x150>http://images.musicnet.com/albums/000/673/181/m.jpeg</Album150x150>
        <Album800x800>http://images.musicnet.com/albums/000/673/181/g.jpeg</Album800x800>
      </Images>
      - <FeaturedArtists>
        - <Artist>
          <MnetId>33357</MnetId>
          <Name>The Police</Name>
          - <Images>
            <Artist180x80>http://images.musicnet.com/artists/000/033/357/b.jpeg</Artist180x80>
          </Images>
        </Artist>
      </FeaturedArtists>
      - <Tracks>
        - <Track>
          <MnetId>673183</MnetId>
          <Title>Message In A Bottle</Title>
          - <Artist>
            <MnetId>33357</MnetId>
            <Name>The Police</Name>
            - <Images>
              <Artist180x80>http://images.musicnet.com/artists/000/033/357/b.jpeg</Artist180x80>
              <Artist190x230>http://images.musicnet.com/artists/000/033/357/c.jpeg</Artist190x230>
              <Artist375x250>http://images.musicnet.com/artists/000/033/357/a.jpeg</Artist375x250>
              <Artist200x>http://images.musicnet.com/artists/000/033/357/f.jpeg</Artist200x>
            </Images>
          </Artist>
          <Genre>Rock</Genre>
          <ExplicitLyrics>>false</ExplicitLyrics>
          <Duration>4:49</Duration>
          <ReleaseDate>01-01-2002</ReleaseDate>
          <TrackNumber>1</TrackNumber>
          <DiscNumber>1</DiscNumber>
          - <Mp3Rights>
            <CanSampleStream>true</CanSampleStream>
            <CanPurchaseDownload>true</CanPurchaseDownload>
            <AlbumPurchaseOnly>>false</AlbumPurchaseOnly>
          </Mp3Rights>
          <FeaturedArtists i:nil="true" />
          - <SampleLocations>
            - <MediaLocation>
              <Location>rtmp://DMUS.flash-ie.musicnet.com:80/Sample</Location>
              <Resource>mp3://spl/000/673/183/spl_024</Resource>
            </MediaLocation>
          </SampleLocations>
          <Bitrate>320kb</Bitrate>
          <PopularityRanking>16635</PopularityRanking>
        </Track>
        - <Track>
          <MnetId>673185</MnetId>
          <Title>Can't Stand Losing You</Title>
          - <Artist>
            <MnetId>33357</MnetId>
            <Name>The Police</Name>
            - <Images>
              <Artist180x80>http://images.musicnet.com/artists/000/033/357/b.jpeg</Artist180x80>
              <Artist190x230>http://images.musicnet.com/artists/000/033/357/c.jpeg</Artist190x230>
              <Artist375x250>http://images.musicnet.com/artists/000/033/357/a.jpeg</Artist375x250>
              <Artist200x>http://images.musicnet.com/artists/000/033/357/f.jpeg</Artist200x>
            </Images>
          </Artist>
        </Track>
      </Tracks>
    </Album>
  </Albums>
</AlbumSearchResults>

```

```

.....
                                <Genre>Rock</Genre>
.....
<Artist190x230>http://images.musicnet.com/artists/000/033/357/c.jpeg</Artist190x230>
<Artist375x250>http://images.musicnet.com/artists/000/033/357/a.jpeg</Artist375x250>
<Artist200x>http://images.musicnet.com/artists/000/033/357/f.jpeg</Artist200x>
                                </Images>
                                </Artist>
                                <Genre>Rock</Genre>
                                <ExplicitLyrics>>false</ExplicitLyrics>
                                <Duration>4:47</Duration>
                                <ReleaseDate>01-01-2002</ReleaseDate>
                                <TrackNumber>18</TrackNumber>
                                <DiscNumber>1</DiscNumber>
                                - <Mp3Rights>
                                  <CanSampleStream>>true</CanSampleStream>
                                  <CanPurchaseDownload>>true</CanPurchaseDownload>
                                  <AlbumPurchaseOnly>>false</AlbumPurchaseOnly>
                                </Mp3Rights>
                                <FeaturedArtists i:nil="true" />
                                - <SampleLocations>
                                  - <MediaLocation>
                                    <Location>rtmp://DMUS.flash-ie.musicnet.com:80/Sample</Location>
                                    <Resource>mp3://spl/000/673/217/spl_024</Resource>
                                  </MediaLocation>
                                </SampleLocations>
                                <Bitrate>320kb</Bitrate>
                                <PopularityRanking>108937</PopularityRanking>
                                </Track>
                                </Tracks>
                                <LabelOwnerId>5</LabelOwnerId>
                                - <PriceTag>
                                  <Currency>USD</Currency>
                                  <Amount>11.66</Amount>
                                  <IsSetPrice>true</IsSetPrice>
                                  <WholesalePrice>9.84</WholesalePrice>
                                </PriceTag>
                                <Bitrate>320kb</Bitrate>
                                </Album>
                                </Albums>
</AlbumSearchResults>

```

4. PHP: Add a First Item to a New Cart

Use Case:

The following code uses MediaNet's MN Open API (Cart, namely Cart.AddItem API call) to add a first item (a track) to a new cart. It also illustrates how to use your shared secret to build a signature parameter for the API call. The code is written in PHP. At the core of this code is a call to Cart.AddItem. The PHP code (contained in index.php) wraps the **Cart.AddItem API call** in and does a HTTP call to the MN Open web site to obtain an XML response of all the albums and tracks for an artist, in this case sting. A **curl_exec** function call is the center of the PHP code. For a full listing see PHP listing below.

API and Parameters

Please see Open API Implementation Guide Part Two: Technical Details of the MN Open API, under Cart.AddItem.

The actual item to add to the new cart is specified in the XML payload of the PHP Post call. The XML to add **<MnetId>12296213</MnetId>** item, typically obtained from a search call to find album tracks, to the cart, is shown here:

```

<AddItems xmlns="http://api.mndigital.com">
  <Items>
    <LineItem>
      <MnetId>12296213</MnetId>
      <ItemType>Track</ItemType>
      <Format>MP3</Format>
      <Price>0.99</Price>
    </LineItem>
  </Items>
</AddItems>

```

Some API calls require a signature parameter added to the query string. You can do this by simply adding a call to PHP hash_hmac('md5', **\$theQuery**, \$yoursharedsecret);

```
$theQuery = "method=Cart.AddItem";  
// this is your shared secret fir IE  
$secret = "yoursharedsecret";  
// This is your API Key  
$APIKEY="yourapikey";  
$theparams= "&format=xml&APIKEY=".$APIKEY;  
$theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,  
$secret);
```

In order to send an XML payload using the PHP call, you need to set the following options:

```
curl_setopt($curl, CURLOPT_POST, true);  
// set the Content-Type to application/xml  
curl_setopt($curl, CURLOPT_HTTPHEADER, array ("Content-Type:  
application/xml"));  
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
```

All we have to do now is post the HTTP query and parse the XML results.

We then take the **xml** returned into a for-loop. You can of course load this into your favorite XML DOM and XPath it, or iterate over the nodes as you like, in any way you like.

Ideally, check for <Success>true</Success> first.

Save the cart ID (<CartId>39209</CartId>) if you like to add a second item to the same cart.

Check to make sure there are no duplicate tracks in the cart using

<DuplicateItemInCart>>false</DuplicateItemInCart>.

Check to make sure the <Count>1</Count> matches your expectation.

For tracks that are sold only as part of an album, check also:

```
<CartAlbumContainsTrack>>false</CartAlbumContainsTrack>  
<CartTrackInAlbum>>false</CartTrackInAlbum>
```

A full listing of the Index.php HTML page, the output, and the XML returned from the MN Open API call is included below for completeness.

Screen of IndexPost.php

```
PHP - C:\inetpub\wwwroot\WIND-PHP1\IndexPost.php - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

index1.php indexPost.php
10 // depending on the API call you like to make
11 // and the parameters specific to the API call
12 // Please note signatures may be required for other API calls
13 $theQuery = "method=Cart.AddItem";
14 // this is your shared secret fir IE
15 $secret = "yoursharedsecret";
16 // This is your API Key
17 $APIKEY="yourapikey";
18 $theparams = "&format=xml&APIKEY=".$APIKEY;
19 $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams, $secret);
20 // Build the XML payload for post
21 // Ideally, build this XML payload in a more general fashion
22 $data = utf8_encode('
23 <AddItems xmlns="http://api.mmdigital.com">
24 <Items>
25 <LineItem>
26 <MnetId>12296213</MnetId>
27 <ItemType>Track</ItemType>
28 <Format>MP3</Format>
29 <Price>0.99</Price>
30 </LineItem>
31 </Items>
32 </AddItems>');
33 $url = "http://ie-api.mmdigital.com?".$theQuery."&amp;rrp=10";
34 // for debug
35 echo $url."<br /><br />";
36 // This is standard curl call of PHP
37 $curl = curl_init();
38 curl_setopt($curl, CURLOPT_URL, $url);
39 curl_setopt($curl, CURLOPT_POST, true);
40 // set the Content-Type to application/xml
41 curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
42 curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
43 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
44 // for debug
45 $response = curl_exec ($curl);
46 if (curl_errno($curl))
47 {
48     print "error";
49     print curl_error ($curl);
50 }
51
52 curl_close ($curl);
53 //echo "<br />".$response."<br />";
54 // Now that we have the XML response,
55 // let's load it and find the titles for albums and tracks
56 $domDoc = new DOMDocument();
57 $dom = $domDoc->loadXML($response);
58 // echo $response;
59
```

Code Listing PHP:

```
<html>
<head>
</head>
<body>
Adding a first item number to a new cart.
<br /><br />
<?php
    // Initialize the query parameters for Cart.AddItem
    // The query is fully spelled out to illustrate the API call,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=Cart.AddItem";
    // this is your shared secret fir IE
    $secret = "yoursharedsecret";
    // This is your API Key
    $APIKEY="yourapikey";
    $theparams= "&format=xml&APIKEY=".$APIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
    // Build the XML payload for post
    // The query is fully spelled out to illustrate the API call , build this XML payload in a
more general fashion
    $data = utf8_encode('
        <AddItems xmlns="http://api.mndigital.com">
            <Items>
                <LineItem>
                    <MnetId>12296213</MnetId>
                    <ItemType>Track</ItemType>
                    <Format>MP3</Format>
                    <Price>0.99</Price>
                </LineItem>
            </Items>
        </AddItems>');
    $url = "http://ie-api.mndigital.com?".$theQuery;
    // for debug
    echo $url."<br /><br />";
    // This is standard curl call of PHP
    $curl = curl_init();
    curl_setopt ($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    // set the Content-Type to application/xml
    curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    // for debug
    $response = curl_exec ($curl);
    if (curl_errno($curl))
    {
        print "error";
        print curl_error($curl);
    }

    curl_close ($curl);
    //echo "<br />".$response."<br />";
    // Now that we have the XML response,
    // let's load it and find the cart id
    $domDoc = new DOMDocument(); ;
    $dom = $domDoc->loadXML($response);
    // echo $response;
    // this should not happen
    if (!$dom) {
        echo "Error while parsing the response";
        exit;
    }
}
```

```
// We have a valid XML doc loaded
$CartItemCount = $domDoc->getElementsByTagName('CartItemCount'); // All Cart Items
for ($i = 0; $i < $CartItemCount->length; $i++) {
    $Nodes = $CartItemCount->item($i)->getElementsByTagName('CartId'); // Searching for
    CartId
    echo "CartId returned from Cart.AddItem API: ".$CartItemCount->item($i)-
    >getElementsByTagName('CartId')->item(0)->nodeValue. "<br />";
}
?>
</body>
</html>
```

Output

Adding a first item number to a new cart.

CartId returned from Cart.AddItem API: 39194 (or XXXXX)

Response

The xml returned from curl_exec (\$curl)

```
<CartItemCount xmlns="http://api.mndigital.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Success>true</Success>
  <CartId>39209</CartId>
  <Count>1</Count>
  <Status>
    <DuplicateItemInCart>false</DuplicateItemInCart>
    <CartAlbumContainsTrack>false</CartAlbumContainsTrack>
    <CartTrackInAlbum>false</CartTrackInAlbum>
  </Status>
</CartItemCount>
```


5. PHP: Add an Item to a New Cart and Perform a Purchase

Use Case:

The following code uses MediaNet's MN Open API (Cart, namely Cart.AddItem & Cart.Purchase API calls) to add a first item (a track) to a new cart and continue to purchase it. The code is written in PHP. At the core of this code is a call to Cart.AddItem. The PHP code (contained in index.php) wraps the **Cart.AddItem & Cart.Purchase API calls** using HTTP calls to the MN Open web APIs. A **curl_exec** function call is the center of the PHP code. For a full listing see PHP listing below.

API and Parameters

Please see Open API Implementation Guide Part Two: Technical Details of the MN Open API, under Cart.AddItem

The actual item to add to the new cart is specified in the XML payload of the PHP Post call. The XML to add `<MnetId>12296213</MnetId>` item, typically obtained from a search call to find album tracks, to the cart, is shown here:

```
<AddItems xmlns="http://api.mndigital.com">
  <Items>
    <LineItem>
      <MnetId>12296213</MnetId>
      <ItemType>Track</ItemType>
      <Format>MP3</Format>
      <Price>0.99</Price>
    </LineItem>
  </Items>
</AddItems>
```

Check for `<Success>>true</Success>` first.

Save the cart ID (`<CartId>39209</CartId>`) if you like to add a second item to the same cart.

Check to make sure there are no duplicate tracks in the cart using `<DuplicateItemInCart>>false</DuplicateItemInCart>`.

Check to make sure the `<Count>1</Count>` matches your expectation.

For tracks that are sold only as part of an album, check also:

```
<CartAlbumContainsTrack>>false</CartAlbumContainsTrack>
<CartTrackInAlbum>>false</CartTrackInAlbum>
```

A full listing of the Index.php HTML page, the output, and the XML returned from the MN Open API call, is included below for completeness.

The actual cart to complete purchase, is specified in `&cartId=".$cartIDRet` and the credit card information is included in the XML payload of the PHP Post call. The XML to add `<MnetId>12296213</MnetId>` item, typically obtained from a search call to find album tracks, to the cart, is shown here:

```
$data = utf8_encode('
  <Purchase xmlns="http://api.mndigital.com">
    <User>
      <FirstName>First</FirstName>
      <LastName>Last</LastName>
      <EmailAddress>test@mndigital.com</EmailAddress>
      <AllowHtmlEmail>>true</AllowHtmlEmail>
      <BillingAddress>
        <AddressLine1>2401 Elliott Ave.</AddressLine1>
        <AddressLine2>Suite 300</AddressLine2>
        <City>Seattle</City>
        <State>WA</State>
```

```

        <PostalCode>98121</PostalCode>
        <Country>US</Country>
    </BillingAddress>
    <CreditCard>
        <FirstName>First</FirstName>
        <LastName>Last</LastName>
        <Number>4111111111111111</Number>
        <CVV>000</CVV>
        <CardType>Visa</CardType>
        <ExpirationMonth>5</ExpirationMonth>
        <ExpirationYear>2010</ExpirationYear>
    </CreditCard>
</User>
</Purchase>');

```

A full listing of the Index.php HTML page, the output, and the XML returned from the MN Open API call, is included below for completeness.

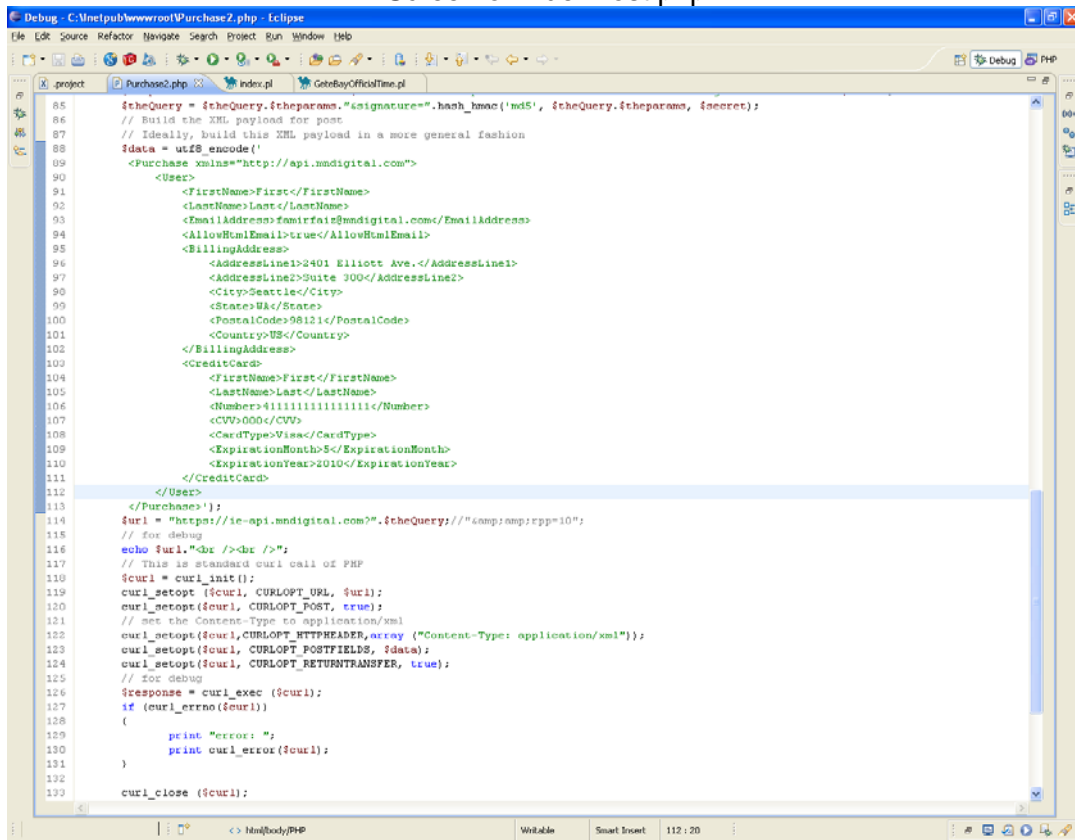
Please note the HTTPS in the URL for the purchase command.

```
$url = "https://ie-api.mndigital.com?".$theQuery;
```

Cart.Purchase requires that SSL is configured. Otherwise you can get an error:

```
error: SSL certificate problem, verify that the CA cert is OK. Details: error:14090086:SSL
routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

Screen of IndexPost.php



```

85 $theQuery = $theQuery.$theParams."&signature=".hash_hmac('md5', $theQuery.$theParams, $secret);
86 // Build the XML payload for post
87 // Ideally, build this XML payload in a more general fashion
88 $data = utf8_encode('
89 <Purchase xmlns="http://api.mndigital.com">
90 <User>
91 <FirstName>First</FirstName>
92 <LastName>Last</LastName>
93 <EmailAddress>famirfai@mndigital.com</EmailAddress>
94 <AllowHtmlEmail>true</AllowHtmlEmail>
95 <BillingAddress>
96 <AddressLine1>2401 Elliott Ave.</AddressLine1>
97 <AddressLine2>Suite 300</AddressLine2>
98 <City>Seattle</City>
99 <State>WA</State>
100 <PostalCode>98121</PostalCode>
101 <Country>US</Country>
102 </BillingAddress>
103 <CreditCard>
104 <FirstName>First</FirstName>
105 <LastName>Last</LastName>
106 <Number>4111111111111111</Number>
107 <CVV>000</CVV>
108 <CardType>Visa</CardType>
109 <ExpirationMonth>5</ExpirationMonth>
110 <ExpirationYear>2010</ExpirationYear>
111 </CreditCard>
112 </User>
113 </Purchase>');
114 $url = "https://ie-api.mndigital.com?".$theQuery."&amp;amp;http=10";
115 // for debug
116 echo $url."<br /><br />";
117 // This is standard curl call of PHP
118 $curl = curl_init();
119 curl_setopt($curl, CURLOPT_URL, $url);
120 curl_setopt($curl, CURLOPT_POST, true);
121 // set the Content-Type to application/xml
122 curl_setopt($curl, CURLOPT_HTTPHEADER, array ("Content-Type: application/xml"));
123 curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
124 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
125 // see debug
126 $response = curl_exec ($curl);
127 if (curl_errno($curl))
128 {
129     print "error: ";
130     print curl_error($curl);
131 }
132
133 curl_close ($curl);

```

Code Listing PHP:

PHP Post example of using Cart.AddItem

```
<html>
<head>
</head>
<body>
Adding a first item number to a new cart and purchasing the item.
<br /><br />
<?php
    // Initialize the query parameters for Cart.AddItem
    // The query is fully spelled out to illustrate the API call, build this query in a more
general fashion,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=Cart.AddItem";
    // this is your shared secret fir IE
    $secret = "sharedsecret";
    // This is your API Key
    $APIKEY="yourapiKey";
    $theparams= "&format=xml&APIKEY=".$APIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
    // Build the XML payload for post
    // Ideally, build this XML payload in a more general fashion
    $data = utf8_encode('
        <AddItems xmlns="http://api.mndigital.com">
            <Items>
                <LineItem>
                    <MnetId>12296213</MnetId>
                    <ItemType>Track</ItemType>
                    <Format>MP3</Format>
                    <Price>0.99</Price>
                </LineItem>
            </Items>
        </AddItems>');
    $url = "http://ie-api.mndigital.com?".$theQuery;
    // for debug
    echo $url."<br /><br />";
    // This is standard curl call of PHP
    $curl = curl_init();
    curl_setopt ($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    // set the Content-Type to application/xml
    curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    // for debug
    $response = curl_exec ($curl);
    if (curl_errno($curl))
    {
        print "error";
        print curl_error($curl);
    }

    curl_close ($curl);
    //echo "<br />".$response."<br />";
    // Now that we have the XML response,
    // let's load it and find the cart id
    $domDoc = new DOMDocument(); ;
    $dom = $domDoc->loadXML($response);
    // echo $response;
    // this should not happen
    if (!$dom) {
```

```

        echo "Error while parsing the response";
        exit;
    }
    // We have a valid XML doc loaded
    $CartItemCount = $domDoc->getElementsByTagName('CartItemCount'); // All Cart Items
    for ($i = 0; $i < $CartItemCount->length; $i++) {
        $Nodes = $CartItemCount->item($i)->getElementsByTagName('CartId'); // Searching for
CartId
        $cartIDRet = $CartItemCount->item($i)->getElementsByTagName('CartId')->item(0)-
>nodeValue;
        echo "CartId returned from Cart.AddItem API: ".$cartIDRet. "<br />";
    }

    // Initialize the query parameters for Cart.Purchase
    // The query is fully spelled out to illustrate the API call , build this query in a more
general fashion,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=Cart.Purchase";
    // this is your shared secret fir IE
    $secret = "YourSharedSecret";
    // This is your API Key
    $APIKEY="YourAPIKey";
    $theparams=
"&format=xml&cartId=".$cartIDRet."&userIp=0.0.0.0&userDomain=testdomain.mndigital.com&APIKEY=".$A
PIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
    // Build the XML payload for post
    // The query is fully spelled out to illustrate the API call , build this XML payload in a
more general fashion
    $data = utf8_encode('
        <Purchase xmlns="http://api.mndigital.com">
            <User>
                <FirstName>First</FirstName>
                <LastName>Last</LastName>
                <EmailAddress>test@mndigital.com</EmailAddress>
                <AllowHtmlEmail>true</AllowHtmlEmail>
                <BillingAddress>
                    <AddressLine1>2401 Elliott Ave.</AddressLine1>
                    <AddressLine2>Suite 300</AddressLine2>
                    <City>Seattle</City>
                    <State>WA</State>
                    <PostalCode>98121</PostalCode>
                    <Country>US</Country>
                </BillingAddress>
                <CreditCard>
                    <FirstName>First</FirstName>
                    <LastName>Last</LastName>
                    <Number>4111111111111111</Number>
                    <CVV>000</CVV>
                    <CardType>Visa</CardType>
                    <ExpirationMonth>5</ExpirationMonth>
                    <ExpirationYear>2010</ExpirationYear>
                </CreditCard>
            </User>
        </Purchase>');
    $url = "https://ie-api.mndigital.com?".$theQuery;
    // for debug
    echo $url."<br /><br />";
    // This is standard curl call of PHP
    $curl = curl_init();
    curl_setopt ($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    // set the Content-Type to application/xml
    curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));

```

```
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
// for debug
$response = curl_exec ($curl);
if (curl_errno($curl))
{
    print "error: ";
    print curl_error($curl);
}
?>
</body>
</html>
```

Output

Adding a first item number to a new cart.

http://ie-api.mndigital.com?method=Cart.AddItem&format=xml&APIKEY=yourApiKey&signature=45410f35bf9f1acdc9c3ab38d2ac5944

CartId returned from Cart.AddItem API: **39629**

https://ie-api.mndigital.com?method=Cart.Purchase&format=xml&cartId=**39629**&userIp=0.0.0.0&userDomain=yourdomain.com&APIKEY=yourApiKey &signature=60a5e9d8b698c69877fa6f38bce8d1e5

Email sent from Cart.Purchase.

From: support@mndigital.com [mailto:support@mndigital.com]
Sent: November 20, 2013 2:31 PM
To:
Subject: Your MP3 purchase from Open API Test Partner



Order Completed

Thank you for your order!

Here is your receipt for your MP3 purchase from Open API Test Partner. If you have any questions about your purchase, please contact customer support at [\[support@mndigital.com\]](mailto:support@mndigital.com). Include a description of your issue and your order number in your e-mail.

Order Summary

Songs/Albums	Content	Price
 Karaoke: Spin The Black Circle by Pearl Jam	MP3	\$0.99

Tax: \$0.09

Total: \$1.08



Order Number: #326352
Date: 11-20-2013
Time: 14:30

Card Type: Visa
Name: First Last
Card Number: *****1111

If you were not able to complete your downloads, they are [available here](#).

6. PHP: Create a User

Use Case:

The following code uses MediaNet's MN Open API (Cart, namely User.Create API calls). If you would like to show order history for each registered user of your site, use the User.Create and Purchase.UseCreditCard APIs.

Code Listing PHP:

PHP Post example of using User.Create

```
<html>
<head>
</head>
<body>
Create a User.
<br /><br />
<?php
    // Initialize the query parameters for User.Creat
    // The query is fully spelled out to illustrate the API call , build this query in a more
general fashion,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=User.Create";
    // this is your shared secret fir IE
    $secret = "Yoursharedsecret";
    // This is your API Key
    $APIKEY="YourAPIKey";
    $theparams= "&domain=Domain.com&format=xml&APIKEY=".$APIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
    // Build the XML payload for post
    // The query is fully spelled out to illustrate the API call , build this XML payload in a
more general fashion
    $data = utf8_encode('
        <Create xmlns="http://api.mndigital.com">
            <User>
                <FirstName>First</FirstName>
                <LastName>Last</LastName>
                <Username>TestUsername</Username>
                <UserId/>
                <Password>password</Password>
                <EmailAddress>username@mndigital.com</EmailAddress>
                <PartnerPID>000-00-0000</PartnerPID>
                <Role></Role>
                <Association></Association>
                <AllowNewsletter>>false</AllowNewsletter>
                <AllowHtmlEmail>>true</AllowHtmlEmail>
            </User>
        </Create>
    ');
    $url = "https://ie-api.mndigital.com?".$theQuery;
    // for debug
    echo $url."<br /><br />";
    // This is standard curl call of PHP
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    // set the Content-Type to application/xml
    curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
```

```
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
// for debug
$response = curl_exec ($curl);
if (curl_errno($curl))
{
    print "error";
    print curl_error($curl);
}

curl_close ($curl);
//echo "<br />".$response."<br />";
// Now that we have the XML response,
// let's load it and find the cart id
$domDoc = new DOMDocument();;
$dom = $domDoc->loadXML($response);
echo $response;
// this should not happen
if (!$dom) {
    echo "Error while parsing the response";
    exit;
}
// We have a valid XML doc loaded
echo $domDoc->getElementsByTagName('ID'); // User ID created

?>
</body>
</html>
```

Response:

```
<CreateUserResponse xmlns="http://api.mndigital.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Success>true</Success>
  <User>
    <ID>6776122c-40d6-40ca-86eb-c7a7d56b39b3</ID>
    <FirstName>First</FirstName>
    <LastName>Last</LastName>
    <Username>TestUsername</Username>
    <Password>password</Password>
    <EmailAddress>username@mndigital.com</EmailAddress>
    <PartnerPID>000-00-0000</PartnerPID>
    <Role/>
    <Association/>
    <AllowNewsletter>>false</AllowNewsletter>
    <AllowHtmlEmail>>true</AllowHtmlEmail>
  </User>
</CreateUserResponse>
```


7. PHP: Purchase Using a Credit Card for a Specific User

Use Case:

The following code uses MediaNet's MN Open API (Cart, namely User.Create API calls). If you would like to show order history for each registered user of your site, use the User.Create and Purchase.UseCreditCard APIs.

Code Listing PHP:

```
<html>
<head>
</head>
<body>
Create a User.
<br /><br />
<?php
    // Initialize the query parameters for Purchase.UseCreditCard
    // The query is fully spelled out to illustrate the API call , build this query in a more
    general fashion,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=Purchase.UseCreditCard";
    // this is your shared secret fir IE
    $secret = "Yoursharedsecret";
    // This is your API Key
    $APIKEY="YourAPIKey";
    // TestUsername is the name of the user created with User.Create
    $theparams= "&userIp=0.0.0.0&username=TestUsername&format=xml&APIKEY=" . $APIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
    // Build the XML payload for post
    // The query is fully spelled out to illustrate the API call , build this XML payload in a
    more general fashion
    $data = utf8_encode('
    <UseCreditCard xmlns="http://api.mndigital.com">
        <User>
            <FirstName>First</FirstName>
            <LastName>Last</LastName>
            <EmailAddress>test@mndigital.com</EmailAddress>
            <BillingAddress>
                <AddressLine1>2401 Elliott Ave.</AddressLine1>
                <AddressLine2>Suite 300</AddressLine2>
                <City>Seattle</City>
                <State>WA</State>
                <PostalCode>98121</PostalCode>
                <Country>US</Country>
            </BillingAddress>
            <CreditCard>
                <FirstName>First</FirstName>
                <LastName>Last</LastName>
                <Number>4111111111111111</Number>
                <CVV>000</CVV>
                <CardType>Visa</CardType>
                <ExpirationMonth>5</ExpirationMonth>
                <ExpirationYear>2010</ExpirationYear>
            </CreditCard>
        </User>
    </Items>
        <LineItem>
            <MnetId>12296213</MnetId>
            <ItemType>Track</ItemType>
            <Format>MP3</Format>
            <Price>0.99</Price>
```

```

                <Tax>0.09</Tax>
            </LineItem>
        </Items>
        <TotalCharge>1.08</TotalCharge>
    </UseCreditCard>
');
$url = "https://ie-api.mndigital.com?".$theQuery;
// for debug
echo $url."<br /><br />";
// This is standard curl call of PHP
$curl = curl_init();
curl_setopt ($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_POST, true);
// set the Content-Type to application/xml
curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
// for debug
$response = curl_exec ($curl);
if (curl_errno($curl))
{
    print "error";
    print curl_error($curl);
}

curl_close ($curl);
//echo "<br />".$response."<br />";
// Now that we have the XML response,
// let's load it and find the cart id
$domDoc = new DOMDocument();;
$dom = $domDoc->loadXML($response);
echo $response;
// this should not happen
if (!$dom) {
    echo "Error while parsing the response";
    exit;
}
// We have a valid XML doc loaded
echo $domDoc->getElementsByTagName('ID'); // Order ID created

?>
</body>
</html>

```

Response:

```

<?xml version="1.0" encoding="utf-8"?>
<PurchaseUseCreditCardResults xmlns="http://api.mndigital.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <Success>true</Success>
    <Order>
        <ID>326620</ID>
        <Date>12-09-2009</Date>
        <Time>12:28</Time>
        <SubTotal>0.99</SubTotal>
        <Tax>0.09</Tax>
        <Total>1.08</Total>
        <Items>
            <LineItem>
                <MnetId>12296213</MnetId>
                <ItemType>Track</ItemType>
                <Price>0.99</Price>
                <Format>MP3</Format>
                <Description>
                    <TrackMnetId>12296213</TrackMnetId>
                    <TrackTitle>Karaoke: Spin The Black Circle</TrackTitle>
                </Description>
            </LineItem>
        </Items>
    </Order>
</PurchaseUseCreditCardResults>

```

```

<ArtistMnetId>43089</ArtistMnetId>
<ArtistName>Pearl Jam</ArtistName>
<AlbumMnetId>12296211</AlbumMnetId>
<AlbumTitle>Karaoke Tribute: More Tribute To Pearl Jam</AlbumTitle>
<AlbumLabel>APM Music</AlbumLabel>
<Images>
  Images section .....
</Images>
<AlbumLabelOwnerId>122</AlbumLabelOwnerId>
<TrackDiscNumber>1</TrackDiscNumber>
<TrackNumber>1</TrackNumber>
</Description>
<DownloadLocations>
  <MediaLocation>
    <MnetId>12296213</MnetId>
    <Location>http://DMUS.download-
ie.musicnet.com/getFile?filename=Pearl+Jam+-+Karaoke+Tribute%3A+More+Tribute+To+Pearl+J+-+1+-+1+-+
+Karaoke%3A+Spin+The+Black+Circle.mp3&amp;file=NA&amp;token=99e83113f7e7ddcdf12217731e78497e28f06
dfbaa86983d35bd80d0e65048145a44a963ac131363946168e28ccb9fa6370eafec1c3c961d8e626bc0a7c68bdee60d66
fbda8a39b7fb6610f89de61c1f7f736e84a40a928e873d897b18aa4ff320773a4f2753368acacafc76d2742eb8262154d
2126d632649d11775bef0f608</Location>
    <FlashDownload>>false</FlashDownload>
  </MediaLocation>
</DownloadLocations>
<AssetCode>MP3_DOWNLOAD_UENC_320kb_070</AssetCode>
</LineItem>
</Items>
<User>
  <ID>6776122c-40d6-40ca-86eb-c7a7d56b39b3</ID>
  <FirstName>First</FirstName>
  <LastName>Last</LastName>
  <EmailAddress>test@mndigital.com</EmailAddress>
  <AllowHtmlEmail>>true</AllowHtmlEmail>
  <BillingAddress>
    <AddressLine1>2401 Elliott Ave.</AddressLine1>
    <AddressLine2>Suite 300</AddressLine2>
    <City>Seattle</City>
    <State>Washington</State>
    <PostalCode>98121</PostalCode>
    <Country>US</Country>
  </BillingAddress>
  <CreditCard>
    <FirstName>First</FirstName>
    <LastName>Last</LastName>
    <Number>1111</Number>
    <CardType>Visa</CardType>
    <ExpirationMonth>5</ExpirationMonth>
    <ExpirationYear>2010</ExpirationYear>
    <CardID>0</CardID>
  </CreditCard>
</User>
<OrderStatus>Successful</OrderStatus>
<PaymentMethod>CreditCard</PaymentMethod>
</Order>
</PurchaseUseCreditCardResults>

```

8. JavaScript (Internet Explorer): Search for an Artist's Albums and Tracks

Use Case:

The following code uses MediaNet's MN Open API (Search API, namely Search.GetAlbums API call) to find all albums and tracks by an artist, in this example by "Sting". The code is written in IE/JavaScript. At the core of this IE/JavaScript code is a .Microsoft.XMLHTTP open call to "GET" the results from **Search.GetAlbums API**. The IE/JavaScript code (contained in index.htm) wraps the **Search.GetAlbums API call** and does a HTTP GET call to the MN Open web site to obtain an XML response of all the albums and tracks for an artist, in this case sting. For a full listing see IE/JavaScript listing below.

```
xmlhttp.open("GET", "http://ie-api.mndigital.com?" + theQuery, true);
```

API and Parameters

Please see Open API Implementation Guide Part Two: Technical Details of the MN Open API, under **Search.GetAlbums API call** and its parameters

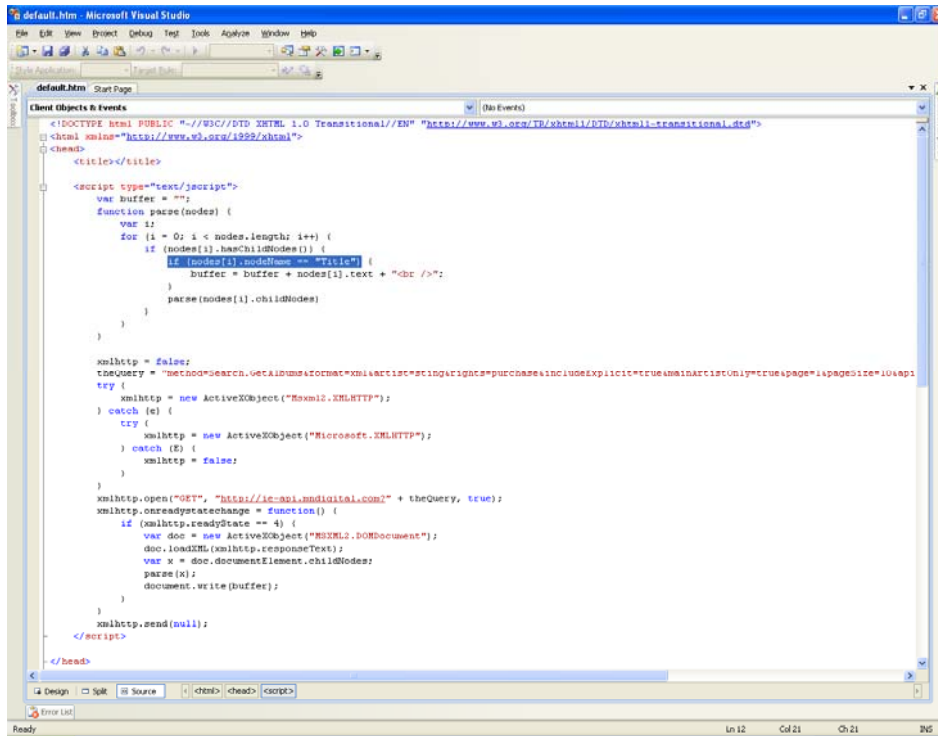
When the document is ready, i.e. the XML results are returned from the API call, a MSXML2.DOMDocument XML DOM object is created and the XML results are loaded (loadXML).

```
xmlhttp.onreadystatechange = function() {  
    if (xmlhttp.readyState == 4) {  
        var doc = new ActiveXObject("MSXML2.DOMDocument");  
        doc.loadXML(xmlhttp.responseText);  
        var x = doc.documentElement.childNodes;  
        parse(x);  
        document.write(buffer);  
    }  
}
```

A helper routine then looks for tags in the XML DOM:
if (nodes[i].nodeName == "Title")

A full listing of the Index.htm HTML page and the output is included below for completeness.

Screen of Index.htm



Code Listing:

Full HTML listing of Index.htm

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <script type="text/jscript">
    var buffer = "";
    function parse(nodes) {
      var i;
      for (i = 0; i < nodes.length; i++) {
        if (nodes[i].hasChildNodes()) {
          if (nodes[i].nodeName == "Title") {
            buffer = buffer + nodes[i].text + "<br />";
          }
          parse(nodes[i].childNodes)
        }
      }
    }

    xmlhttp = false;
    theQuery =
"method=Search.GetAlbums&format=xml&artist=sting&rights=purchase&includeExplicit=true&mainArtistO
nly=true&page=1&pageSize=10&apiKey=YourAPIKey";
    try {
      xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
      try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
      } catch (E) {
        xmlhttp = false;
      }
    }
  </script>
</head>

```

```
xmlhttp.open("GET", "http://ie-api.mndigital.com?" + theQuery, true);
xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4) {
        var doc = new ActiveXObject("MSXML2.DOMDocument");
        doc.loadXML(xmlhttp.responseText);
        var x = doc.documentElement.childNodes;
        parse(x);
        document.write(buffer);
    }
}
xmlhttp.send(null);
</script>
```

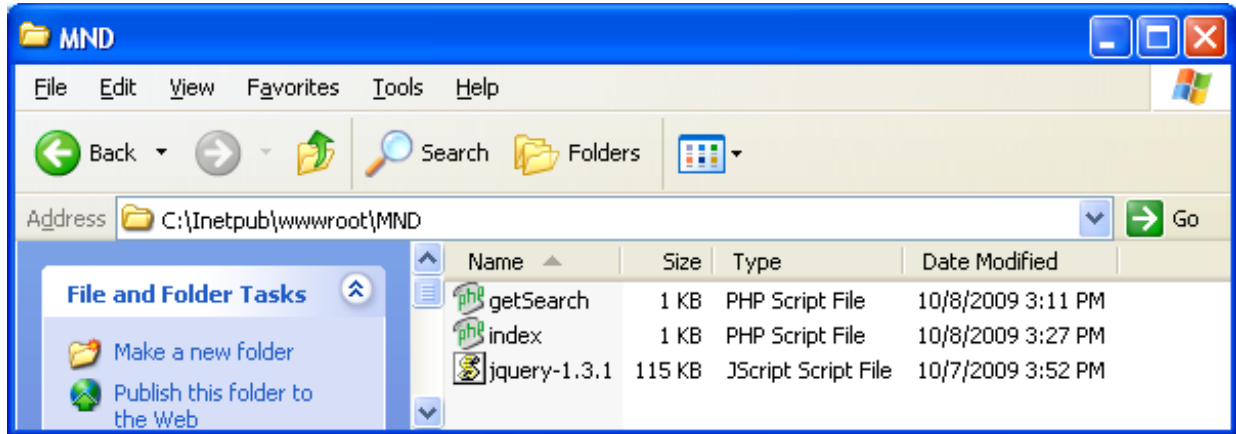
```
</head>
<body>
</body>
</html>
```

9 . PHP/jQuery: Search for an Artist's Albums and Tracks

Use Case:

The following code uses MediaNet Search API (Search.GetAlbums API call) to look for all albums and tracks by an artist, in this case by "Sting". At the core of this code is a call to **Search.GetAlbums**.

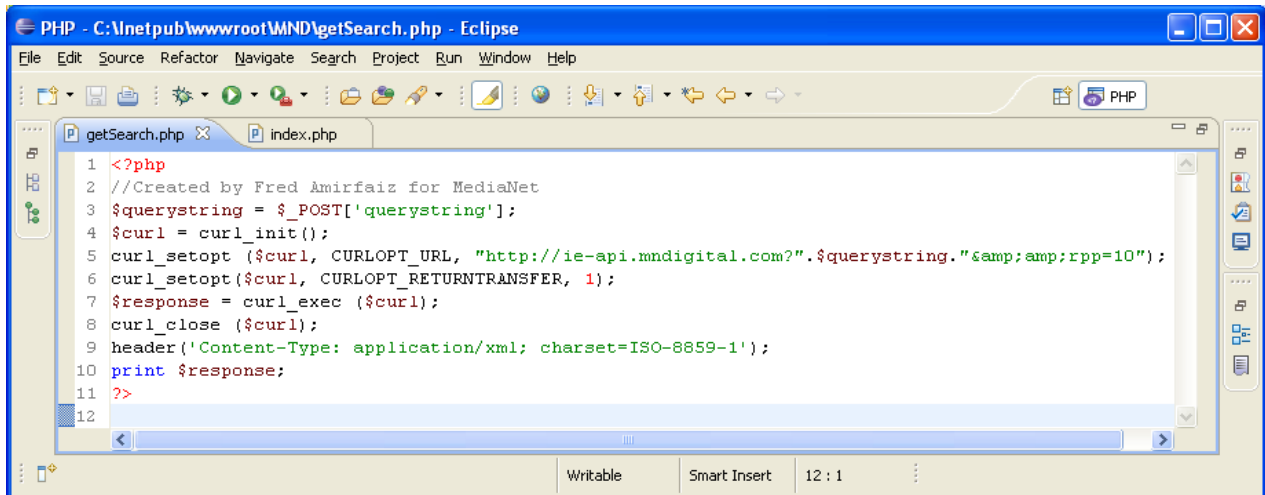
Project files:



The PHP code (contained in getSearch.php) wraps the **Search.GetAlbums API call** in a code module that simply does a POST HTTP call to the MN Open web site and obtains an XML response of all the albums and tracks for an artist, in this case sting. A **curl_init** function call is the center of the PHP code. For a full listing see PHP listing below.

```
$querystring = $_POST['querystring'];  
$curl = curl_init();  
curl_setopt ($curl, CURLOPT_URL, "http://ie-api.mndigital.com?.$querystring");  
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);  
$response = curl_exec ($curl);  
curl_close ($curl);  
header('Content-Type: application/xml; charset=ISO-8859-1');  
print $response;
```

GetSearch.php helper page



```

1 <?php
2 //Created by Fred Amirfaiz for MediaNet
3 $querystring = $_POST['querystring'];
4 $curl = curl_init();
5 curl_setopt ($curl, CURLOPT_URL, "http://ie-api.mndigital.com?". $querystring. "&amp; rpp=10");
6 curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
7 $response = curl_exec ($curl);
8 curl_close ($curl);
9 header('Content-Type: application/xml; charset=ISO-8859-1');
10 print $response;
11 ?>
12

```

Think of the code (contained in Index.php) as a JQuery helper function.
 The line : `<script type="text/javascript" src="jquery-1.3.1.js"></script>` This line enables JQuery
 Adds JQuery functionality to your PHP page.

Function:

`$(document).ready(function() { // JQuery function to execute when the HTML document is ready
 mndSearch();`

Simply specifies a function to run when the page is loaded. Think of it as a Page_load in C# .NET, or a script block that executes when the document DOM is fully loaded.

API and Parameters

Please see Open API Implementation Guide Part Two: Technical Details of the MN Open API, under **Search.GetAlbums API call** and the parameters

```

$('Title', xml).each(function(i) {
    // Get the text for each 'Title' nodes in the returned xml
    // this includes Albums titles and nested track titles
    var title = $(this).text();
    //Make an HTML list set of the titles
    displayResults = displayResults + "<li> " + title + "</li>";
});

```

A full listing of the helper code, and the full listing of the Index.php HTML page, the output, and the XML is shown below for completeness.

Code Listing 1 PHP:

The full listing of Index.php is shown here:

```
<script type="text/javascript" src="jquery-1.3.1.js"></script> This line enables JQuery
Check for the latest version and documentation on http://docs.jquery.com/Main\_Page

<script type="text/javascript" >
$(document).ready(function() { // JQuery function to execute when the HTML document is ready
mndSearch();
});

function mndSearch()
{
    var displayResults = ""; // Initialize display Results
    // Initialize the query parameters for Search.GetAlbums API call
    var theQuery =
"method=Search.GetAlbums&format=xml&artist=sting&rights=purchase&includeExplicit=true&mainArtistO
nly=true&page=1&pageSize=10&apiKey=APIKey";
    // Display the query for show
    $("#AlbumAndTrackContainer").html(theQuery);

    // Execute a JQuery HTTP post using the HTTP query string
    // When the HTTP post is completed, execute function(xml)
    // on the XML blob returned from the HTTP post call

    $.post("getSearch.php?", {querystring:theQuery}, function(xml) {
    // look for Title nodes in the returned xml
    $("Title", xml).each(function(i) {
        // Get the text for each 'Title' nodes in the returned xml
        // this includes Albums titles and nested track titles
        var title = $(this).text();
        //Make an HTML list set of the titles
        displayResults = displayResults + "<li> " + title + "</li>";
    });
    // Show the HTML list set of the titles in the AlbumAndTrackContainer <div>
    $("#AlbumAndTrackContainer").html(displayResults);
    }, "xml"); // make the format XML
}
</script>
```

Code Listing 1 PHP:

Full listing of Index.php

```
<html>
<head>
<script type="text/javascript" src="jquery-1.3.1.js"></script>
<script type="text/javascript" >
// When the document is ready, execute mndSearch() function
$(document).ready(function() {
    mndSearch();
});

function mndSearch()
{
    var displayResults = ""; // Initialize display Results
    // Initialize the query parameters for search
    var theQuery =
"method=Search.GetAlbums&format=xml&artist=sting&rights=purchase&includeExplicit=true&mainArtistO
nly=true&page=1&pageSize=10&apiKey=yourApiKey";
    // Display the query for show
    $("#AlbumAndTrackContainer").html(theQuery);

    // Execute a jQuery HTTP post using the HTTP query
```

```
// When the post is complete, execute function(xml)
// on the xml blob returned from the HTTP call

$.post("getSearch.php?", {querystring:theQuery}, function(xml) {
// look for Title nodes in the returned xml
$('Title', xml).each(function(i) {
// Get the text for each 'Title' nodes in the returned xml
// this includes Albums titles and nested track titles
var title = $(this).text();
//Make an HTML list set of the titles
displayResults = displayResults + "<li> " + title + "</li>";
});
// Show the HTML list set of the titles in the AlbumAndTrackContainer <div>
$("#AlbumAndTrackContainer").html(displayResults);
}, "xml"); // make the format XML
}
</script>
</head>
<body>
Searching for artist "sting" for all Albums and all tracks.
<br /><br />
<div id="AlbumAndTrackContainer"></div>
<p>
</body>
</html>
```



```
Debug - MNDPerf/index.pl - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

#####

# Make the API call
my $httpRequest = HTTP::Request->new($urlMethod, $server.$method.$postFormat.$params.$apiKey);
#Send the request and get HTTP:Response object
my $httpResponse = $objUserAgent->request($httpRequest);

#See if there were any errors getting the response
if ($httpResponse->is_error)
{
    #There were errors
    print $httpResponse->error_as_HTML;
}
else # No Errors
{
    #Create a new DOM:Parser
    my $parser = new XML::DOM::Parser();
    my $responseDoc = $parser->parse($httpResponse->content);
    #Look for Title nodes in the XML
    my $albumNodes = $responseDoc->getElementsByTagName('Album', 1);

    print "<B>For Artist Sting, Medianet Search API returned the following albums and title(s):</B>,<BR />";
    if($albumNodes->getLength() > 0)
    {
        #go through each XML Album node
        for(my $i = 0; $i < $albumNodes->getLength(); $i++)
        {
            #Get this Album node
            my $thisNode = $albumNodes->item($i);
            print "<BR />Album: " . $thisNode->getElementsByTagName('Title')->item(0)->getChildAtIndex(0)->getNodeValue() . "<br />";
            #Get the tracks and print track titles
            my $Titles = $thisNode->getElementsByTagName('Title', 1);
            for ($j = 1; $j < $Titles->getLength(); $j++) {
                print "\n\nTrack: " . $Titles->item($j)->getChildAtIndex(0)->getNodeValue() . "<br />";
            }
        }
        #for(my $i = 0; $i < $albumNodes->getLength(); $i++)
        #if($albumNodes->getLength() > 0)
        $responseDoc->dispose();
    }
    #else # No Errors
    exit;
}

#####

# Builds the XML request body to send to the server
# containing the data needed by the API
sub buildAPIxmlBody()
```

Code Listing PERL:

Full HTML of Index.Pperl

```
use LWP::UserAgent;

use HTTP::Request;
use HTTP::Response;
use XML::DOM;
use CGI qw(:standard);
use CGI::Carp qw(fatalsToBrowser);

#####

#UserAgent - used to process the web requests.
my $objUserAgent = LWP::UserAgent->new;

#Tell IIS what type of content we are returning, See http://support.microsoft.com/kb/145661
print "Content-Type: text/html\n\n";

#Get the HTTP::Headers object needed for calls to the MNDigital API
my $httpHeader = geteMNDHeaders();

#Get the body of the XML request for this call
my $requestXmlBody = buildAPIXmlBody();

# The query is fully spelled out to illustrate the API call build this in a more generic manner
to cover all APIs
#Create a HTTP:Request to the IE server, using the headers and body specified
my $uRLMethod = 'GET';
my $server = "http://ie-api.mndigital.com?";
my $method = 'method=Search.GetAlbums';
my $postFormat = '&format=xml';
my $params =
'&title=&artist=sting&artistMnetId=&rights=purchase&keyword=&includeExplicit=true&mainArtistOnly=
true&page=1&pageSize=10';
my $apiKey = '&apiKey=yourApiKey';
# print $method, $server.$method.$postFormat.$params.$apiKey;

#####

# Make the API call
my $httpRequest = HTTP::Request->new($uRLMethod, $server.$method.$postFormat.$params.$apiKey);
#Send the request and get HTTP::Response object
my $httpResponse = $objUserAgent->request($httpRequest);

#See if there were any errors getting the response
if ($httpResponse->is_error)
{
    #There were errors
    print $httpResponse->error_as_HTML;
}
else # No Errors
{
    #Create a new DOM::Parser
    my $parser = new XML::DOM::Parser();
    my $responseDoc = $parser->parse($httpResponse->content);
    #Look for Title nodes in the XML
    my $albumNodes = $responseDoc->getElementsByTagName('Album', 1);

    print "<B>For Artist Sting, Medianet Search API returned the following albums and
title(s):</B>,<BR />";
    if($albumNodes->getLength() > 0)
    {
        #go through each XML Album node
        for(my $i = 0; $i < $albumNodes->getLength(); $i++)
```

```

    {
        #Get this Album node
        my $thisNode = $albumNodes->item($i);
        print "<BR />Album: " . $thisNode->getElementsByTagName('Title')->item(0)-
>getChildAtIndex(0)->getNodeValue() . "<br />";
        #Get the tracks and print track titles
        my $Titles = $thisNode->getElementsByTagName('Title', 1);
        for ($j = 1; $j < $Titles->getLength(); $j++) {
            print "&nbsp;&nbsp;&nbsp;&nbsp;Track: " . $Titles->item($j)-
>getChildAtIndex(0)->getNodeValue() . "<br />";
        }
    } #for(my $i = 0; $i < $albumNodes->getLength(); $i++)
} #if($albumNodes->getLength() > 0)
$responseDoc->dispose();
} #else # No Errors
exit;

#####

# Builds the XML request body to send to the server
# containing the data needed by the API

sub buildAPIXmlBody()
{
    #build a string representation of the XML, containing all the post data
    my $xmlBody = ""; # there is no post data for this API
    #return the XML string
    return $xmlBody;
}

#####
# Construct and return HTTP::Headers for the API call

sub geteMNDHeaders()
{
    #Create new HTTP::Headers object
    my $httpHeader = HTTP::Headers->new;
    #set all the headers with their values
    $httpHeader->push_header('Content-Type: text/html\n\n');
    #return the object
    return $httpHeader;
}

```

Output:

Output from Index.Perl

For Artist Sting, Medianet Search API returned the following albums and title(s):

Album: If On A Winter's Night
 Track: Gabriel's Message
 Track: Soul Cake
 Track: There Is No Rose Of Such Virtue
 Track: The Snow It Melts The Soonest
 Track: Christmas At Sea
 Track: Lo How A Rose E'er Blooming
 Track: Cold Song
 Track: The Burning Babe
 Track: Now Winter Comes Slowly
 Track: The Hounds Of Winter
 Track: Balulalow
 Track: Cherry Tree Carol
 Track: Lullaby For An Anxious Child
 Track: Hurdy Gurdy Man
 Track: You Only Cross My Mind In Winter
 Album: The Very Best Of Sting And The Police (2002 Brits Version)

Track: Message In A Bottle
Track: Can't Stand Losing You
Track: An Englishman In New York
Track: Every Breath You Take
Track: Seven Days
Track: Walking On The Moon
Track: Fields Of Gold
Track: Fragile
Track: Every Little Thing She Does Is Magic
Track: De Do Do Do, De Da Da Da
Track: If You Love Somebody Set Them Free
Track: Brand New Day
Track: Desert Rose
Track: If I Ever Lose My Faith In You
Track: When We Dance (Edit)
Track: Don't Stand So Close To Me (Original Album Version)
Track: Roxanne
Track: So Lonely

Truncated to save space

Album: ...Nada Como El Sol - Selecciones Especiales En Espanol Y Portugues

Track: Mariposa Libre
Track: Frágil (Portuguese)
Track: Si Estamos Juntos
Track: Ellas Danzan Solas (Cueca Solo)
Track: Fragilidad

Album: ...Nothing Like The Sun

Track: The Lazarus Heart
Track: Be Still My Beating Heart
Track: An Englishman In New York
Track: History Will Teach Us Nothing
Track: They Dance Alone
Track: Fragile
Track: We'll Be Together
Track: Straight To My Heart
Track: Rock Steady
Track: Sister Moon
Track: Little Wing
Track: The Secret Marriage

Album: Absolute Sting: The Unauthorised Interview

Track: Absolute Sting: The Unauthorised Interview

Album: Brand New Day

Track: A Thousand Years
Track: Desert Rose
Track: Big Lie Small World
Track: After The Rain
Track: Perfect Love... Gone Wrong
Track: Tomorrow We'll See
Track: Prelude To The End Of The Game
Track: Fill Her Up
Track: Ghost Story
Track: Brand New Day

11. PHP: Add an Item to a New Cart and Perform a Purchase

Use Case:

The following code uses MediaNet's MN Open API (Cart, namely Cart.AddItem & Cart.Purchase API calls) to add a first item (a track) to a new cart and continue to purchase it. The code is written in PHP. At the core of this code is a call to Cart.AddItem. The PHP code (contained in index.php) wraps the **Cart.AddItem & Cart.Purchase API calls** using HTTP calls to the MN Open web APIs. A **curl_exec** function call is the center of the PHP code. For a full listing see PHP listing below.

API and Parameters

Please see Open API Implementation Guide Part Two: Technical Details of the MN Open API, under **Cart.AddItem API call** and its parameters

```
$theQuery = "method=Cart.AddItem";  
// this is your shared secret fir IE  
$secret = "yoursharedsecret";  
// This is your API Key  
$APIKEY="yourapikey";  
$theparams= "&format=xml&APIKEY=".$APIKEY;  
$theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,  
$secret);
```

The actual item to add to the new cart is specified in the XML payload of the PHP Post call. The XML to add `<MnetId>12296213</MnetId>` item, typically obtained from a search call to find album tracks, to the cart, is shown here:

```
<AddItems xmlns="http://api.mndigital.com">  
  <Items>  
    <LineItem>  
      <MnetId>12296213</MnetId>  
      <ItemType>Track</ItemType>  
      <Format>MP3</Format>  
      <Price>0.99</Price>  
    </LineItem>  
  </Items>  
</AddItems>
```

Ideally, check for `<Success>>true</Success>` first.

Save the cart ID (`<CartId>39209</CartId>`) if you like to add a second item to the same cart.

Check to make sure there are no duplicate tracks in the cart using

`<DuplicateItemInCart>>false</DuplicateItemInCart>`.

Check to make sure the `<Count>1</Count>` matches your expectation.

For tracks that are sold only as part of an album, check also:

```
<CartAlbumContainsTrack>>false</CartAlbumContainsTrack>  
<CartTrackInAlbum>>false</CartTrackInAlbum>
```

A full listing of the Index.php HTML page, the output, and the XML returned from the MN Open API call, is included below for completeness.

Cart.Purchase

Same as cart add item with addition of :

`&cartId=39629` Returned from the first Cart.AddItem call, used in subsequent Cart.AddItem calls and in Cart.Purchase call

`&userIp=0.0.0.0` for tracking and reporting, use your IP address

`&userDomain=testdomain.mndigital.com` for tracking and reporting, use your Domain name

The actual cart to complete purchase, is specified in `&cartId=". $cartIDRet` and the credit card information is included in the XML payload of the PHP Post call. The XML to add `<MnetId>12296213</MnetId>` item, typically obtained from a search call to find album tracks, to the cart, is shown here:

```
$data = utf8_encode('
    <Purchase xmlns="http://api.mndigital.com">
        <User>
            <FirstName>First</FirstName>
            <LastName>Last</LastName>
            <EmailAddress>test@mndigital.com</EmailAddress>
            <AllowHtmlEmail>true</AllowHtmlEmail>
            <BillingAddress>
                <AddressLine1>2401 Elliott Ave.</AddressLine1>
                <AddressLine2>Suite 300</AddressLine2>
                <City>Seattle</City>
                <State>WA</State>
                <PostalCode>98121</PostalCode>
                <Country>US</Country>
            </BillingAddress>
            <CreditCard>
                <FirstName>First</FirstName>
                <LastName>Last</LastName>
                <Number>4111111111111111</Number>
                <CVV>000</CVV>
                <CardType>Visa</CardType>
                <ExpirationMonth>5</ExpirationMonth>
                <ExpirationYear>2010</ExpirationYear>
            </CreditCard>
        </User>
    </Purchase>');
```

A full listing of the Index.php HTML page, the output, and the XML returned from the MN Open API call, is included below for completeness.

Code Listing PHP:

PHP Post example of using Cart.AddItem

```
<html>
<head>
</head>
<body>
Adding a first item number to a new cart.
<br /><br />
<?php
    // Initialize the query parameters for Cart.AddItem
    // The query is fully spelled out to illustrate the API call , build this query in a more
    general fashion,
    // depending on the API call you like to make
    // and the parameters specific to the API call
    // Please note signatures may be required for other API calls
    $theQuery = "method=Cart.AddItem";
    // this is your shared secret fir IE
    $secret = "sharedsecret";
    // This is your API Key
    $APIKEY="yourapiKey";
    $theParams= "&format=xml&APIKEY=". $APIKEY;
    $theQuery = $theQuery.$theParams."&signature=".hash_hmac('md5', $theQuery.$theParams,
$secret);
    // Build the XML payload for post
    // The query is fully spelled out to illustrate the API call , build this XML payload in a
    more general fashion
    $data = utf8_encode('
        <AddItems xmlns="http://api.mndigital.com">
            <Items>
                <LineItem>
                    <MnetId>12296213</MnetId>
                    <ItemType>Track</ItemType>
```

```

                <Format>MP3</Format>
                <Price>0.99</Price>
            </LineItem>
        </Items>
    </AddItems>');
$url = "http://ie-api.mndigital.com?".$theQuery;
// for debug
echo $url."<br /><br />";
// This is standard curl call of PHP
$curl = curl_init();
curl_setopt ($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_POST, true);
// set the Content-Type to application/xml
curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
// for debug
$response = curl_exec ($curl);
if (curl_errno($curl))
{
    print "error";
    print curl_error($curl);
}

curl_close ($curl);
//echo "<br />".$response."<br />";
// Now that we have the XML response,
// let's load it and find the cart id
$domDoc = new DOMDocument(); ;
$dom = $domDoc->loadXML($response);
// echo $response;
// this should not happen
if (!$dom) {
    echo "Error while parsing the response";
    exit;
}
// We have a valid XML doc loaded
$CartItemCount = $domDoc->getElementsByTagName('CartItemCount'); // All Cart Items
for ($i = 0; $i < $CartItemCount->length; $i++) {
    $Nodes = $CartItemCount->item($i)->getElementsByTagName('CartId'); // Searching for
CartId
    $cartIDRet = $CartItemCount->item($i)->getElementsByTagName('CartId')->item(0)-
>nodeValue;
    echo "CartId returned from Cart.AddItem API: ".$cartIDRet. "<br />";
}

// Initialize the query parameters for Cart.Purchase
// The query is fully spelled out to illustrate the API call , build this query in a more
general fashion,
// depending on the API call you like to make
// and the parameters specific to the API call
// Please note signatures may be required for other API calls
$theQuery = "method=Cart.Purchase";
// this is your shared secret fir IE
$secret = "Yoursharedsecret";
// This is your API Key
$APIKEY="YourAPIKey";
$theparams=
"&format=xml&cartId=".$cartIDRet."&userIp=0.0.0&userDomain=testdomain.mndigital.com&APIKEY=".$A
PIKEY;
    $theQuery = $theQuery.$theparams."&signature=".hash_hmac('md5', $theQuery.$theparams,
$secret);
// Build the XML payload for post
// The query is fully spelled out to illustrate the API call , build this XML payload in a
more general fashion
    $data = utf8_encode('
        <Purchase xmlns="http://api.mndigital.com">

```

```
        <User>
          <FirstName>First</FirstName>
          <LastName>Last</LastName>
          <EmailAddress>test@mndigital.com</EmailAddress>
          <AllowHtmlEmail>true</AllowHtmlEmail>
          <BillingAddress>
            <AddressLine1>2401 Elliott Ave.</AddressLine1>
            <AddressLine2>Suite 300</AddressLine2>
            <City>Seattle</City>
            <State>WA</State>
            <PostalCode>98121</PostalCode>
            <Country>US</Country>
          </BillingAddress>
          <CreditCard>
            <FirstName>First</FirstName>
            <LastName>Last</LastName>
            <Number>4111111111111111</Number>
            <CVV>000</CVV>
            <CardType>Visa</CardType>
            <ExpirationMonth>5</ExpirationMonth>
            <ExpirationYear>2010</ExpirationYear>
          </CreditCard>
        </User>
      </Purchase>');
$url = "https://ie-api.mndigital.com?"..$theQuery;
// for debug
echo $url."<br /><br />";
// This is standard curl call of PHP
$curl = curl_init();
curl_setopt ($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_POST, true);
// set the Content-Type to application/xml
curl_setopt($curl,CURLOPT_HTTPHEADER,array ("Content-Type: application/xml"));
curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
// for debug
$response = curl_exec ($curl);
if (curl_errno($curl))
{
    print "error: ";
    print curl_error($curl);
}
?>
</body>
</html>
```

Output

Adding a first item number to a new cart.

http://ie-api.mndigital.com?method=Cart.AddItems&format=xml&APIKEY=yourApiKey&signature=45410f35bf9f1acdc9c3ab38d2ac5944

CartId returned from Cart.AddItems API: **39629**

https://ie-api.mndigital.com?method=Cart.Purchase&format=xml&cartId=**39629**&userIp=0.0.0.0&userDomain=testdomain.mndigital.com&APIKEY=yourApiKey &signature=60a5e9d8b698c69877fa6f38bce8d1e5

Email sent from Cart.Purchase.

From: support@mndigital.com [mailto:support@mndigital.com]
Sent: Friday, November 20, 2009 2:31 PM
To:
Subject: Your MP3 purchase from Open API Test Partner



Order Completed

Thank you for your order!

Here is your receipt for your MP3 purchase from Open API Test Partner. If you have any questions about your purchase, please contact customer support at [\[support@mndigital.com\]](mailto:support@mndigital.com). Include a description of your issue and your order number in your e-mail.

Order Summary

Songs/Albums	Content	Price
 Karaoke: Spin The Black Circle by Pearl Jam	MP3	\$0.99

Tax: \$0.09
Total: \$1.08

Order Number:	#326352	Card Type:	Visa
Date:	11-20-2009	Name:	First Last
Time:	14:30	Card Number:	*****1111

If you were not able to complete your downloads, they are [available here](#).

12. Java: Searching for Streaming Tracks

Use Case:

The following Android Java code uses MediaNet's MN Open API (Search, namely Search.GetTracks API call) to look for a sample music Track called "fields of gold". The Java code wraps the Search. GetTracks API call using a HTTP GET call (via a URL object) to the MN Open integration environment to obtain the sample stream for a track. A URL `getInputStream` call is the center of the code. The XML is then parsed using standard SAX (or XML DOM) and the *Location* of the sample is then sent to the built in player object.

Code Listing JAVA:

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
import java.net.URLConnection;
import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.webkit.URLUtil;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.InputSource;
import org.xml.sax.XMLReader;

private class MN_OpenAPIDemo extends Activity {
    private static final String TAG = "MN_OpenAPIDemo";

    private VideoView mStreamPlayer;
    private EditText mPath;
    private String mURLPath;
    private String current;

    @Override
    private void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);
        mStreamPlayer = (VideoView) findViewById(R.id.surface_view);

        mPath = (EditText) findViewById(R.id.path);
        try {
            // For a fuller application, build this URL in a more generic manner
            // all Search.GetTracks parameters are URL encoded
            // Note &rights=Sample
            // The URL is included in raw form here to show the parameters
            mURLPath = getPlaySource("http://ie-
api.mndigital.com/?method=Search.GetTracks&format=xml&title=fields%20of%20gold&album=&albumMnetId
=&artist=&artistMnetId=&keyword=&rights=Stream&includeExplicit=true&mainArtistOnly=true&genre=&pa
ge=1&pageSize=10&blacklistIds=&apiKey=YourAPIKey");
            mPath.setText("fields of gold");
        } catch (IOException e) {
            e.printStackTrace();
        }

        private void playTrackStream() {
```

```

try {
    String path = mURLPath.toString();
    Log.v(TAG, "path: " + path);
    if (path == null || path.length() == 0) {
        Toast.makeText(MN OpenAPIDemo.this, "Streaming coming soon",
            Toast.LENGTH_LONG).show();
    } else {
        if (path.equals(current) && mStreamPlayer != null) {
            mStreamPlayer.start();
            mStreamPlayer.requestFocus();
            return;
        }
        current = path;
        mStreamPlayer.setVideoPath(getLocalAudioSource(path));
        mStreamPlayer.start();
        mStreamPlayer.requestFocus();
    }
} catch (Exception e) {
    Toast.makeText(MN OpenAPIDemo.this, "Streaming error: " + e.getMessage(),
        Toast.LENGTH_LONG).show();
    if (mStreamPlayer != null) {
        mStreamPlayer.stopPlayback();
    }
}
}

private String getLocalAudioSource(String path) throws IOException {
    URL url = new URL(path);
    URLConnection cn = url.openConnection();
    cn.connect();
    InputStream stream = cn.getInputStream();
    if (stream == null)
        throw new RuntimeException("stream");
    File temp = File.createTempFile("streamplay", "mnd");
    temp.deleteOnExit(); // erase local file when play is complete
    String tempPath = temp.getAbsolutePath();
    FileOutputStream out = new FileOutputStream(temp);
    byte buf[] = new byte[128];
    do {
        int numread = stream.read(buf);
        if (numread <= 0)
            break;
        out.write(buf, 0, numread);
    } while (true);
    try {
        stream.close();
    } catch (IOException ex) {
        Log.e(TAG, "error: " + ex.getMessage(), ex);
    }
    return tempPath;
}

private String getPlaySource(String path) throws IOException {
    try {
        URL url = new URL(path);
        URLConnection cn = url.openConnection();
        cn.connect();
        InputStream istrm = url.openStream();
        InputSource isrc = new InputSource(istrm);
        // can also load the XML returned from API call into an XML DOM
        SAXParserFactory spf = SAXParserFactory.newInstance();
        SAXParser sp = spf.newSAXParser();
        XMLReader xr = sp.getXMLReader();
        XMLPareseHandler myXMLPareseHandler = new XMLPareseHandler();
        xr.setContentHandler(myXMLPareseHandler);
        xr.parse(isrc);
        ParsedDataSet parsedExampleDataSet = myXMLPareseHandler.getParsedData();
        return parsedDataSet.Location;
    }
}

```

```

    } catch (Exception e) {
        Log.e(TAG, "Service temporarily unavailable", e);
    }
    return null;
}
}

```

Response:

The API call returns the following typical XML response

```

- <TrackSearchResults xmlns="http://api.mndigital.com" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  - <Debug>
    <Version i:nil="true" />
    <QueryString i:nil="true" />
    <Trace>Time: 11:09:48 AM 0.0000000 0.0000000 Timer Initialized 0.0000000 0.0000000
Volt.OpenApi.Packages.SearchPackage GetTracks</Trace>
  </Debug>
  <Success>true</Success>
  <ResultsReturned>10</ResultsReturned>
  <TotalResults>182</TotalResults>
  - <Tracks>
    - <Track>
      <MnetId>540999</MnetId>
      <Title>Fields Of Gold</Title>
      - <Artist>
        <MnetId>31068</MnetId>
        <Name>Sting</Name>
        - <Images>
          <Artist180x80>http://images.musicnet.com/artists/000/031/068/b.jpeg</Artist180x80>
          <Artist190x230>http://images.musicnet.com/artists/000/031/068/c.jpeg</Artist190x230>
          <Artist375x250>http://images.musicnet.com/artists/000/031/068/a.jpeg</Artist375x250>
          <Artist200x>http://images.musicnet.com/artists/000/031/068/f.jpeg</Artist200x>
        </Images>
      </Artist>
      <Genre>Pop</Genre>
      <ExplicitLyrics>>false</ExplicitLyrics>
      <Duration>3:40</Duration>
      <ReleaseDate>01-01-1994</ReleaseDate>
      <TrackNumber>3</TrackNumber>
      <DiscNumber>1</DiscNumber>
      - <PriceTag>
        <Currency>USD</Currency>
        <Amount>1.29</Amount>
        <IsSetPrice>>true</IsSetPrice>
        <WholesalePrice>1.01</WholesalePrice>
      </PriceTag>
      - <Mp3Rights>
        <CanSampleStream>>true</CanSampleStream>
        <CanPurchaseDownload>>true</CanPurchaseDownload>
        <AlbumPurchaseOnly>>false</AlbumPurchaseOnly>
      </Mp3Rights>
      <FeaturedArtists i:nil="true" />
    - <Album>
      <MnetId>540993</MnetId>
      <Title>Fields Of Gold: The Best Of Sting 1984-1994</Title>
      <Artist i:nil="true" />
      <Genre>Pop</Genre>
      <ExplicitLyrics>>false</ExplicitLyrics>
      <Label>A&M</Label>
      <Duration>66:29</Duration>
      <ReleaseDate>01-01-1994</ReleaseDate>
      <NumberOfTracks>14</NumberOfTracks>
      - <Images>
        <Album75x75>http://images.musicnet.com/albums/000/540/993/s.jpeg</Album75x75>
        <Album150x150>http://images.musicnet.com/albums/000/540/993/m.jpeg</Album150x150>
        <Album800x800>http://images.musicnet.com/albums/000/540/993/g.jpeg</Album800x800>
      </Images>
    </Album>
  </Track>
</Tracks>
</TrackSearchResults>

```

```

        <FeaturedArtists i:nil="true" />
        <LabelOwnerId>5</LabelOwnerId>
        <Bitrate>320kb</Bitrate>
    </Album>
    - <SampleLocations>
        - <MediaLocation>
            <Location>rtmp://DMUS.flash-ie.musicnet.com:80/Sample</Location>
            <Resource>mp4:/spl/000/540/999/spl_029.mp4</Resource>
            <Type>s_mp4</Type>
        </MediaLocation>
        - <MediaLocation>
            <Location>http://DMUS.download-
ie.musicnet.com/getFile?file=&af=mp4&token=21f9c8afe2ac9c3e7b720469757f944ce725c3587afc2c8a4482de462f97e1a54756
e352ee95441596dba4bbf2eeel681ac811c74e6d2fd56c8ed52067aba6547679dac0def63dc909a2d2318e1d7bf0ca1d57de0bec1484d3b
7b099a15cf5ee</Location>
            <Type>p_mp4</Type>
        </MediaLocation>
        - <MediaLocation>
            <Location>rtmp://DMUS.flash-ie.musicnet.com:80/Sample</Location>
            <Resource>mp3:/spl/000/540/999/spl_024</Resource>
            <Type>s_mp3</Type>
        </MediaLocation>
        - <MediaLocation>
            <Location>http://DMUS.download-
ie.musicnet.com/getFile?file=&af=mp3&token=566ec9f84da892edb274cfe25518a0d5d1711f7a741730087b96a7d9e964a7824c07
09c0fb603887f238fe6d6d15e67e295cfe9750b6a0ef99aed75b84c54d0a1f4e8be594dd886bcc166a83bbda9c05426f48129f7b0fddf80
3fe6bd6262997</Location>
            <Type>p_mp3</Type>
        </MediaLocation>
    </SampleLocations>
    <Bitrate>320kb</Bitrate>
    <PopularityRanking>11782</PopularityRanking>
</Track>
.....
</TrackSearchResults>

```


13. DOT NET: Search for an Artist's Tracks

Use Case:

The following DOT NET code uses MediaNet's MN Open API (Search, namely Search.GetTracks API call) to look for a Track by "sting". The code wraps the Search. GetTracks API call using a WebRequest / WebResponse to the MN Open integration environment to obtain the list of tracks by sting. A URL *getInputStream* call is the center of the code. The XML is then parsed using standard XML DOM (XmlDocument) and the *Title* of the tracks and albums is then printed using Response.Write.

Code Listing DOT NET:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Xml;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Create a request.
        string requestUrl = "http://ie-
api.mndigital.com?method=Search.GetTracks&format=xml&title=&album=&albumMnetId=&artist=sting&arti
stMnetId=&keyword=&rights=purchase&includeExplicit=true&mainArtistOnly=true&genre=&page=1&pageSiz
e=10&blacklistIds=&apiKey=yourAPIKey";

        WebRequest request = WebRequest.Create(requestUrl);
        request.Method = "GET";
        WebResponse response = request.GetResponse();
        Response.Write("Executed: " + requestUrl + "<br /><br />");
        Response.Write("WebRequest Status: " + ((HttpWebResponse)response).StatusDescription +
"<br /><br />");
        Stream dataStream = response.GetResponseStream();
        StreamReader reader = new StreamReader(dataStream);
        XmlDocument doc = new XmlDocument();
        doc.Load(reader);
        XmlNodeList nodes = doc.GetElementsByTagName("Title");
        for (int i = 0; i < nodes.Count; i++)
        { //Every other title is a track and album title in the XML response
            XmlNode node = nodes[i++];
            string nodeName = node.InnerText;
            Response.Write("Track: " + nodeName + ", from the album: ");
            node = nodes[i];
            nodeName = node.InnerText;
            Response.Write(nodeName + "<br />");
        }
        reader.Close();
        dataStream.Close();
        response.Close();
    }
}
```

Output:

```
Executed: http://ie-  
api.mndigital.com?method=Search.GetTracks&format=xml&title=&album=&albumMnetId=&artist=sting&artistMnetId=&keyword=&rights=p  
urchase&includeExplicit=true&mainArtistOnly=true&genre=&page=1&pageSize=10&blacklistIds=&apiKey=yourApiKey  
WebRequest Status: OK
```

```
Track: Desert Rose, from the album: The Very Best Of Sting And The Police (2002 Brits Version)  
Track: All For Love, from the album: The Best Of Me  
Track: Every Little Thing She Does Is Magic, from the album: The Very Best Of Sting And The Police (2002 Brits Version)  
Track: Roxanne, from the album: The Very Best Of Sting And The Police (2002 Brits Version)  
Track: Brand New Day, from the album: The Very Best Of Sting And The Police (2002 Brits Version)  
Track: Every Breath You Take, from the album: The Very Best Of Sting And The Police (2002 Brits Version)  
Track: Fields Of Gold, from the album: Fields Of Gold: The Best Of Sting 1984-1994  
Track: I'm So Happy I Can't Stop Crying, from the album: Toby Keith: 35 Biggest Hits  
Track: Fields Of Gold, from the album: Ten Summoner's Tales  
Track: If I Ever Lose My Faith In You, from the album: ...All This Time
```

DOT NET: Add an Item to a Cart (XML Format)

Use Case:

The following DOT NET code wraps the Cart.AddItem API call using a WebRequest / WebResponse to the MN Open integration environment to obtain a cart id.

Code Listing DOT NET:

```
using System;  
using System.Web;  
using System.Web.UI;  
using System.IO;  
using System.Net;  
using System.Text;  
using System.Xml;  
  
public partial class _Default : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        string requestUrl = "https://ie-  
api.mndigital.com?method=Cart.AddItem&format=xml&cartId=&apiKey=yourAPIKey&signature=beeb5c2f4ba  
e539c37eb5079ff6ae2b4";  
        string postData = "<AddItems  
xmlns='http://api.mndigital.com'><Items><LineItem><MnetId>12296213</MnetId><ItemType>Track</ItemT  
ype><Format>MP3</Format><Price></Price><Tax></Tax></LineItem></Items></AddItems>";  
  
        WebRequest request = WebRequest.Create(requestUrl);  
        request.Method = "POST";  
        byte[] byteArray = Encoding.UTF8.GetBytes(postData);  
        request.ContentType = "application/xml";  
        request.ContentLength = byteArray.Length;  
        Stream dataStream = request.GetRequestStream();  
        dataStream.Write(byteArray, 0, byteArray.Length);  
        dataStream.Close();  
        WebResponse response = request.GetResponse();  
        Response.Write("Executed: " + requestUrl + "<br /><br />");  
        Response.Write("WebRequest Status: " + ((HttpWebResponse)response).StatusDescription +  
"<br /><br />");  
        dataStream = response.GetResponseStream();  
        StreamReader reader = new StreamReader(dataStream);  
        XmlDocument doc = new XmlDocument();
```

```
doc.Load(reader);
XmlNodeList nodes = doc.GetElementsByTagName("CartId");
for (int i = 0; i < nodes.Count; i++)
{
    XmlNode node = nodes[i];
    string nodeName = node.InnerText;
    Response.Write("CartId: " + nodeName );
}
reader.Close();
dataStream.Close();
response.Close();
}
}
```

Output:

Executed: <https://ie-api.mndigital.com?method=Cart.AddItem&format=xml&cartId=&apiKey=yourAPIKey&signature=beeb5c2f4bae539c37eb5079ff6ae2b4>

WebRequest Status: OK

CartId: 43210

15. DOT NET: Add an Item to a Cart (JSON Format)

Use Case:

The following DOT NET code wraps the Cart.AddItem API call using a WebRequest / WebResponse using JSON, to the MN Open integration environment to obtain a cart id.

Code Listing DOT NET:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using System.Net;
using System.Text;
using System.Xml;
using System.Web.Script.Serialization;
using System.Security.Cryptography;

public partial class _Default : System.Web.UI.Page
{
    JavaScriptSerializer serializer;

    protected void Page_Load(object sender, EventArgs e)
    {
        string apiKey = "YourAPIkey";
        string sharedsecret = "YourSecret";
        string requestUrl = "https://ie-api.mndigital.com?";
        string message = "method=Cart.AddItem&format=json&cartId=&apiKey=" + apiKey;
        System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();
        byte[] keyByte = encoding.GetBytes(sharedsecret);

        HMACMD5 hmacmd5 = new HMACMD5(keyByte);
        byte[] messageBytes = encoding.GetBytes(message);
        byte[] hashmessage = hmacmd5.ComputeHash(messageBytes);
        string thisSignature = ByteToString(hashmessage);
        requestUrl = requestUrl + message + "&signature=" + thisSignature;
        string postData = "{ \"Items\": [{ \"MnetId\": 12296213, \"ItemType\": \"Track\",
        \"Format\": \"MP3\", \"Price\": \"\", \"Tax\": \"\" } , ]}";

        WebRequest request = WebRequest.Create(requestUrl);
        request.Method = "POST";
        byte[] byteArray = Encoding.UTF8.GetBytes(postData);
        request.ContentType = "application/json";
        request.ContentLength = byteArray.Length;
        Stream dataStream = request.GetRequestStream();
        dataStream.Write(byteArray, 0, byteArray.Length);
        dataStream.Close();
        WebResponse response = request.GetResponse();
        Response.Write("Executed: " + requestUrl + "<br /><br />");
        Response.Write("JSON Payload sent: " + postData + "<br /><br />");
        Response.Write("WebRequest Status: " + ((HttpWebResponse)response).StatusDescription +
        "<br /><br />");
        dataStream = response.GetResponseStream();

        StreamReader reader = new StreamReader(dataStream);
        string responseFromServer = reader.ReadToEnd();
        Response.Write(responseFromServer);
        dataStream.Close();
        response.Close();
    }
}
```

```
}  
protected string ByteToString(byte[] buff)  
{  
    string sbinary = "";  
    for (int i = 0; i < buff.Length; i++)  
    {  
        sbinary += buff[i].ToString("X2");  
    }  
    return (sbinary);  
}  
}
```

Output:

Executed: https://ie-api.mndigital.com?method=Cart.AddItem&format=json&cartId=&apiKey=YourAPIkey&signature=570257D5B92B972D38F71248B4E4B4EE

JSON Payload sent: { "Items": [{ "MnetId": 12296213, "ItemType": "Track", "Format": "MP3", "Price": "", "Tax": "" } ,]}

WebRequest Status: OK

{"Success":true,"CartId":"44808","Count":1,"Status":{"DuplicateItemInCart":false,"CartAlbumContainsTrack":false,"CartTrackInAlbum":false}}

MediaNet makes no warranties whatsoever as to the fitness of this or any sample code provided or as to its suitability for any purpose(s), including the purpose for which I/we may claim that it functions. MediaNet includes these sample codes in the hope that it will be useful to you in jumpstarting your development, and in the belief that it is free from serious errors. However, we make no guarantees in either of these regards. In addition to the above disclaimers, these code samples are distributed under the terms of the MediaNet MN Open Terms of Use; see <http://mndigital.com/legal/>.