|  |
| --- |
| LyricJAM |
| API Documentation |
| Working Copy |
|  |
| **ss23** |
| **12/29/2010** |

|  |
| --- |
| The document is only a working copy, and as such, doesn’t outline any specific ways of calling the API. For the code examples within, think of them as pseudo XML. |

**Authentication**

Registration of API accounts will be optional for most calls, with calls that require authentication detailed as such (and also returning an error detailing the requirement).

When authentication is needed, a username and an API key will need to be specified. Consider the following example:

<Username>ss23</Username>  
 <APIKey>c3MyM3J1bGVz</APIKey>  
 <Request:AdministrativeCommand>  
 <Input>Foo</Input>  
 </Request>

This will perform the request as the user ‘ss23’, identified by the API Key of ‘c3MyM3J1bGVz’.

**Errors**

A special kind of error that can’t be handled the standard way is one which prevents the API knowing what format the client is requesting data. For example, a request is made with the content “foo”, which tells the API nothing about whether an error should be returned in JSON, XML, or some other language. In such a case, a default will need to be specified, probably JSON. See the following example:

Example Request:

FOO BAR BAZ

Example Response:

<Error>  
 <ErrorCode>201</ErrorCode>  
 <ErrorMessage>The request send to the server was malformed</ErrorMessage>  
 <Input>FOO BAR BAZ</Input>  
 </Error>

Whenever an error occurs (note that this is for fatal errors; warnings and other information can be handled differently), the response will have an error code, an error message (designed for human readability), and the entire request sent to the server (to help developers diagnose errors in the sending of their requests).

Error codes should follow a standard similar to HTTP error codes, where the first digit of the three is a specified to the kind of error it is.

1XX – Server errors e.g. maintenance being performed.  
2XX – Input errors e.g. malformed input (invalid XML/JSON or just garbage input).  
3XX – Authorization errors e.g. authorization required, or invalid password.  
4XX – Request type errors e.g. invalid request, request not implemented.  
5XX – Request specific errors e.g. invalid lyric ID, invalid number of parameters.

**SearchLyrics**

Parameters:

* <string> <required> Artist
* <string> <required> Track
* <string> Album
* <integer> Year
* <string> Track Length
* <integer> Track Number
* <integer> Disk Number

Example Request:

<Request:SearchLyrics>  
 <Artist>Jimmy Eat World</Artist>  
 <Track>The Middle</Track>  
 <Album>Bleed American</Album>  
</Request>

Response Parameters:

* <integer> Certainty
* <uuid> ID
* <string> Lyrics
* <string> Artist
* <string> Track
* <string> <optional> Album
* <integer> <optional> Year
* <string> <optional> Track Length
* <integer> <optional> Track Number
* <integer> <optional> Disk Number

Example Response:

<Response:SearchLyrics>  
 <Certainty>0.95</Certainty>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 <Lyrics>Lyrics go here</Lyrics>  
 <Artist>Jimmy Eat World</Artist>  
 <Track>The Middle</Track>  
 <Album>Bleed American</Album>  
 <Year>2001</Year>  
 <TrackLength>00:02:28</TrackLength>  
 <TrackNumber>3</TrackNumber>  
 </Response>

**GetLyrics**

Parameters:

* <uuid> <required> ID

Example Request

<Request:GetLyrics>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 </Request>

Response Parameters:

* <integer> Certainty
* <uuid> ID
* <string> Lyrics
* <string> Artist
* <string> Track
* <string> <optional> Album
* <integer> <optional> Year
* <string> <optional> Track Length
* <integer> <optional> Track Number
* <integer> <optional> Disk Number

Example Response:

<Response:SearchLyrics>  
 <Certainty>0.95</Certainty>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 <Lyrics>Lyrics go here</Lyrics>  
 <Artist>Jimmy Eat World</Artist>  
 <Track>The Middle</Track>  
 <Album>Bleed American</Album>  
 <Year>2001</Year>  
 <TrackLength>00:02:28</TrackLength>  
 <TrackNumber>3</TrackNumber>  
 </Response>

**ReportLyrics**

Parameters:

* <uuid> <required> ID

Example Request:

<Request:ReportLyrics>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 </Request>

Response Parameters:

* <uuid> ID

Example Request

<Response:ReportLyrics>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>   
</Response>

**UpdateLyrics**

Parameters:

* <uuid> <required> ID
* <string> <required> Lyrics

Example Request:

<Request:UpdateLyrics>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 <Lyrics>These are the correct lyrics silly!</Lyrics>  
 </Request>

Response Parameters:

* <uuid> ID

Example Response:

<Response:UpdateLyrics>  
 <ID>74acd988-12a0-11e0-b7c0-6c42065ab167</ID>  
 </Response>