Limnu

External API v1.4.3

2016/12/19

Table of Contents

Overview

This is the current documentation for the Limnu External API, for third party developers who wish to integrate Limnu into their app. If you have any questions, comments, or feedback for improving this API, please contact **`support@limnu.com`**.

Notes for Early Developers

Thank you for using the Limnu External API! You're part of the early third party developer group because you have a product that we at Limnu believe in, and we value your feedback.

The Limnu External API, the associated payment models, the authentication with the API Keys, and so forth are all in active development. If there's anything that would make this system better for you and your products, please let us know.

In this release, here are some areas in which we're actively seeking your input:

- Do the existing API calls provide you all the functionality/information you need?

- Do you ever need to attach your own metadata to a board?
- The current per-board pricing model ($2 per board, per month, not prorated) is designed for apps which tend to keep boards around for long periods of time (ie: a month or more). If your app tends to have boards which only last for short periods of time, we'd like to talk to you about a new pricing model which fits your needs better.
- Does it make more sense for you to attach the API key to a user or a team?

We plan on moving this document out to a public web page once the primary needs are addressed.

Limnu API and API Key notes

All the functions in the Limnu External API require the use of an API Key issued from Limnu. An API key may be issued to either a team or a user account. Currently, API keys may only be issued by Limnu support manually. In the near future we will be adding the ability to generate an API key from the app.

IMPORTANT: Use of this API with your API key can result in additional charges to your account; (for example, if you are on a per-board payment model and your API key is used to call **`boardCreate`**). As such, we strongly recommend that you do not make your API key available within your exposed clients, and keep it only on your restricted access servers.

If for any reason you ever need to retire your current API key and replace it with a new one, this can easily be done by contacting Limnu support. (This is another feature we plan to expose on the app soon.)

If you have a valid API key, but your account is not in good standing (that is, you are not paying for your account, or you have no trial credit attached to your account), then all Limnu External API functions with your API key will fail.

Limnu External API commonalities

All Limnu External API v1 calls adhere to the following rules:

1. The URL for a function is **`https://api.limnu.com/v1/funcName`**
2. All functions are all of type POST.
3. The body of the POST request is a JSON object.
4. The response to the POST request is a JSON object.
5. All responses will have a boolean **`success`** attribute in the JSON object.
6. If the call fails, **`success`** will be false, and the additional **`reason`** attribute will be included in the JSON object for describing the failure.

Function: userActive

Overview

`userActive` returns a list of boards that a user is currently connected to.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`userId`**: The userId of the user you wish to examine, as issued by userCreate.

Response attribute details: users

`boards` contains all the boards that the user in the input parameters is currently connected to. Each board contains the number of times the user has connected to the board, along with the total number of seconds the user has spent connected to the board (ie: since the page was most recently loaded).

There may be a delay of a few seconds after a user disconnects from a board before the results of userActive is updated.

The data is structured like this:

```
  "boards":{
    "B_B7LyhbF2QGaCV4": {"seconds": 3280 },
    "B_uh86rnYhQFOLuf": {"seconds": 3100 }
  }
```

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "userId": "U_8jksmRr4bqmjPl",
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "boards":{
    "B_B7LyhbF2QGaCV4": {"seconds": 3280 },
    "B_uh86rnYhQFOLuf": {"seconds": 3100 }
  }
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such user: U_8jksmRr4bqmjPl"
}
```

Command Line Testing

Here's an example of how to test **`userActive`** from the command line:

```
curl \
  https://api.limnu.com/v1/userActive \
  --request POST \
  --header "Content-Type: application/json" \
  --data @userActive.json
```

The contents of the file named `userActive.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "userId": "U_8jksmRr4bqmjPl"
}
```

Function: userCreate

Overview

`userCreate` creates a Limnu user, and returns a user ID and authentication token. You should create a single Limnu user for each of your application's users, and store the ID and token in your own user records. The ID and token are required to connect to a board, and (in the future) will be used for things like analytics, access restriction, and so forth.

Besides the **`apiKey`**, the only mandatory parameter is the user's **`displayName`**. Limnu does not require your app to turn over any information which would uniquely identify your users.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`displayName`**: The user's displayed name.

Response attribute details: userId

`userId` is the permanent, unique identifer for the new Limnu user. You should store this in your application user's record.

Response attribute details: token

`token` is the authentication token required as a parameter to the board URL so that the user can access the board (see the boardCreate call for details). You should store this in your application's user record.

Response attribute details: secret

`secret` is a permanent code which will be required for changing the Limnu user's attributes. You should store this in your application's user record. You should *not* allow the user to see this code.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "displayName": "Mr. Grue"
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "userId": "U_wli1TdLjTladq9",
  "token": "eyJ0eXAiOiJKV1QiLCJhbG",
  "userId": "s_xpRO3TbfR7KAbT"
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "Backend error. Please contact support@limnu.com"
}
```

Command Line Testing

Here's an example of how to test **`boardImageURLUpload`** from the command line:

```
curl \
  https://api.limnu.com/v1/userCreate \
  --request POST \
  --header "Content-Type: application/json" \
  --data @userCreate.json
```

The contents of the file named `boardImageURLUpload.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "displayName": "Mr. Grue"
}
```

Function: userUsage

Overview

`userUsage` returns a list of boards that have been used by a user, and how much time the user has spent on each board.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`userId`**: The userId of the user you wish to examine, as issued by userCreate.

Response attribute details: users

`boards` contains all the boards that have been used by the user in the input parameters. Each board contains the number of times the user has connected to the board, along with the total number of seconds the user has spent connected to the board. The data is structured like this:

```
  "boards":{
    "B_B7LyhbF2QGaCV4":{"connectionCount":2,"secondsUsed":3280},
    "B_uh86rnYhQFOLuf":{"connectionCount":3,"secondsUsed":3100}
  }
```

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "userId": "U_8jksmRr4bqmjPl",
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "boards":{
    "B_B7LyhbF2QGaCV4":{"connectionCount":2,"secondsUsed":3280},
    "B_uh86rnYhQFOLuf":{"connectionCount":3,"secondsUsed":3100}
  }
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such user: U_8jksmRr4bqmjPl"
}
```

Command Line Testing

Here's an example of how to test **userUsage** from the command line:

```
curl \
  https://api.limnu.com/v1/userUsage \
  --request POST \
  --header "Content-Type: application/json" \
  --data @userUsage.json
```

The contents of the file named `userUsage.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "userId": "U_8jksmRr4bqmjPl"


}
```

Function: boardActive

Overview

boardActive returns a list of users that are currently connected to a given board.

Parameters

Input parameters:

1. **apiKey**: Private API key issued by Limnu to developers.
2. **boardId**: The boardId of the board you wish to examine, as issued by boardCreate.

Response attribute details: users

users contains the list of all users that are currently connected to the board in the input parameters. Each user record contains the number of seconds the user has been connected to the board (ie: since the last time the page was loaded).

There may be a delay of a few seconds after a user disconnects from a board before the results of boardActive are updated.

The data is structured like this:

```
  "users":{
    "U_8jksmRr4bqmjPl":{"seconds":3280},
    "U_ARSfi7vp5pxSCC":{"seconds":3100}
  }
```

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "users":{
    "U_8jksmRr4bqmjPl":{"seconds":3280},
    "U_ARSfi7vp5pxSCC":{"seconds":3100}
  }
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **`boardActive`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardActive \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardActive.json
```

The contents of the file named `boardActive.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4"
}
```

Function: boardCreate

Overview

boardCreate will create a new Limnu board, owned by your account.

The board will be a “shared link” board, meaning that anyone who has the URL for the board may access it. This is to reduce friction for your users; they are not required to create a Limnu account to access the board.

If your account is on a payment model that involves paying per board, use of this function will *increase* the amount you are charged.

Parameters

Input parameters:

1. **apiKey**: API key issued by Limnu to developers.
2. **boardName**: The user-visible name for the new Limnu board. The board name does not need to be unique.
3. **disallowSharing**: (Optional, defaults to false). If true, the “Board Share” menu will not be available from the menu on the board itself, and will also not be available on the board’s row on the board list page.
4. **whiteLabel**: (Optional, defaults to false). If true, all mentions of Limnu will be removed from the board, as well mentions of accounts, subscriptions, and so forth. The entire hamburger menu in the bottom left corner of the board will be removed, as all the actions in that menu are meant for Limnu subscribers. **IMPORTANT:** Using this feature will result in an increase in the of the boards. For details, please contact **support@limnu.com.**

In addition to the common output values, a successful **boardCreate** response will also include:

1. **boardId**: The unique identifier for the new board.
2. **boardUrl**: Contains the browser URL to get to the new Limnu board.
3. **claimUrl**: Contains the browser URL for a Limnu user to be able to claim ownership of the new board.

Response attribute details: boardId

boardId is the permanent, unique identifer for the new board. It is used in the URL for accessing the board, and it is also used by other API functions to manipulate an existing board.

Response attribute details: boardUrl

boardUrl is used to navigate to the board in a window, tab, iframe, or other webview. It may be used by an unlimited number of users, unless someone claims the board and makes it private.

boardUrl will look like this:

```
https://limnu.com/d/draw.html?b=B_Ezl0gvE7QoqFz1&
```

If the **boardUrl** is used as-is, any users who click on it who are not currently authenticated with Limnu will have to go through the normal login-or-signup flow. However, it is possible to skip this flow (thereby reducing friction for the user, letting users preserve identity between boards, and allow Limnu to measure usage on your boards) by attaching the **t** (for “token”) URL parameter with the token generated by userCreate, as shown in **orange** below:

```
https://limnu.com/d/draw.html?b=B_Ezl0gvE7QoqFz1&t=eyJ0eXAiOiJKV1QiLCJhbGG&
```

If a user is not already authenticated with Limnu, a new Limnu guest user will be created with that name. This way, the user’s name from the third party app can be used in Limnu without any user effort.

Response attribute details: claimUrl

The **claimUrl** allows a user with a Limnu account to claim ownership of a new board. The user will become the first admin on the board, and subsequently be able to grant admin controls to other users.

Use of **claimUrl** is optional. However, it may only be used once, and only by someone with a Limnu account. If a user who does not have a Limnu account navigates to **claimUrl**, they will be prompted to sign up for an account before claiming the board.

A **claimUrl** will look like this:

```
https://limnu.com/d/draw.html?b=B_Ezl0gvE7QoqFz1&bc=N_v2wioJCRfeBQki&
```

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardName": "My first API-created Limnu board!"
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "boardId": "B_B7LyhbF2QGaCV4",
  "boardUrl": "https://limnu.com/d/draw.html?b=B_B7LyhbF2QGaCV4&",
  "claimUrl": "https://limnu.com/d/draw.html?b=B_B7LyhbF2QGaCV4&bc=N_ESsI8D0tRFaMMO&"
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "Malformed URL"
}
```

Command Line Testing

Here’s an example of how to test **`boardCreate`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardCreate \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardCreate.json
```

The contents of the data file named `boardCreate.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardName": "My first API-created board!"
}
```

Function: boardDelete

Overview

`boardDelete` will delete the Limnu board specified in the input parameters, provided that board is owned by your account.

If your account is on a payment model that involves paying per board, use of this function will prevent any further usage from accruing for the board in question.

Note that this will not delete the activity history for the board as reported by **`boardUsage`** and **`userUsage`**. That data will be preserved as it is how we bill usage.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`boardId`**: The boardId of the board you wish to delete, as issued by **`boardCreate`**.
3. **`redirectUrl`**: (Optional). If this value is true, users actively connected to the board when **`boardDelete`** is called will be redirected to **`redirectUrl`**.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4"
}
```

Example of JSON output from a successful call:

```
{
  "success": true
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **`boardDelete`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardDelete \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardDelete.json
```

The contents of the data file named `boardDelete.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4"
}
```

Function: boardList

Overview

`boardList` lists all boards owned by your account . If you have used **`boardCreate`** to delete some of your boards, those boards will not be returned by **`boardList`**.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.

In addition to the common output values, a successful **`boardCreate`** response will also include:

1. **`boardList`**: Contains a list of boards owned by the account associated with the API Key.

Response attribute details: boardList

boardList contains the list of boards owned by this account which have not been deleted (eg: with a function like **boardDelete**). The list is structured like this:

```
  [ { "boardId": "B_B7LyhbF2QGaCV4",
      "createdAt": 1464553780,
      "boardName": "My first API-created Limnu board!" },
    … with more board objects as needed.
  ]
```

All boards owned by the account associated with the API Key will be listed. If a board has been deleted (eg: through a function like **boardDelete**), the board will not be listed. You can think of this function as listing only the boards you are currently being charged for.

The **boardId** field matches the boardId returned from the **boardCreate** function.

The **createdAt** field is the time (in UTC seconds) when the board was created.

The **boardName** field contains the name given to the board when it was created.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A"
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "boardList": [
    { "boardId": "B_B7LyhbF2QGaCV4",
      "createdAt": 1464553780,
      "boardName": "My first API-created Limnu board!" },
    { "boardId": "B_FGe9olQ9Sa2c8r",
      "createdAt": 1466886636,
      "boardName": "Another board I made" }
  ]
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "Bad API key"
}
```

Command Line Testing

Here's an example of how to test **`boardList`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardList \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardList.json
```

The contents of the file named `boardList.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A"
}
```

Function: boardRename

Overview

`boardRename` changes the name associated with an existing board.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`boardId`**: The boardId of the board you wish to rename, as issued by boardCreate.
3. **`boardName`**: The new name for the board. It does not need to be unique.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "boardName": "A renamed board"
}
```

Example of JSON output from a successful call:

```
{
  "success": true
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **boardRename** from the command line:

```
curl \
  https://api.limnu.com/v1/boardRename \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardRename.json
```

The contents of the file named `boardRename.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "boardName": "A renamed board"
}
```

Function: boardUsage

Overview

boardUsage returns a list of users that have used a board, and how much time they’ve spent on the board.

If you are on a payment model that is calculated per-board, the data that this function samples from is what Limnu uses to calculate your costs. If a board is used across payment boundaries and as such has partially billing in multiple months, that is not represented in the output of this function.

Parameters

Input parameters:

1. **apiKey**: Private API key issued by Limnu to developers.
2. **boardId**: The boardId of the board you wish to examine, as issued by boardCreate.

Response attribute details: users

users contains the list of all users that have used the board in the input parameters. Each user contains the number of times the user has connected to the board, along with the total number of seconds the user has spent connected to the board. The data is structured like this:

```
  "users":{
    "U_8jksmRr4bqmjPl":{"connectionCount":2,"secondsUsed":3280},
    "U_ARSfi7vp5pxSCC":{"connectionCount":3,"secondsUsed":3100}
  }
```

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "users":{
    "U_8jksmRr4bqmjPl":{"connectionCount":2,"secondsUsed":3280},
    "U_ARSfi7vp5pxSCC":{"connectionCount":3,"secondsUsed":3100}
  }
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **`boardUsage`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardUsage \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardUsage.json
```

The contents of the file named `boardUsage.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4"
}
```

Function: boardImageURLUpload

Overview

`boardImageURLUpload` retrieves an image from a URL and places it on the board. The image is placed at the “origin”; that is, the location on the board where a user is placed the first time they visit the board. Like all Limnu assets, anyone can move or resize the image once it is on the board.

Parameters

Input parameters:

1. **apiKey**: Private API key issued by Limnu to developers.
2. **boardId**: The boardId of the board you wish to upload into, as issued by boardCreate.
3. **imageURL**: The URL for an web-available image.

Call attribute details: imageURL

It is important that the URL you provide to this function actually goes to an image, and not a full web page. Many image servers and image search engines provide URLs that take you to a web page instead of the bare image. Passing in a URL that does not point to a bare image is the most common source of failures for this function.

It's also important that the URL you provide be publicly accessible on the internet. If the image is only available within your own network, the call will fail. If the image can only be accessed after logging into a service, the call will fail. Passing in a URL that not publicly accessible on the internet is the second most common source of failures for this function.

Limnu currently supports JPEG, PNG, and GIF formats. (GIFs containing animations will only show a single frame on a Limnu board.) If you would like Limnu to support other formats, please contact **support@limnu.com**.

Response attribute details: fileId

fileId is the permanent, unique identifer for the new image.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "imageURL": "http://www.brycecanyon.org/wp-content/uploads/2016/01/main-01d.jpg"
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "fileId": "I_X7LD9O4CTGFHN2fbDUsG"
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **`boardImageURLUpload`** from the command line:

```
curl \
  https://api.limnu.com/v1/boardImageURLUpload \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardImageURLUpload.json
```

The contents of the file named `boardImageURLUpload.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "imageURL": "http://www.brycecanyon.org/wp-content/uploads/2016/01/main-01d.jpg"
}
```

Function: boardImageFileUpload

Overview

`boardImageURLUpload` allows for direct upload of images files (as opposed to boardImageURLUpload above) to a board. The image is placed at the “origin”; that is, the location on the board where a user is placed the first time they visit the board. Like all Limnu assets, anyone can move or resize the image once it is on the board.

Limnu currently supports JPEG, PNG, and GIF formats. (GIFs containing animations will only show a single frame on a Limnu board.) If you would like Limnu to support other formats, please contact **`support@limnu.com`**.

Parameters

Input parameters:

1. **`apiKey`**: Private API key issued by Limnu to developers.
2. **`boardId`**: The boardId of the board you wish to upload into, as issued by boardCreate.
3. **`contentType`**: The content type of the image.
4. **`data_base64`**: The image data itself, encoded in base64 format.

Call attribute details: contentType

This parameter should contain the content type string that matches the file you are uploading. If your **`contentType`** parameter does not match the actual file, this call will fail.

Call attribute details: data_base64

Your binary image must be converted to the JSON-friendly “Base 64” format. To convert your image, you can use the **base64** commmand line program on MacOS, the **base64** command line program on Linux, the **base64** library in Python, or any of a myriad of other tools many of which are listed on the Base64 Wikipedia page.

The end result will be a file approximately 33% larger than your original image. This entire file should be put into the value for the **data_base64** key in the API call. This means that your API call will be large, and depending on the speed of your connection to the net, it may take some time to complete.

As a result, if your image passes the requirements of **boardImageURLUpload** above, it’s going to be faster and easier to use that call instead.

Response attribute details: fileId

fileId is the permanent, unique identifer for the new image.

Example data

Example of correct JSON input:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "contentType": "image/png",
  "data_base64": "iVBORw0KGgoAAAANSUhEUgAAACQAAAAkCAIAAABuYg/PAAAACXBIWXMAAAs..."
}
```

Example of JSON output from a successful call:

```
{
  "success": true,
  "fileId": "I_X7LD9O4CTGFHN2fbDUsG"
}
```

Example of JSON output from a failed call:

```
{
  "success": false,
  "reason": "No such board: B_B7LyhbF2QGaCV4"
}
```

Command Line Testing

Here’s an example of how to test **boardImageFileUpload** from the command line:

```
curl \
  https://api.limnu.com/v1/boardImageFileUpload \
  --request POST \
  --header "Content-Type: application/json" \
  --data @boardImageFileUpload.json
```

The contents of the file named `boardImageFileUpload.json` are:

```
{
  "apiKey": "K_isR4E6qcQamY0NxFuKa9aw7k5hssRBQXuqpStgR1qU1A",
  "boardId": "B_B7LyhbF2QGaCV4",
  "imageURL": "http://www.brycecanyon.org/wp-content/uploads/2016/01/main-01d.jpg"
}
```

Changelog

v1.0.0 - 2016/06/26

- Initial version.

v1.0.1 - 2016/06/26

- Formatting and minor changes based on feedback.

v1.1.0 - 2016/09/28

- Added: boardImageURLUpload
- Added: boardImageFileUpload
- Changed all ‘curl’ examples to use a file for the parameter data instead of in JSON on the command line, which has all kinds of problems.
- Added this Changelog.

v1.2.0 - 2016/10/03

- Added: userCreate
- Updated boardCreate to use tokens from userCreate

v1.3.0 - 2016/10/17

- Added: userUsage
- Added: boardUsage
- Minor corrections to existing documentation

v1.4.0 - 2016/11/10

- Added: userActive
- Added: boardActive
- Minor corrections to existing documentation

v1.4.1 - 2016/12/15

- Added optional parameter ‘disallowSharing’ to boardCreate.

v1.4.2 - 2016/12/19

- Yeah, the ‘disallowSharing’ parameter is supposed to be on boardCreate, not on boardActive. For real this time.

v1.4.3 - 2017/02/14

- Added ‘whiteLabel’ parameter to boardCreate.
- Added ‘urlRedirect’ parameter to boardDelete.
- Clarified board cost calculation in boardUsage and boardDelete.