API Errors

KBPublisher uses conventional HTTP response codes to indicate success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that resulted from the provided information (e.g. a required parameter was missing, etc.), and codes in the 5xx range indicate an error on server, db error or API settings error.

HTTP Status Code Summary

- 200 OK - Everything worked as expected.
- 400 Bad Request - Often missing a required parameter, bad request.
- 401 Unauthorized - No valid API key provided,bad signature.
- 402 Request Failed - Parameters were valid but request failed.
- 404 Not Found - The requested item doesn't exist.
- 500, 502, 503, 504 Server errors - something went wrong on server.

Error Codes

In addition to descriptive error text, error messages contain machine-parseable codes. While the text for an error message may change, the codes will stay the same. The following table describes the codes which may appear when working with the API:

Code	Text	HTTP Code	Description
3	Authentication failed	401	Could not authenticate user.
4	Authorization failed	401	Bad signature, api request could not be authorized.
21	API is available via SSL only	400	Use https to send a request.
22	You cannot access this resource using (%s) request	400	HTTP request methods (POST, PUT or DELETE) is not allowed.
23	Sorry, that page does not exist	400	The specified resource was not found.
24	Sorry, that method does not exist	400	The specified method was not found.
25	Missing or invalid argument(s)	400	Missing a required parameter or invalid parameter.
11	Database error	500	Database error
28	API is not available	503	API is disabled.
29	API is temporarily unavailable	503	API does not respond.
31	Not found	404	The requested item doesn't exist.

If you see an error response which is not listed in the above table, then fall back to the HTTP status code in order to determine the best way to address the error.

Error Messages

When the KBPublisher API returns error messages, it does so in your requested format. For example, an error from a JSON method might look like this:

```
{
   "errors":[
      {
         "errorCode": 25,
         "errorMessage": "Missing or invalid argument(s)",
         "errorInfo": "Required argument(s): timestamp"
      }
   ]
}
```

The corresponding XML response would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
   <error>
      <errorCode>25</errorCode>
      <errorMessage>Missing or invalid argument(s)</errorMessage>
      <errorInfo>Required argument(s): timestamp</errorInfo>
   </error>
</errors>
```

errorInfo field is an optional in error response.