Overview of the Digimarc Barcode Manager Web Services API

DIGIMARC

Contents

Introduction

The Digimarc Barcode Manager is Digimarc's web portal through which customers create and manage Digimarc Barcodes. This document describes the web services API exposed by the Digimarc Barcode Manager. The Barcode Manager web services API is intended for programmatic access to most of the core features of the Barcode Manager. Applications can use this API to automate the tasks associated with creating a link between media containing a Digimarc Barcode and the related content.

NOTE: Historically, the Digimarc Barcode has been termed a *service* in the Digimarc Barcode Manager. The Barcode Manager API and this document retain this term for backward compatibility.

In order to use and understand the Barcode Manager API it is assumed that you are very familiar with the HTTP protocol and with RESTful web services. The Barcode Manager API is designed generally with a hi-RESTful approach in that it will allow actions on resources by utilizing standard HTTP Verbs. Your choice of development platform/tooling must support access to HTTP Headers and Verbs, Basic Authentication, SSL, and XML or JSON parsing.

Web Services Environments

As mentioned above, the web services API provides programmatic access to most of the features of the Digimarc Barcode Manager. In reality, there are two Barcode Managers:

- The ***Labs*** Barcode Manager enables you to safely develop and test your web services application.
- The ***Live*** Barcode Manager is the real-world environment in which the production version of your web services application operates.

These two versions of the Barcode Manager reside on different servers and require distinct credentials. The following table lists the URLs for the Barcode Manager itself and its web services API for both Live and Labs versions:

Digimarc Discover Platform Base URLs

Environment	Barcode Manager URL	Web Services API Base URL
Live	portal.digimarc.net	api.digimarc.net
Labs	labs-portal.digimarc.net	labs-api.digimarc.net

You will do all your development and testing on the Labs web services API. When your app is ready for production, you will then move to the Live API. The discussions in this document apply equally to both Live and Labs versions.

Web Services Credentials

Your web services app requires credentials to submit requests to the Barcode Manager web services API. You should first obtain credentials through the Labs Barcode Manager. If you try to get credentials through the Live Barcode Manager without having gotten Labs credentials, you will be referred to the Labs Barcode Manager.

To obtain credentials:

1. Log in to the Labs Barcode Manager.

2. Click the **Developers** tab.
3. Click the **Web API** bar.
4. Click **Create New Web API Application**.
5. Enter your application name and click **Create**.

The credentials consist of a unique *Application Name* and *API Key* combination. You will receive an email containing your API Key to use with the Application Name you provided. The email also contains the *Web Services API Base URL* for the Digimarc Barcode Manager web services API. The use of your credentials is discussed in **Authentication** below.

Method Invocation

To invoke a web services method, combine the Base URL with the URL snippet for the method to produce the request URL. For example:

```
Base URL: labs-api.digimarc.net
Method URL snippet: v2/service/12345
Request URL: https://labs-api.digimarc.net/v2/service/12345
```

The `v2` is a legacy URL component that must be included in every request URL.

String parameters passed to the API are trimmed of any leading or trailing whitespace characters. The API prohibits the use of the greater-than (>) and less-than (<) characters in string parameters. When these or other special characters are included in a string parameter passed to a web service method, the method will return status code `400 Bad Request`.

Data returned by the API is not HTML Encoded. For proper display in an HTML application, the data should be HTML Encoded prior to display.

Authentication

Authentication is required to use the Barcode Manager API, and authentication requires the credentials obtained through the Digimarc Barcode Manager. You submit your credentials using HTTP Basic Authentication (encrypted over HTTPS). The format of the header for Basic Authentication is:

```
Authorization: Basic <string>
```

where `<string>` is the base-64-encoding of `<Application Name>:<API Key>`. For example, if your Application Name is `myWebAPI` and your API Key is `6M8+pjLZhgT7Jtpri0ZeAh+d99hG1nMP`, combining these gives:

```
myWebAPI:6M8+pjLZhgT7Jtpri0ZeAh+d99hG1nMP
```

and the base-64 encoding of this combination is:

```
bXlXZWJBUEk6Nk04K3BqTFpoZ1Q3SnRwcmkwWmVBaCtkOTloRzFuTVA=
```

Then your authentication header is:

```
Authorization: Basic bXlXZWJBUEk6Nk04K3BqTFpoZ1Q3SnRwcmkwWmVBaCtkOTloRzFuTVA=
```

You must provide your credentials every time you make an API request. No sessions are used and there is no option for unsecured communication.

For efficient use of the Barcode Manager API, be sure to send your user credentials preemptively. Some HTTP clients omit the authentication header and make a second request with the header when they get a `401`

`Unauthorized` response that includes a WWW-Authenticate header. Send your credentials directly with each request to avoid unnecessary round trips.

Once your application is thoroughly tested and ready for production, you can obtain production credentials from the Live Digimarc Barcode Manager through the steps indicated above.

NOTE: If you would like to experiment with the Barcode Manager web services API methods using a tool like Postman, you can use your personal Barcode Manager credentials — the username and password you use to log in to the Barcode Manager. This will avoid creating unnecessary projects and Digimarc Barcodes in your Live Barcode Manager account.

Request and Response Body Formats

All methods of the Barcode Manager API support both XML and JSON format in the Request and Response bodies. The caller can control the format using standard HTTP Content Negotiation via the `Content-Type` and `Accept` headers. For example consider the following HTTP headers in a request:

```
Content-Type: application/xml
Accept: application/xml
```

This request will result in the method interpreting the request body as XML and cause it to return any response in XML format. JSON format is similarly supported using `application/json`.

When using XML format for requests the order of request parameters MUST match the order specified in that methods request parameters description. Similarly the response order will match the documented order.

All methods also allow the caller to specify the response format on the query string. These examples show the use of the format query parameter to request JSON or XML responses:

```
https://example.digimarc.net/object?format=json
https://example.digimarc.net/object?name=value&format=xml
```

This query parameter is provided as a convenience for testing method calls within a browser. It shouldn't be relied upon when calling the API programmatically because it is not always possible to override the standard HTTP Content Negotiation.

Gzip Encoding the Response Body

All methods of the Barcode Manager API support the http header `Accept-Encoding: gzip` to request that the body of the response be gzip-compressed. When used the API will respond with the body of the response being gzip-compressed and will include the standard http header `Content-Encoding: gzip` to indicate that you must decompress the body. Note that if the returned body size is below a certain threshold (somewhere around 256 bytes) the response will not be compressed, regardless of any `Accept-Encoding` header, and the `Content-Encoding` header will be omitted. Other encoding schemes are not supported.

Date Formats

All dates used in Request and Response bodies must be in ISO 8601 extended format. Where time is specified it is always UTC time — for example: `2011-04-12T13:00:00Z`. As noted, some fields ignore the time if it is included. Also, as noted, some fields allow for a time offset to be specified — for example: `2011-04-12T13:00:00-07:00`. When time offset is positive — for example: `2014-04-12T13:00:00+04:00` — the plus sign (+) must be replaced by its URL encoding (`%2B`): `2014-04-12T13:00:00%2B04:00`.

Common Response Codes

Standard HTTP Status codes and messages will be returned by API methods for success and failure conditions. For failure conditions the message body will include a short textual description of the first error encountered. The following HTTP Status codes can be returned by any of the API methods (along with error codes that vary amongst API methods as described in those methods):

Status Code	Status Message	Description
401	Unauthorized	Caller is not authorized
429	Retry-After xx	Your rate or resource usage limit has been exceeded
500	Internal Server Error	Unexpected internal error - report to Digimarc

Extended Error Information

All API methods support returning additional error information for failure conditions. You can enable extended error information on a method by setting the query parameter:

```
exerror=1
```

Example:

```
https://example.digimarc.net/v2/project/109441?exerror=1
```

When `exerror` is set, failure conditions return a `WebApiErrorResponse` response body with the following members, plus others according to circumstances:

Name	Type	Description
HttpStatus	integer(32)	The HTTP status code
Code	string(128)	An internal error code
CodeDescription	string(256)	Text description of the first error encountered
Occurred	date-time	The date and time the error occurred
Source	string(128)	The error source — e.g. a parameter name or entity

For example:

```
{
  "HttpStatus": 429,
  "Code": "GEN_RateLimitLimitExceeded",
  "CodeDescription": "Rate Limit Exceeded",
  "Occurred": "2017-02-02T00:29:08Z",
  "Source": "/v2/projects/count",
  . . .
}
```

Rate Limiting

The Barcode Manager API enforces usage limits to avoid individual applications degrading the overall user experience. The API tracks usage for short and long intervals and has distinct limits for each associated with every set of credentials. The time intervals used for short and long usage tracking are currently defined as per minute and per hour; however these intervals could change over time as overall usage patterns change. Intervals start with the first incoming request and thus are not tied to a clock-based minute or hour boundary.

Every call to the Barcode Manager API will return the current rate-limiting status in custom HTTP headers of the response. The headers are named `X-RateLimit-Short` and `X-RateLimit-Long`. Both headers use the same format as shown in the following example:

```
X-RateLimit-Short: Limit=300; Remain=245; Expires=45
X-RateLimit-Long: Limit=7500; Remain=7401; Expires=3219
```

where `Limit` indicates the total number of requests allowed in that interval, `Remain` indicates the number of requests remaining in the interval and `Expires` indicates the number of seconds before the interval expires. So in the example of the short interval above you can see there are 300 requests allowed per interval with 245 remaining in the next 45 seconds. Once an interval expires the `Remain` value is reset to the limit.

If you exceed the limit in an interval you will receive `HTTP 429 Retry-After xx`, where `xx` is a number of seconds. The header is followed by a response body containing additional members (beyond the basic `WebApiErrorResponse` response body shown above) that provide more information. As shown in this example, the additional body members are self-explanatory:

```
{
  "HttpStatus": 429,
  "Code": "GEN_RateLimitLimitExceeded",
  "CodeDescription": "Rate Limit Exceeded",
  "Occurred": "2017-02-02T00:29:08Z",
  "Source": "/v2/projects/count",
  "RetryAfter": 55,
  "ShortRateLimitingDurationInSeconds": 60,
  "NumberOfRequestsAllowedDuringShortRateLimitingDuration": 30,
  "RequestsRemainingForShortDurationRateLimiting": 0,
  "SecondsUntilShortRateLimitingResets": 55,
  "LongRateLimitingDurationInSeconds": 3600,
  "NumberOfRequestsDuringLongRateLimitingDuration": 2500,
  "RequestsRemainingForLongDurationRateLimiting": 2469,
  "SecondsUntilLongRateLimitingResets": 3595}
}
```

Additionally each Barcode Manager account has a limit on the total number of services (Digimarc Barcodes) that can be active at any given time. When this limit is reached a `403 Forbidden` error is returned with `Service Limit Exceeded` in the response body. By default this limit is set to an appropriate level based on your expected usage.

This is also a limit on CPU consumption by users. When this limit is exceeded, further calls result in a `HTTP 429 Retry-After xx`, where `xx` is a number of seconds. The header is followed by a response body containing additional members (beyond the basic `WebApiErrorResponse` response body shown above) that provide more information. As shown in this example, the additional body members are self-explanatory:

```
{
  "HttpStatus": 429,
  "Code": "GEN_UsageLimitExceeded",
  "CodeDescription": "Method usage suspended due to resource constraints",
```

```
  "Occurred": "2017-02-02T23:09:09Z",
  "Source": "/v2/services",
  "RetryAfter": 0,
  "MinimumElapsedMsTracked": 201,
  "MsAllowedInPeriod": 200,
  "PeriodLengthInMinutes": 10
}
```


These limits are primarily in place to protect the system from poorly coded applications and to retain the overall usability of the system, not to unduly restrict you as a user. If you feel overly restricted by these limits, please feel free to contact your Digimarc account representative to discuss higher limits.