

## Picture Mosaics Online Mosaic API

v2.1

### Introduction

Welcome to the Online Mosaic Tool (OMT) API. You can use our API to generate custom mosaics for your customers, fans, and/or followers. To gain access to the API you will need to provide us your servers' IP addresses. Once we have your IP addresses we can generate encrypted tokens that will allow you to gain access to our API.

### Authentication

Once we send over the token, access to the API is activated. For successful API calls, the token will need be to sent over with every call. When setting up the framework, it's best to set the token as a globally accessible variable.

### Glossary

**Source** - the main image that is being built/comprised of the smaller photos

**Cell** or **Cells** - the smaller photos that build/comprise the main source image

**Color Overlay** - color filter applied to a cell that correlates to its location in the mosaic

**Source Overlay** - applying the source image on top of the final mosaic

### Requirements

All photos must be in JPEG format, RGB color mode, and saved as extension "jpg".

When transferring the source photo, the image should be downsized to no larger than 1200 pixels on its largest side. Each cell photo should be downsized to 600 pixels on its largest side. This is to maintain low file sizes. We recommend storing a higher resolution version of these photos locally so the end user can have the ability to crop/zoom/edit the photos.

In order to create, access, send, and/or receive data a unique alphanumeric project ID must be created that is different for every mosaic project. When starting a new project for a user, an API call is required in order to generate an unique project ID. This project ID will be required by all API calls.

### API Calls

PHP version 5.3.X calls differ slightly for 5.4+

All API calls will have json\_encoded data returned. The minimal amount a data return will be all 'success' var which will contain a 0 (fail) or 1 (success).

### Get a Project ID:

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXXXX' );
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/getPid.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_errno($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

#### **Expected Return Data**

```
{
  "pid" : "XXXXXXXX",
  "success" : 1
}
```

### Submitting and Updating Source Image:

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXXXX',
                       'pid' => 'XXXXXXXX',
                       'type' => 'jpg',
                       'name' => 'source', // A project can have only 1 source, should always be called source
                       'file' => '@source.jpg' // only send a file that is 1200px on the longest side
    );
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/postSource.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_errno($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

#### **Expected Return Data**

```
{
  "file" : "source.jpg",
  "success" : 1
}
```

### Adding and Updating Cell Image:

```
<?php
```

```

$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                   'pid' => 'XXXXXXXX',
                   'type' => 'jpg',
                   'name' => '0000001', // A project can 1000's of cells this should be unique for each cell
                   'file' => '@0000001.jpg' // only send a file that is 600px on the longest side
                 );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://www.picturemosaics.com/photo-mosaic-tool/api/addCell.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_errno($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
?>

```

**Expected Return Data**

```

{
  "file" : "0000001.jpg",
  "success" : 1
}

```

**Removing Single Cell Image:**

```

<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                       'pid' => 'XXXXXXXX',
                       'name' => '0000001.jpg'
                     );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'https://www.picturemosaics.com/photo-mosaic-tool/api/removeCell.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_errno($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>

```

**Expected Return Data**

**SUCCESS** - file removed

```

{
  "file" : "0000001.jpg",
  "success" : 1
}

```

**FAILURE** - could not file file

```

{
  "file" : "0000001.jpg",
  "success" : 0
}

```

## Removing All Cell Images:

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                        'pid' => 'XXXXXXX'
                    );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/removeAll.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_erro($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

### Expected Return Data

```
{
  "success" : 1
}
```

## Generate Mosaic Preview: (Estimated time 10-45s)

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                        'pid' => 'XXXXXXX',
                        'mosaicSize' => 'medium', // defaults to medium - sizes: small, medium, large, x-large
                        'cellSize' => 'medium', // defaults to medium - sizes: small, medium, large, x-large
                        'cellRatio' => '1:1', // Supported cell ratios width:height - 1:1, 3:2, 4:3, 3:4, 2:3
                        'colorOverlay' => '25', // default 25
                        'sourceOverlay' => '20', // default 20
                    );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/buildPreview.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_erro($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

### Expected Return Data

```
{
  "grid" : "2500", // total grid size columns * rows
  "cellSizeX" : "0.49", // individual cell width size in inches
  "cellSizeY" : "0.49", // individual cell width size in inches
  "printSize" : "24x18", // recommended print size in inches
  "success" : 1
}
```

### Checking Mosaic Preview Progress: (Estimated time 10-45s)

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                    'type' => 'preview'
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/checkProgress.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_erro($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data** (Preview URL will only be return when percentage is at 100%)

```
{
  "percentage" : "XXX",
  "url" : "https://URL-to-Preview-Image",
}
```

### Generate Mosaic Zoom: (Estimated time 1-3 minutes)

Will use settings from latest preview build

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/buildZoom.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_erro($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data**

```
{
  "success" : 1
}
```

### Checking Mosaic Zoom Progress: (Estimated time 1-3 minutes)

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                        'pid' => 'XXXXXXX',
                        'type' => 'zoom'
                    );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/checkProgress.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_errno($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

**Expected Return Data** (Zoom URL will only be return when percentage is at 100%)

```
{
  "percentage" : "XXX",
  "url" : "https://URL-to-Zoom-iframe",
}
```

### Generate Final File: (Estimated time 5-8 minutes)

Will use settings from latest preview build

```
<?php
    $post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                        'pid' => 'XXXXXXX',
                    );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/buildFinal.php');
    curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_POST, 1 );
    curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
    $postResult = curl_exec($ch);
    if (curl_errno($ch)) {
        print curl_error($ch);
    }
    curl_close($ch);
    print $postResult;
?>
```

**Expected Return Data**

```
{
  "success" : 1
}
```

## Checking Mosaic File Progress: (Estimated time 5-8 minutes)

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                    'type' => 'file'
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/checkProgress.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_errno($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data** (File URL will only be return when percentage is at 100%)

```
{
  "percentage" : "XXX",
  "url" : "https://URL-to-Final-Image"
}
```

## Duplicate Project: (Estimated time 10-45 seconds)

Will duplicate what ever pid you send and send back a new pid to reference the duplicate project.

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,'https://www.picturemosaics.com/photo-mosaic-tool/api/duplicateProject.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_errno($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data** (duplicate\_pid = new copy pid of the duplicated project)

```
{
  "success" : 1,
  "duplicate_pid" : "XXXXXXXXX"
}
```

## Checking Duplicate Progress: (Estimated time 10-45s)

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                    'type' => 'duplicate'
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://www.picturemosaics.com/photo-mosaic-tool/api/checkProgress.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_erro($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data** (Preview URL will only be return when percentage is at 100%)

```
{
  "percentage" : "XXX"
}
```

## Deleting Project:

Will delete an entire project, so be careful

<?php

```
$post_data = array( 'token' => 'XXXXXXXXXXXXXXXXXXXX',
                    'pid' => 'XXXXXXX',
                  );

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://www.picturemosaics.com/photo-mosaic-tool/api/deleteProject.php');
curl_setopt($ch, CURLOPT_BINARYTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POST, 1 );
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
$postResult = curl_exec($ch);
if (curl_erro($ch)) {
    print curl_error($ch);
}
curl_close($ch);
print $postResult;
```

?>

**Expected Return Data**

```
{
  "success" : 1
}
```