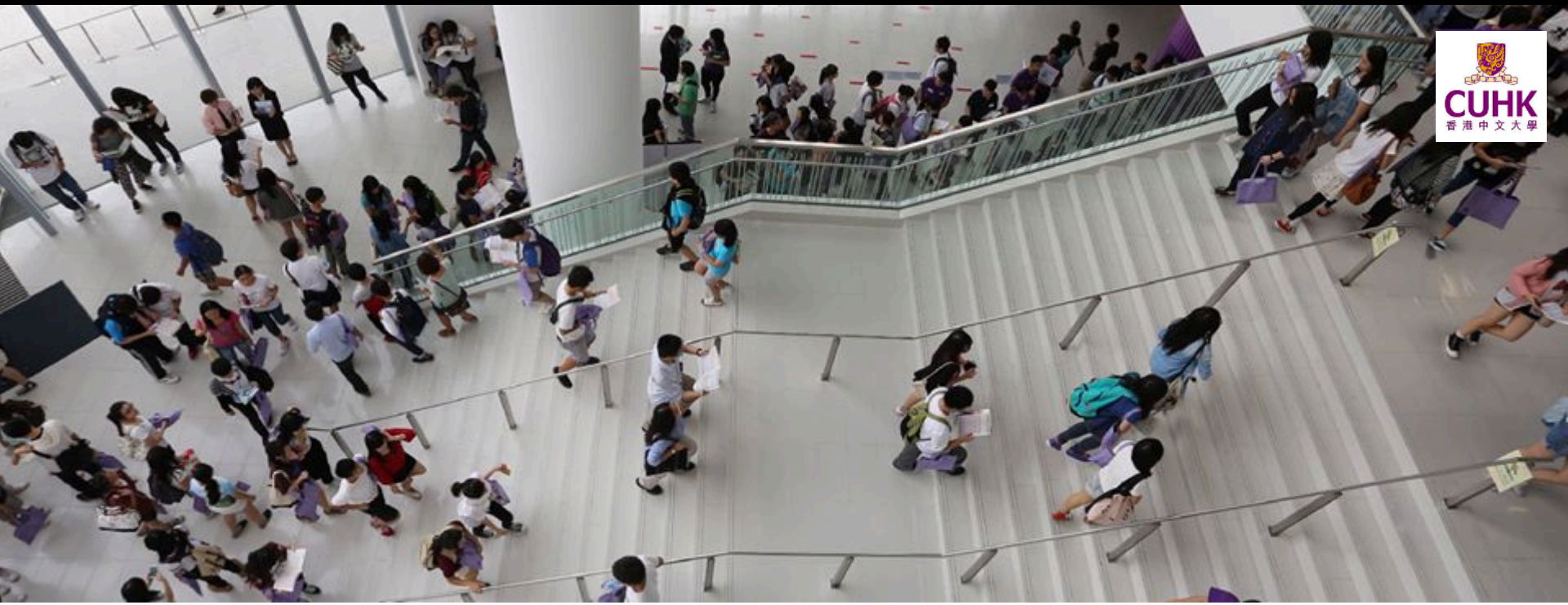


COM5940 NEW MEDIA BUSINESS MODEL & INNOVATION:
LESSON 3 - FRONTEND/BACKEND REST API INTEGRATION

Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

Today's agenda.

1. Manage Flask CRUD data with Flask-Admin.
2. REST API and CRUD support in Airtable.
3. Build mobile app with CRUD and authentication and authorisation support in Bravo Studio.
4. Introduction to Airtable automation.

Search projects

Help Sponsors Log in Register

Flask-Admin 1.6.0

```
pip install Flask-Admin
```

Latest version

Released: Jan 30, 2022

Simple and extensible admin interface framework for Flask

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

GitHub statistics:

- Stars: 5239

Project description

Flask-Admin

The project was recently moved into its own organization. Please update your references to *git@github.com:flask-admin/flask-admin.git*.

localized 96% Run tests passing

Introduction

Flask-Admin is a batteries-included, simple-to-use Flask extension that lets you add admin interfaces to Flask applications. It is inspired by the *django-admin* package, but implemented in such a way that the developer has total control of the look, feel and functionality of the resulting application.

Source: https://pypi.org/project/Flask-Admin/

Useful Links

Flask
Flask-Admin @ github

Quick search

Go

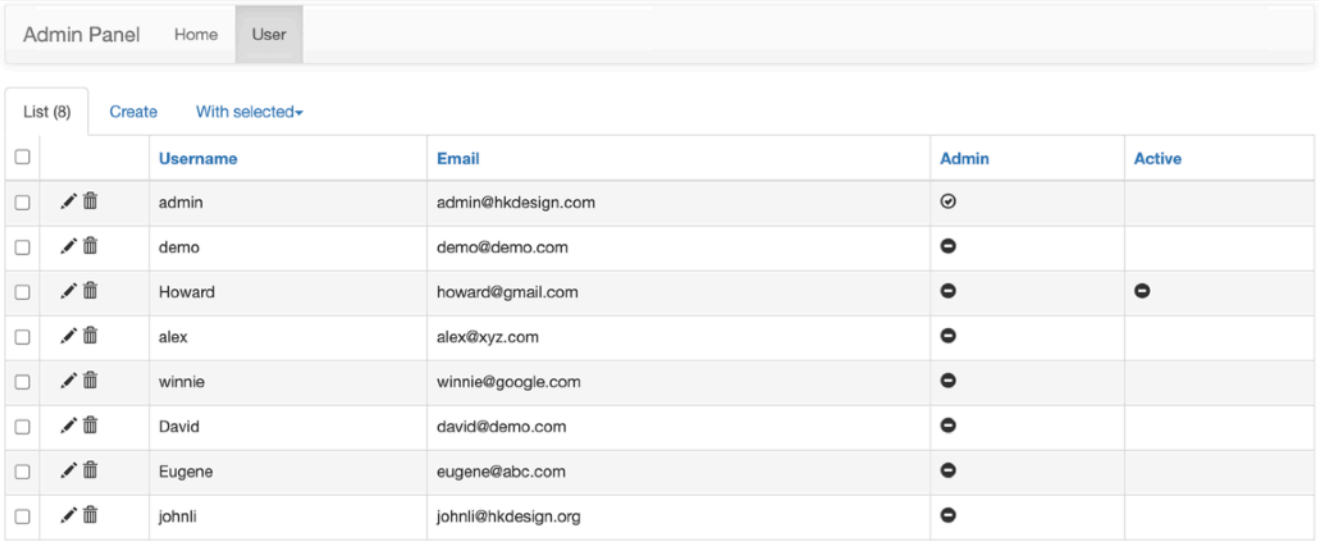
Flask-Admin

Why Flask? As a micro-framework, Flask lets you build web services with very little overhead. It offers freedom for you, the designer, to implement your project in a way that suits your particular application.

Why Flask-Admin? In a world of micro-services and APIs, Flask-Admin solves the boring problem of building an admin interface on top of an existing data model. With little effort, it lets you manage your web service's data through a user-friendly interface.

How does it work? The basic concept behind Flask-Admin, is that it lets you build complicated interfaces by grouping individual views together in classes: Each web page you see on the frontend, represents a method on a class that has explicitly been added to the interface.

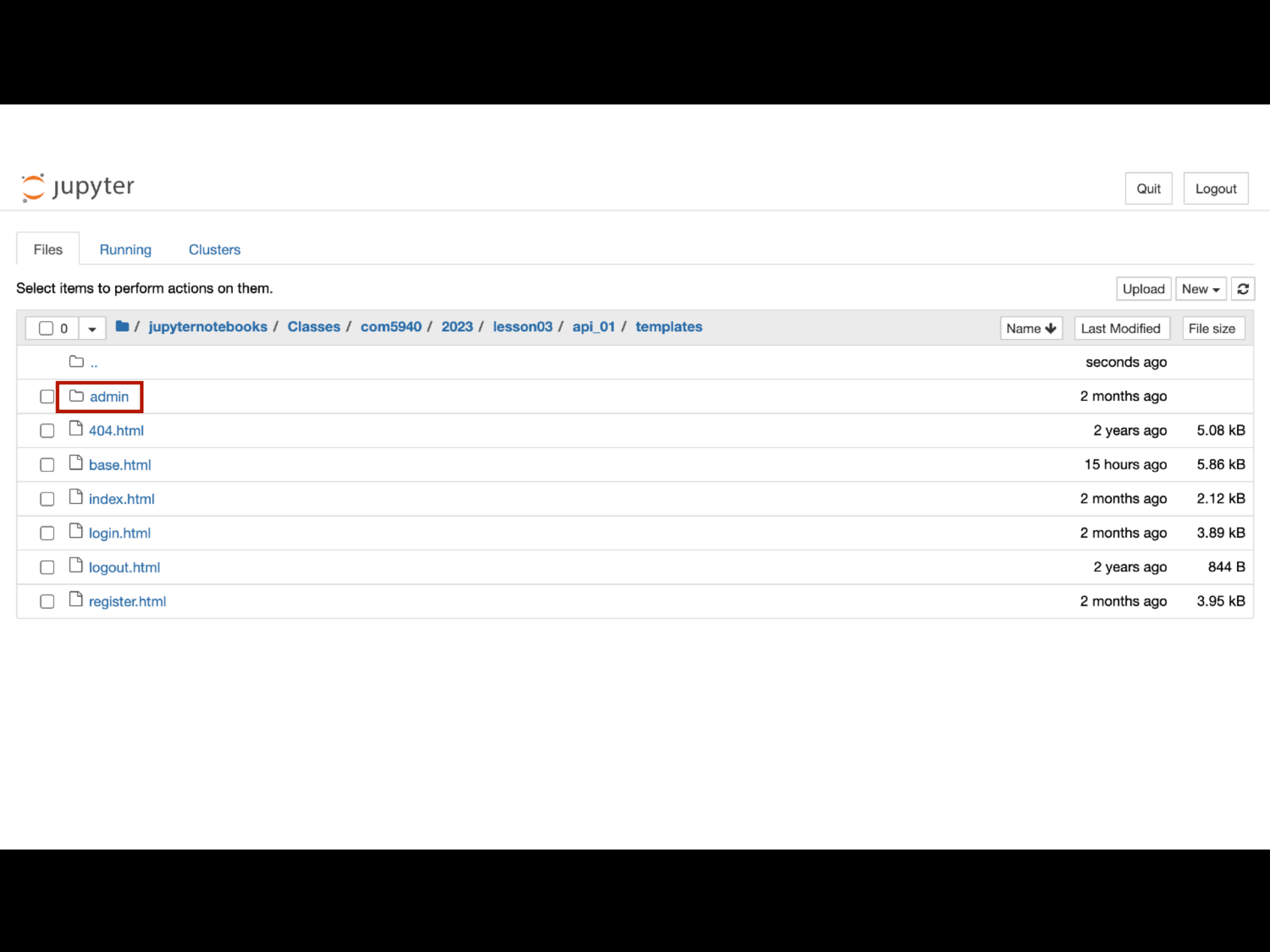
Source: https://flask-admin.readthedocs.io/en/latest/

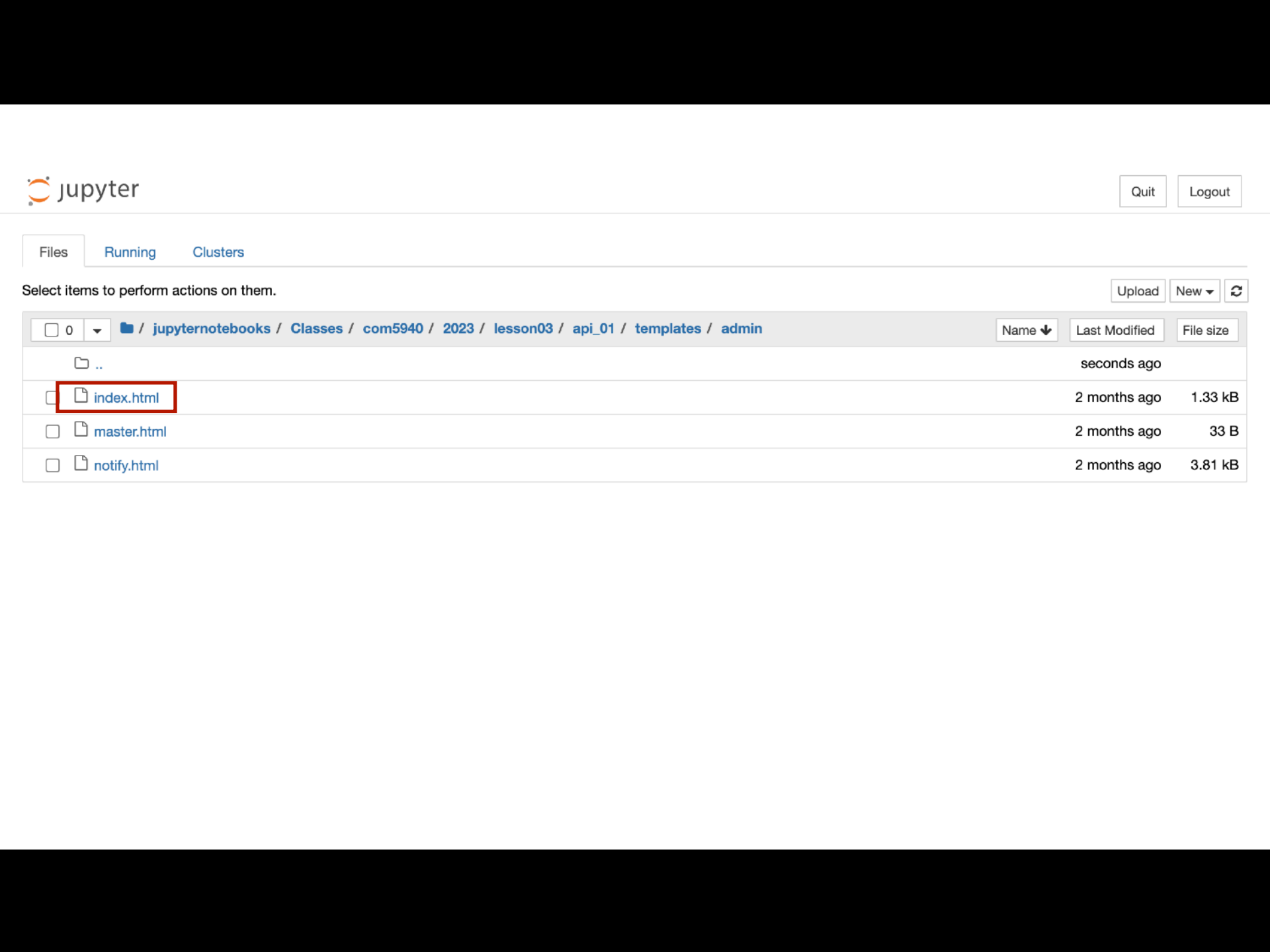

Admin Panel
Home
User
List (8)
Create
With selected
Username
Email
Admin
Active
admin
admin@hkdesign.com
demo
demo@demo.com
Howard
howard@gmail.com
alex
alex@xyz.com
winnie
winnie@google.com
David
david@demo.com
Eugene
eugene@abc.com
johnli
johnli@hkdesign.org

In [*]:

```
##########################################################################################################################

# EXTERNAL MODULES TO BE USED

##########################################################################################################################
import os
import secrets
from flask import Flask, flash, render_template, request, redirect, url_for, session, jsonify, make_response
# For pythonanywhere development, use the following code
# from flask_session import Session
# and remove from flask_session.__init__ import Session
from flask_session.__init__ import Session
from flask_bcrypt import Bcrypt
import jwt
from flask_cors import CORS
from functools import wraps
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime, timedelta
from flask_admin import Admin
from flask_admin.contrib.sqla import ModelView
from flask_admin import BaseView, expose

app = Flask(__name__)
bcrypt = Bcrypt(app)

##########################################################################################################################

# APP CONFIGURATION

##########################################################################################################################

# app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:root@localhost/workshop'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///workshop.db'
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SECRET_KEY'] = 'thisismysecret'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
Session(app)
CORS(app)
cors = CORS(app, resources={r"/api/*": {"origins": "*"}})

UPLOAD_FOLDER = 'static/profile_pics'
```

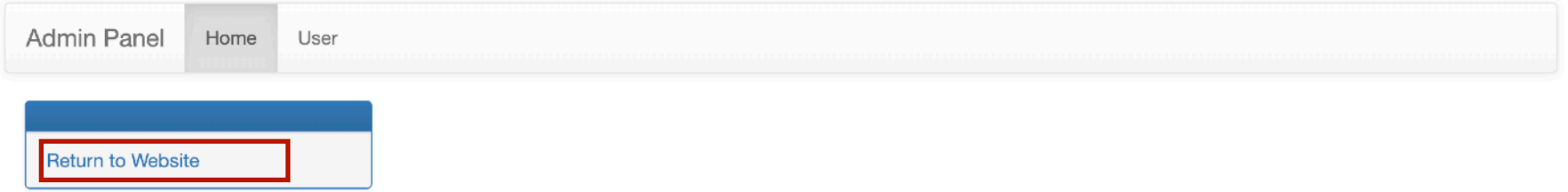
```
87  ##########################################################################################################################
88
89  # DATA MODELS
90
91  ##########################################################################################################################
92
93  db = SQLAlchemy(app)
94
95  class User(db.Model):
96      __tablename__ = 'users'
97      id = db.Column(db.Integer, primary_key=True)
98      username = db.Column(db.String(50))
99      email = db.Column(db.String(255))
100     password = db.Column(db.String(80))
101     bio = db.Column(db.Text, nullable=False)
102     admin = db.Column(db.Boolean)
103     image_file = db.Column(db.String(255), nullable=False, default='default.jpg')
104     active = db.Column(db.Boolean)
105
106 class UserModelView(ModelView):
107     column_exclude_list = ['password', 'bio', 'image_file' ]
108     def is_accessible(self):
109         return (session['logged_in'] and session['admin'] > 0)
110
111     def inaccessible_callback(self, name, **kwargs):
112         return redirect(url_for('login'))
113
114 admin = Admin(app, name='Admin Panel', template_mode='bootstrap3')
115 admin.add_view(UserModelView(User, db.session))
116
```


jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/ jupyternotebooks / Classes / com5940 / 2023 / lesson03 / api_01 / templates
Name
Last Modified
File size
..
seconds ago
admin
2 months ago
404.html
2 years ago
5.08 kB
base.html
15 hours ago
5.86 kB
index.html
2 months ago
2.12 kB
login.html
2 months ago
3.89 kB
logout.html
2 years ago
844 B
register.html
2 months ago
3.95 kB

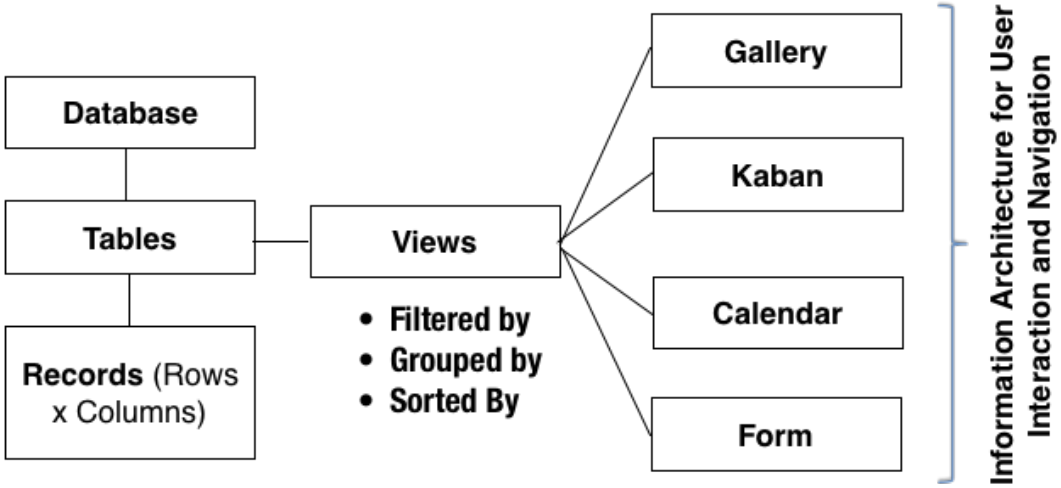

jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/ jupyternotebooks / Classes / com5940 / 2023 / lesson03 / api_01 / templates / admin
Name
Last Modified
File size
..
seconds ago
index.html
2 months ago
1.33 kB
master.html
2 months ago
33 B
notify.html
2 months ago
3.81 kB

```
{% extends 'admin/master.html' %}
{% block head_css %}
{{ super() }}
<link href="{{ url_for('static', filename='sb-admin-2.css') }}" rel="stylesheet">
{% endblock head_css %}

{% block body %}
{{ super() }}
                <div class="col-lg-3 col-md-6">
                    <div class="panel panel-primary">
                        <div class="panel-heading">
                            <div class="row">
                                <div class="col-xs-3">
                                    <i class="fa fa-comments fa-5x"></i>
                                </div>
                                <div class="col-xs-9 text-right">
                                    <div class="huge"></div>
                                    <div></div>
                                </div>
                            </div>
                        </div>
                        <a href="../">
                            <div class="panel-footer">
                                <span class="pull-left">Return to Website</span>
                                <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
                                <div class="clearfix"></div>
                            </div>
                        </a>
                    </div>
                </div>
{% endblock body %}
```

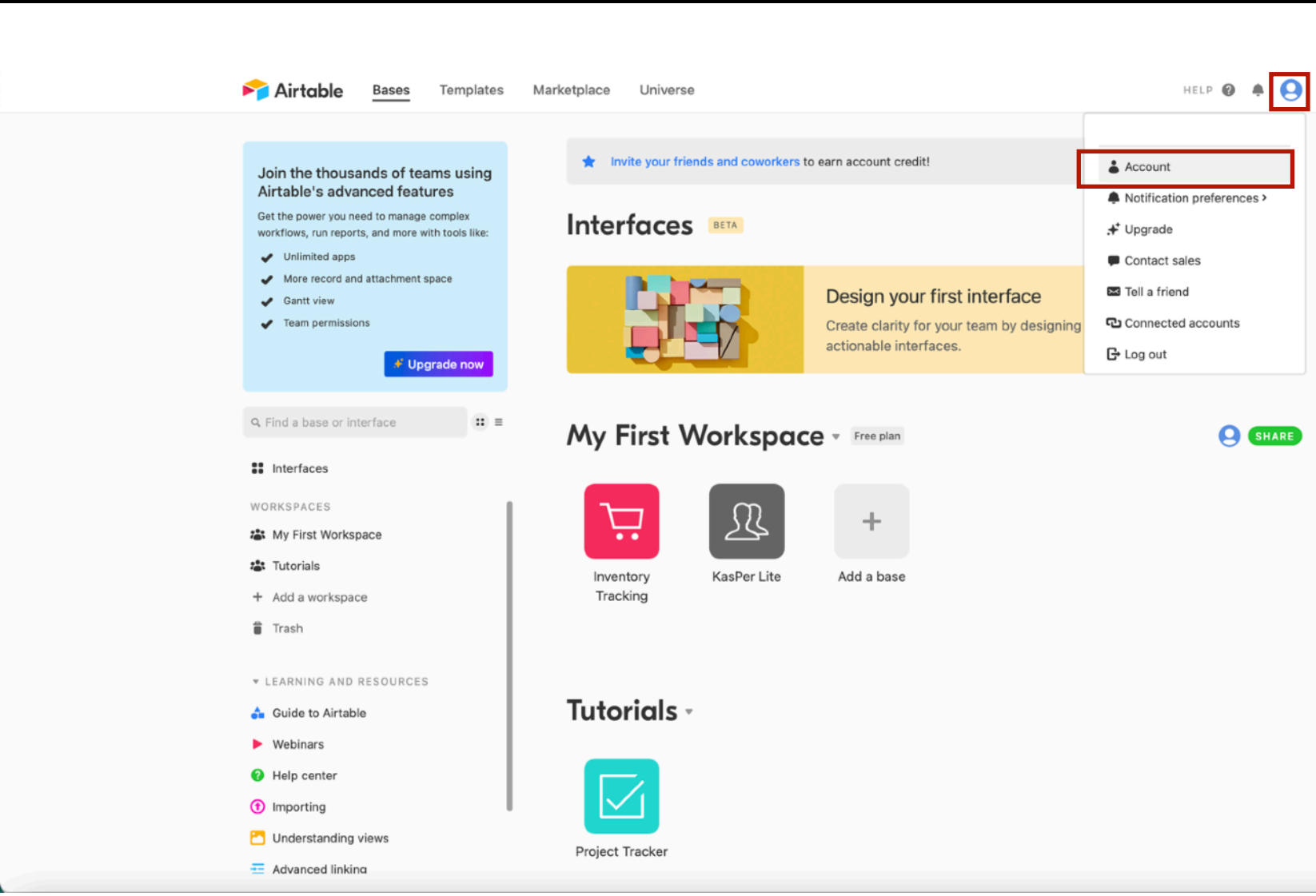
index.html

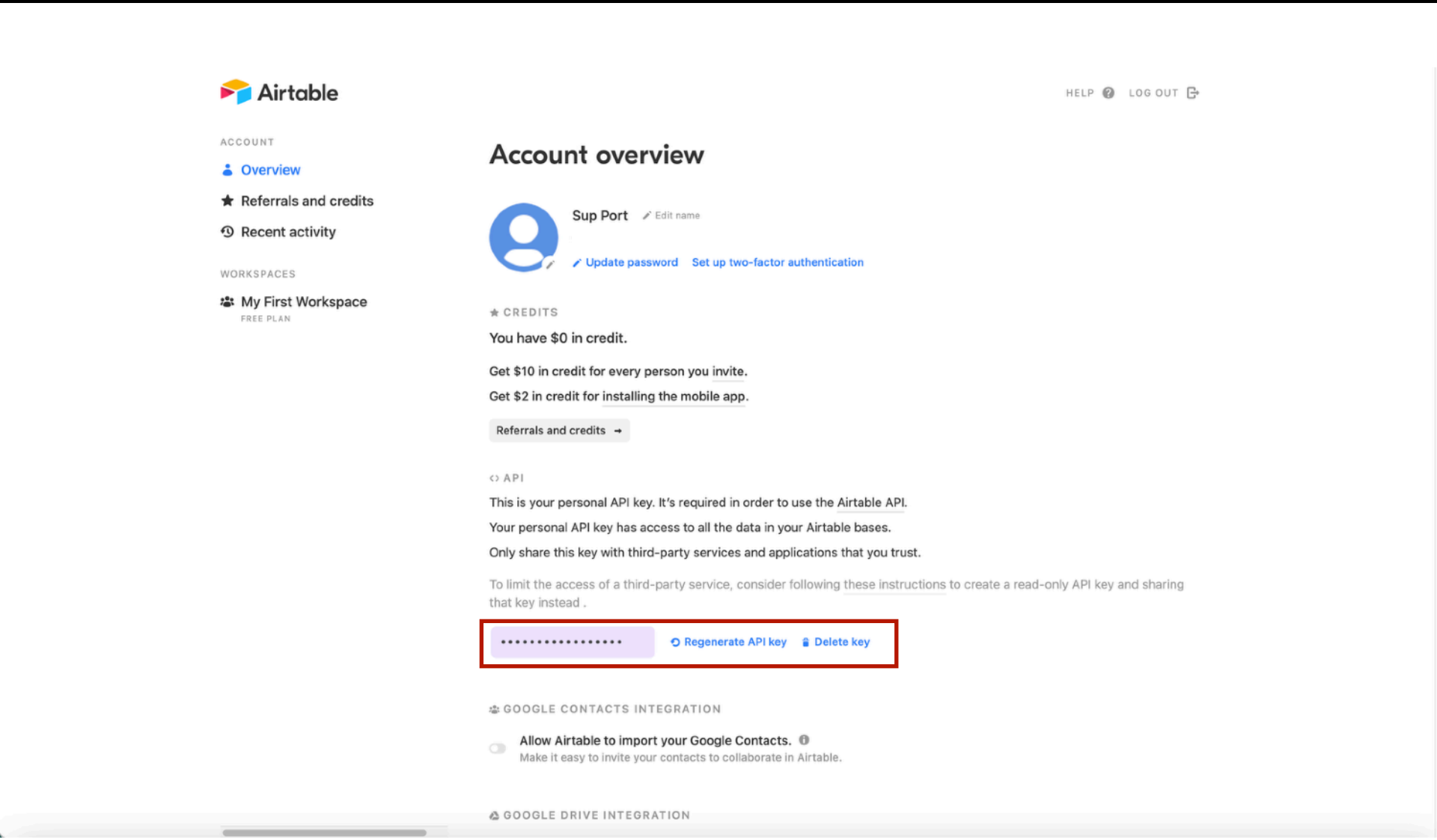

Admin Panel
Home
User
Return to Website

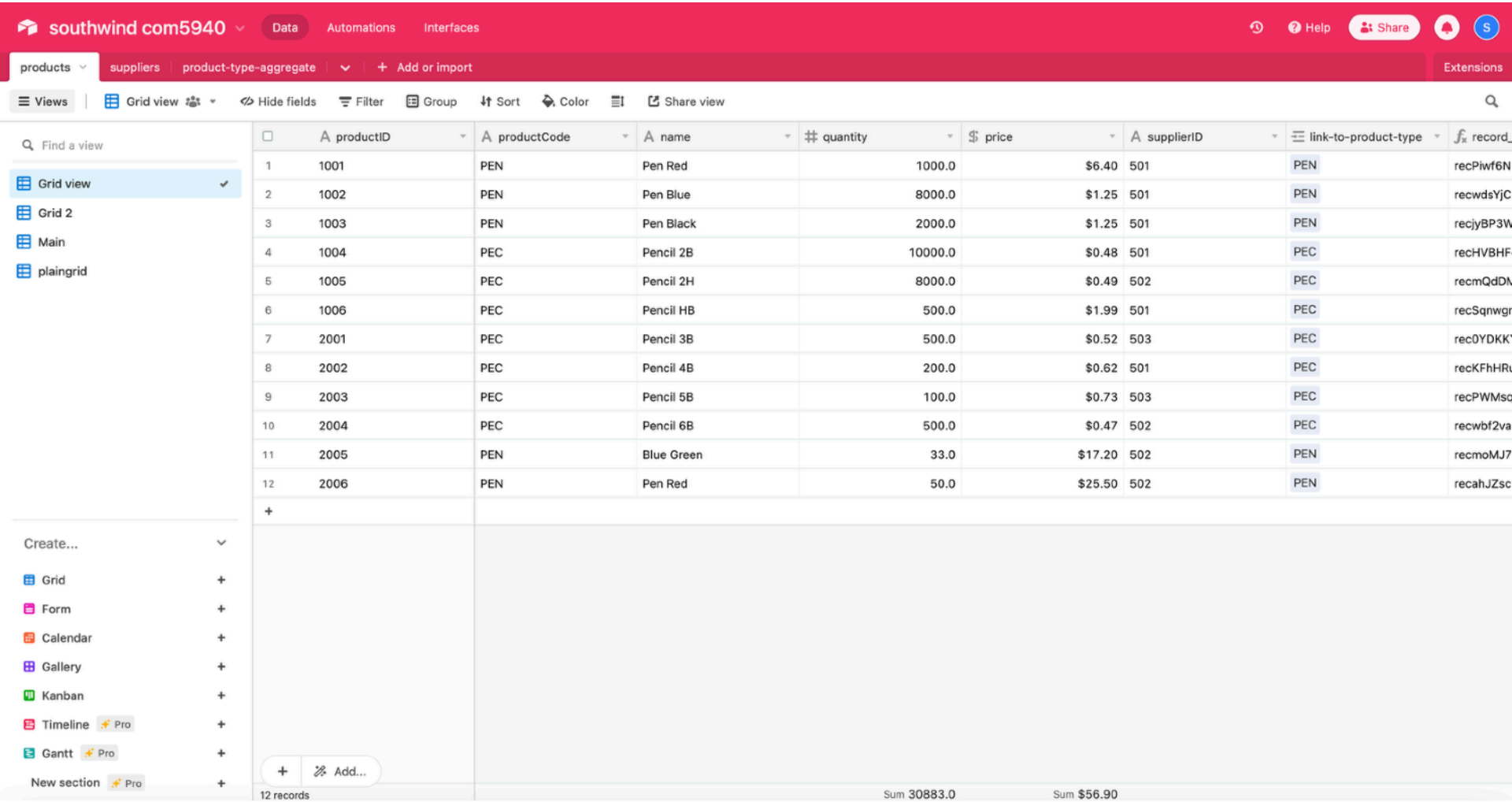
REST API and CRUD support in Airtable


Database
Tables
Records (Rows x Columns)
Views
Filtered by
Grouped by
Sorted By
Gallery
Kaban
Calendar
Form
Information Architecture for User Interaction and Navigation

Airtable API


Airtable
Bases
Templates
Marketplace
Universe
HELP
Join the thousands of teams using Airtable's advanced features
Get the power you need to manage complex workflows, run reports, and more with tools like:
Unlimited apps
More record and attachment space
Gantt view
Team permissions
Upgrade now
Invite your friends and coworkers to earn account credit!
Account
Notification preferences
Upgrade
Contact sales
Tell a friend
Connected accounts
Log out
Interfaces
BETA
Design your first interface
Create clarity for your team by designing actionable interfaces.
Find a base or interface
Interfaces
WORKSPACES
My First Workspace
Tutorials
Add a workspace
Trash
LEARNING AND RESOURCES
Guide to Airtable
Webinars
Help center
Importing
Understanding views
Advanced linking
My First Workspace
Free plan
SHARE
Inventory Tracking
KasPer Lite
Add a base
Tutorials
Project Tracker

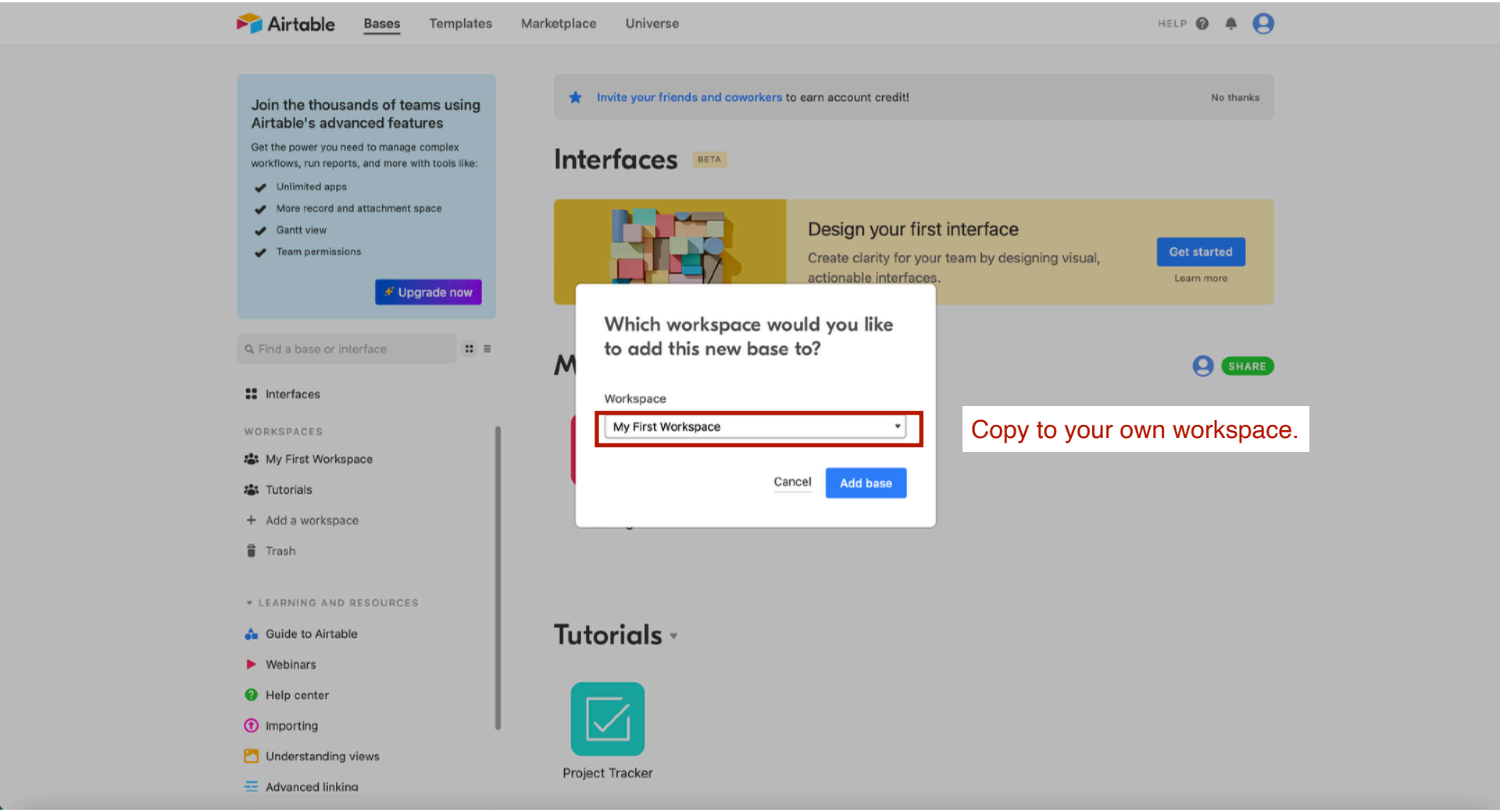

Airtable
HELP
LOG OUT
ACCOUNT
Overview
Referrals and credits
Recent activity
WORKSPACES
My First Workspace
FREE PLAN
Account overview
Sup Port
Edit name
Update password
Set up two-factor authentication
CREDITS
You have $0 in credit.
Get $10 in credit for every person you invite.
Get $2 in credit for installing the mobile app.
Referrals and credits
API
This is your personal API key. It's required in order to use the Airtable API.
Your personal API key has access to all the data in your Airtable bases.
Only share this key with third-party services and applications that you trust.
To limit the access of a third-party service, consider following these instructions to create a read-only API key and sharing that key instead .
Regenerate API key
Delete key
GOOGLE CONTACTS INTEGRATION
Allow Airtable to import your Google Contacts.
Make it easy to invite your contacts to collaborate in Airtable.
GOOGLE DRIVE INTEGRATION



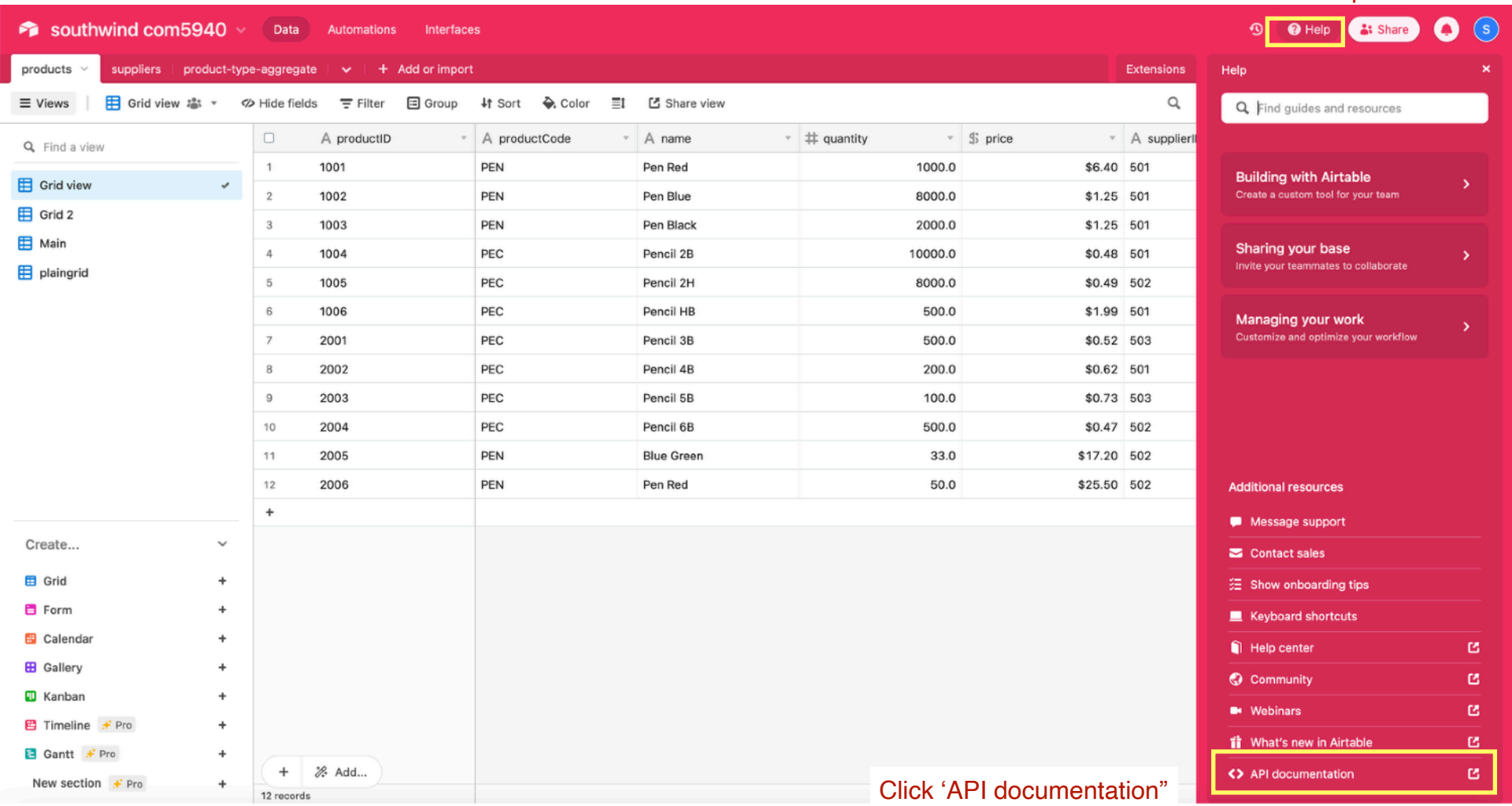
	productID	productCode	name	quantity	price	supplierID	link-to-product-type	record_
1	1001	PEN	Pen Red	1000.0	$6.40	501	PEN	recPiwf6N
2	1002	PEN	Pen Blue	8000.0	$1.25	501	PEN	recwdsYjC
3	1003	PEN	Pen Black	2000.0	$1.25	501	PEN	recjyBP3W
4	1004	PEC	Pencil 2B	10000.0	$0.48	501	PEC	recHVBHF
5	1005	PEC	Pencil 2H	8000.0	$0.49	502	PEC	recmQdDM
6	1006	PEC	Pencil HB	500.0	$1.99	501	PEC	recSqnwgr
7	2001	PEC	Pencil 3B	500.0	$0.52	503	PEC	rec0YDKK
8	2002	PEC	Pencil 4B	200.0	$0.62	501	PEC	recKFhHRu
9	2003	PEC	Pencil 5B	100.0	$0.73	503	PEC	recPWMsc
10	2004	PEC	Pencil 6B	500.0	$0.47	502	PEC	recwbf2va
11	2005	PEN	Blue Green	33.0	$17.20	502	PEN	recmoMJ7
12	2006	PEN	Pen Red	50.0	$25.50	502	PEN	recahJZsc

Click the following link to share the database.

https://airtable.com/shrArG9LXv5vj4pbH


Airtable
Bases
Templates
Marketplace
Universe
HELP
Join the thousands of teams using Airtable's advanced features
Get the power you need to manage complex workflows, run reports, and more with tools like:
Unlimited apps
More record and attachment space
Gantt view
Team permissions
Upgrade now
Invite your friends and coworkers to earn account credit!
No thanks
Interfaces
BETA
Design your first interface
Create clarity for your team by designing visual, actionable interfaces.
Get started
Learn more
Find a base or interface
Interfaces
WORKSPACES
My First Workspace
Tutorials
Add a workspace
Trash
LEARNING AND RESOURCES
Guide to Airtable
Webinars
Help center
Importing
Understanding views
Advanced linking
Which workspace would you like to add this new base to?
Workspace
My First Workspace
Cancel
Add base
SHARE
Copy to your own workspace.
Tutorials
Project Tracker

Click ‘Help”



Click ‘API documentation”

So

Checkbox click ‘show API key”

Open base

- INTRODUCTION
- METADATA
- RATE LIMITS
- AUTHENTICATION
- PRODUCTS TABLE
  - Fields
  - List records
  - Retrieve a record
  - Create records
  - Update records
  - Delete records
- SUPPLIERS TABLE
- PRODUCT-TYPE-AGGREGATE TABLE
- ERRORS

List products records

To list records in `products`, issue a **GET** request to the `products` endpoint. Note that table names and table ids can be used interchangeably. Using table ids means table name changes do not require modifications to your API request.

Returned records do not include any fields with "empty" values, e.g. `""`, `[]`, or `false`.

You can filter, sort, and format the results with the following query parameters. Note that these parameters need to be URL encoded. You can use our API URL encoder tool to help with this. If you are using a helper library like Airtable.js, these parameters will be automatically encoded.

Note: Airtable's API only accepts request with a URL shorter than 16,000 characters. Encoded formulas may cause your requests to exceed this limit. To fix this issue you can instead make a POST request to `/v0/{baseId}/{tableIdOrName}/listRecords` while passing the parameters within the body of the request instead of the query parameters. See our support article on this for more information.

fields
`array of strings`
optional

Only data for fields whose names are in this list will be included in the result. If you don't need every field, you can use this parameter to reduce the amount of data transferred.

For example, to only return data from `productID` and `productCode`, send these two query parameters:

`fields%5B%5D=productID&fields%5B%5D=productCode`

You can also perform the same action with field ids (they can be found in the fields section):

`fields%5B%5D=fldBWQ5jhL2MRTPUv&fields%5B%5D=fldspti48PE3H6`

Note: `%5B%5D` may be omitted when specifying multiple fields, but must always be included when specifying only a single field.

filterByFormula
`string`
optional

A formula used to filter records. The formula will be evaluated for each record, and if the result is not `0`, `false`, `""`, `NaN`, `[]`, or `#Error!` the record will be included in the response. We recommend testing your

curl | JavaScript

Here is your end-point.

EXAMPLE REQUEST

```
curl "https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products?maxRecords=3&view=Grid%20view" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

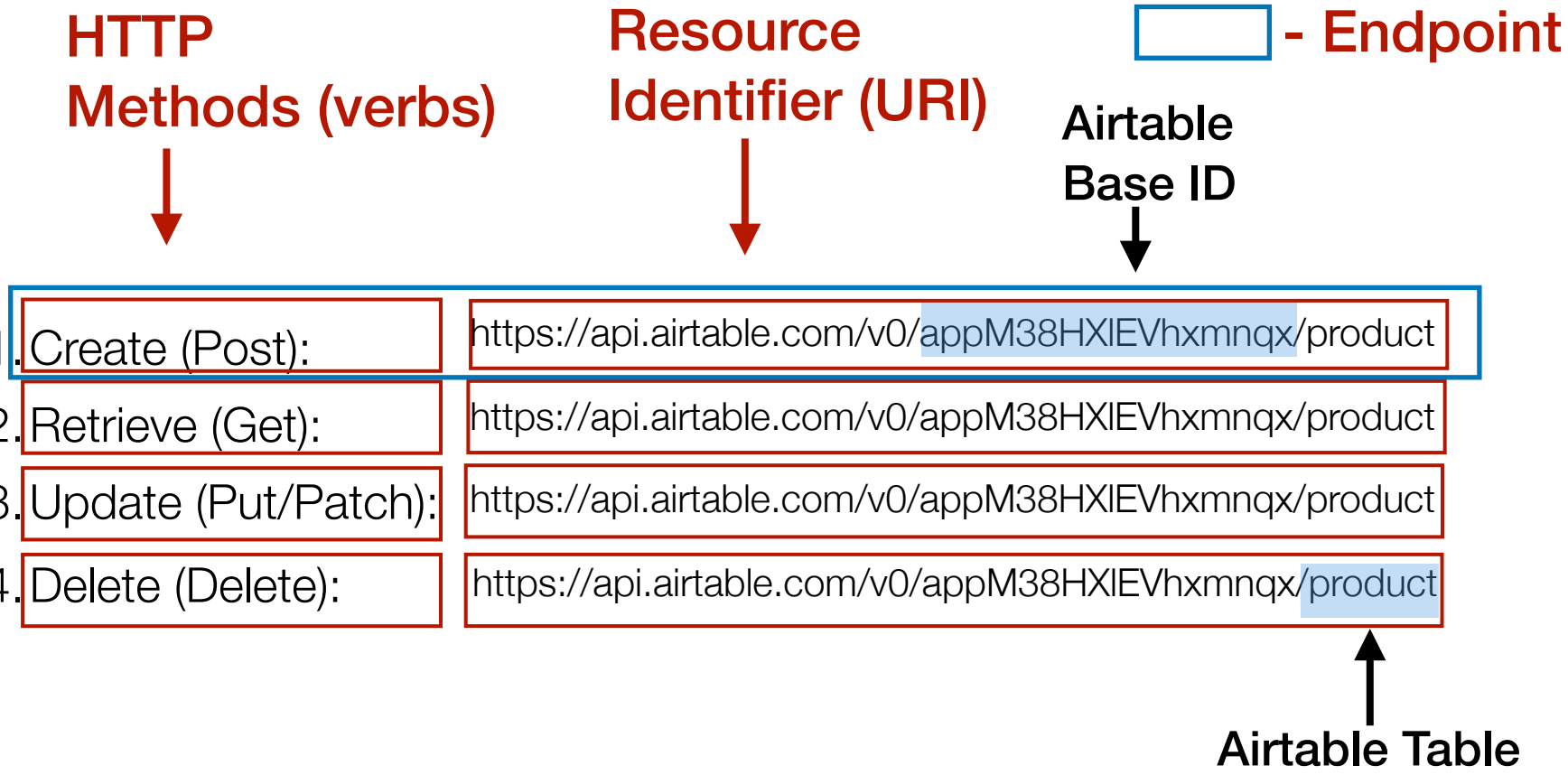
Here is your key.

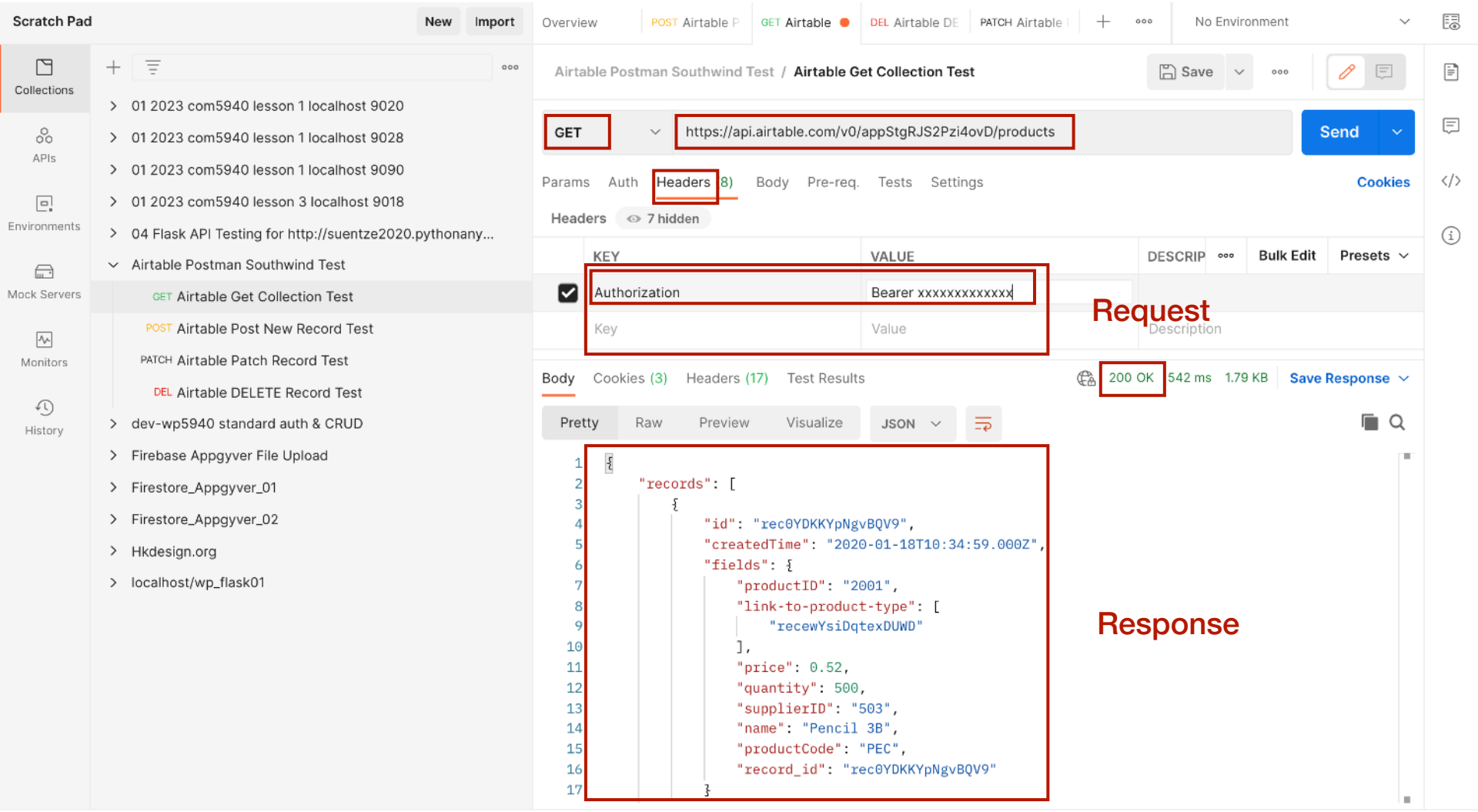
EXAMPLE RESPONSE

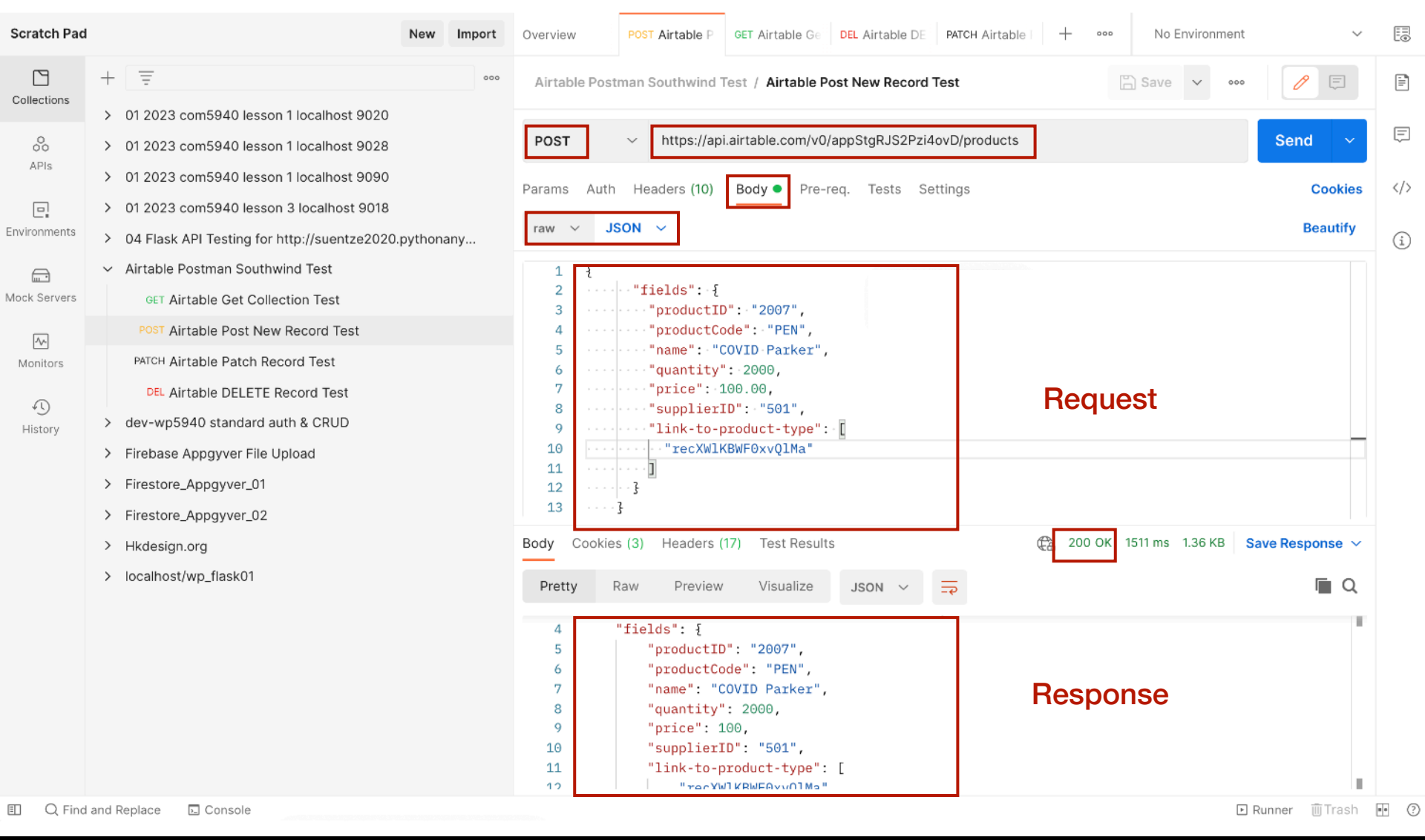

```
{
    "records": [
        {
            "id": "recPiwf6NEqheccgC",
            "createdTime": "2020-01-18T10:34:59.000Z",
            "fields": {
                "productID": "1001",
                "link-to-product-type": [
                    "recXWlKBWF0xvQlMa"
                ],
                "price": 6.4,
                "quantity": 1000,
                "supplierID": "501",
                "name": "Pen Red",
                "productCode": "PEN",
                "record_id": "recPiwf6NEqheccgC"
            }
        },
        {
            "id": "recwdsYjC7OASwEfU",
            "createdTime": "2020-01-18T10:34:59.000Z",
            "fields": {
                "productID": "1002",
                "link-to-product-type": [
                    "recXWlKBWF0xvQlMa"
                ],
                "price": 1.25,
                "quantity": 8000,
                "supplierID": "501",
                "name": "Pen Blue",
                "productCode": "PEN",
                "record_id": "recwdsYjC7OASwEfU"
            }
```

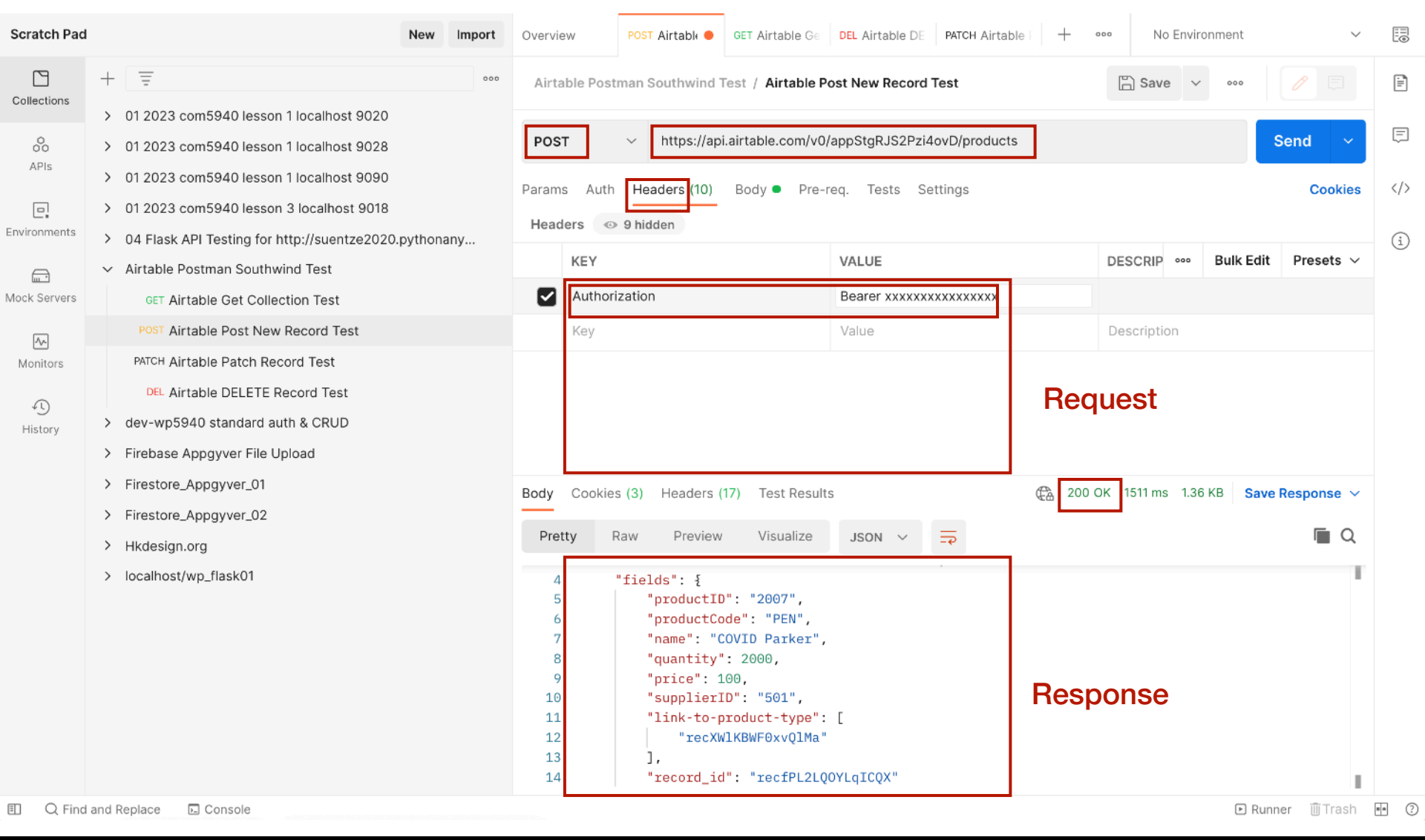

Fields JSON data

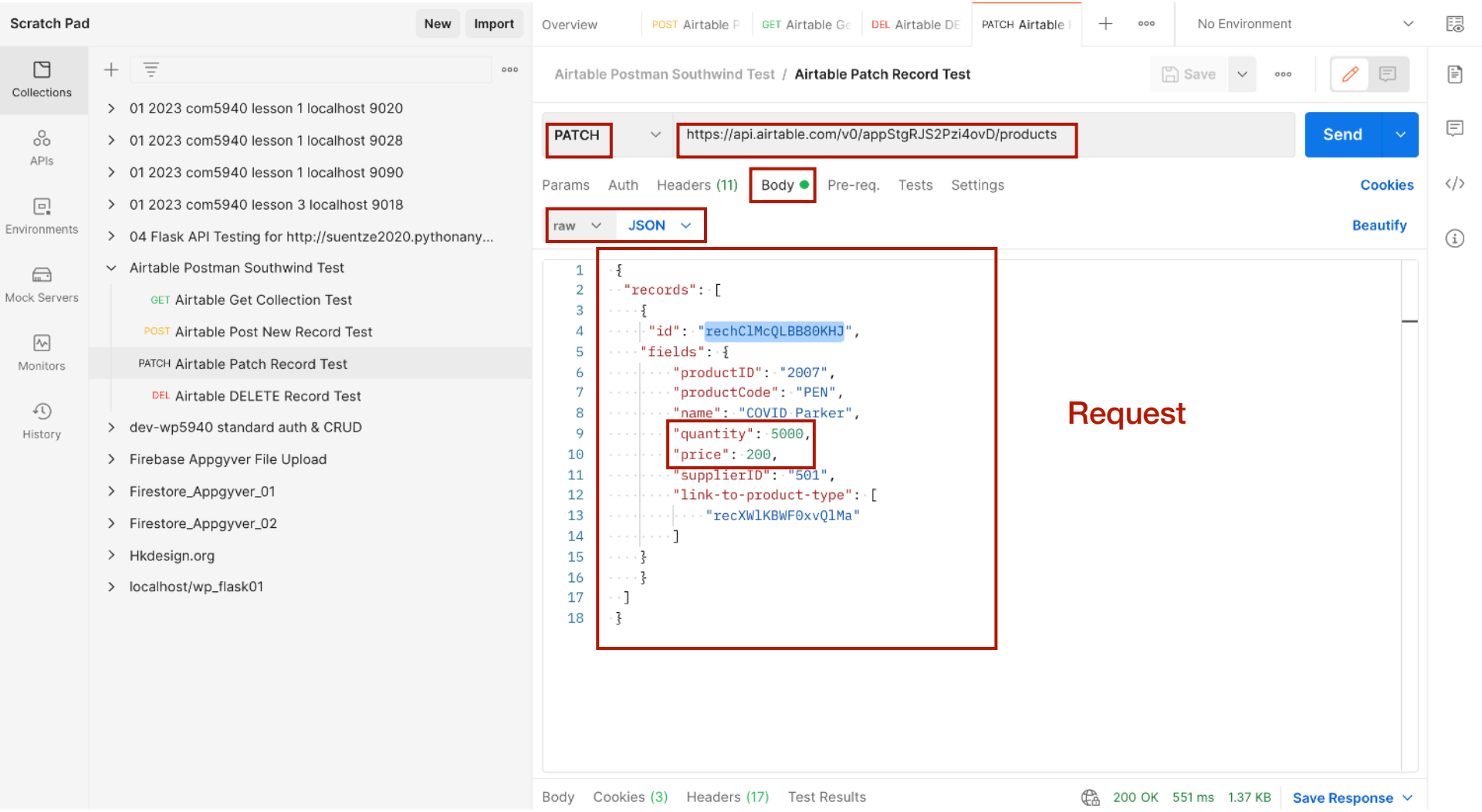
Basic RESTful Methods and Structure for Airtable

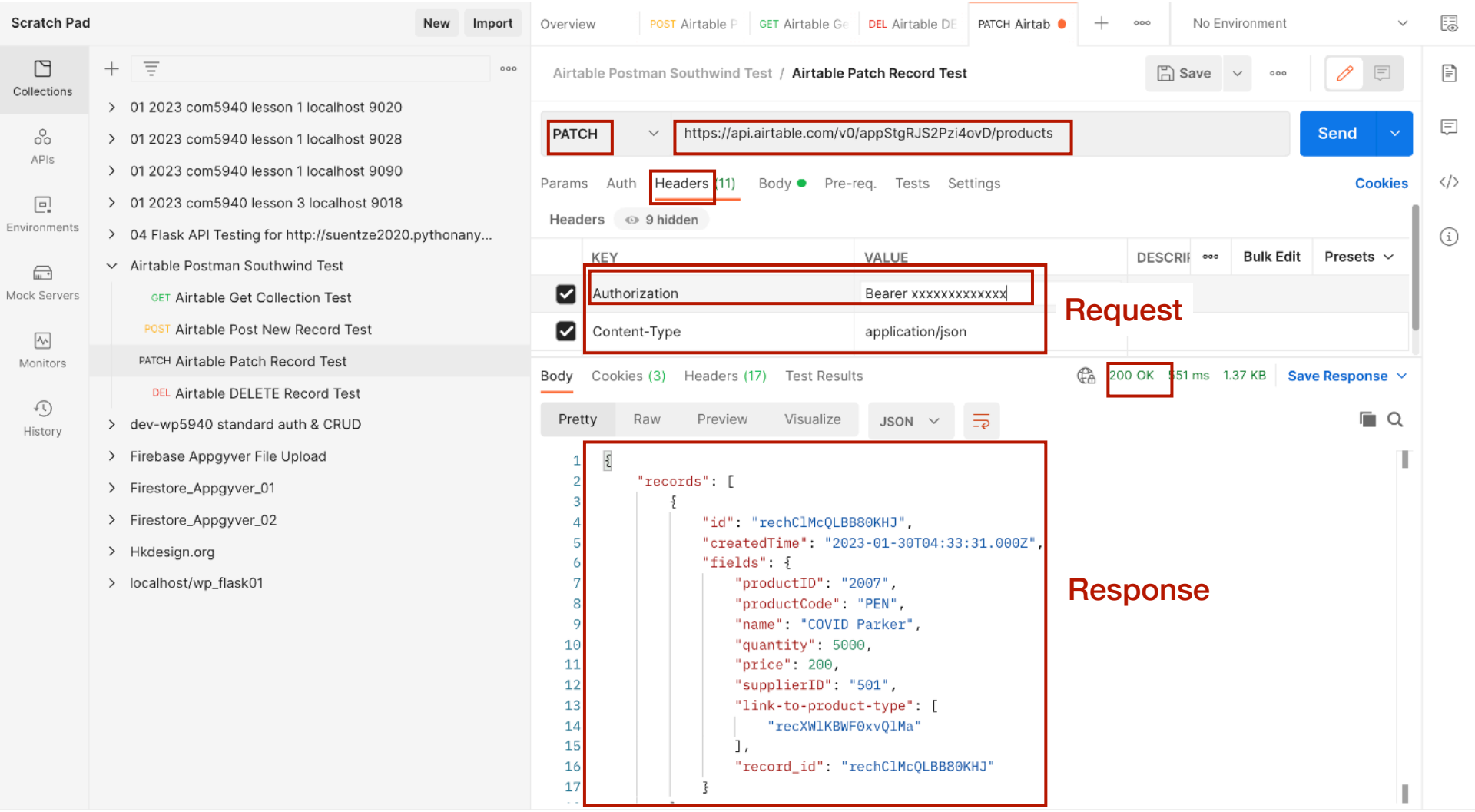


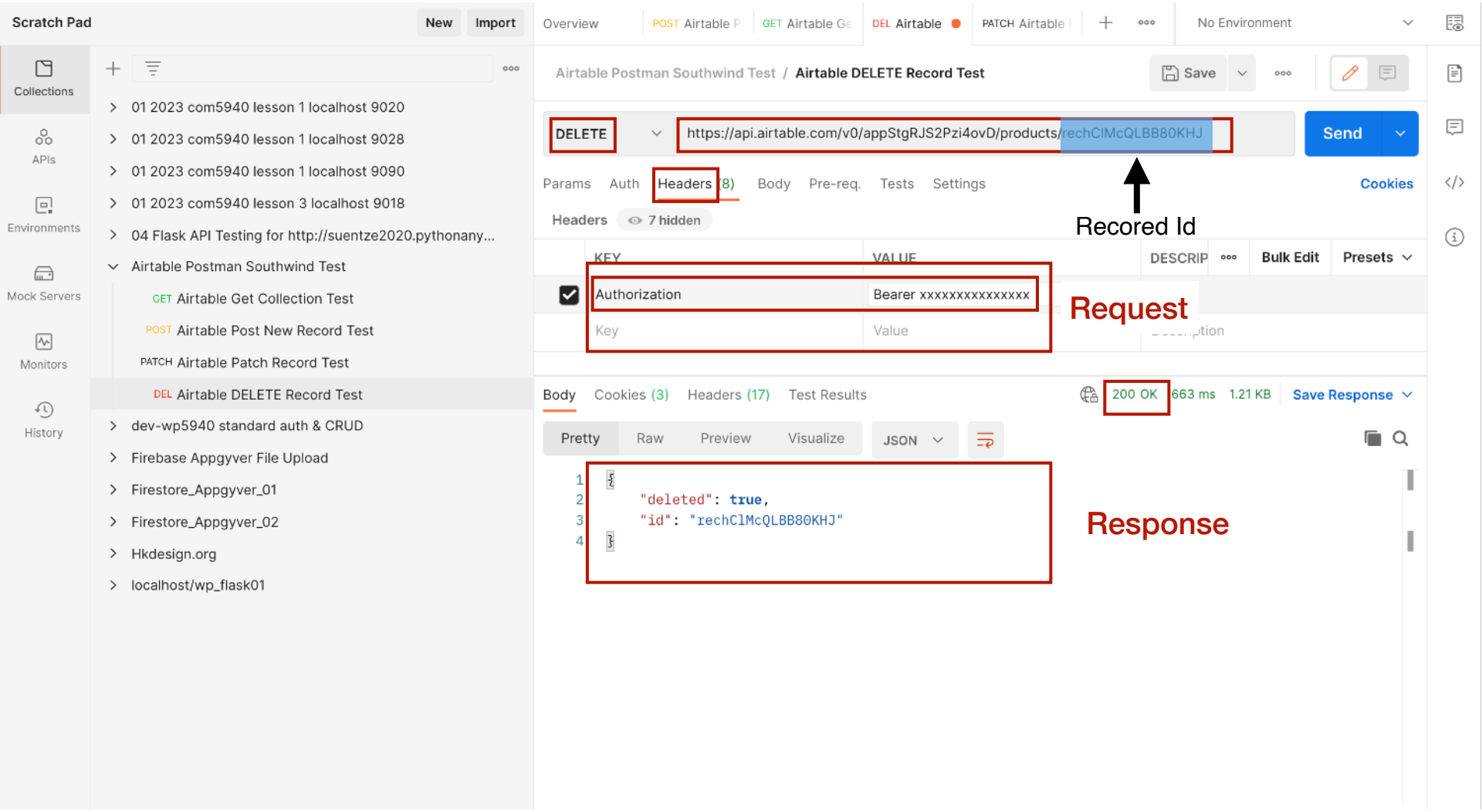

Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Overview
POST Airtable P
GET Airtable
DEL Airtable DE
PATCH Airtable
No Environment
Airtable Postman Southwind Test / Airtable Get Collection Test
Save
GET
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
Headers
7 hidden
KEY
VALUE
DESCRIP
Bulk Edit
Presets
Authorization
Bearer xxxxxxxxxxxx
Key
Value
Description
Request
Body
Cookies (3)
Headers (17)
Test Results
200 OK
542 ms
1.79 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"records": [
{
"id": "rec0YDKKYpNgvBQV9",
"createdTime": "2020-01-18T10:34:59.000Z",
"fields": {
"productID": "2001",
"link-to-product-type": [
"recewYsiDqtexDUWD"
],
"price": 0.52,
"quantity": 500,
"supplierID": "503",
"name": "Pencil 3B",
"productCode": "PEC",
"record_id": "rec0YDKKYpNgvBQV9"
}
Response


Scratch Pad
New
Import
Overview
POST Airtable P
GET Airtable Ge
DEL Airtable DE
PATCH Airtable
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Airtable Postman Southwind Test / Airtable Post New Record Test
Save
POST
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products
Send
Params
Auth
Headers (10)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
{
"fields": {
"productID": "2007",
"productCode": "PEN",
"name": "COVID Parker",
"quantity": 2000,
"price": 100.00,
"supplierID": "501",
"link-to-product-type": [
"recXWlKBWF0xvQlMa"
]
}
}
Request
Body
Cookies (3)
Headers (17)
Test Results
200 OK
1511 ms
1.36 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
"fields": {
"productID": "2007",
"productCode": "PEN",
"name": "COVID Parker",
"quantity": 2000,
"price": 100,
"supplierID": "501",
"link-to-product-type": [
"recXWlKBWF0xvQlMa"
Response
Find and Replace
Console
Runner
Trash


Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Overview
No Environment
Airtable Postman Southwind Test / Airtable Post New Record Test
Save
POST
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products
Send
Params
Auth
Headers (10)
Body
Pre-req.
Tests
Settings
Cookies
Headers
9 hidden
KEY
VALUE
DESCRIP
Bulk Edit
Presets
Authorization
Bearer xxxxxxxxxxxxxxx
Key
Value
Description
Request
Body
Cookies (3)
Headers (17)
Test Results
200 OK
1511 ms
1.36 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
"fields": {
"productID": "2007",
"productCode": "PEN",
"name": "COVID Parker",
"quantity": 2000,
"price": 100,
"supplierID": "501",
"link-to-product-type": [
"recXWlKBWF0xvQlMa"
],
"record_id": "recfPL2LQOYLqICQX"
Response
Find and Replace
Console
Runner
Trash


Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Overview
POST Airtable P
GET Airtable Ge
DEL Airtable DE
PATCH Airtable
No Environment
Airtable Postman Southwind Test / Airtable Patch Record Test
Save
PATCH
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products
Send
Params
Auth
Headers (11)
Body
Pre-req.
Tests
Settings
Cookies
raw
JSON
Beautify
{
"records": [
{
"id": "rechClMcQLBB80KHJ",
"fields": {
"productID": "2007",
"productCode": "PEN",
"name": "COVID Parker",
"quantity": 5000,
"price": 200,
"supplierID": "501",
"link-to-product-type": [
"recXWlKBWF0xvQlMa"
]
}
}
]
}
Request
Body
Cookies (3)
Headers (17)
Test Results
200 OK
551 ms
1.37 KB
Save Response


Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Overview
POST Airtable P
GET Airtable Ge
DEL Airtable DE
PATCH Airtab
No Environment
Airtable Postman Southwind Test / Airtable Patch Record Test
Save
PATCH
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products
Send
Params
Auth
Headers (11)
Body
Pre-req.
Tests
Settings
Cookies
Headers
9 hidden
KEY
VALUE
DESCRI
Bulk Edit
Presets
Authorization
Bearer xxxxxxxxxxxx
Content-Type
application/json
Request
Body
Cookies (3)
Headers (17)
Test Results
200 OK
951 ms
1.37 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"records": [
{
"id": "rechClMcQLBB80KHJ",
"createdTime": "2023-01-30T04:33:31.000Z",
"fields": {
"productID": "2007",
"productCode": "PEN",
"name": "COVID Parker",
"quantity": 5000,
"price": 200,
"supplierID": "501",
"link-to-product-type": [
"recXWlKBWF0xvQlMa"
],
"record_id": "rechClMcQLBB80KHJ"
}
Response


Scratch Pad
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
01 2023 com5940 lesson 1 localhost 9020
01 2023 com5940 lesson 1 localhost 9028
01 2023 com5940 lesson 1 localhost 9090
01 2023 com5940 lesson 3 localhost 9018
04 Flask API Testing for http://suentze2020.pythonany...
Airtable Postman Southwind Test
GET Airtable Get Collection Test
POST Airtable Post New Record Test
PATCH Airtable Patch Record Test
DEL Airtable DELETE Record Test
dev-wp5940 standard auth & CRUD
Firebase Appgyver File Upload
Firestore_Appgyver_01
Firestore_Appgyver_02
Hkdesign.org
localhost/wp_flask01
Overview
POST Airtable P
GET Airtable Ge
DEL Airtable
PATCH Airtable
No Environment
Airtable Postman Southwind Test / Airtable DELETE Record Test
Save
DELETE
https://api.airtable.com/v0/appStgRJS2Pzi4ovD/products/rechClMcQLBB80KHJ
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
Cookies
Headers
7 hidden
Recored Id
KEY
VALUE
DESCRIP
Bulk Edit
Presets
Authorization
Bearer xxxxxxxxxxxxxx
Request
Key
Value
Body
Cookies (3)
Headers (17)
Test Results
200 OK
663 ms
1.21 KB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"deleted": true,
"id": "rechClMcQLBB80KHJ"
}
Response

Airtable REST API in Flask

Retrieve Airtable Records

In [3]:

```
1  import requests
2  import pandas as pd
3  headers = {
4  'Authorization': 'Bearer Keyxxxxxxxxxxxxxx',
5  }
6
7  params = (
8  ('view', 'Grid view'),
9  )
10 # Using the Southwind Base stored in the COM5940 workspace in Firefox Airtable Account (bernard@intechnigence.com)
11 r = requests.get('https://api.airtable.com/v0/appswwta5yYhNVokp/products?api_key=keyXxxxxxxxxxxxxxxx', headers=header
12 print("Status Code:",r.status_code)
13 dict = r.json()
14 dataset = []
15 productID = []
16 productCode = []
17 result = []
18 for data in dict['records']:
19      field_content = data['fields']
20      dataset.append(field_content)
21 print(dataset)
22 print('\n')
23 print(pd.DataFrame(dataset))
24 print('\n')
25 for field in dataset:
26     productID.append(field.get('productID'))
27     productCode.append(field.get('productCode'))
28     result.append(field.get('name'))
29 print(result)
30 print(pd.DataFrame(result))
```

GET method

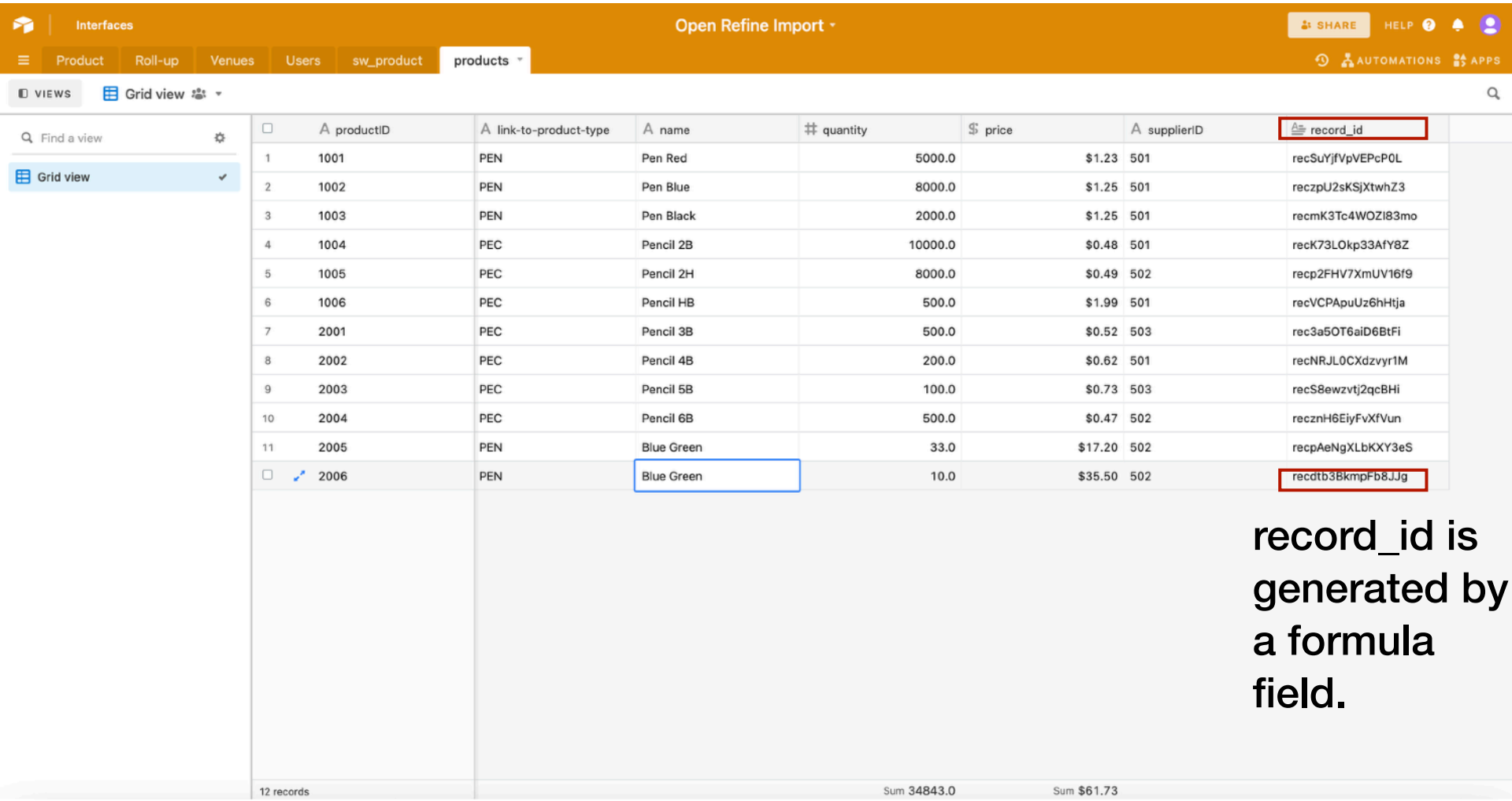
```
Status Code: 200
[{'name': 'Pen Red', 'link-to-product-type': 'PEN', 'productCode': 'PEN', 'record_id': 'recSuYjfVpVEPcP0L', 'price':
1.23, 'quantity': 5000, 'productID': '1001', 'supplierID': '501'}, {'name': 'Pen Blue', 'link-to-product-type': 'PE
N', 'productCode': 'PEN', 'record_id': 'reczpU2sKSjXtwhZ3', 'price': 1.25, 'quantity': 8000, 'productID': '1002', 'su
```

Create Airtable Record

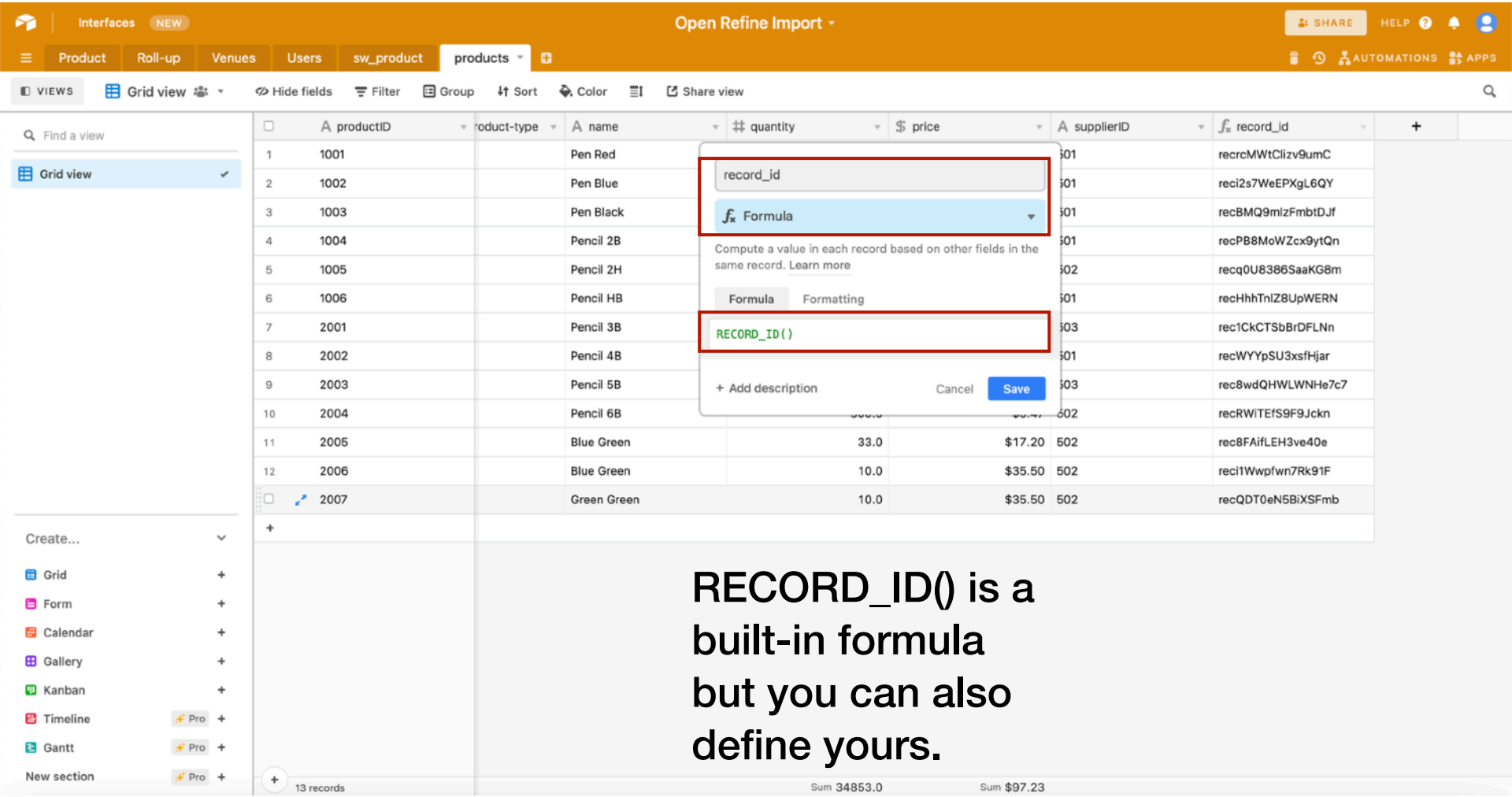
In [7]:

```
#
# Add Record
#
# Using the Southwind Base stored in the COM5940 workspace in Firefox Airtable Account (bernard@intechnigence.com)
#

import requests
import pandas as pd

productID = '2007'
productCode = 'PEN'
name =  'Green Green'
quantity = 10
price = 35.5
supplierID = '502'
link_to_product_type = 'PEN'

mydict =  {
    "productID": productID,
    "productCode": productCode,
    "name": name,
    "quantity": quantity,
    "price": price,
    "supplierID": supplierID,
    "link-to-product-type": link_to_product_type
    }

data = {"fields": mydict}
headers = {'Authorization': 'Bearer keyxxxxxxxxx', 'Content-Type': 'application/json; charset=utf-8'}
# r = requests.post(url = API_ENDPOINT, data = data)
r = requests.post('https://api.airtable.com/v0/appxxxxxxxxxx/products',json=data,headers=headers)
print("Status Code:",r.status_code)
print(r.text)

params = (
('view', 'Grid view'),
)
# Using the Southwind Base stored in the COM5940 workspace
r = requests.get('https://api.airtable.com/v0/appxxxxxxxxxx/products?api_key=keyxxxxxxxxx&sortField=productID&sortDi
print("Status Code:",r.status_code)
```

POST method



record_id is generated by a formula field.



RECORD_ID() is a built-in formula but you can also define yours.

Update Airtable Record

In [10]:

```
#
# Update Records
#
# Using the Southwind Base stored in the COM5940 workspace in Firefox Airtable Account (bernard@intechnigence.com)
#

import requests
import pandas as pd

#productID = '2007'
#productCode = 'PEN'
#name =  'Pen Green'
#quantity = 10
#price = 25.5
#supplierID = '502'

data = {
  "records": [
    {
      "id": "recQDT0eN5BiXSFmb",
      "fields": {
        "productID": "2007",
        "productCode": 'PEC',          # From PEN to PEC
        "name": 'Pen Red',             # From Green Green to Pen Red
        "quantity": 50,                # From 10 to 50
        "price": 25.50,                # From 35.50 to 25.50
        "supplierID": '501',           # From 502 to 501
        "link-to-product-type": 'PEC' # From PEN to PEC
      }
    }
  ] # end records
} # end data
headers = {'Authorization': 'Bearer keyxxxxxxxxxx', 'Content-Type': 'application/json; charset=utf-8'}
r = requests.put('https://api.airtable.com/v0/appxxxxxxxxx/products',json=data,headers=headers)
print("Status Code:",r.status_code)
print(r.text)
params = (
('view', 'Grid view'),
)
# Using the Southwind Base stored in the COM5940 workspace
r = requests.get('https://api.airtable.com/v0/appxxxxxxxxxxx/products?api_key=keyxxxxxxxxxx&sortField=productID&s
print("Status Code:",r.status_code)
```

Record Id of record to be updated.

PUT method

Delete Airtable Record

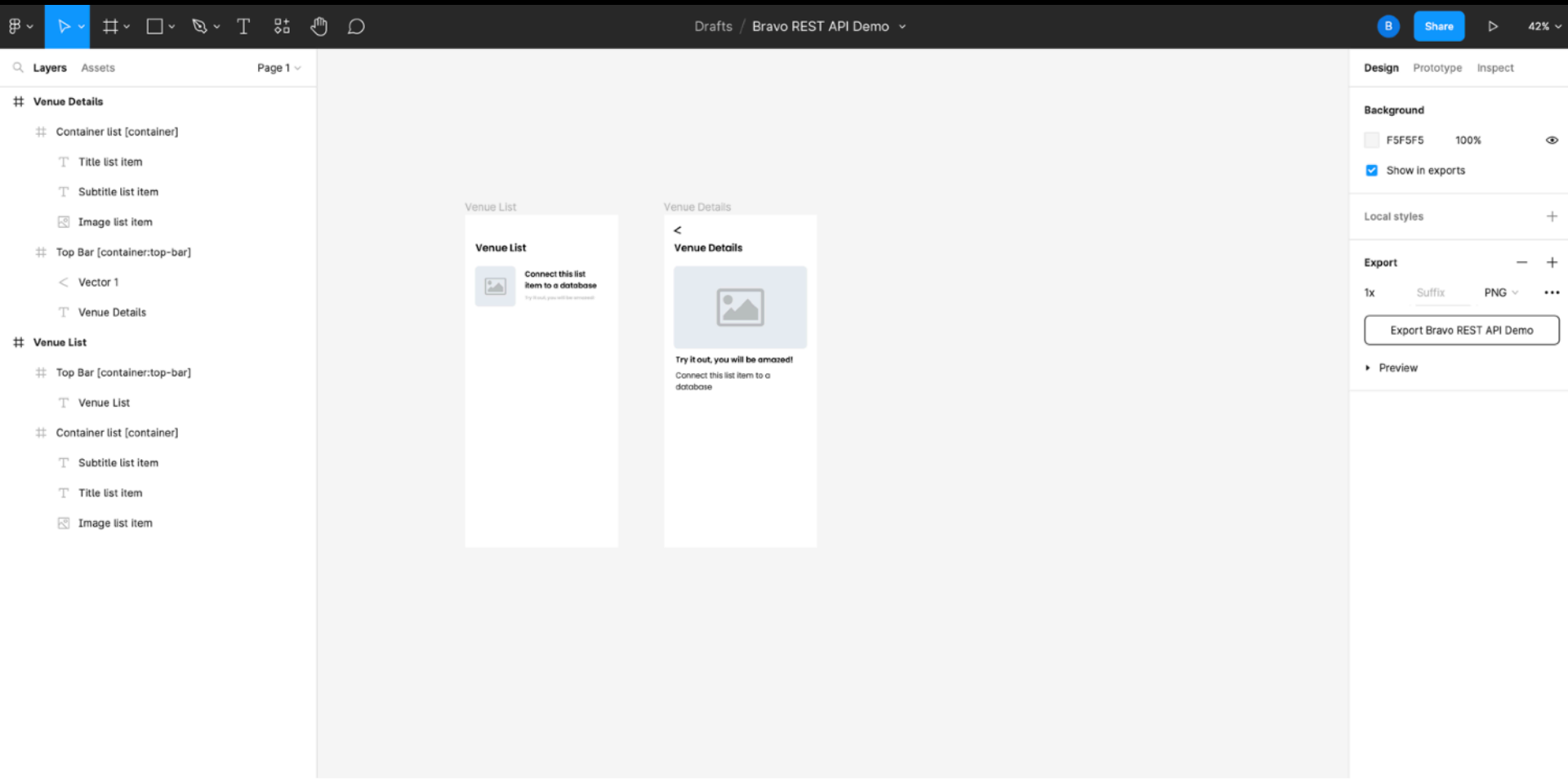
In [11]:

```
#
# Delete Record
#
# Using the Southwind Base stored in the COM5940 workspace in Firefox Airtable Account (bernard@intechnigence.com)
#
import requests
import pandas as pd

headers = {'Authorization': 'Bearer keyxxxxxxxxx', 'Content-Type': 'application/x-www-form-urlencoded'}
r = requests.delete('https://api.airtable.com/v0/appsxxxxxxxxxx/products/recQDT0eN5BiXSFmb',headers=headers)
print("Status Code:",r.status_code)
print(r.text)

params = (
('view', 'Grid view'),
)
# Using the Southwind Base stored in the COM5940 workspace
r = requests.get('https://api.airtable.com/v0/appxxxxxxxxx/products?api_key=keyxxxxxxxxx&sortField=productID&sortI
print("Status Code:",r.status_code)
dict = r.json()
dataset = []
productID = []
productCode = []
result = []
for data in dict['records']:
     field_content = data['fields']
     dataset.append(field_content)
print(dataset)
print('\n')
print(pd.DataFrame(dataset))
```

DELETE method

Record Id of record to be deleted.

Build mobile app with CRUD and authentication and authorisation support in Bravo Studio.

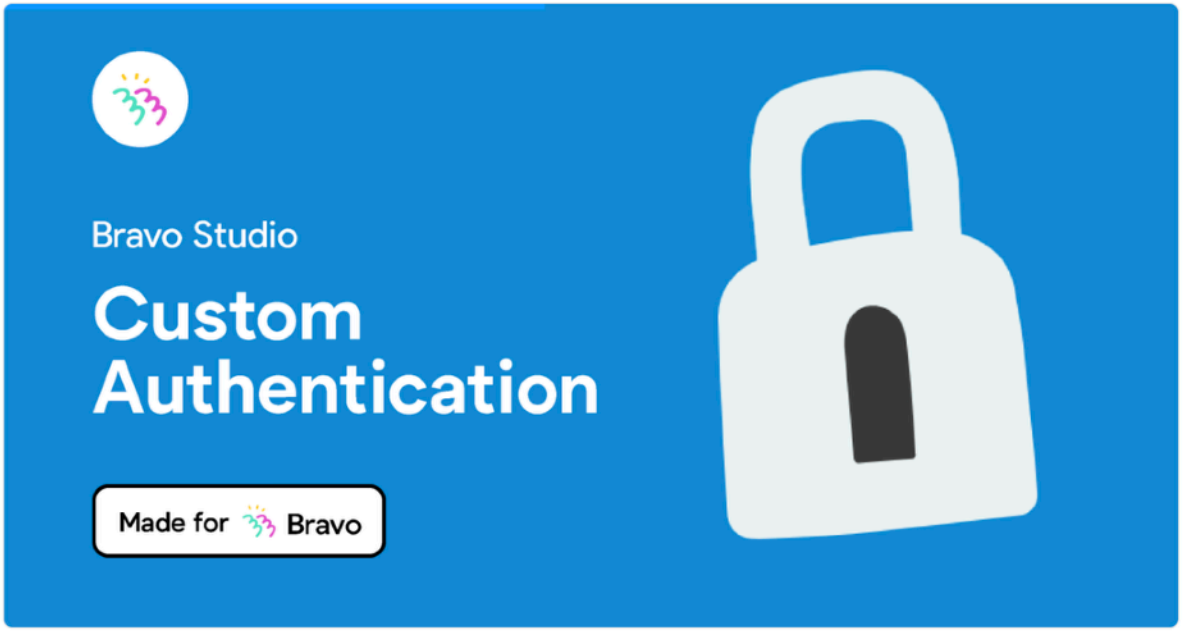

Drafts / Bravo REST API Demo
Share
42%
Layers
Assets
Page 1
Venue Details
Container list [container]
Title list item
Subtitle list item
Image list item
Top Bar [container:top-bar]
Vector 1
Venue Details
Venue List
Top Bar [container:top-bar]
Venue List
Container list [container]
Subtitle list item
Title list item
Image list item
Design
Prototype
Inspect
Background
F5F5F5
100%
Show in exports
Local styles
Export
1x
Suffix
PNG
Export Bravo REST API Demo
Preview
Venue List
Venue List
Connect this list item to a database
Venue Details
Venue Details
Try it out, you will be amazed!
Connect this list item to a database

Community | Search Community | Publish | B

Bravo Sample: Custom Authentication

By Bravo Studio

5 | Get a copy 280



https://www.figma.com/community/file/1046060070731163579

Drafts / Bravo Sample: Custom Authentication (Community)

Share

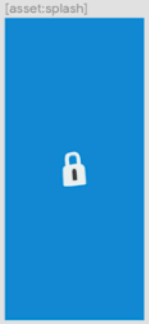
Layers Assets App pages

Pages

- App pages
- Thumbnail [skip]
- Venue Details
- Venue List
- [asset:icon]
- [asset:splash]
- Sign up
- Home page
- Login [page:login]

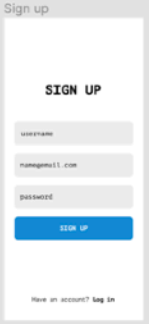
[asset:icon]

[asset:splash]



Login [page:login]

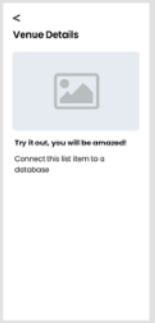
Sign up



Home page

Venue List

Venue Details



Design Prototype Inspect

Background

E5E5E5 100%

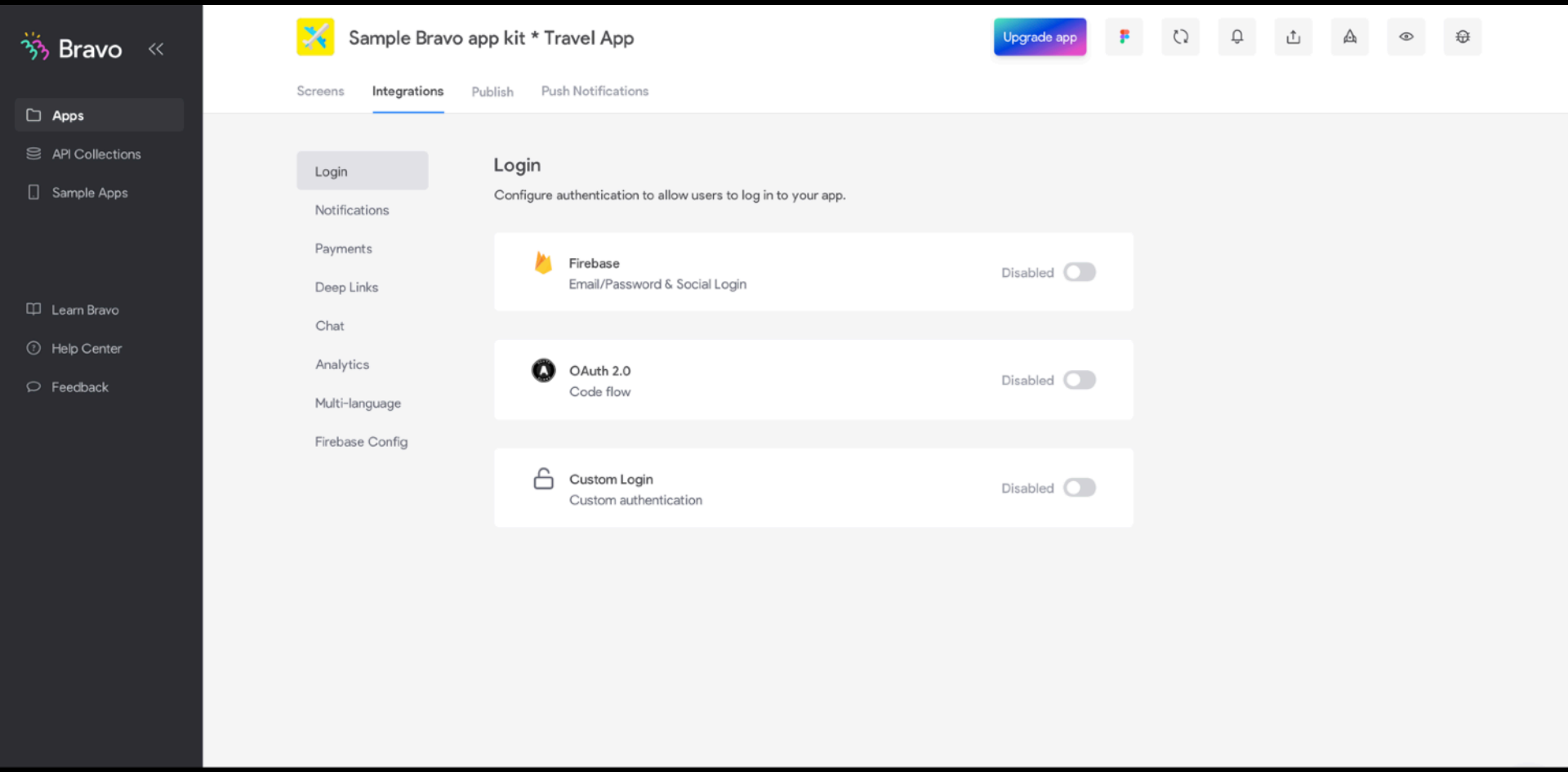
Local styles

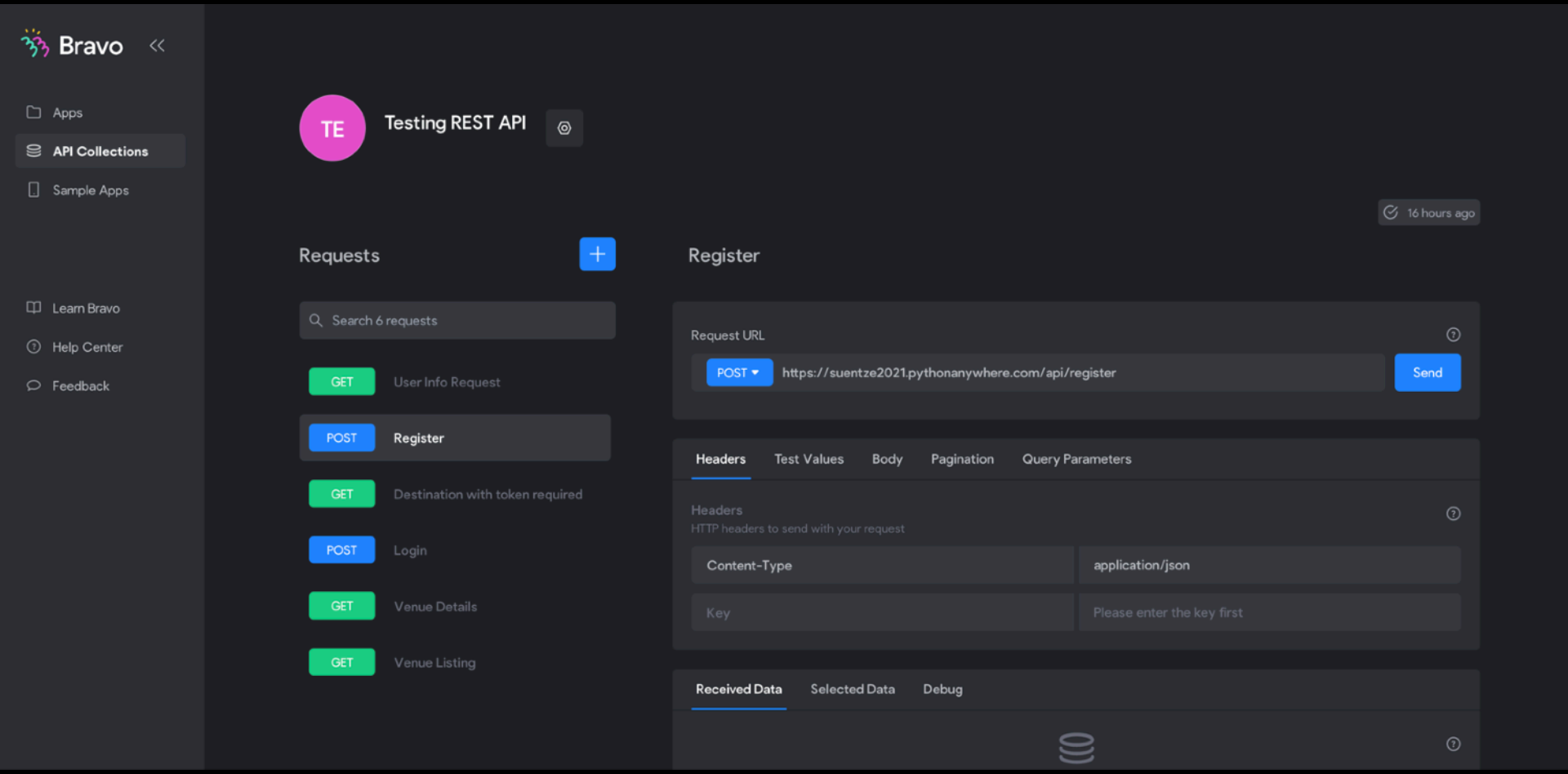
Export

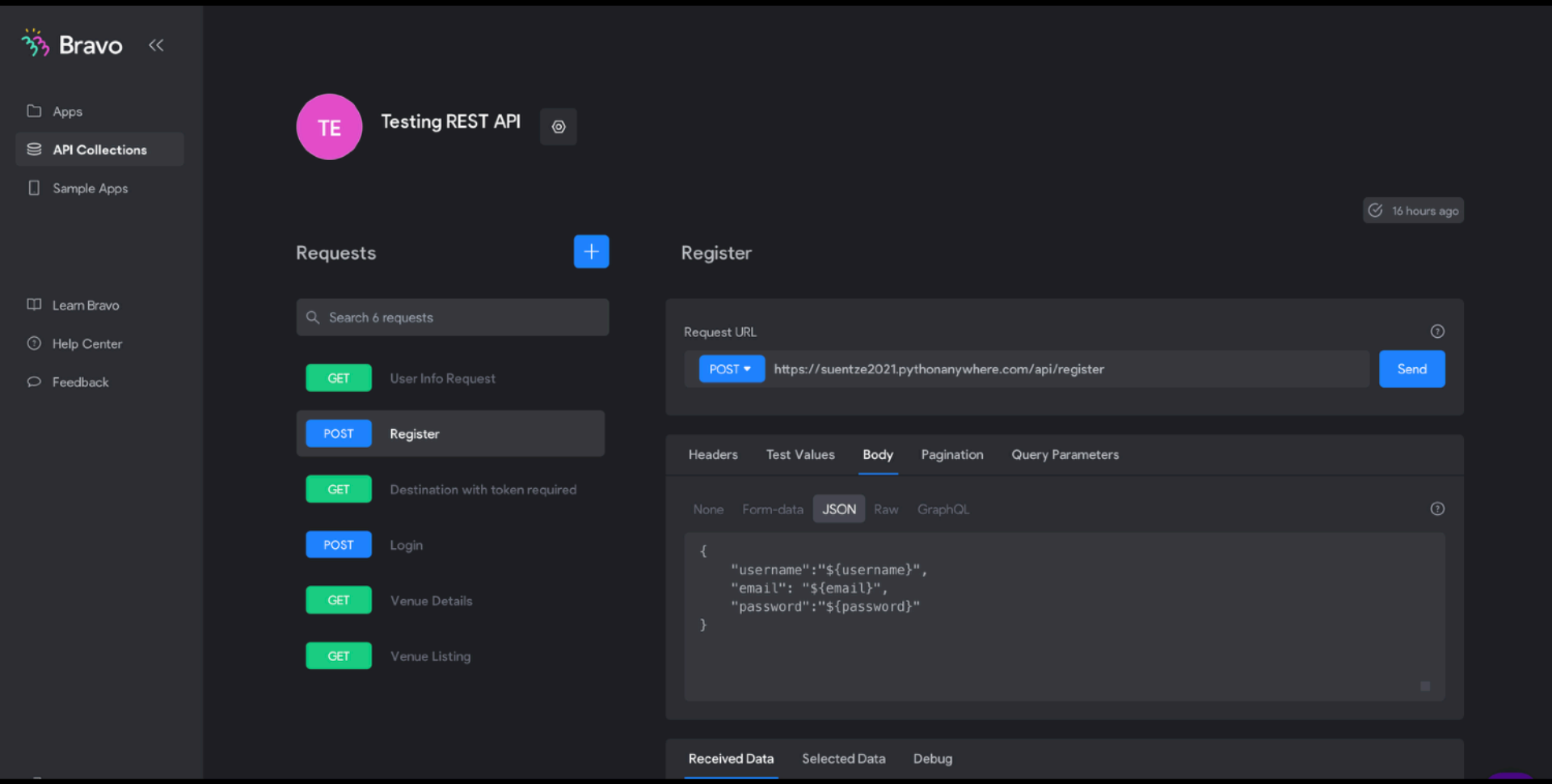
Plugin

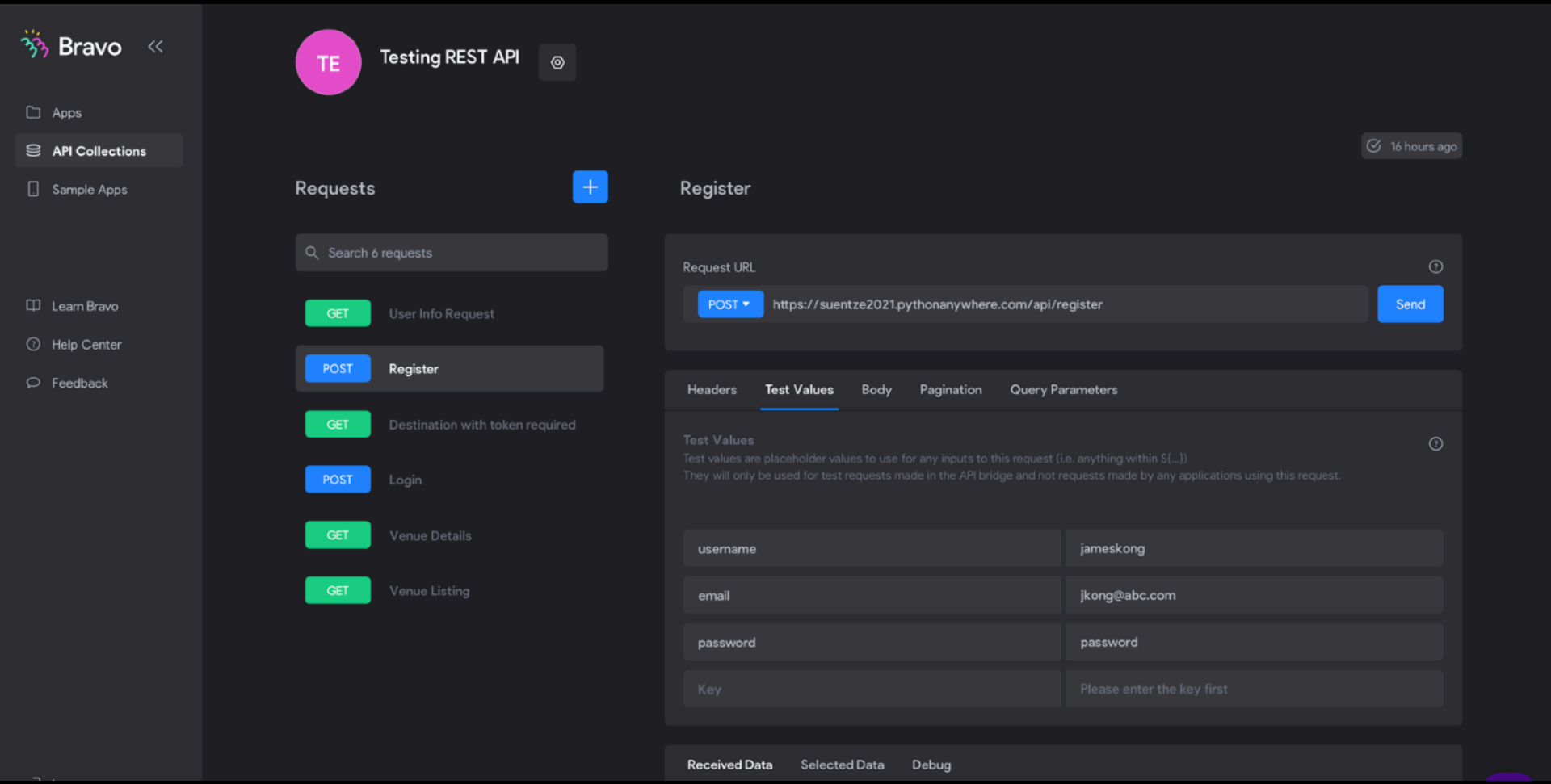
Convertify

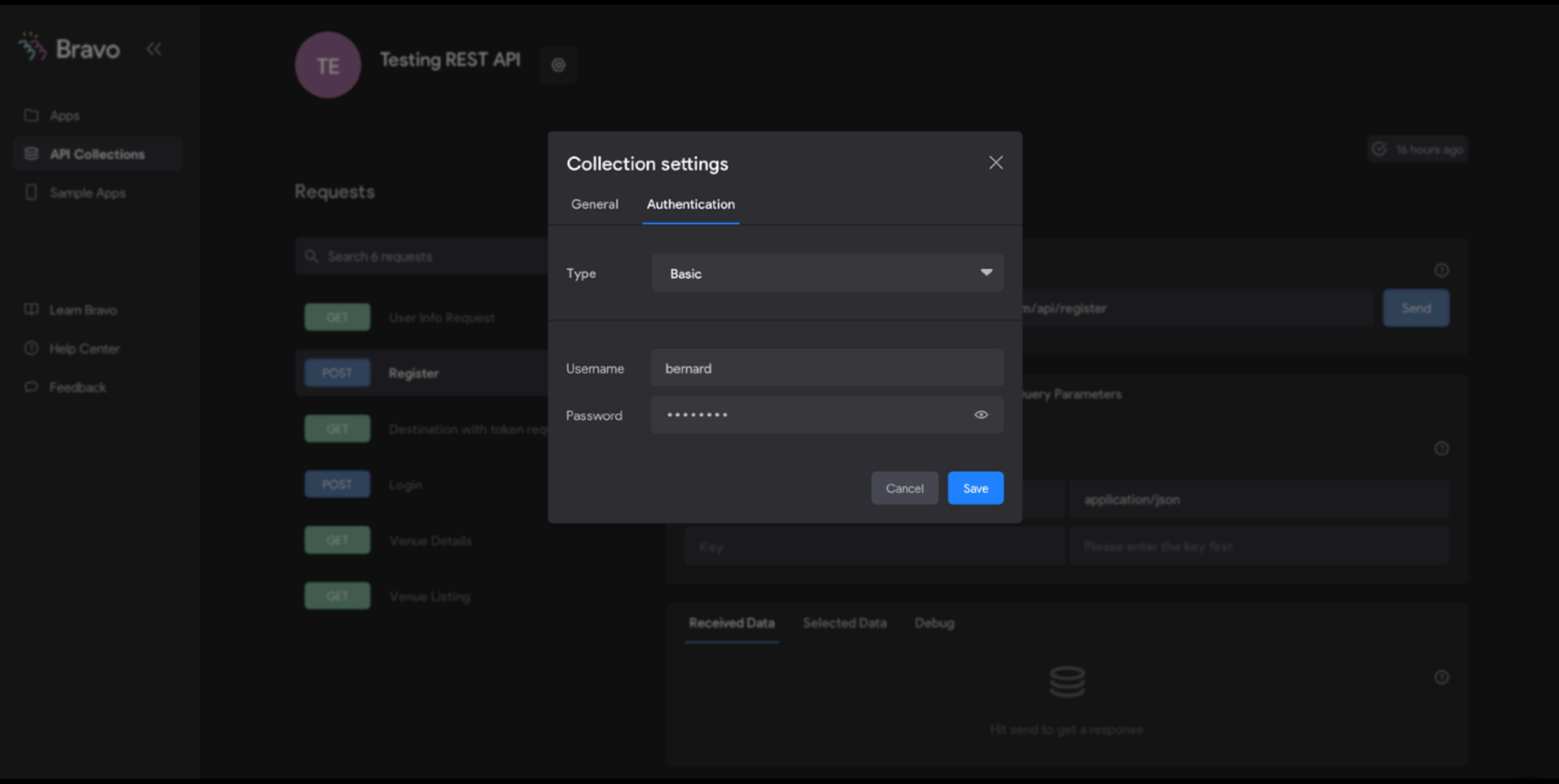
Convert to Sketch/Adobe XD

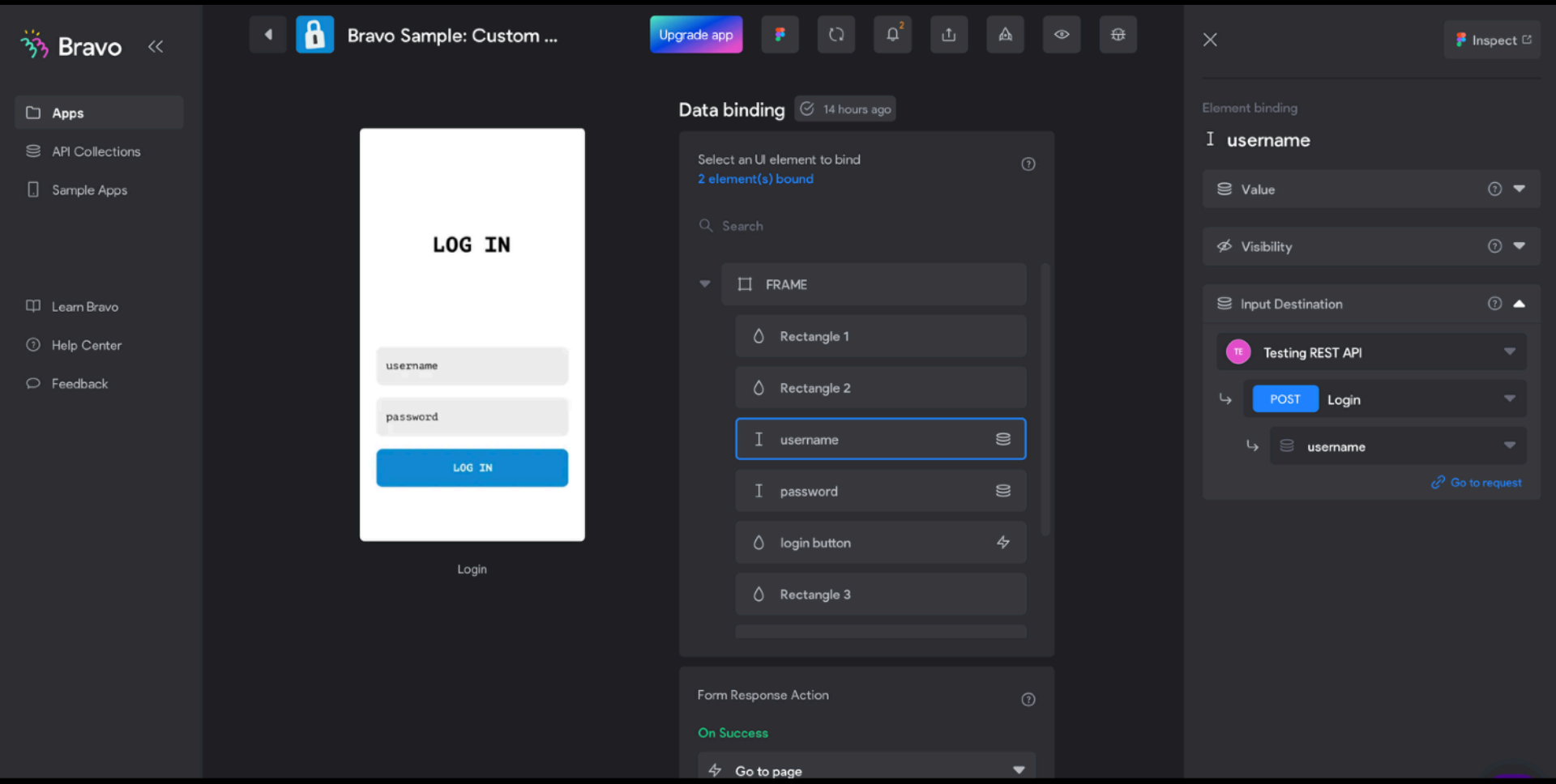

Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
Sample Bravo app kit * Travel App
Upgrade app
Screens
Integrations
Publish
Push Notifications
Login
Notifications
Payments
Deep Links
Chat
Analytics
Multi-language
Firebase Config
Login
Configure authentication to allow users to log in to your app.
Firebase
Email/Password & Social Login
Disabled
OAuth 2.0
Code flow
Disabled
Custom Login
Custom authentication
Disabled

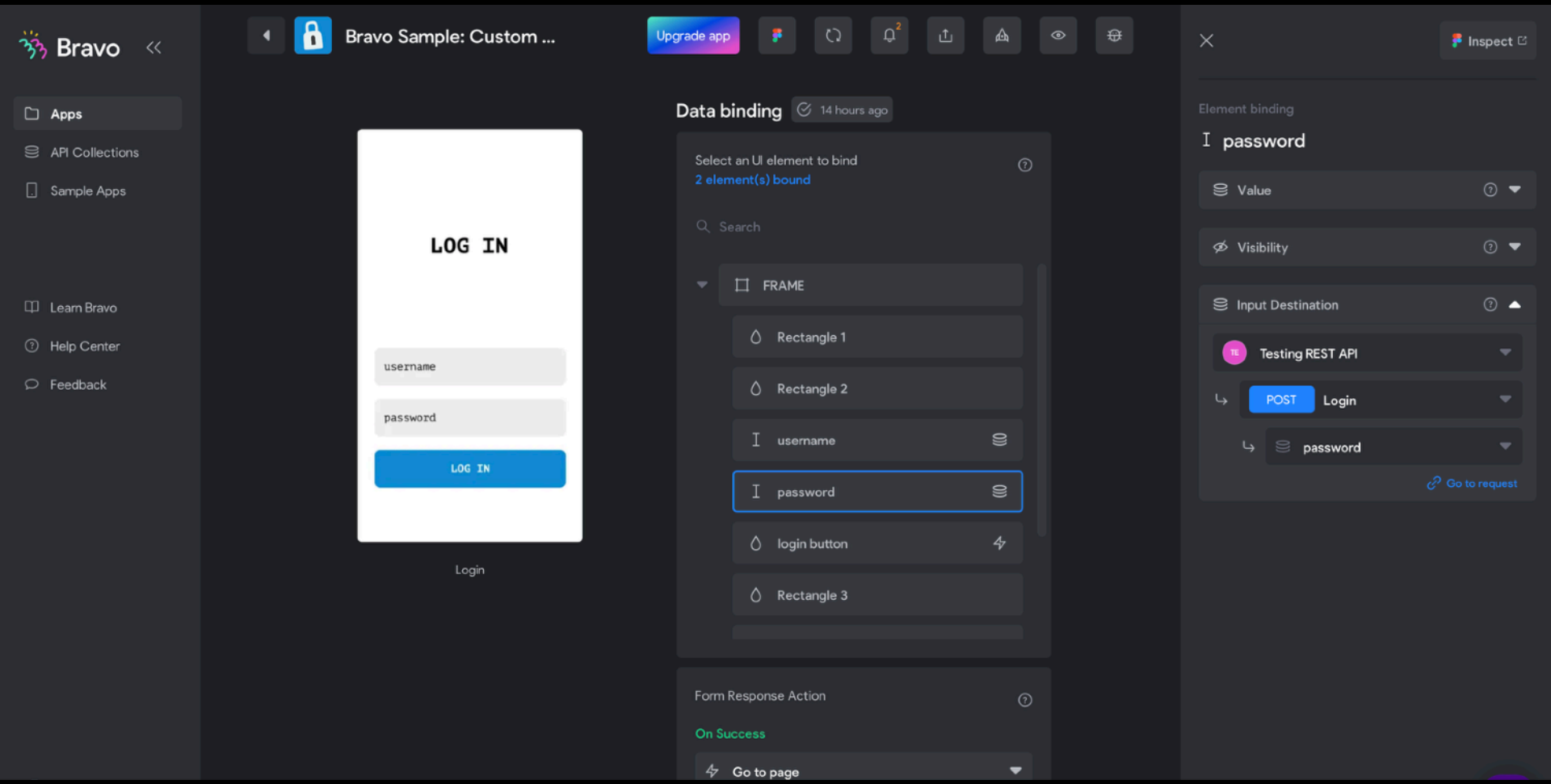

Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
TE
Testing REST API
16 hours ago
Requests
Search 6 requests
GET
User Info Request
POST
Register
GET
Destination with token required
POST
Login
GET
Venue Details
GET
Venue Listing
Register
Request URL
POST
https://suentze2021.pythonanywhere.com/api/register
Send
Headers
Test Values
Body
Pagination
Query Parameters
Headers
HTTP headers to send with your request
Content-Type
application/json
Key
Please enter the key first
Received Data
Selected Data
Debug


Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
TE
Testing REST API
16 hours ago
Requests
Search 6 requests
GET User Info Request
POST Register
GET Destination with token required
POST Login
GET Venue Details
GET Venue Listing
Register
Request URL
POST https://suentze2021.pythonanywhere.com/api/register
Send
Headers Test Values Body Pagination Query Parameters
None Form-data JSON Raw GraphQL
{
"username":"${username}",
"email": "${email}",
"password":"${password}"
}
Received Data Selected Data Debug


Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
TE
Testing REST API
16 hours ago
Requests
Search 6 requests
GET
User Info Request
POST
Register
GET
Destination with token required
POST
Login
GET
Venue Details
GET
Venue Listing
Register
Request URL
POST
https://suentze2021.pythonanywhere.com/api/register
Send
Headers
Test Values
Body
Pagination
Query Parameters
Test Values
Test values are placeholder values to use for any inputs to this request (i.e. anything within ${...})
They will only be used for test requests made in the API bridge and not requests made by any applications using this request.
username
jameskong
email
jkong@abc.com
password
password
Key
Please enter the key first
Received Data
Selected Data
Debug


Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
TE
Testing REST API
Requests
Search 6 requests
GET
User Info Request
POST
Register
GET
POST
Login
GET
Venue Details
GET
Venue Listing
16 hours ago
Send
application/json
Received Data
Selected Data
Debug
Hit send to get a response
Collection settings
General
Authentication
Type
Basic
Username
bernard
Password
Cancel
Save


Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
Bravo Sample: Custom ...
Upgrade app
LOG IN
username
password
LOG IN
Login
Data binding
14 hours ago
Select an UI element to bind
2 element(s) bound
Search
FRAME
Rectangle 1
Rectangle 2
username
password
login button
Rectangle 3
Form Response Action
On Success
Go to page
Inspect
Element binding
username
Value
Visibility
Input Destination
Testing REST API
POST
Login
username
Go to request


Bravo
Apps
API Collections
Sample Apps
Learn Bravo
Help Center
Feedback
Bravo Sample: Custom ...
Upgrade app
LOG IN
username
password
LOG IN
Login
Data binding
14 hours ago
Select an UI element to bind
2 element(s) bound
Search
FRAME
Rectangle 1
Rectangle 2
username
password
login button
Rectangle 3
Form Response Action
On Success
Go to page
Inspect
Element binding
password
Value
Visibility
Input Destination
Testing REST API
POST
Login
password
Go to request

Airtable Automation

Click the following link to share the database.

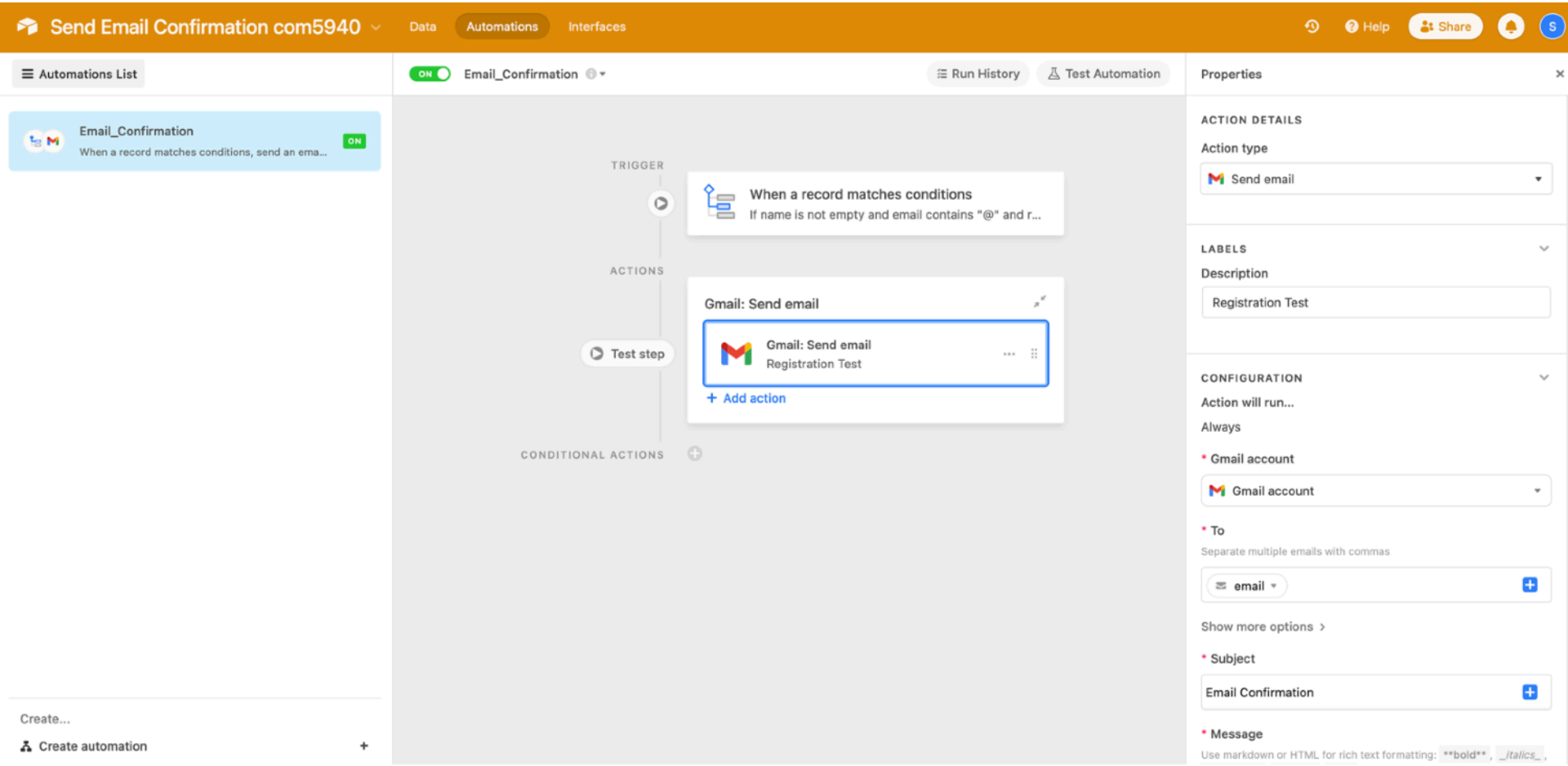
https://airtable.com/shr3ELfNWDo0GUvtU

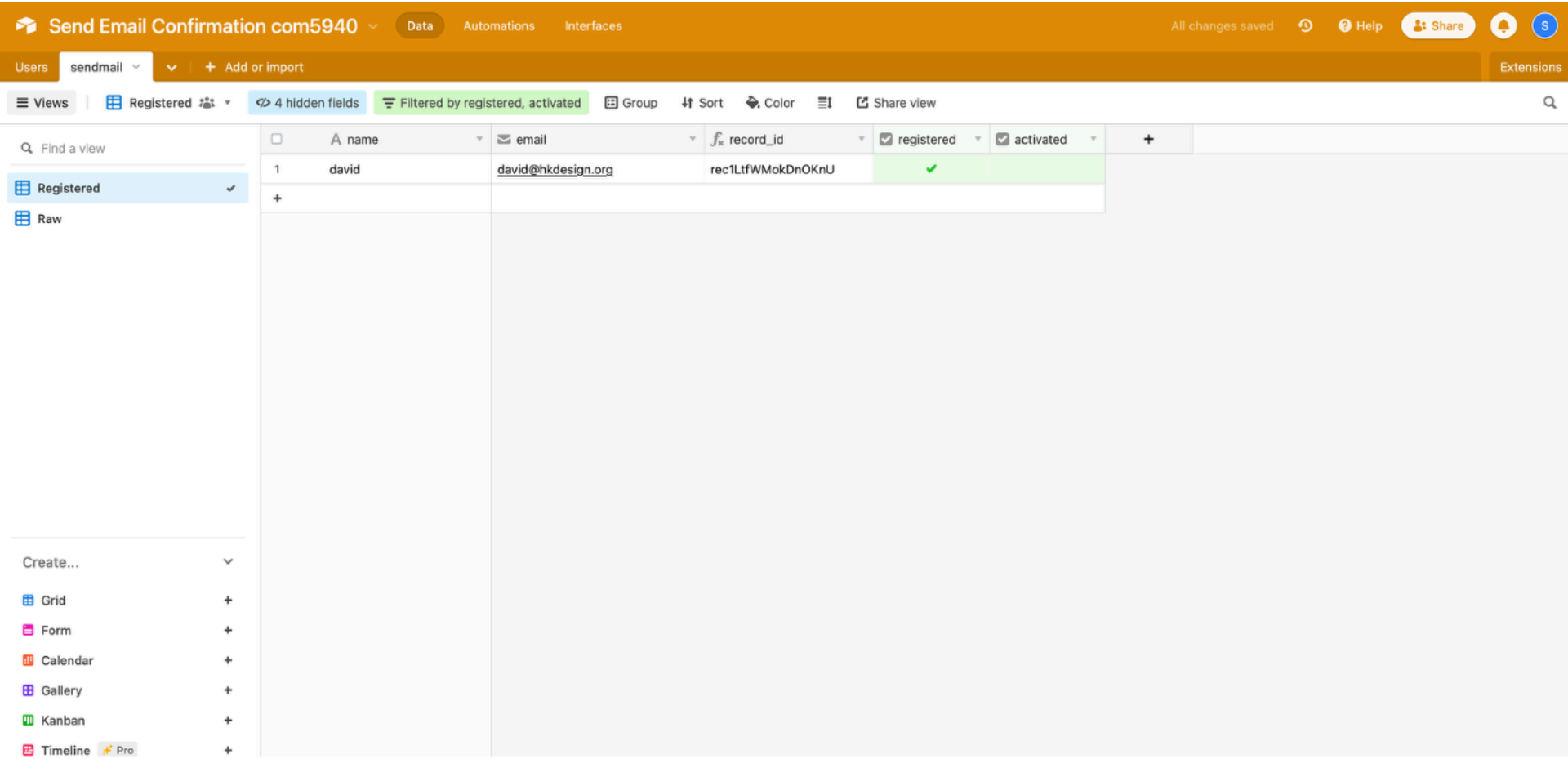
```python
@app.route('/api/register', methods=['POST'])
def register():
    data = request.get_json()
    hashed_password = bcrypt.generate_password_hash(data['password'])
    new_user = User(username=data['username'], email=data['email'], password=hashed_password, admin=False)
    token = s.dumps(data['email'], salt='email-confirm')
    link = url_for('confirmemail', token=token, _external=True)
    message = "Click the link to confirm your registration:"+link
    username = data['username']
    email = data['email']
    registered = True

    mydict =  {
        "name": username,
        "email": email,
        "registered": registered,
        "message": message
        }

    data = {"fields": mydict}
    headers = {'Authorization': 'Bearer xxxxxxxxxxx', 'Content-Type': 'application/json; charset=utf-8'}
    # r = requests.post(url = API_ENDPOINT, data = data)
    r = requests.post('https://api.airtable.com/v0/appajvHoXUME1ZmI2/sendmail',json=data,headers=headers)
    print("Status Code:",r.status_code)
    print(r.text)

    db.session.add(new_user)
    db.session.commit()
    return jsonify({'message' : 'Registration completed!'})
```

Email notification

```python
@app.route('/confirmemail/<token>')
def confirmemail(token):
    try:
        email  = s.loads(token, salt='email-confirm', max_age=3600)
        user = User.query.filter_by(email=email).first()
        if user:
            user.activated = True
            db.session.commit()
        # return 'The token works!'
        message = 'Thank you for authenticating your email!'
    except SignatureExpired:
        # return '<h1>The confirmation token is expired!</h1>'
        message = 'The confirmation token is expired!'

    return render_template('confirmation.html', message=message)
```

Email confirmation


Send Email Confirmation com5940
Data
Automations
Interfaces
Help
Share
Automations List
Email_Confirmation
When a record matches conditions, send an ema...
ON
Create...
Create automation
Email_Confirmation
Run History
Test Automation
TRIGGER
When a record matches conditions
If name is not empty and email contains "@" and r...
ACTIONS
Gmail: Send email
Test step
Gmail: Send email
Registration Test
Add action
CONDITIONAL ACTIONS
Properties
ACTION DETAILS
Action type
Send email
LABELS
Description
Registration Test
CONFIGURATION
Action will run...
Always
Gmail account
Gmail account
To
Separate multiple emails with commas
email
Show more options
Subject
Email Confirmation
Message
Use markdown or HTML for rich text formatting: **bold**, _italics_,


Send Email Confirmation com5940
Data
Automations
Interfaces
All changes saved
Help
Share
Users
sendmail
Add or import
Extensions
Views
Registered
4 hidden fields
Filtered by registered, activated
Group
Sort
Color
Share view
Find a view
Registered
Raw
name
email
record_id
registered
activated
1
david
david@hkdesign.org
rec1LtfWMokDnOKnU
Create...
Grid
Form
Calendar
Gallery
Kanban
Timeline
Pro

Task Automation

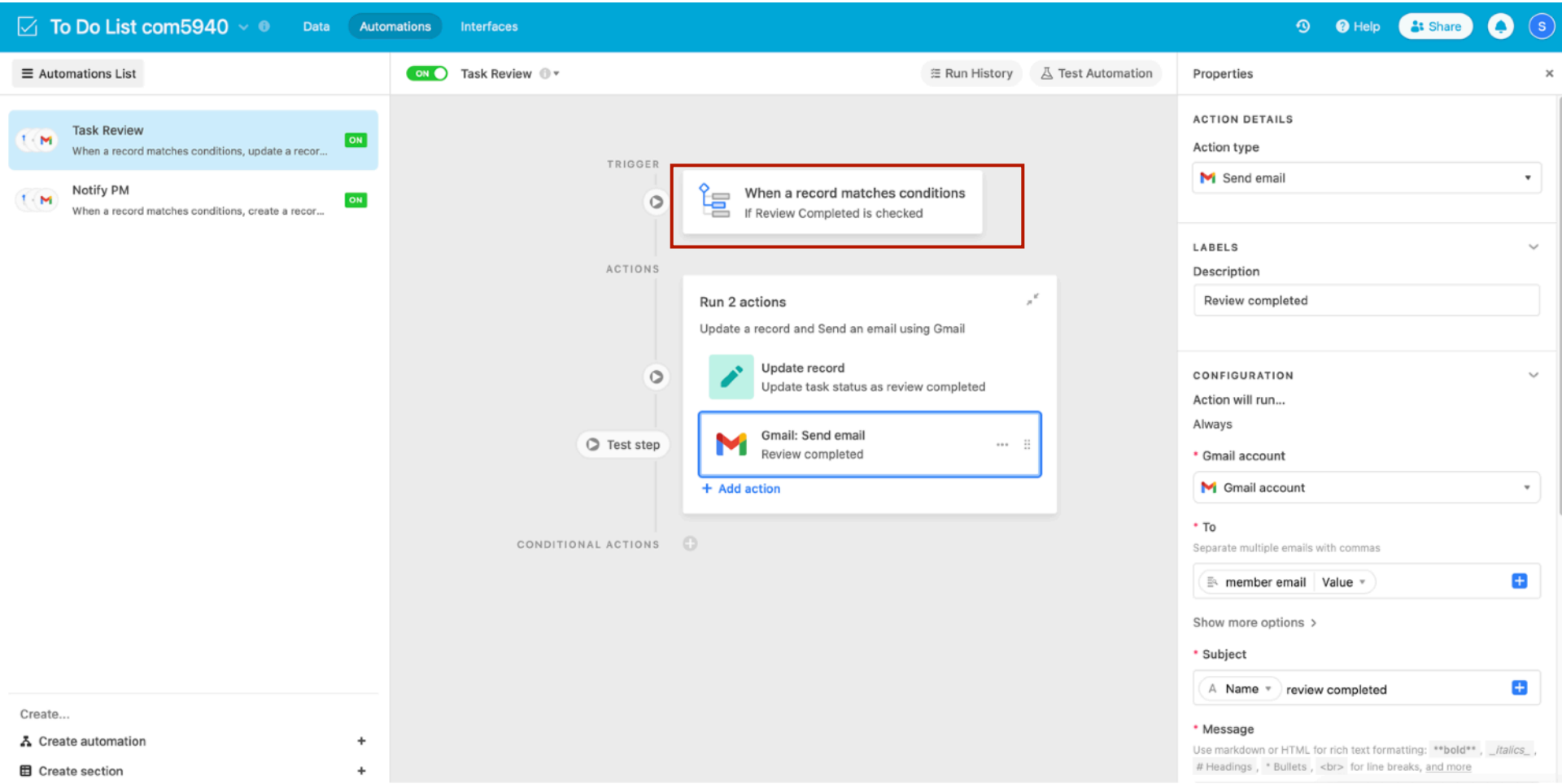
Click the following link to share the database.

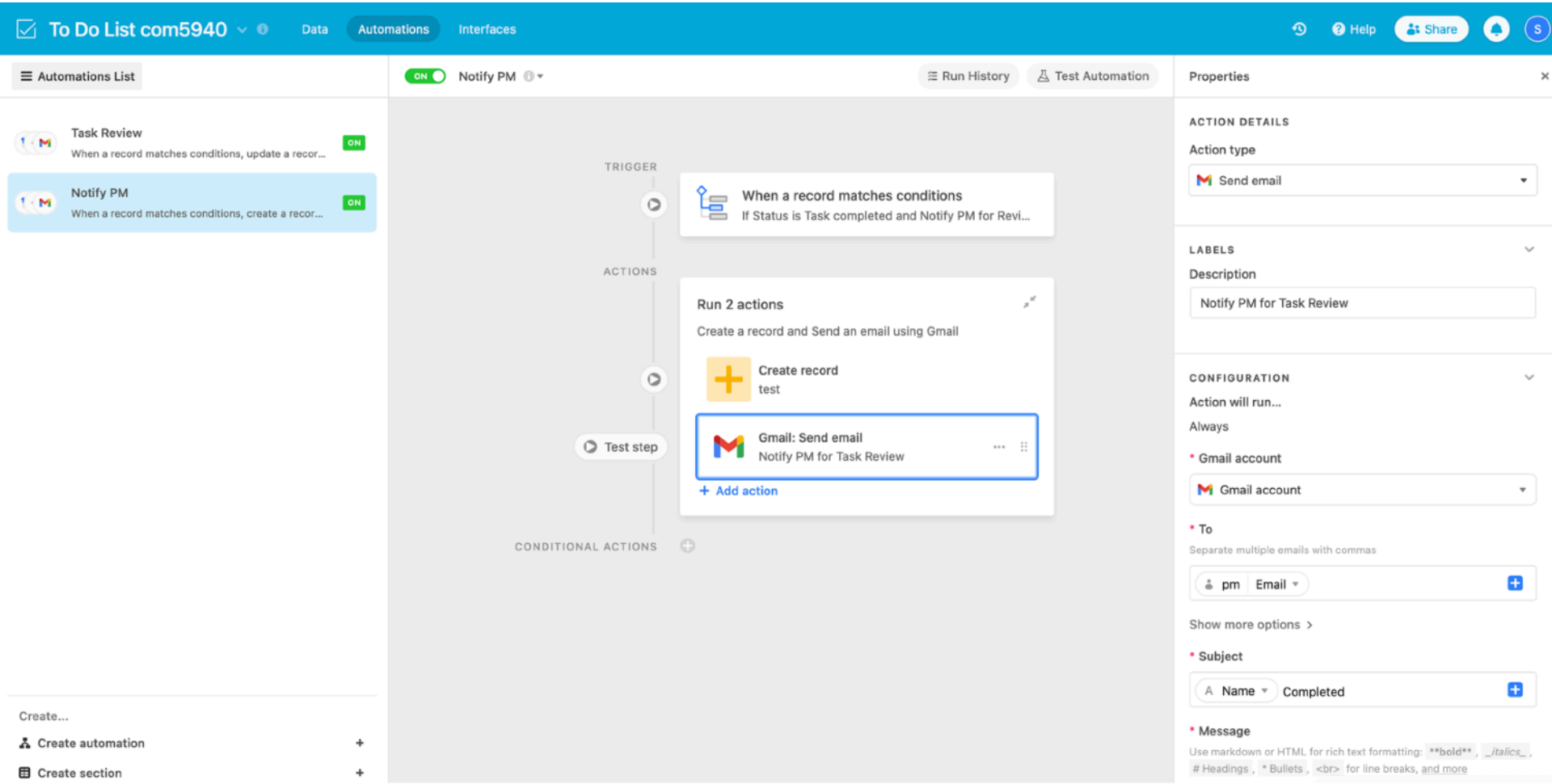
https://airtable.com/shrAGjaWzOlj2qrhL

In [6]:

```
#########################
#
#  Patch Request
#
#########################

import requests
import pandas as pd

data = {
  "records": [
    {
      "id": "rec0G040YicU6Iv4M",
      "fields": {
        "Notify PM for Review": True,
        "Status": 'Task in progress',
        "Date": '1/30/2023'
      }
    }
  ] # end records
} # end data

headers = {'Authorization': 'Bearer XXXXXXXXXXXXXXXXX' , 'Content-Type': 'application/json; charset=utf-8'}
r = requests.patch('https://api.airtable.com/v0/appnTdfkawngXQEyT/Tasks',json=data,headers=headers)
print("Status Code:",r.status_code)
print(r.text)
params = (
('view', 'Raw Grid'),
)
# Using the Southwind Base stored in the COM5940 workspace
r = requests.get('https://api.airtable.com/v0/appnTdfkawngXQEyT/Tasks', headers=headers, params=params)
print("Status Code:",r.status_code)
dict = r.json()
dataset = []
productID = []
productCode = []
result = []
for data in dict['records']:
     field_content = data['fields']
     dataset.append(field_content)
print(dataset)
print('\n')
print(pd.DataFrame(dataset))
```

Update Airtable “Single Select” Status field.


To Do List com5940
Data
Automations
Interfaces
Help
Share
S
Automations List
ON
Task Review
Run History
Test Automation
Properties
Task Review
When a record matches conditions, update a recor...
ON
Notify PM
When a record matches conditions, create a recor...
ON
TRIGGER
When a record matches conditions
If Review Completed is checked
ACTIONS
Run 2 actions
Update a record and Send an email using Gmail
Update record
Update task status as review completed
Test step
Gmail: Send email
Review completed
Add action
CONDITIONAL ACTIONS
ACTION DETAILS
Action type
Send email
LABELS
Description
Review completed
CONFIGURATION
Action will run...
Always
* Gmail account
Gmail account
* To
Separate multiple emails with commas
member email
Value
Show more options
* Subject
Name
review completed
* Message
Use markdown or HTML for rich text formatting: **bold** , _italics_ , # Headings , * Bullets ,
 for line breaks, and more
Create...
Create automation
Create section


To Do List com5940
Data
Automations
Interfaces
Help
Share
S
Automations List
Task Review
When a record matches conditions, update a recor...
ON
Notify PM
When a record matches conditions, create a recor...
ON
Create...
Create automation
Create section
ON
Notify PM
Run History
Test Automation
TRIGGER
When a record matches conditions
If Status is Task completed and Notify PM for Revi...
ACTIONS
Run 2 actions
Create a record and Send an email using Gmail
Create record
test
Test step
Gmail: Send email
Notify PM for Task Review
Add action
CONDITIONAL ACTIONS
Properties
ACTION DETAILS
Action type
Send email
LABELS
Description
Notify PM for Task Review
CONFIGURATION
Action will run...
Always
Gmail account
Gmail account
To
Separate multiple emails with commas
pm
Email
Show more options
Subject
Name
Completed
Message
Use markdown or HTML for rich text formatting: **bold** , _italics_ , # Headings , * Bullets ,
 for line breaks, and more

Demo

Problem Set #1

1. Create a Postman folder consisted of CRUD and authentication tests to your Flask database from last term's final project.
2. Export the Postman folder and include with your Jupyter Notebook.

Problem Set #2

1. Create an Airtable base (i.e. derived from your SQLite database) and table (e.g. blog tables)
2. Create a Jupyter Notebook which includes the use of the Python requests module to access Airtable.
3. Perform CRUD requests demonstrating how to exchange data between a Python application and Airtable.

Thank you for your time!