



cloudmine

Connected Health Cloud Platform Walkthrough

Copyright 2018 CloudMine, Inc. All rights reserved.

All brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

CloudMine LLC.
1217 Sansom St
FL 6
Philadelphia, PA 19107

Version 1.0

Table of Contents

1. Introduction
2. Document Purpose
3. CloudMine Applications
 1. Creating an Application
 2. Manually Creating Data with the CloudMine Connected Health Cloud
 3. Manually Searching for Objects in the Connected Health Cloud
4. CloudMine REST API
 1. Introduction to the CloudMine REST API
 2. Create API
 3. Fetch API
 4. Update API
5. Access Control & Security
 1. User-Level Security & ACL Maps
 2. App-Level Security and API Keys
 3. The `__class__` attribute
6. Creating your First Server-Side JavaScript Snippet
 1. Introduction
 2. Creating your Snippet
 3. Executing Snippets
 4. Providing Inputs to Snippets
7. Developer Resources & Documentation
 1. Online Documentation
 2. GitHub Resources
 3. Engineers
8. Appendix A: Getting Started with Postman
 1. Building a Request
 2. Evaluating the response
9. Appendix B: Sample Curl Calls

1. Introduction

CloudMine's Connected Health Cloud is comprised of a set of APIs (Application Programming Interfaces), SDKs (Software Development Kits), and third-party integration services that developers may use to enhance web and mobile apps with pre-made, backend services such as identity management, data storage, and push notifications. These services are accessed via REST APIs and client-side SDKs. The CloudMine backend is extendable with JavaScript and Java. CloudMine's Connected Health Cloud can integrate where necessary with existing relational databases and other enterprise applications.

2. Document Purpose

The purpose of this guide is to give your technical team members a working knowledge of the following topics:

- Creating Applications on the CloudMine platform: **Connected Health Cloud**
- Creating, updating, and fetching data via the CloudMine REST API
- Building and running server-side JavaScript Snippets
- Additional documentation and engineering resources

3. CloudMine Applications

Prerequisites for working with CloudMine Applications:

- ✓ A Connected Health Cloud organization; if you do not have an org, please reach out to support@cloudmineinc.com to get started!
- ✓ A fundamental understanding of REST APIs
- ✓ A fundamental understanding of JSON and JavaScript
- ✓ A REST client
 - CloudMine recommends Postman, available for free as a Chrome extension. For more information, see <https://www.getpostman.com/> and Appendix A below.
 - For developers familiar with curl, Appendix B includes examples for each API call described below.

An Introduction to your Connected Health Cloud Organization

A CloudMine Connected Health Cloud Organization is your dedicated HIPAA-compliant development environment. Whether you are creating applications, building hosted sites or leveraging secure environments for integration logic, organizations will house all the necessary pieces to successfully achieve your digital health goals. When an organization is created, you will define your organization owner. The org owner has the ability to invite developers, create teams and define access to all developers within your Connected Health Cloud.

An Introduction to CloudMine Applications

An Application or “App” is the foundational concept in the CloudMine platform and each App encapsulates the data stored inside of it. CloudMine allows developers to create any number of Apps. Each App is assigned a unique **Application Identifier (App ID)** that differentiates it from the rest.

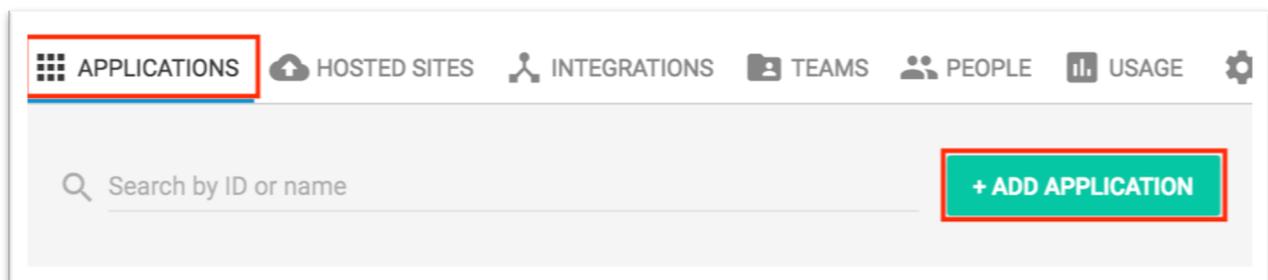
There are two levels of data access in CloudMine. An object can have *App-level* data, which means anyone with the App ID and API Key can access it. An object can also have *User-level* data, which means that only the logged-in user can access it.

Data stored in a particular CloudMine Application, at both the App and User levels, are separated from data in other Apps. Within an individual Application, however, it is possible to set permission levels to give users a range of access to data-objects in the App.

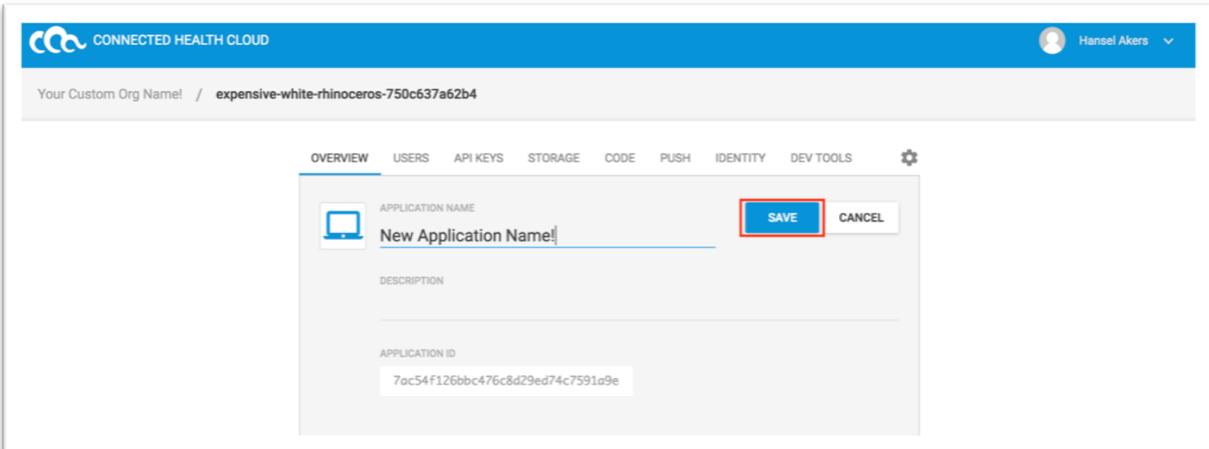
1. Creating an Application

To get started in the Connected Health Cloud, create an application.

1. **Navigate** to <https://chc.cloudmine.io/> and enter your username and password credentials to view your Connected Health Cloud organization.
2. After navigating to the 'Applications' tab, click the 'Add Application' button.



3. On the Application Overview page, name the App and click **Save**.

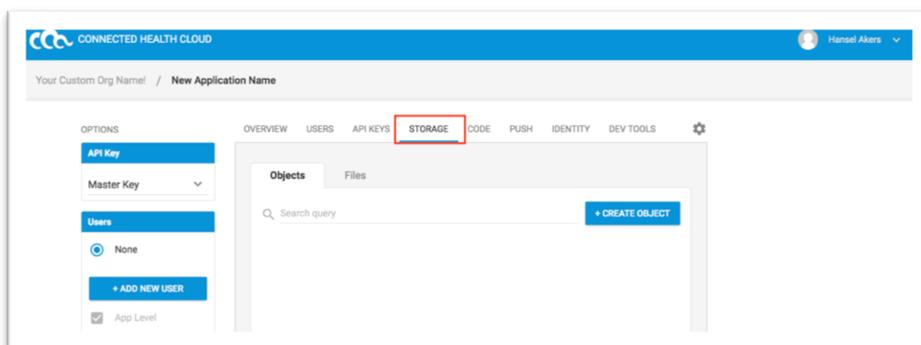


Congratulations on creating your first CloudMine application!

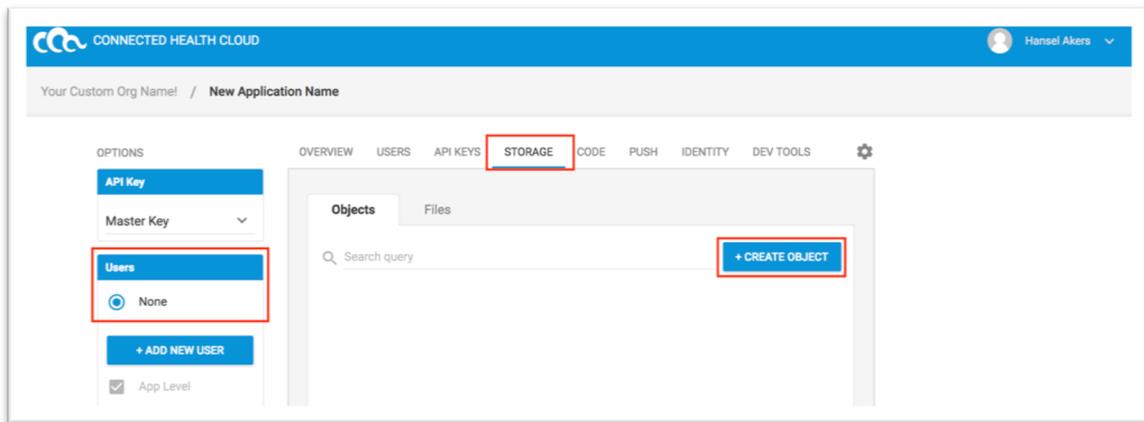
2. Manually Creating Data in the CHC Dashboard

Once an application has been created, populate the application with data. There are different ways to enter data into a CloudMine application, but the easiest way to get started is to create an object with the CHC dashboard.

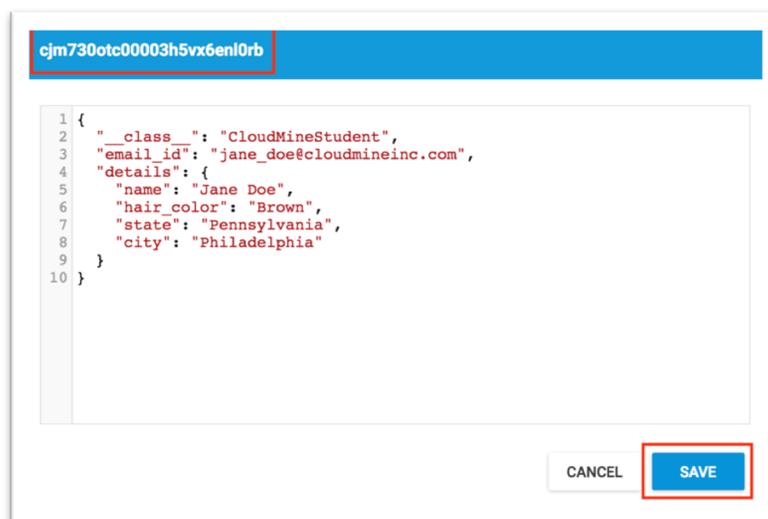
1. From the **Applications tab** in the main nav-bar, select the appropriate app.
2. Click the **Storage tab** in the application nav bar.



3. Verify **None** is selected in the **Users panel** on the left side of the dashboard. This indicates that the object is being created at the application level. Click the **Create Object** button.

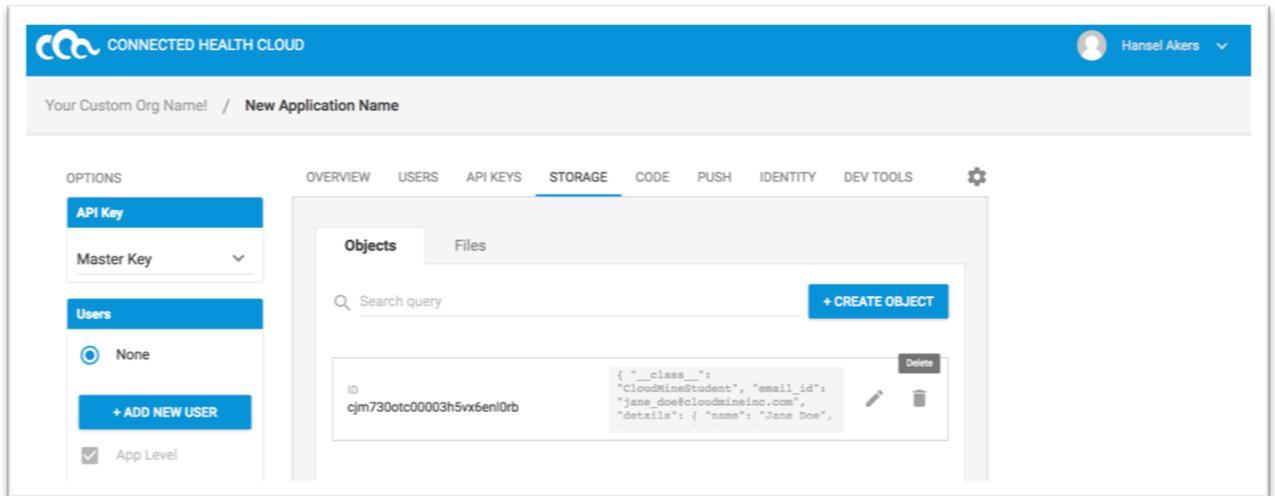


4. The new object has been assigned a unique ID, which is displayed at the top of the page. Use the object editor to enter the JSON body of the object.



Click **Save**. If the JSON is valid, you will be redirected to the object list view. If the JSON is invalid, an error message will appear.

5. To verify that the object was successfully saved, confirm the object is listed in the Object **Storage browser**.



Congratulations on creating your first CloudMine object!

3. Manually Searching for Objects in the Connected Health Cloud

Once data has been manually created, use the dashboard to search for and retrieve specific data.

1. Click on **Applications** in the main nav-bar and select the appropriate application.
2. Click on the **Storage tab** from the top nav-bar.
3. Enter query terms in the **Object query** input box and hit enter.
 - The search query depends on the JSON object that was entered into the CloudMine app. A basic search query for a field in the top-layer of the object might look like this:

```
`email_id = "jane_doe@cloudmineinc.com"``
```

- Fields in nested objects are accessed using dot-notation:

```
`outerField.innerField[name = "Jane"]`
```

- CloudMine supports Regular Expressions in the search query string. This is accomplished by enclosing a Regular Expression in slashes:

```
`name = /Jane/`
```

Way to go! You have successfully completed your first CloudMine search!

4. CloudMine REST API

In addition to providing manual data entry options through the Connected Health Cloud, CloudMine also features a rich REST API, as well as fully-functional iOS, Android, and Node.js SDKs. Use the REST API as a starting point to learn how to programmatically manage data entry and collection on CloudMine.

This guide presents all REST API examples and instructions using **Postman** (<https://www.getpostman.com>), a free Google Chrome extension. For more in-depth Getting Started assistance with Postman, please review Appendix A. CloudMine conforms to all standard REST protocols, so any REST client can be used complete the exercises in this guide. See Appendix B for curl examples for each request.

1. An Introduction to the **CloudMine REST API**

CloudMine authenticates API calls for App-level data using a combination of **App ID** and **API Key**. These two pieces of information are required for all REST calls on CloudMine's platform. User-level data, discussed below, will require a user's **Session Token** for authentication.

This guide uses colors to represent different aspects of the API calls:

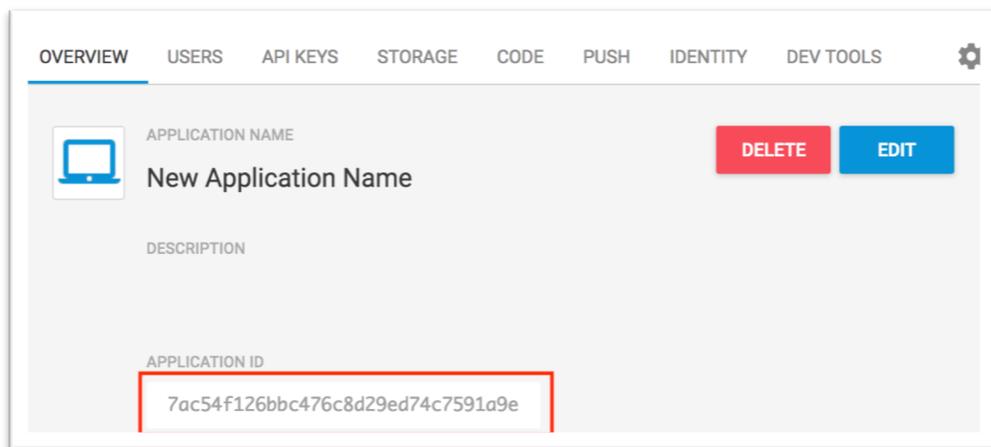
— **Blue text** indicates that the value needed to complete the API calls are available in the CloudMine Connected Health Cloud dashboard or via a different API call.

— **Orange text** indicates that the requisite details need to be provided by the developer making the API request.

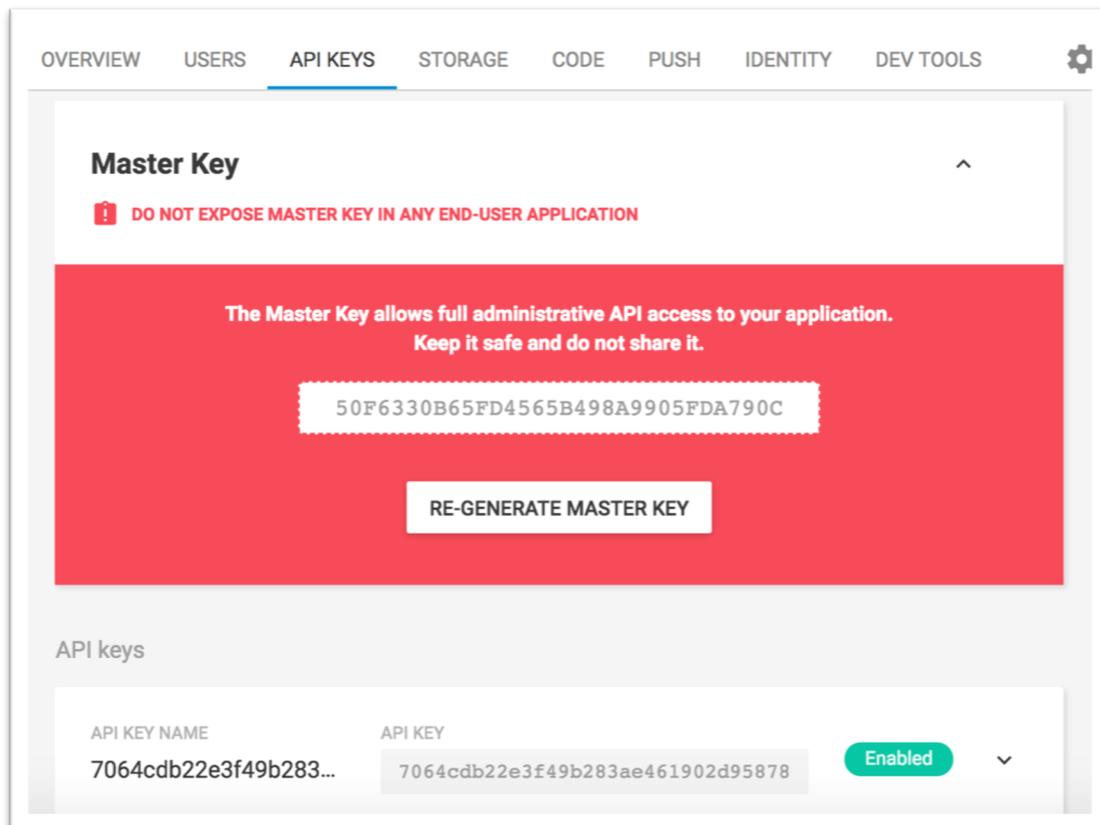
2. **Create API**

The Create API allows programmers to make new objects in their CloudMine applications.

1. An App ID and API Key will be necessary before making a request. To find the App ID, log into the Connected Health Cloud dashboard and click on **Applications** on the main nav-bar.
2. Click on the respective application that you will be using.
3. From the **Application overview**, make note of the **App ID**.



- Next, click on the **API Keys** link in the main nav-bar. There is an option to click the **Master Key** drop-down, **which has *all* available permissions on the platform for the specific application you are in**, or use the unnamed API Key that is automatically generated when an application is created. Take note of one of these keys.



Note: The **Master Key** is the equivalent of 'root' level access to a CloudMine application. If it is ever compromised, a new **Master Key** can be generated in the CloudMine app by clicking **Regenerate Key**.

- Open Postman in order to build the **Object Create** request using the following details:

- Request type

```
`PUT`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`
```

- Request body (raw type JSON)

```
`{  
  "student_123456": {  
    "__class__": "CloudMineStudent",  
    "email_id":  
    "jane_doe@cloudmineinc.com",  
    "details": {  
      "name": "Jane Doe",  
      "hairColor": "brown",  
      "state": "Pennsylvania",  
      "city": "Philadelphia"  
    }  
  }  
}`
```

- Expected Response body

```
`{  
  "success": {  
    "student_123456": "created"  
  },  
  "errors": {}  
}`
```

Note: It is important to provide a unique, top-level key for the objects created in CloudMine, such as 'student_123456' in this example. This key will effectively operate as the object's ID.

Congratulations on creating a new object with the REST API!

3. Fetch API

The **Fetch** API enables developers to retrieve data that has already been created in a CloudMine application. For this example, we will retrieve the data created in the previous steps.

1. The object Fetch request is built with the following details:

- Request type

```
`GET`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request parameters

```
`keys: student_123456`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`
```

2. Once the details are configured in the REST client, send or submit the request. The response should look something like the following:

```
`{ "success":  
  {"student_123456":  
    { "__class__": "CloudMineStudent",  
      "email_id": "JaneDoe@CloudMine.com",  
      "details": {  
        "name": "Jane Doe",  
        "hair_color": "brown",  
        "state": "Pennsylvania",  
        "city": "Philadelphia" }  
      }  
    },  
    "errors": {}  
  }`
```

4. Update API

The **Update** API is used to change an existing piece of data on the CloudMine platform.

1. To update an object, configure the client with the following:

- Request type

```
`POST`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`
```

- Request body

```
`{ "student_123456":  
  { "details":  
    { "company": "CloudMine" }  
  }  
}`
```

2. Once the REST client is configured with these details, send or submit the request. A successful response should look similar to the following:

```
`{ "success":  
    {"student_123456": "updated"},  
    "errors":{}}  
}`
```

Great! You have updated a CloudMine object!

Note: During an update call, CloudMine will automatically create any key-value pair that does not already exist in an object. If the top-level key (represented here by “student_123456”) does not already exist in the CloudMine application, a new object will be created with that key.

5. Access Control & Security

CloudMine employs two different methods for securing data: 1) user-level Access Control Lists (ACLs), and 2) application-level API keys. At the user-level, ACLs can be set up to grant a range of access levels and might define:

- Users with access to a specific object or objects
- The type of access allowed (read, write, update, delete) to a given object
- Whether anonymous or unauthenticated requests for objects are permitted.

CloudMine's recommendation is to ensure that ACLs grant as few permissions as possible.

1. User-Level Security & ACL Maps

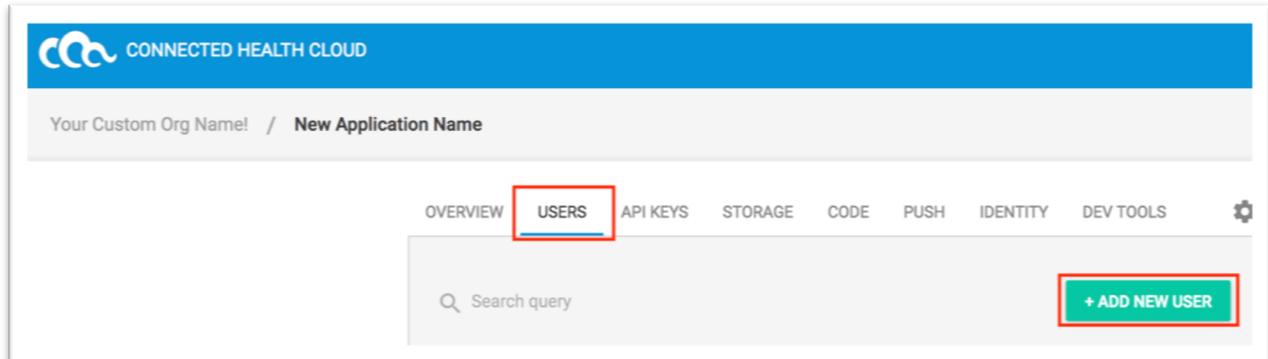
To help demonstrate the power and use of ACLs, imagine a basic app that allows employees to create a profile and describe their personal healthy habits. In this app, you would want to store a HealthyHabits attribute for each user. To encourage some *healthy* competition, however, you also might want users to be able to share their HealthyHabits activities with other users. To accomplish this requirement, we can create ACLs to define which users in the organization have rights to view another's HealthyHabits data. Since this scenario requires two different users, you first need to create two user accounts (Mary and John in this example). It is possible to create these users directly via the API, however, this example will demonstrate creating the users via the CloudMine application.

For the purposes of this exercise, let's say that Mary wants to share her HealthyHabits with her colleague, John.

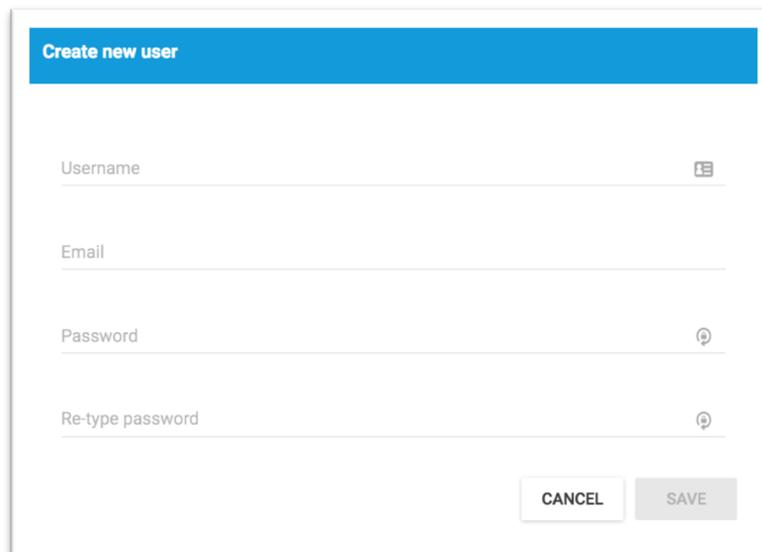
Since this process will make use of a few different APIs, review the high-level overview of the user/developer stories, the required APIs, and the context with which the API needs to be invoked in the following chart:

Step	Developer Story	User Story	User Context	API	Request Type	Prerequisite Data
1	Mary's SessionToken is needed for her to log in and create a new HealthyHabits object	User logs into the App	Mary	User Log In	POST	X-CloudMine-ApiKey Mary's authorization
2	Create a new HealthyHabits object for Mary	User submits HealthyHabits	Mary	Create User Object	PUT	X-CloudMine-ApiKey X-CloudMine-SessionToken
3	Locate John's User Id	User searches for colleague	John	User Search	GET	X-CloudMine-ApiKey
4	Create ACL with shared user Id	User shares healthy habits	Mary	Create Object Update Object	POST	X-CloudMine-ApiKey X-CloudMine-SessionToken
5	Verify that Mary's HealthyHabits are shared with John	Verify Shared Users' Access	John	Fetch Object	GET	

1. First, log into the CloudMine Connected Health Cloud dashboard.
2. Once authenticated, navigate to the intended **application**, and click on the **Users tab** in the main nav-bar.

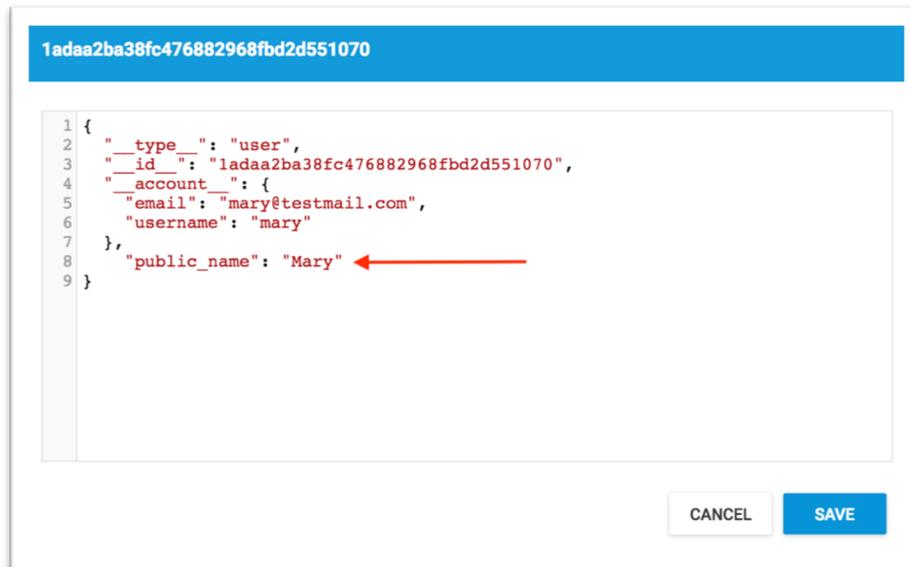


3. To create a new user click **+Add New User**. User creation requires a username, email, password and a password confirmation field. Be sure to note the passwords used for each sample user.

A screenshot of the 'Create new user' form. The form has a blue header with the text 'Create new user'. Below the header are four input fields: 'Username', 'Email', 'Password', and 'Re-type password'. Each field has a small icon to its right: a person icon for Username, an envelope icon for Email, and a key icon for Password and Re-type password. At the bottom right of the form are two buttons: 'CANCEL' and 'SAVE'.

4. After each user is created, double-click on a user to edit each in the **Users** screen. When a user is created in the Connected Health Cloud, the ``email`` and ``username`` are stored in the ``__account__`` hash. Since

this information is often sensitive, data stored in the `__account__` hash map is not exposed to the public API by default and cannot be searched. In order to search for a user later in this exercise, edit the user object to add something that you can search for. After the `__account__` hash, add a key and value `"public_name": "Mary"` and click **Save**. Each user should look something like this when created:



```
1 {
2   "__type__": "user",
3   "__id__": "1adaa2ba38fc476882968fbd2d551070",
4   "__account__": {
5     "email": "mary@testmail.com",
6     "username": "mary"
7   },
8   "public_name": "Mary" ←
9 }
```

CANCEL SAVE

5. Next Mary will need to be authenticated against CloudMine in order to get her **Session Token**.

- Request type

```
`POST`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/account/login`
```

- Request headers

```
`Content-Type: application/json`  
`Authorization: Basic bWFyeTpwYXNzd29yZDE=\n`  
`X-CloudMine-ApiKey: {your_api_key}`
```

Note: the encoded text in the authorization header is a Base64 representation of `{username}:{password}` or `{email}:{password}`. You can get a Base64 representation of the username and password by using [this tool](#).

- Successful response body:

```
`{  
  "session_token":  
  "9f26ec65c0ea481fb498c755ae9f0f20",  
  "expires": "Mon, 19 Mar 2018 15:12:30 GMT",  
  "profile": {"__type__": "user",  
              "__id__":  
              "fda23bb33c724fe4a74322a1f0443959"  
            }  
}`
```

Note: record the session_token as it will be required later.

6. If Mary wants to submit her HealthyHabits data, create a `HeathyHabits` object that will be associated with/belong to her user.

- Request type

```
`PUT`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/user/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-SessionToken: {marys_session_token}`  
`X-CloudMine-ApiKey: {your_api_key}`
```

- Request body

```
`{  
  "HealthyHabits_12345": {  
    "activities": ["running", "lifting  
weights", "cycling"]  
  }  
}`
```

- Successful response body

```
`{"success":  
  "HealthyHabits_12345": "created",  
  "errors": {}  
}`
```

7. Now search for John and find his **User ID** so that it can be included in the ACL created for Mary. As mentioned above, data in the `__account__` hash are not exposed to the API and cannot be searched, so it will need to be searched.

- Request type

```
`GET`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/account/  
search`
```

- Request parameters

```
`p: [public_name=/john/i]`
```

Note: the query parameter is a Regular Expression

- Request headers

```
`X-CloudMine-ApiKey: {your_api_key}`
```

- Successful response body

```
`{  
  "success": {  
    "4351ac9f5b024830a2e428c5c611de11": {  
      "__type__": "user",  
      "__id__":  
"4351ac9f5b024830a2e428c5c611de11",  
      "public_name": "John"  
    }  
  },  
  "errors": {}  
}`
```

Note: record John's ID so that you can include it in the next step.

8. Now create the ACL list for Mary's HealthyHabits data.

- Request type

```
`POST`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/user/access`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`  
`X-CloudMine-SessionToken: {marys_session_token}`
```

- Request body

```
`{ "members": [{"johns_user_id"}], "permissions":  
["r"] }`
```

- Response body

```
`{  
  "5b304290a2484882b7f3144ffe83f4c2": {  
    "__type__": "acl",  
    "__id__":  
"5b304290a2484882b7f3144ffe83f4c2",  
    "permissions": [  
      "r"  
    ],  
    "members": [  
      "4351ac9f5b024830a2e428c5c611de11"  
    ],  
    "segments": {}  
  }  
}`
```

- **Note:** record the ACL's `__id__` value found in the response, which will be required in the next step to connect Mary's ACL and her HealthyHabits object.
9. Now that both Mary's HealthyHabits data and the ACL has been generated, add the ACL ID to the HealthyHabits object.

- Request type

```
`POST`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/user/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`  
`X-CloudMine-SessionToken: {marys_session_token}`
```

- Request body

```
`{ "HealthyHabits_12345":  
  { "__access__":  
    ["5b304290a2484882b7f3144ffe83f4c2"] }  
}`
```

Note: `__access__` needs to be populated with the ACL's `__id__` value.

10. Using John's session token, Mary's HealthyHabits object can now be accessed.

- Request type

```
`GET`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/user/text`
```

- Request parameters

```
`keys: HealthyHabits_12345`
```

- Request headers

```
`X-CloudMine-ApiKey: {your_api_key}`  
`X-CloudMine-SessionToken: {johns_session_token}`
```

- Successful request body

```
`{"success":  
  {"HealthyHabits_12345":  
    {"activities": ["running", "lifting  
weights"], "  
      __access__":  
      ["5b304290a2484882b7f3144ffe83f4c2"]}  
    },  
    "errors": {},  
    "meta": {  
      "HealthyHabits_12345" {  
  
        "owner": "fda23bb33c724fe4a74322a1f0443959",  
        "permissions": ["r"]  
      }  
    }  
  }`
```

Note: Mary is the “owner” of this object.

2. App-Level Security and API Keys

CloudMine secures app-level data with API keys. An application may have many different API keys, and each key can be configured with different permission levels and can be given permission to interact with only certain objects. As a best practice, API keys should be “containerized” or configured to interact with as few objects as possible.

In the current example, to encourage collaboration, the App should have some mechanism by which users can submit suggestions for living a healthier lifestyle. Users should be able to query these suggestions and optionally add them to their own HealthyHabits data if they like. For the sake of this exercise, the name of these suggestions will be **HealthyHabitsSuggestions**.

In order to create the correct permissions, first create a new API key in the App using the Connected Health Cloud.

1. Log into the CloudMine Connected Health Cloud and select the **New Application Name** project.
2. Click the **API Keys tab** in the main nav-bar and click **+Create API Key** to make a new key. A newly generated key will show up in the API Keys list. The new key will have a randomly generated ID. Click on the drop down arrow to fully customize an API Key’s general information as well as access rights.

3. Click the key's row to change the key's name to HealthyHabitsSuggestions, and then click the pencil icon that appears under the status column. Then, uncheck all of the permissions for **All objects** and **All files**.

26c187c90fa94a91be637e47889bde8d

Name HealthyHabitsSuggestions

Key 26c187c90fa94a91be637e47889bde8d

Permissions settings

All objects CREATE + READ UPDATE DELETE

All files CREATE + READ UPDATE DELETE

Push Enabled

Schedule Allowed

Delete All

CANCEL SAVE

4. While in the edit API modal, scroll down and click the **Add** button. From there, enter `[__type__="HealthyHabitsSuggestions"]`` in the **filter** field and check the “create, read, and update” icons for the filter rule, and then click **Save**.

Custom access rules + ADD RULE

Filter X

[_type__="HealthyHabitsSuggestions"]

CREATE + READ UPDATE DELETE

CANCEL SAVE

- Now that the new API key has been created, create the HealthyHabitsSuggestions object via the REST API.

- Request type

```
`PUT`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey`
```

Note: this should be the new, restricted key.

- Request body

```
`{  
  "HealthyHabitsSuggestions": {  
    "Hiking": { "RelatedActivities":  
["Camping", "Climbing", "Fishing", "TrailRunning"]  
},  
    "Running": {  
"RelatedActivities":["CrossCountry", "Track",  
"TrailRunning"] },  
    "__type__": "HealthyHabitsSuggestions"  
  }  
}`
```

- Response Body

```
`{"success":  
  {"HealthyHabitsSuggestions": "created"},  
  "errors":{}}  
}`
```

6. To confirm that the new custom API Key rules are working, try to create an object without the correct `__type__` value. If the value in the `__type__` field is incorrect, the filter in the API Key should deny permission to create the object.

- Request type

```
`PUT`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey:{new_ApiKey}`
```

Note: this should be the new restricted API key.

- Request body

```
`{  
  "HealthyHabitsSuggestionsTest": {  
    "Hiking": {  
"RelatedActivities":["Camping", "Campground",  
"Outdoors", "Tents"] },  
    "Running": { "RelatedActivities":["Cross  
Country", "Track", "Field"] },  
    "__type__": "HealthyHabitsSuggestionsTest"  
  }  
}`
```

- Response Body

```
`{  "success": {},  
    "errors": {"HealthyHabitsSuggestionsTest":  
"permission denied"}  
}`
```

Great! You have used an API Key to assign a specific permission to an object!

3. The `__class__` Attribute

As discussed above, API Keys can be designed to allow access and affect only objects that match specific criteria; one field many of our customers use to perform these mappings is the `__class__` attribute. CloudMine treats `__class__` as a special field and suggests that this specific attribute is appropriately populated and maintained. Many of our customers use this attribute for different use cases, such as:

- Logical Separation of Disparate Data Types
- Access Control around Various Object Types
- Easily identify data in the Connected Health Cloud Dashboard
- Reporting & Analytics

To facilitate these use cases, the CloudMine platform automatically indexes values with a `__class__` key so that searching with this attributed is optimized.

The `__class__` attribute is exposed via all CloudMine SDKs and APIs, as well as across all types of objects (files, images, etc.)

6. Creating your First Server-Side JavaScript Snippet

1. Introduction

In some cases, an app may need to perform operations that are better executed on the server-side instead of the client-side in order to improve performance, data fidelity, and user experience. CloudMine enables developers to create and invoke custom JavaScript Snippets, or free-form code blocks, that are executed on the server.

2. Creating a Snippet

1. First, navigate to the Connected Health Cloud Dashboard and the App that will implement the snippet.
2. Click the **Server Code** button in the left nav-bar, select **JavaScript**. On the next page, click the **Make one** button to start working on a new snippet.
3. Give the snippet a name in the **Snippet Name** input field and click **Save**.

This guide will use: **GettingStartedSnippet**

Enter the following code into the code editor on the snippet page and click **Save**.

```
// 'data' is automatically available with the
request info
var total = null;
//Subtract one since the '__type__' value does
not count as a suggestion
total = --
(Object.keys(data.success.HealthyHabitsSuggest
ions).length);
//Return the total to the client
```

```
exit( { num_suggestions: total } );
```

3. Executing Snippets

To verify that the Snippet works as expected, use a REST API call to evoke the code and verify the output. Note that, with this example, data is not being passed into the Snippet, but rather the Snippet is performing a static function and returning its results.

1. Build the Object Fetch request with the following details:

- Request type

```
`GET`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/{app_id}/text`
```

- Request Parameters

```
`f: GettingStartedSnippet`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {your_api_key}`
```

2. After the REST client is configured with these details, **Send** or **Submit** the request. The response should look similar to the following:

```
`{  
  "result":{"num_suggestions":2},  
  "success": {"HealthyHabitsSuggestions":  
    {"Hiking":{"RelatedActivities":["Camping",  
    "Campground", "Outdoors",  
    "Tents"]},"Running":{"RelatedActivities":  
    ["Cross Country","Track","Field"]}},  
  "__type__":" HealthyHabitsSuggestions"} },  
  "errors":{}  
}`
```

Awesome! You have created and run your first CloudMine Snippet!

3. The ``data.success`` value might not always be wanted in the response body. The following parameter can be added to the Snippet call to silence the ``data.success`` value in the response.

```
`result_only: true`
```

4. The response body with the ``result_only`` parameter should look something like this:

```
`{  
  "result": {"num_suggestions":2}  
}`
```

4. Providing Inputs to Snippets

If the function built requires input, the input will need to be included in the parameters in the API call.

Consider a case where an end user should be allowed to see how many HealthyHabitsSuggestions have been shared with them. Rather than performing this calculation on the client side (in the browser), expose an API on the server that returns the count for the specified user. This API might require the user's login credentials to verify that the correct user is getting the correct count. Since the number of employees sharing HealthyHabits data might be relatively large, it's easy to imagine that this function might be best implemented on the server.

1. To begin, navigate to the **JavaScript Server Code** section of the 'New Application Name' application.
2. Click the **+New Code Snippet** button.
3. Provide a name for this snippet. This guide will use: **CountSharedUserObjects**.

4. Copy and paste the following code block into the code editor and click **Save**.

```
`//Count Shared Objects

//Get variables from Data structure

var email = data.params.email;
var password = data.params.password;
var ApiKey = data.apikey;

//Establish a CM WS Session

var ws = new cloudmine.WebService({
  appid: '{your_app_id}',
  apikey: ApiKey
});

//Authenticate the user for their shared count

ws.login(email, password).on('success',
function(data, response){
  //Perform a search; we only want the results
  JSON in the response
  ws.get({results_only: true}).on('success',
function(data2, response){
  // One of these is the ACL, we need to
  subtract
  var count = (Object.keys(data2).length);
  //Return the count to the caller
  exit({"count": count});
  });
});
`
```

5. Now that the Snippet is saved, issue an API request to obtain Mary's account value

- Request type

```
`PUT`
```

- Request URL

```
`https://api.cloudmine.io/v1/app/run/CountSharedUserObjects`
```

- Request parameters

```
`{"result_only":true,  
"email":"mary@testmail.com", "password":  
"password1"}`
```

- Request headers

```
`Content-Type: application/json`  
`X-CloudMine-ApiKey: {Your api_key}`
```

6. A successful response body would look something like:

```
`{  
  "result": {"count":1}  
}`
```

Note: Snippets can also be called by using the following endpoint.

```
/app/{app_id}/run/{snippet_name}
```

Great! You have successfully passed inputs into a server-side snippet!

7. Developer Resources & Documentation

Throughout the process of building applications on CloudMine, specific use cases will present more advanced challenges than those included in this guide. To help address potential issues, CloudMine maintains the following resources:

1) Online Documentation

CloudMine's recommendation when beginning with the platform is to consult our Documentation Site, which describes each of our APIs and their use. Before sending an inquiry to our support and engineering team, please review our documentation. <https://cloudmine.io/docs/>

2) Online Getting Started Guides

CloudMine's recommendation when beginning with the platform is to walk through our Getting Started Guides, which walk through our platform, our APIs and their use. Before sending an inquiry to our support and engineering team, please walk through our guides.

<https://cloudmineinc.com/CHC/GSG/>

3) GitHub Resources

CloudMine maintains a GitHub repository with working code samples. The samples were created by our engineering team and are an excellent starting point for getting up and running on CloudMine.

<https://github.com/cloudmine>

4) Engineers

The CloudMine engineering team participates in a rotating support cycle to ensure your success with the CloudMine platform. To request a session with an engineer, please contact your CloudMine account team.

8. Appendix A: Getting Started with Postman

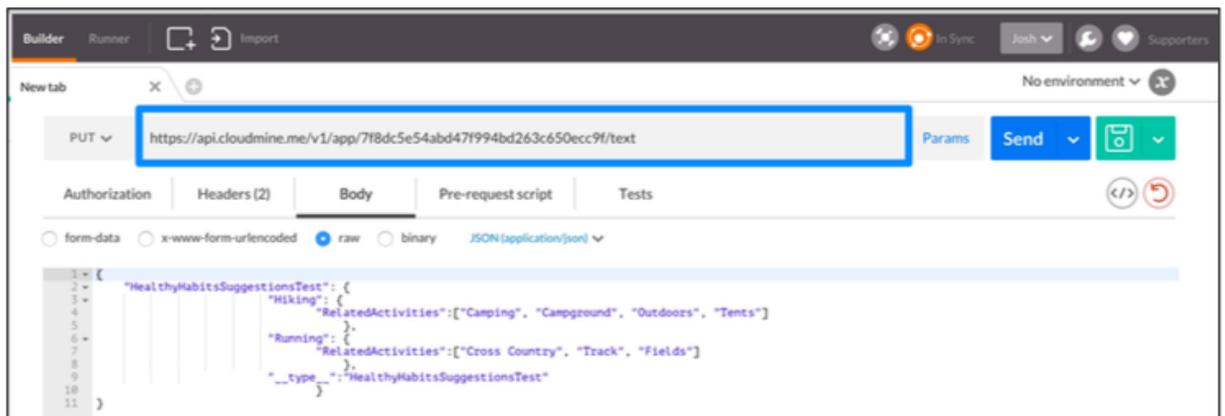
This section will review how to make basic REST API calls with Postman. Please note that CloudMine supports all REST clients; Postman is just a single example of a REST client. To install the extension, please begin here:

<https://www.getpostman.com>.

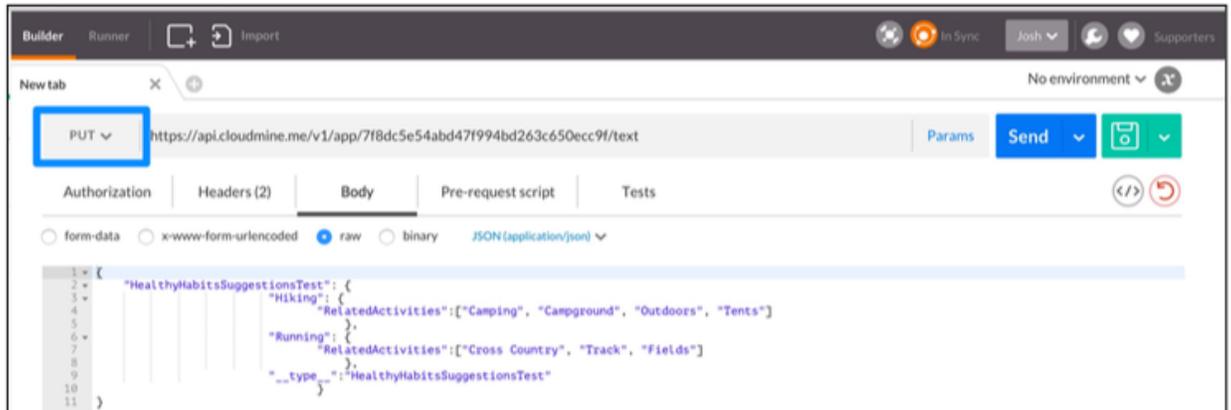
1. Building a Request

A REST request is comprised of various components: URLs, headers, parameters, and bodies. We will review how to define and specify each of these items within Postman.

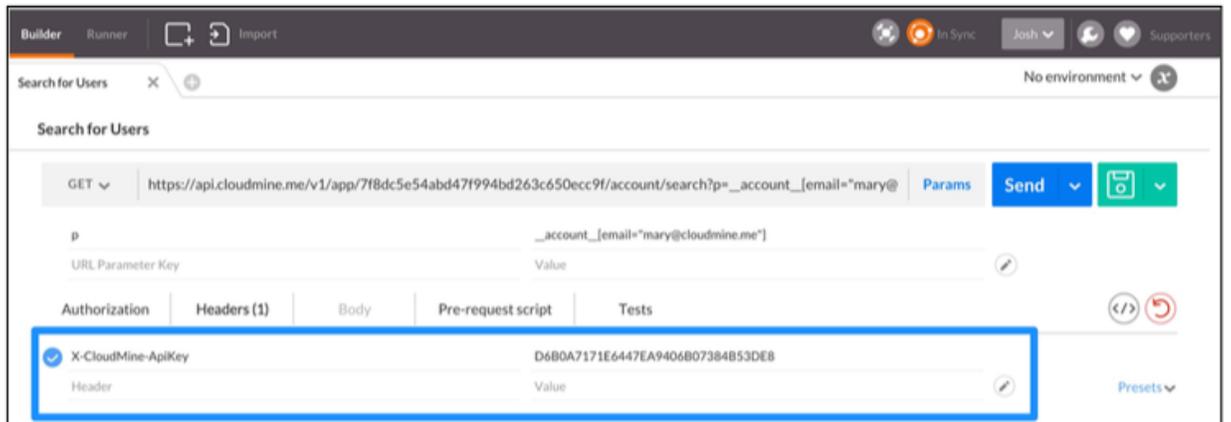
1. The most foundational component of a REST API is arguably its URL. Within Postman, the request URL is specified here:



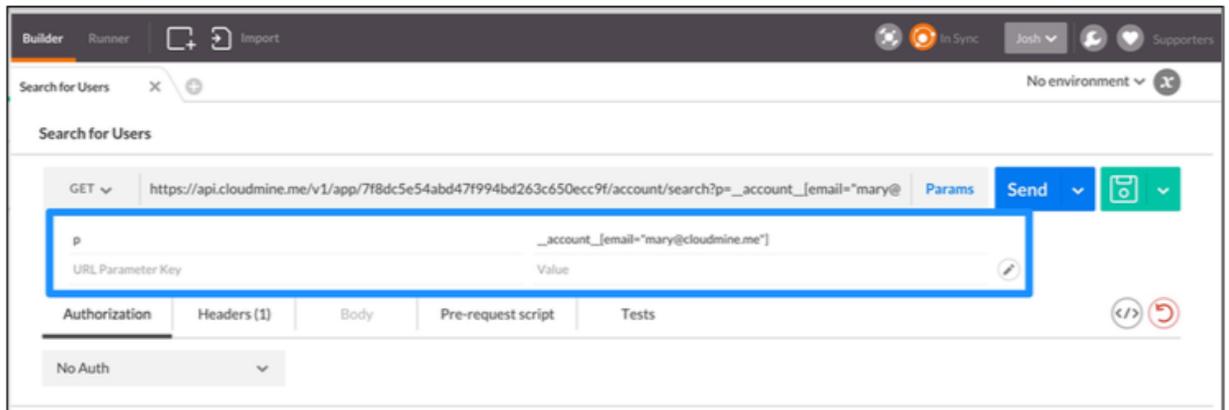
2. Once the endpoint URL is established, specify the type of request that should be executed. Different types of requests output different results and require different parameters. For example, a PUT request may require a **Request Body**, whereas a GET request may only need one or two **Request Headers**.



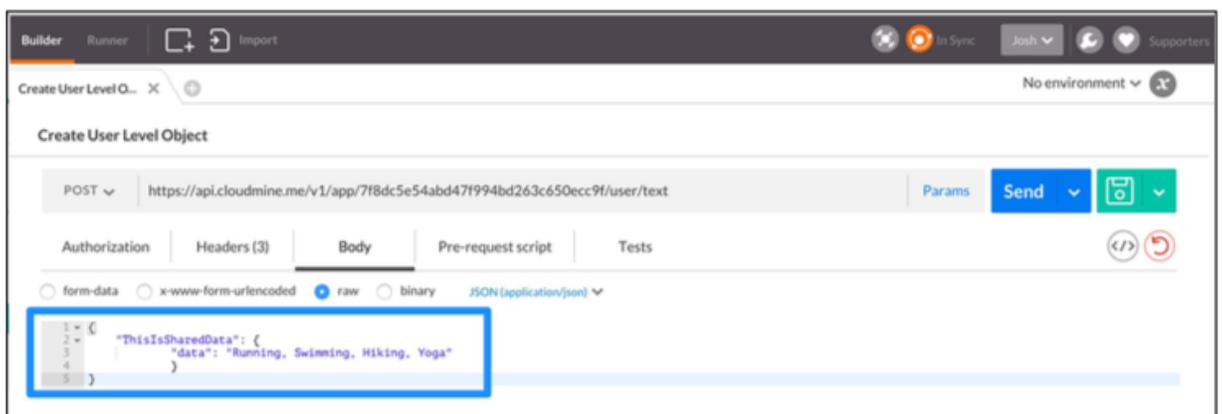
3. Headers provide contextual information about their quest. To specify a header, select the **Headers** button within the Postman app.



- Many times, developers need to include additional data points within their request. These data points can be interpreted by the API as input, or parameters. Parameters exist so that developers can provide additional data to the API to obtain the desired result. To specify any required parameters, select the blue **Params** button next to the URL.

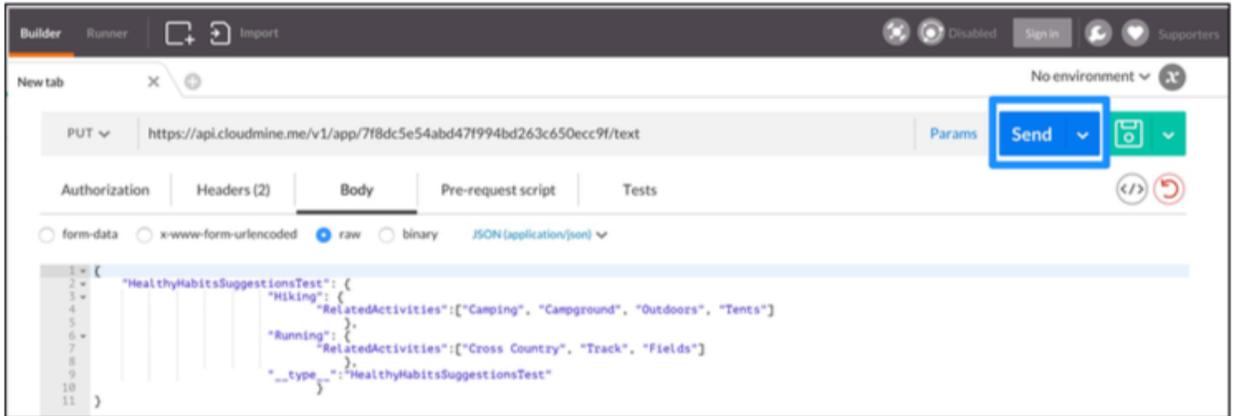


- Finally, depending on the type of request, it may be necessary to specify a body. This body is in addition to any headers, parameters, and the URL. The body of the request can be specified by first clicking **Body**. Once selected, create the **Request Body** within the editor window.



Note: REST typically only supports including a **Request Body** for PUT and POST type requests.

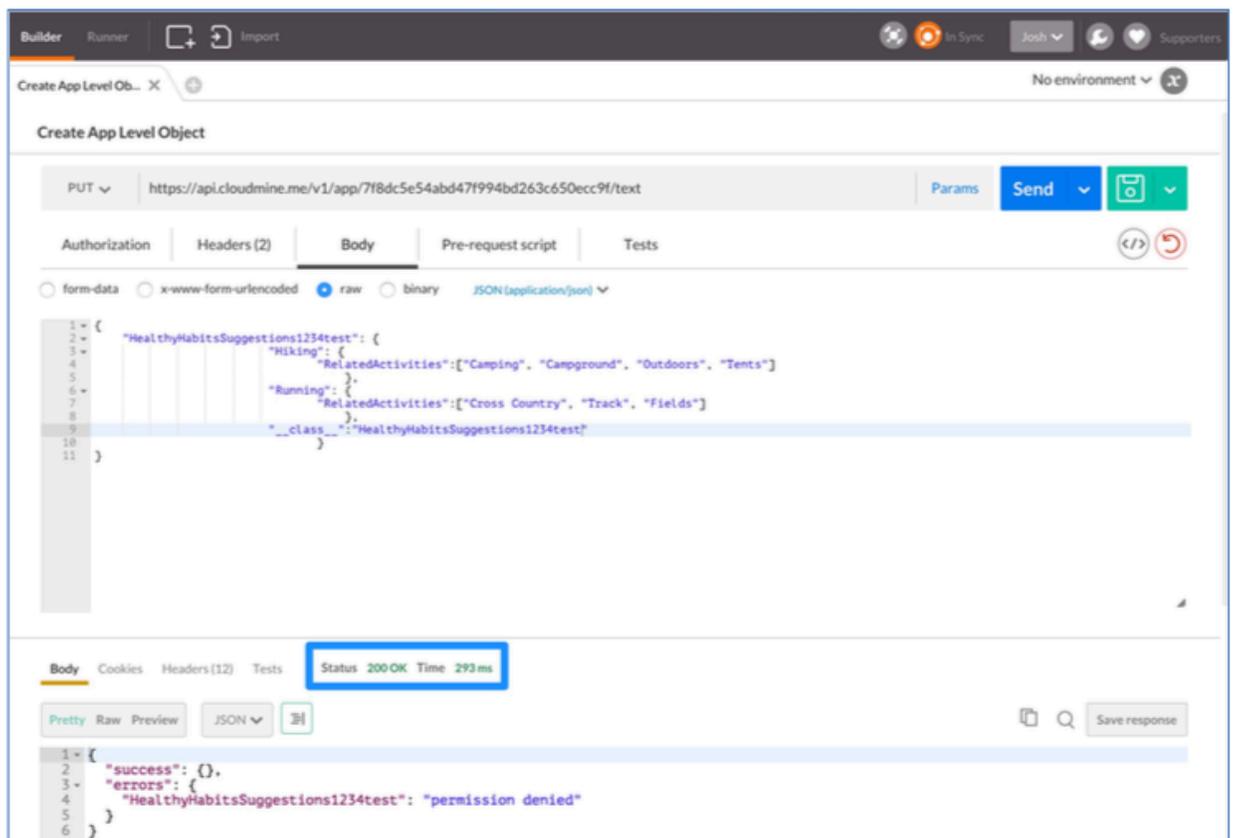
6. After having finished specifying these details about a request, select **Send**.



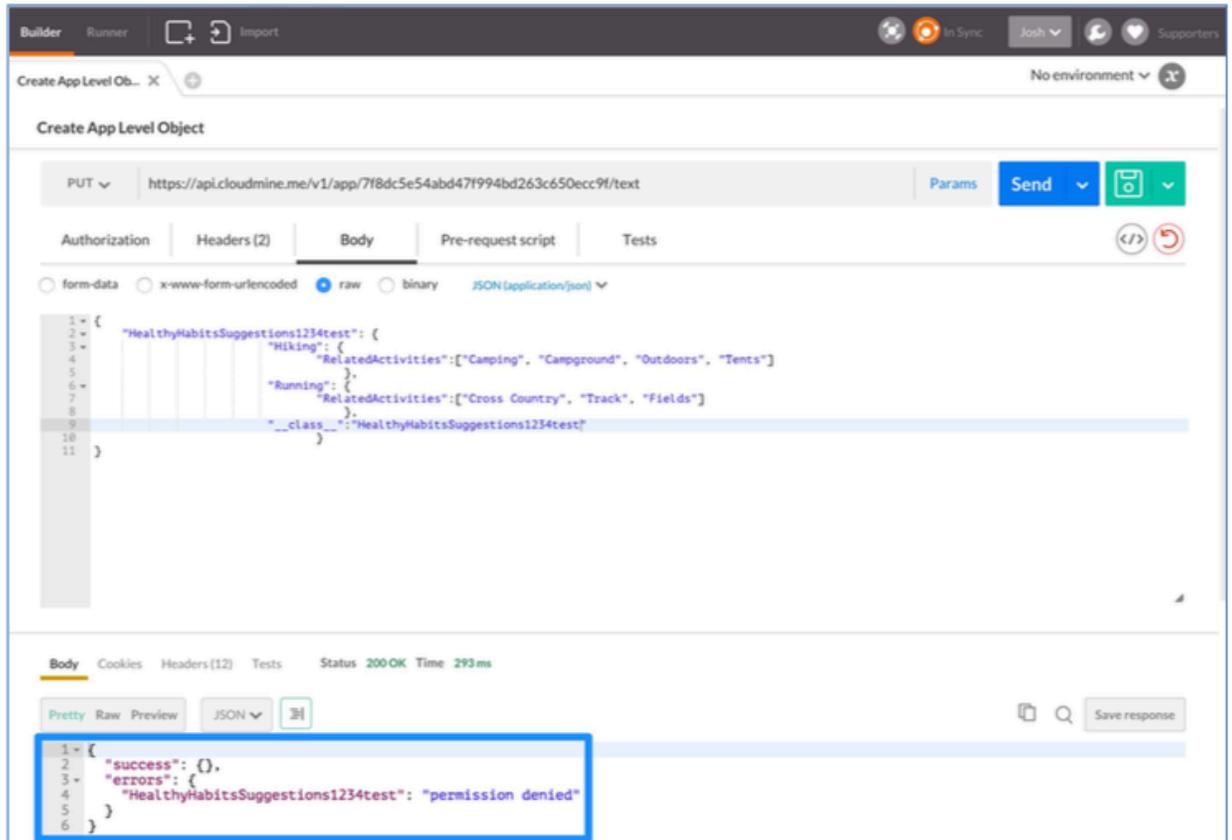
2. Evaluating the Response

When the server responds, it will either issue a successful response (typically a 200 status code) or an unsuccessful response (any non-200 code). The API response will be comprised of two components: the **Response Code** and the **Response Body**.

1. Postman displays the **Response Code** in the bottom half of the application window. It also notes how long the server needed to generate a response.



2. Postman displays the **Response Body** immediately below the **Response Code**.



Note: In this example, the response code is 200. Some folks may be confused by this response, especially given that the response text is Permission Denied. Despite not being able to create this object, the server understood our request and attempted to carry out the action, so the call is classified as successful.

9. Appendix B: Sample Curl Calls

Create an object:

```
curl -d '{ "student_123456": { "__class__":  
"CloudMineStudent", "email_id":  
"JaneDoe@CloudMine.com", "details": { "name": "Jane  
Doe", "hair_color": "brown", "state": "Pennsylvania",  
"city": "Philadelphia" } } }' -H "Content-Type:  
application/json" -H "X-CloudMine-ApiKey:  
cc7dd665d44244f993c77d0f1e12f9af" -X PUT  
'https://api.cloudmine.io/v1/app/d8c6b12278c38343128b1d  
ea00e546c7/text'
```

Fetch an object:

```
curl -H "X-CloudMine-ApiKey: {your_api_key}" -X GET  
'https://api.cloudmine.io/v1/app/{app_id}/text?keys=stu  
dent_123456'
```

Update an object:

```
curl -d '{ "student_123456": { "details": { "company":  
"CloudMine" } } }' -H "Content-Type: application/json"  
-H "X-CloudMine-ApiKey: {your_api_key}" -X POST  
'https://api.cloudmine.io/v1/app/{app_id}/text'
```

Authenticate a user:

```
authentication = {username}:{password} in Base64
encoding

curl -H "Content-Type: application/json" -H
"Authorization: Basic {authentication}" -H "X-
CloudMine-ApiKey: {your_api_key}" -X POST
'https://api.cloudmine.io/v1/app/{app_id}/account/login'
```

Create user object:

```
curl -d '{ "HealthyHabits_12345": {"activities":
["running", "lifting weights", "cycling"]} }' -H
"Content-Type: application/json" -H "X-CloudMine-
SessionToken: {user_session_token}" -H "X-CloudMine-
ApiKey: {your_api_key}" -X POST
'https://api.cloudmine.io/v1/app/{app_id}/user/text'
```

Search for a user (cannot search for data in `__account__`):

```
curl -g -H "X-CloudMine-ApiKey: {your_api_key}" -X GET
https://api.cloudmine.io/v1/app/{app_id}/account/search
?p=[public_name=/John/]
```

Create access list for user:

```
curl -d '{ "members": [{"user_id}], "permissions":
["r"]} ' -H "Content-Type: application/json" -H "X-
CloudMine-SessionToken: {user_session_token}" -H "X-
CloudMine-ApiKey: {your_api_key}" -X POST
https://api.cloudmine.io/v1/app/{app_id}/user/access

5b9a963cdd974704b94545c0da36df61
```

Add access list to user's object:

```
curl -d '{ "HealthyHabits_12345": { "__access__": [{"user_id"}] } }' -H "Content-Type: application/json" -H "X-CloudMine-SessionToken: {user_session_token}" -H "X-CloudMine-ApiKey: {your_api_key}" -X POST https://api.cloudmine.io/v1/app/{app_id}/user/text
```

Fetch other user's object (via `__access__` object):

```
curl -H "X-CloudMine-ApiKey: {your_api_key}" -H "X-CloudMine-SessionToken: {user_session_token}" -X GET 'https://api.cloudmine.io/v1/app/{app_id}/user/text?key s=HealthyHabits_12345'
```

Create object for custom API rule:

```
curl -d '{ "HealthyHabitsSuggestions": { "Hiking": { "RelatedActivities": ["Camping", "Climbing", "Fishing", "TrailRunning"] }, "Running": { "RelatedActivities": ["CrossCountry", "Track", "TrailRunning"] }, "__type__": "HealthyHabitsSuggestions" } }' -H "Content-Type: application/json" -H "X-CloudMine-ApiKey: {restricted_api_key}" -X PUT 'https://api.cloudmine.io/v1/app/{app_id}/text'
```

Test custom API rule:

```
curl -d '{ "HealthyHabitsSuggestionsTest": { "Hiking": { "RelatedActivities":["Camping", "Campground", "Outdoors", "Tents"] }, "Running": { "RelatedActivities":["Cross Country", "Track", "Field"] }, "__type__": "HealthyHabitsSuggestionsTest" } }' -H "Content-Type: application/json" -H "X-CloudMine-ApiKey: {restricted_api_key}" -X PUT 'https://api.cloudmine.io/v1/app/{app_id}/text'
```

Execute a JS call (GET):

```
curl -H "Content-Type: application/json" -H "X-CloudMine-ApiKey: {your_api_key}" -X GET 'https://api.cloudmine.io/v1/app/{app_id}/text?f=GettingStartedSnippet'
```

Execute a JS call (GET), but suppress the `data.success` value:

```
curl -H "Content-Type: application/json" -H "X-CloudMine-ApiKey: {your_api_key}" -X GET 'https://api.cloudmine.io/v1/app/{app_id}/text?f=GettingStartedSnippet&result_only=true'
```

Execute a JS call (PUT) that input data to snippet:

```
curl -d '{"result_only":true, "email":"mary@testmail.com", "password":"password1"}' -H "Content-Type: application/json" -H "X-CloudMine-ApiKey: {your_api_key}" -X PUT 'https://api.cloudmine.io/v1/app/{app_id}/run/CountSharedUserObjects'
```