



Source:Jose Aragones

Integrated Data-Driven Story Telling: Lesson 4 - Connect JavaScript Front-end to Flask Back-end

Objectives Recap

- **Integrate front-end with back-end**
- **Integrate design with computation**
- **Integrate multiple data sources**
- **Integrate story with data**
- **Integrate with other disciplines**
- **Integrate with latest software development practices**

Today's Agenda

- 1. What is RESTful API?**
- 2. Leverage Airtable RESTful API for Database Model**
- 3. Understand HTTP Status Codes**
- 4. Running RESTful API Test in Postman**
- 5. Running RESTful API Test in JupyterNotebook**
- 6. Web Authentication and Authorization**
- 7. Flask Login Module**
- 8. Running RESTful API in Flask**
- 9. Pass Python Data to JQuery/Bootstrap Objects**

Quick Recap of Last Week's Topics

POPULAR FULL STACK FRAMEWORK FOR SOFTWARE DEVELOPMENT

API (e.g. RestFUL)



BACK-END TECHNOLOGY

后台

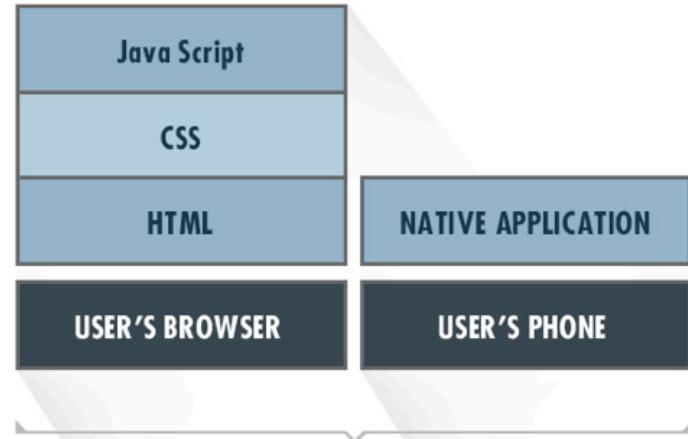
WHAT IS YOUR “CLOUD” AND “STACK” STRATEGY?

云架构及前端和后端的全栈策略

{ JSON }



THE INTERNET



FRONT-END TECHNOLOGY

前台

Python for Web Development Using Flask

Running Flask in Local Mode

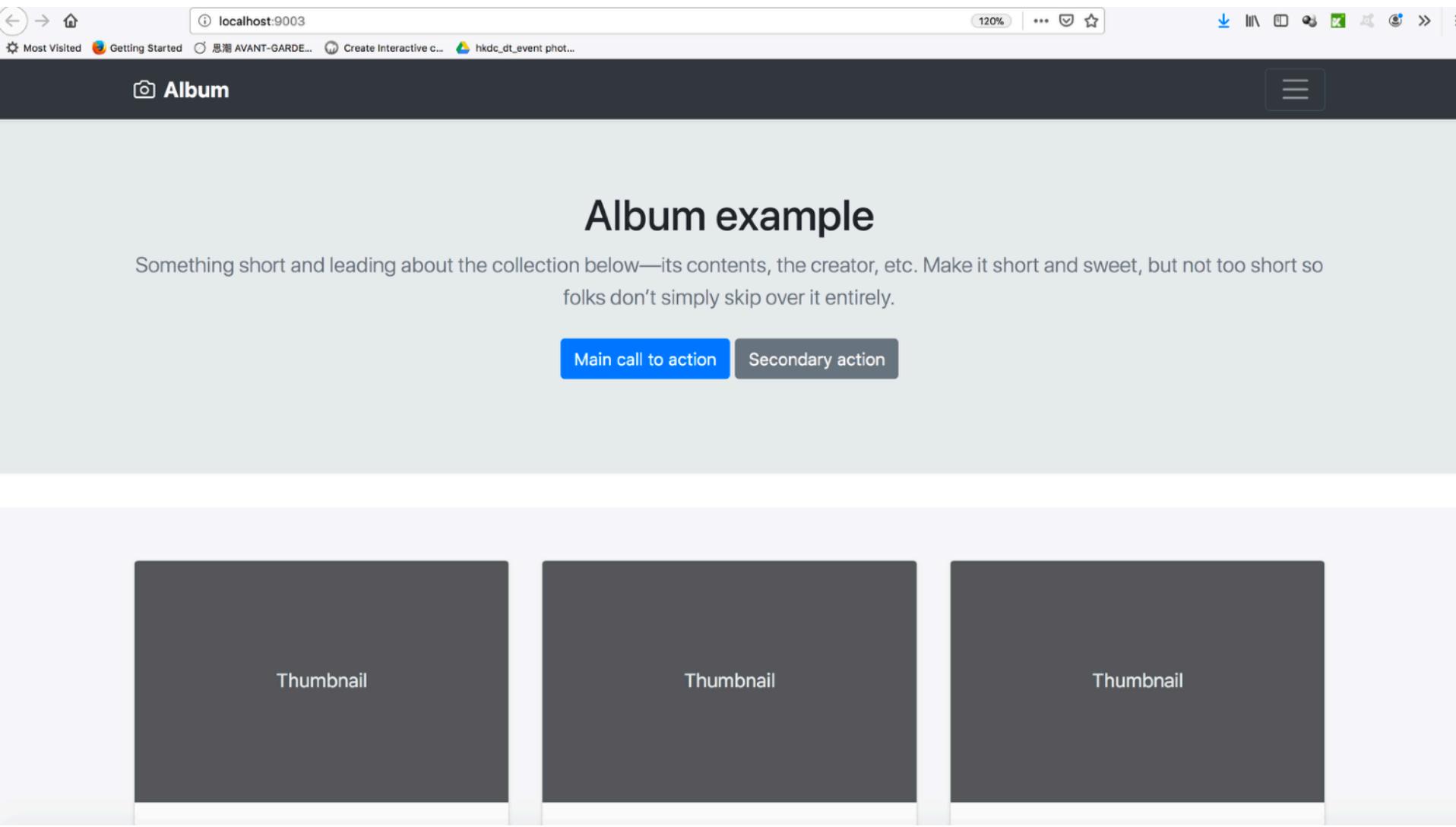
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [*]:

```
1 import os
2 from werkzeug.wsgi import SharedDataMiddleware
3
4 from flask import Flask, render_template
5
6 app = Flask(__name__)
7
8 # The following line is only needed for running inside Jupyter Notebook
9 root_path = os.path.sep.join(app.instance_path.split(os.path.sep)[:-1])
10
11 @app.route("/")
12 def home():
13     return render_template('myhome.html')
14
15 if __name__ == '__main__':
16     from werkzeug.serving import run_simple
17     app.wsgi_app = SharedDataMiddleware(app.wsgi_app, {
18         '/static': root_path+'/flask_apps/static',
19         '/templates': root_path+'/flask_apps/templates'
20     })
21     run_simple('localhost', 9003, app)
```

* Running on <http://localhost:9003/> (Press CTRL+C to quit)

In []:



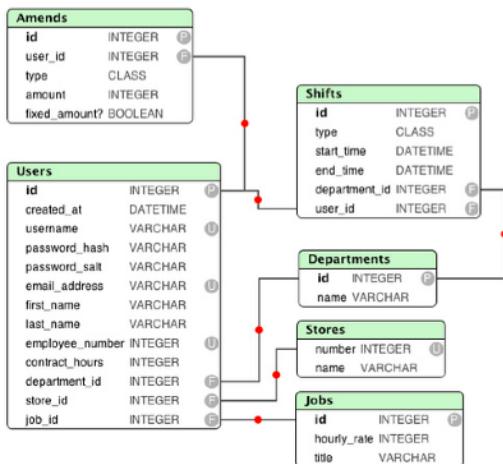
- ▶ **Web template development in Python: Introducing Flask Jinja2 (i.e. render_template library)**
- ▶ **Retrieving information from an Airtable base into a Python Flask program through the Airtable API**

MVC Framework Using the Python Flask

Source: commons.wikimedia.org



Source: [Caius Durling \(Flickr\)](https://flic.kr)



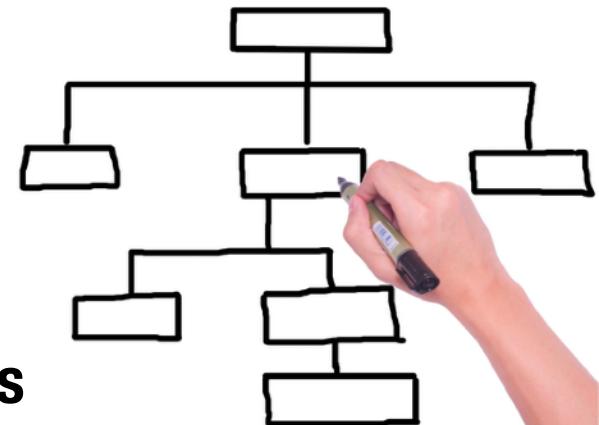
M(odel)
(e.g. Airtable, Pandas, ScikitLearn)

V(iew) (Flask templates)

Source: pexels.com



Routing control to
pre-defined functions



C(ontroller)
(e.g. Flask Routes)

URL Routing in Flask

```
from flask import Flask, render_template  
@app.route('/')  
def main():  
    :  
    return render_template('index.html')  
  
@app.route('/about_us')  
def about_us():  
    :  
    return render_template('aboutus.html')
```

Python Code (control flow and
data model)



Html Templates (view)

```
<!DOCTYPE html>  
<head>  
<meta charset="utf8">  
<title>Main Page</title> </head>  
<body>  
<h1>This is the main page</h1>  
    :  
    :  
</body> </html>
```

index.html

```
<!DOCTYPE html>  
<head>  
<meta charset="utf8">  
<title>About Us</title> </head>  
<body>  
<h1>This is the about us page</h1>  
    :  
    :  
</body> </html>
```

aboutus.html

Running Flask Codes on PythonAnywhere



Host, run, and code Python in the cloud!

Get started for free. Our basic plan gives you access to machines with a full Python environment already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at \$5/month.

[Start running Python online in less than a minute! »](#)

[Watch our one-minute video »](#)

Not convinced? Read what our users are saying!

The screenshot shows the PythonAnywhere dashboard with a green header bar indicating 'All done! Your web app is now set up. Details below.' Below this, there's a list of web apps: 'example.pythonanywhere.com' (disabled), 'www.mydomain.com' (disabled), and 'www.myotherdomain.com' (selected). A button 'Add a new web app' is visible. To the right, there's a section for 'Configuration for www.myotherdomain.com' with a 'Reload' button and a note about DNS setup: 'How to point your domain at your website.' It shows a CNAME entry: 'CNAME: webapp-4.pythonanywhere.com'. Below this is a 'Traffic' section with two line graphs: 'Hits per month' and 'Hits per day', both showing low activity levels.

Start hosting quickly

Just write your application. No need to configure or maintain a web server — everything is set up and ready to go.

[More »](#)

Develop anywhere

Take your development environment with you! If you have a browser and an Internet connection, you've got everything you need.

[More »](#)

Teach and learn

PythonAnywhere is a fully-fledged Python environment, ready to go, for students and teachers — concentrate on teaching, not on installation hassles.

Amazing support

Need help with PythonAnywhere? If you get in touch, you can talk directly with the development team. Help for developers, from developers.

[More »](#)

[Dashboard](#) [Consoles](#) [Files](#) **Web** [Tasks](#) [Databases](#)

All done! Your web app is now set up. Details below.

X

bssuen.pythonanywhere.com[+ Add a new web app](#)Configuration for [bssuen.pythonanywhere.com](#)

Reload:

[↻ Reload bssuen.pythonanywhere.com](#)

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details](#).

This site will be disabled on **Tuesday 17 December 2019**[Run until 3 months from today](#)[Paying users' sites stay up forever without any need to log in to keep them running.](#)

Traffic:

How busy is your site?

This month (previous month)	1 (0)
Today (yesterday)	1 (0)
Hour (previous hour)	1 (0)

Want some more data? [Paying accounts](#) get pretty charts ;)



/home/bssuen/ mysite

Directories

[New directory](#)

__pycache__
static/
templates/



Files

[New file](#)

flask_app.py 2019-09-17 09:27 186 bytes
[Upload a file](#)
100MiB maximum size

**Template Tags
(i.e. comments, expressions,
statements)**

- **{#... comment#}** : template tag for comments
- **{{... expression...}}** : template tag for expressions
- **{%...statement...%}** : template tag for statements

Passing Values to Template Variables

```
from flask import Flask, render_template  
app = Flask(__name__)  
  
@app.route("/")  
def index():  
    user = {"name": "Bernard"}  
    return render_template("result.html", user=user, title="Home Page")
```

code variable and value

template file

template variables

Passing values to template variables.

template variables

```
44-
45   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false"
46     <span class="navbar-toggler-icon"></span>
47   </button>
48 </div>
49 </div>
50 </header>

51-
52 <main role="main">
53   <section class="jumbotron text-center">
54     <div class="container">
55       <!--h1 class="jumbotron-heading">Album example</h1-->
56       <h1 class="jumbotron-heading">{{user.name}}'s {{title}}</h1>
57       <p class="lead text-muted">Something short and leading about the collection below its contents, the creator, etc. Make it short and sweet, but n
58       <p>
59         <a href="https://getbootstrap.com/docs/4.0/examples/album/#" class="btn btn-primary my-2">Main call to action</a>
60         <a href="https://getbootstrap.com/docs/4.0/examples/album/#" class="btn btn-secondary my-2">Secondary action</a>
61       </p>
62     </div>
63   </section>
64
65 <div class="album py-5 bg-light">
66   <div class="container">
67
68     <div class="row">
69       <div class="col-md-4">
70         <div class="card mb-4 box-shadow">
71           
72           <div class="card-body">
73             <p class="card-text">This is a wider card with supporting text below as a natural lead-in to additional content. This content is a littl
74             <div class="d-flex justify-content-between align-items-center">
75               <div class="btn-group">
76                 <button type="button" class="btn btn-sm btn-outline-secondary">View</button>
77                 <button type="button" class="btn btn-sm btn-outline-secondary">Edit</button>
78               </div>
79               <small class="text-muted">9 mins</small>
80             </div>
81           </div>
82         </div>
83       </div>
84     </div>
```

Bernard's Home Page

Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.

Main call to action Secondary action

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

[View](#) [Edit](#) 9 mins

Thumbnail

Thumbnail

Thumbnail

REST API Lab

GROWTH STRATEGY

The Strategic Value of APIs

by Bala Iyer and Mohan Subramaniam

JANUARY 07, 2015

McKinsey&Company
Digital McKinsey

How We Help Clients Our Insights Careers Our People Contact Us

Article
October 2017

Management's next frontier: Making the most of the ecosystem economy

By Jürgen Metzert and Anand Swaminathan

API經濟來了：從開放創新到Open API

by 《數位時代》整合行銷部 2014.11.04



McKinsey&Company
Digital McKinsey

How We Help Clients Our Insights Careers Our People Contact Us

What it really takes to capture the value of APIs

≡ Deloitte.
Insights

Article

API economy

From systems to business services

IBM Cloud 雲端平台 > 瞭解雲端

API 經濟

在您的營運模式中加入 API，
藉此催化創新並觸角伸向新對象

[觀賞影片 \(01:25\)](#)[閱讀 API 入門電子書](#)

關於 IBM

參加評估

API (e.g. REST)



BACK-END TECHNOLOGY

后台

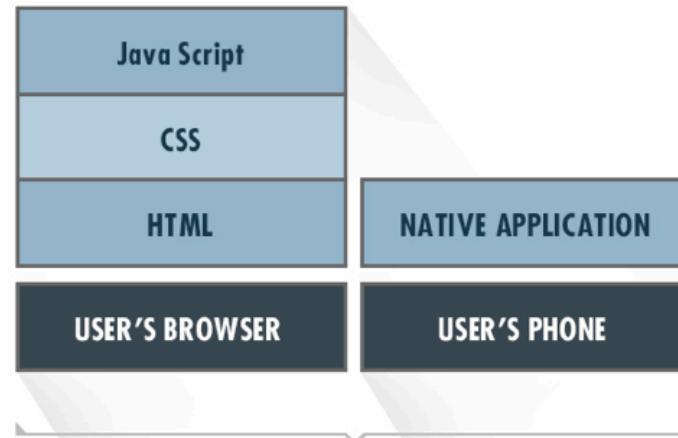
WHAT IS YOUR “CLOUD” AND “STACK” STRATEGY?

云架构及前端和后端的全栈策略

{ JSON }



THE INTERNET



FRONT-END TECHNOLOGY

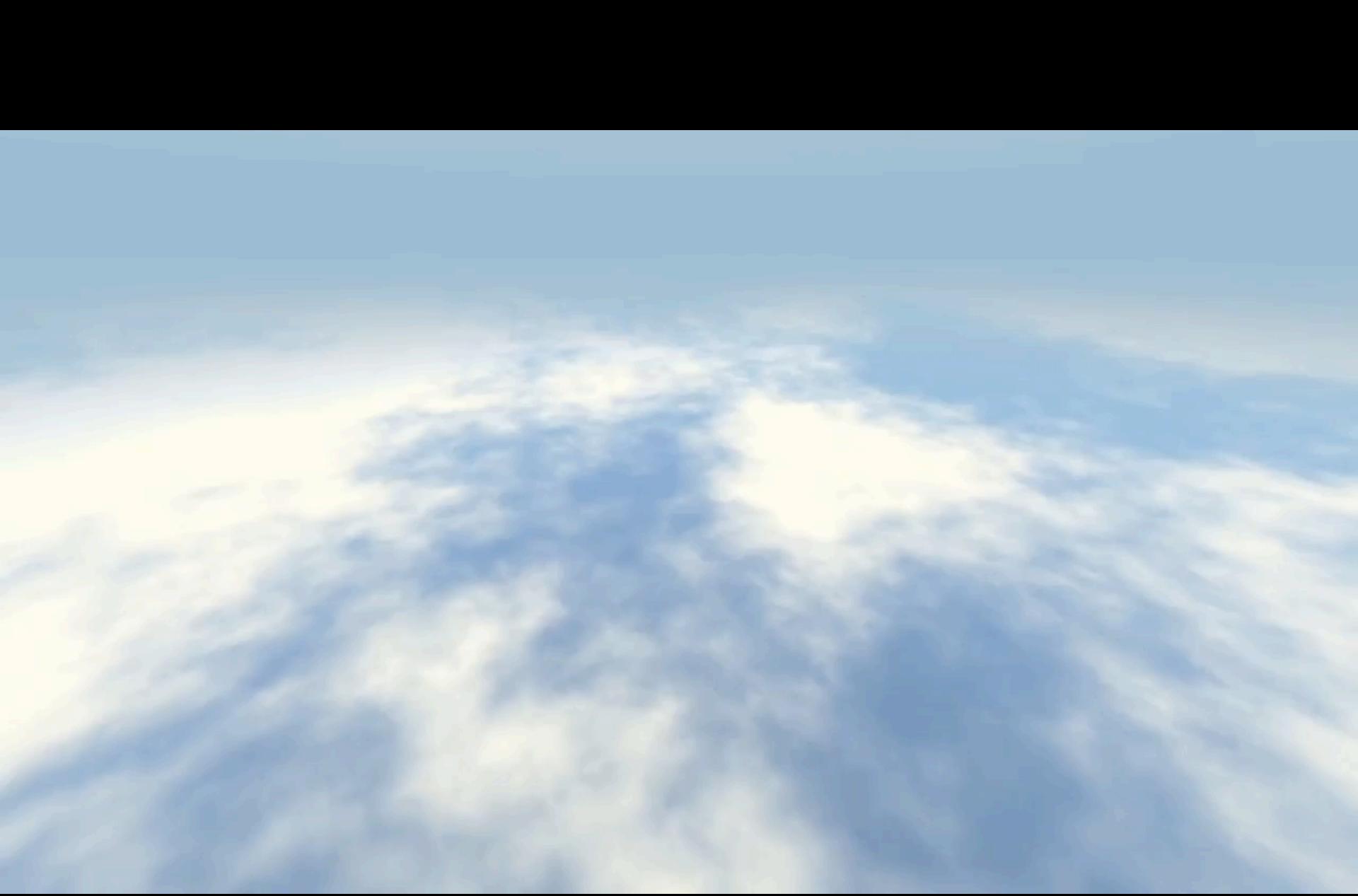
前台



{World of Internet}

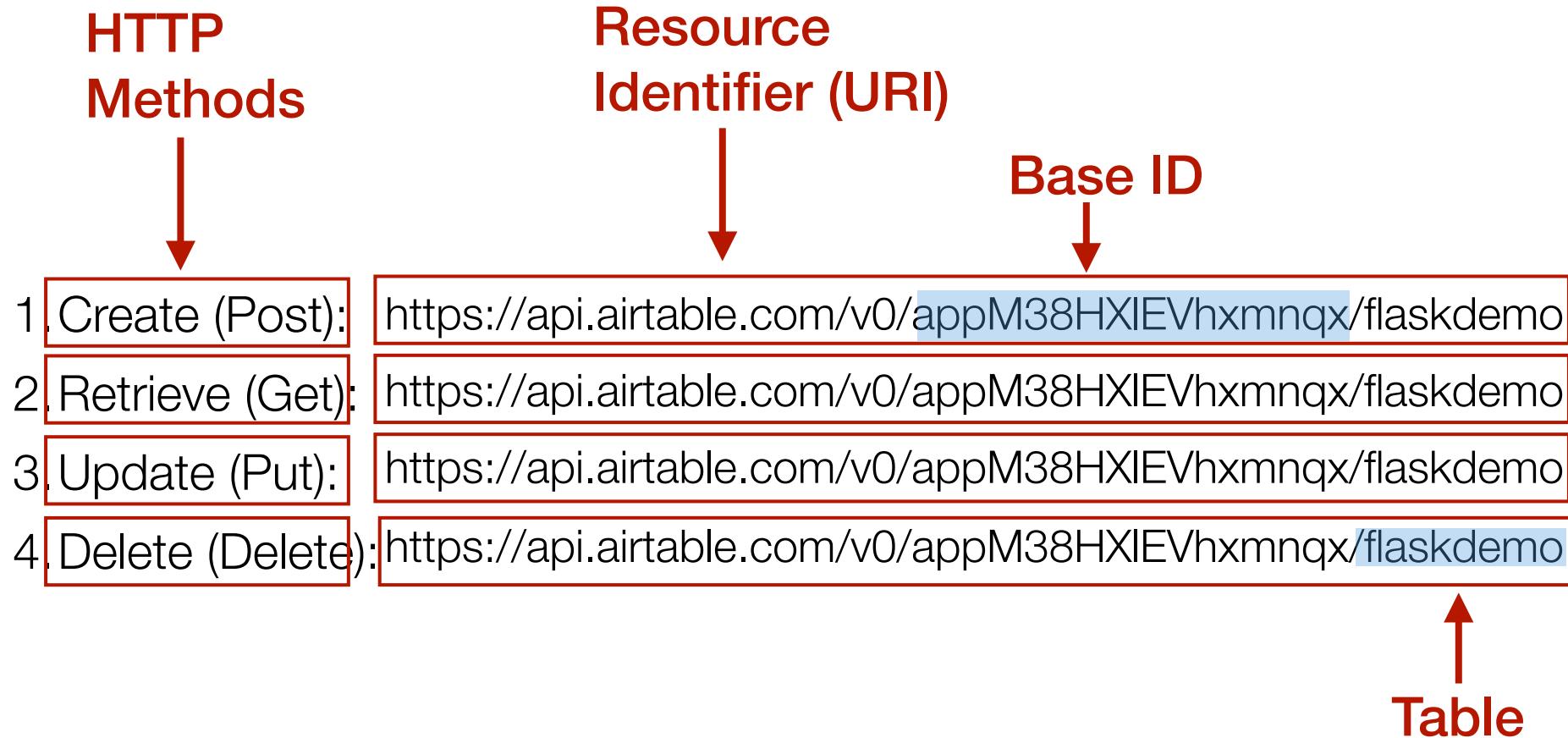


https://www.youtube.com/watch?v=lvlnf9HW_Jw



<https://www.youtube.com/watch?v=u-RnFs9dby4>

Basic RESTful Methods and Structure for Airtable





Airtable RESTful API

INTRODUCTION

The [Project Tracker](#) API provides an easy way to integrate your [Project Tracker](#) data in Airtable with any external system. The API closely follows REST semantics, uses JSON to encode objects, and relies on standard HTTP codes to signal operation outcomes.

The API documentation below is specifically generated for your base. We recommend that you use the [graphical Airtable interface](#) to add a few records of example data for each table. These records will be displayed in the documentation examples generated below.

The ID of this base is [appM38HX1EVhxmnqx](#).

Please note: if you make changes to a field (column) name or type, the API interface for those fields will change correspondingly. Therefore, please make sure to update your API implementation accordingly whenever you make changes to your Airtable schema from the graphical interface.

Official API client:

- JavaScript: [airtable.js](#) (Node.js + browser)

Community-built API clients:

- Ruby: [airrecord](#)
- .NET: [airtable.net](#)

RATE LIMITS

The API is limited to 5 requests per second. If you exceed this rate, you will receive a 429 status code and will need to wait 30 seconds before subsequent requests will succeed.

The [official JavaScript client](#) has built-in retry logic.

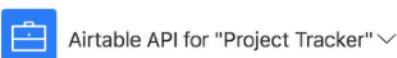
If you anticipate a higher read volume, we recommend using a caching proxy. This rate limit is the same for all plans and increased limits are not currently available.

curl

JavaScript

 show API key

Airtable RESTful API



Open base...

List flaskdemo records

To list records in `flaskdemo`, issue a **GET** request to the `flaskdemo` endpoint.

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

You can use the following parameters to filter, sort, and format the results:

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 `fame` and `lname`, send these two query parameters:

`fields=`
`fields=fame`
`fields=lname`

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.

If combined with `view`, only records in that view which satisfy the formula will be returned.

For example, to only include records where `fame` isn't empty, pass in:
`NOT({fame} = '')`

maxRecords
`number`
optional
The maximum total number of records that will be returned in your requests. If this value is larger than `pageSize` (which is 100 by default), you may have to load multiple pages to reach this total. See the Pagination section below for more.

pageSize
`number`
optional
The number of records returned in each request. Must be less than or equal to 100. Default is 100. See the Pagination section below for more.

[curl](#) [JavaScript](#)

show API key

EXAMPLE REQUEST

```
curl "https://api.airtable.com/v0/appM38HXLEVhxmnqx/flaskdemo?maxRecords=3&view=Grid%20view" \
-H "Authorization: Bearer keyTcsTzckqyBT1k8"
```

EXAMPLE RESPONSE

```
{
  "records": [
    {
      "id": "recp0HXSrfrMIVQia",
      "fields": {
        "fame": "John",
        "lname": "Chan",
        "student_id": "12349",
        "date_of_birth": "2009-02-21"
      },
      "createdTime": "2019-09-19T14:04:24.000Z"
    },
    {
      "id": "recASxs4aPaHLejj3",
      "fields": {
        "fame": "Emily",
        "lname": "Wong",
        "student_id": "67891",
        "date_of_birth": "2008-04-03"
      },
      "createdTime": "2019-09-19T14:04:24.000Z"
    },
    {
      "id": "rec8vF2VbIg0Tdx3F",
      "fields": {
        "fame": "Mary",
        "lname": "Lu",
        "student_id": "99999",
        "date_of_birth": "2001-08-22"
      },
      "createdTime": "2019-09-19T14:32:08.000Z"
    }
  ]
}
```



The Wikimedia Foundation wants to make talk pages accessible to all, to improve the depth, quality and the health on the wikis.
We invite all Wikimedians to consult on how to achieve this goal. Join us!



[Help with translations!]

List of HTTP status codes

From Wikipedia, the free encyclopedia

This is a list of [Hypertext Transfer Protocol](#) (HTTP) response status codes. Status codes are issued by a server in response to a client's request made to the server. It includes codes from IETF [Request for Comments](#) (RFCs), other specifications, and some additional codes used in some common applications of the [Hypertext Transfer Protocol](#) (HTTP). The first digit of the status code specifies one of five standard classes of responses. The message phrases shown are typical, but any human-readable alternative may be provided. Unless otherwise stated, the status code is part of the HTTP/1.1 standard ([RFC 7231](#)).^[1]

The [Internet Assigned Numbers Authority](#) (IANA) maintains the official registry of HTTP status codes.^[2]

Microsoft [Internet Information Services](#) (IIS) sometimes uses additional decimal sub-codes for more specific information,^[3] however these sub-codes only appear in the response payload and in documentation, not in the place of an actual HTTP status code.

All HTTP response status codes are separated into five classes (or categories). The first digit of the status code defines the class of response. The last two digits do not have any class or categorization role. There are five values for the first digit:

HTTP

[Persistence](#) · [Compression](#) · [HTTPS](#) · [QUIC](#)

Request methods

[OPTIONS](#) · [GET](#) · [HEAD](#) · [POST](#) · [PUT](#) ·
[DELETE](#) · [TRACE](#) · [CONNECT](#) · [PATCH](#)

Header fields

[Cookie](#) · [ETag](#) · [Location](#) · [HTTP referer](#) · [DNT](#) ·
· [X-Forwarded-For](#)

Status codes

[301 Moved Permanently](#) · [302 Found](#) ·
[303 See Other](#) · [403 Forbidden](#) ·
[404 Not Found](#) ·
[451 Unavailable For Legal Reasons](#)

Security access control methods

[Basic access authentication](#) ·
[Digest access authentication](#)

V · T · E

- **200 - OK** : Means that the request has been completed successfully and the server has returned the response. Usually returned after a successful **GET** request.
- **201 - Created** : Usually returned after a successful **POST** request. Summarizes that the resource has been created.
- **400 - Bad Request** : It's returned from the server when a request was sent with some missing or invalid parameters. Usually returned in response to **POST** or **PUT** requests.
- **401 - Unauthorized** : Means that user was not authorized to perform certain action. For example, a user tried to create or delete a resource without providing authentication credentials.
- **403 - Forbidden** : Means that server understood the request but refused to complete it due to authentication. It happens when a user provides authentication credentials but it doesn't have sufficient rights to perform the action.
- **404 - Not Found** : The most (in)famous of all the status codes. Summarizes that a resource, the user was looking for, was not found.
- **405 - Method not Allowed** : Means that an HTTP verb supplied in the request was not supported by the resource. An example might be a user trying to update a read-only resource.
- **410 - Gone** : Means that a resource has been moved to another location. An example might be trying to delete an already deleted resource that has been moved to trash.
- **500 - Internal Server Error** : It's returned when a server encounters an unexpected condition and doesn't complete the request.

REST API Test Using Postman

<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddgomop?hl=en>

Home > Apps > Postman



Postman

Offered by: www.getpostman.com

★★★★★ 9,096 | [Extensions](#) | 3,876,124 users

Runs offline

[Launch app](#)

Overview

Reviews

Support

Related



Cloud SWF Player with ...

★★★★★ 2,741



Text

★★★★★ 1,229



Chrome Apps & Extensi...

★★★★★ 1,387



Advanced REST client

★★★★★ 12,153



Vysor

★★★★★ 5,661



Secure Shell App

★★★★★ 3,142



Caret

★★★★★ 1,250



BrowserStack Local

★★★★★ 272



Web Server for Chrome

★★★★★ 1,353



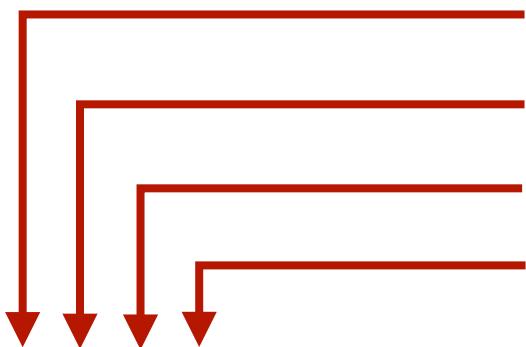
Robot Mesh Connect App

69,129 users

<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdfgehcddcbnccddomop?hl=en>

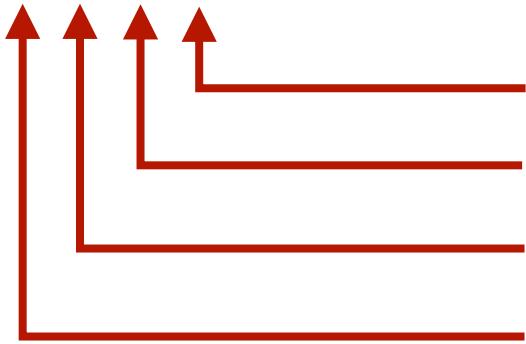
Download REST API Postman Testing Cheat Sheet

https://github.com/ybsuen/JOUR4046/blob/master/Flask_Postman_RESTful_Test.pdf



Create
Retrieve
Update
Delete

CRUD Operations in RESTful API via Postman



Delete
Update
Get
Post

GET (LIST Records) Method

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

POST https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

POST localhost:9007/addblog

▼ August 4

POST 127.0.0.1:5000/addblog

POST 127.0.0.1:5000/

POST localhost:9007/addblog

Request

Builder Team Library

Sync Off

Sign In

Filter

127. http http http http http http No Environment

GET https://api.airtable.com/v0/appM38HXIEVhxmnqx... Send Save

Headers (1)

Key	Value
Authorization	Bearer key

Body Cookies Headers (11) Test Results Status: 200 OK Time: 518 ms

Pretty Raw Preview JSON

Response

```
{ "records": [ { "id": "rec8vF2VbIg0Tdx3F", "fields": { "lname": "Lu", "student_id": "99999", "date_of_birth": "2001-08-22", "fname": "Mary", "full_name": "Mary Lu", "record_id": "rec8vF2VbIg0Tdx3F" }, "createdTime": "2019-09-19T14:32:08.000Z" }, { "id": "recb1ZFR8ZL0H89ti", "fields": { "lname": "Lu", "student_id": "99999", "date_of_birth": "2001-08-22", "fname": "Mary", "full_name": "Mary Lu", "record_id": "recb1ZFR8ZL0H89ti" } } ] }
```

GET (Specific Record) Method

VblgOTdx3F

GET https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo/recpOHXSrfRmlvQia

GET https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo/rec8vF2VbIgOTdx3F

GET https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

DEL https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo/recpOHXSrfRmlvQia

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

Builder Team Library SYNC OFF Sign In Filter http http http http http http No Environment GET https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo/rec8vF2VbIgOTdx3F Params Send Save Headers (1) Authorization Key Value Description Bulk Presets Headers (11) Body Cookies Test Results Status: 200 OK Time: 631 ms Request Response

```
{  
  "id": "rec8vF2VbIgOTdx3F",  
  "fields": {  
    "lname": "Lu",  
    "student_id": "99999",  
    "date_of_birth": "2001-08-22",  
    "fname": "Mary",  
    "full_name": "Mary Lu",  
    "record_id": "rec8vF2VbIgOTdx3F"  
  },  
  "createdTime": "2019-09-19T14:32:08.000Z"  
}
```

POST Method

ble.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST localhost:9007/addblog

▼ August 4

POST 127.0.0.1:5000/addblog

POST 127.0.0.1:5000/

POST localhost:9007/addblog

POST localhost:9007/addblog

POST localhost:9007/addblog

POST localhost:9007/addblog

POST localhost:9007/addblog

Filter Collections Clear all

Builder Team Library

Sync Off Sign In

Chrome apps are being deprecated. Download our free native apps for continued support and better performance. [Learn more](#)

POST https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo

Params Send Save

Headers (2)

Key	Value	Description	...	Bulk	Presets ▾
Authorization	Bearer key				
Content-Type	application/json				
New key	Value	Description			

Response

Hit the Send button to get a response.

Do more with requests

Share Mock Monitor Document



Chrome apps are being deprecated. [Download our free native apps](#) for continued support and better performance. [Learn more](#)

X

POST Method

Collections

Clear all

- PUT https://api.airtable.com/v0/appM38HXIEVhxmnx/flaskdemo
- PUT https://api.airtable.com/v0/appM38HXIEVhxmnx/flaskdemo
- POST https://api.airtable.com/v0/appM38HXIEVhxmnx/flaskdemo
- POST localhost:9007/addblog

▼ August 4

- POST 127.0.0.1:5000/addblog
- POST 127.0.0.1:5000/
- POST localhost:9007/addblog

POST https://api.airtable.com/v0/appM38HXIEVhxmnx... Params Send Save

Authorization Headers (2) Body (selected) Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "records": [  
3     {  
4       "fields": {  
5         "fname": "Bernard",  
6         "lname": "Suen",  
7         "student_id": "121212",  
8         "date_of_birth": "1997-7-1"  
9       }  
10      ]  
11    }  
12 }
```

Request

Status: 200 OK Time: 436 ms

Body Cookies Headers (11) Test Results

Pretty Raw Preview JSON ↻

```
1 {  
2   "records": [  
3     {  
4       "id": "recC41RXQdItIVvDM",  
5       "fields": {  
6         "lname": "Suen",  
7         "student_id": "121212"  
8       }  
9     }  
10   }  
11 }
```

Response

PUT Method

ble.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/rec8vF2
VblgOTdx3F

GET https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

DEL https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/recpOH
XSrfRmlvQia

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST localhost:9007/addblog

Builder Team Library SYNC OFF Sign In

Chrome apps are being deprecated. [Download](#) our free native apps for continued support and better performance. [Learn more](#)

Filter Collections Clear all

PUT https://api.airtable.com/v0/appM38HXIEVhxmnqx... Params Send Save

Headers (2) Authorization: Bearer key Content-Type: application/json

Request

Status: 200 OK Time: 727 ms

Pretty Raw Preview JSON

```
[{"records": [{"id": "rect6erpDSPNty1TN", "fields": {"lname": "Trump", "student_id": "5354", "date_of_birth": "1948-02-21", "fname": "Donald", "full_name": "Donald Trump", "record_id": "rect6erpDSPNty1TN"}, "createdTime": "2019-09-19T15:18:41.000Z"}]}
```

PUT Method

table.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/rec8vF2
VblgOTdx3F

GET https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

DEL https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/recpOH
XSrfRmlvQia

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST localhost:9007/addblog

Builder Team Library SYNC OFF Sign In No Environment Filter Collections Clear all PUT https://api.airtable.com/v0/appM38HXIEVhxmnq... Params Send Save Authorization Headers (2) Body (raw JSON (application/json)) Tests Code

Request

```
{
  "records": [
    {
      "id": "rect6erpDSPNty1TN",
      "fields": {
        "fname": "Donald",
        "lname": "Trump",
        "student_id": "5354",
        "date_of_birth": "1948-02-21"
      }
    }
  ]
}
```

Body Cookies Headers (11) Test Results Status: 200 OK Time: 727 ms

Response

```
{
  "records": [
    {
      "id": "rect6erpDSPNty1TN",
      "fields": {
        "lname": "Trump"
      }
    }
  ]
}
```

DELETE Method

_collections

Clear all

.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/rec8vF2
VblgOTdx3F

GET https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

DEL https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo/recpOH
XSrfRmlvQia

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

PUT https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST https://api.airtable.com/v0/appM3
8HXIEVhxmnqx/flaskdemo

POST localhost:9007/addblog

Builder Team Library SYNC OFF Sign In

Chrome apps are being deprecated. Download our free native apps for continued support and better performance. Learn more

DELETE https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo/recb1ZFR8ZL0HB9ti

Headers (2)

Authorization: Bearer key

Content-Type: application/x-www-form-urlencoded

Params Send Save

Code Bulk Edit Presets

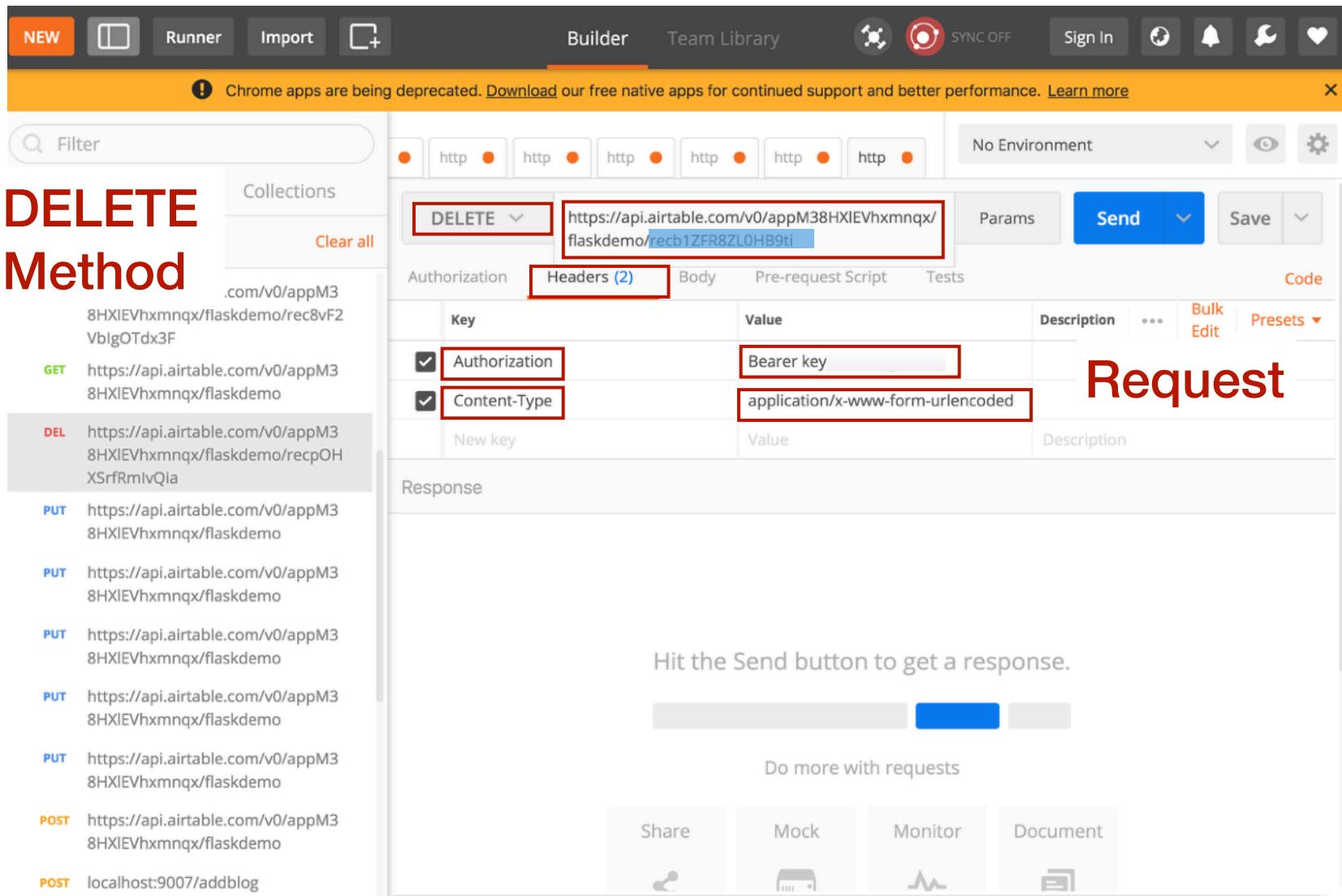
Request

Response

Hit the Send button to get a response.

Do more with requests

Share Mock Monitor Document



DELETE Method

Request

Response

The screenshot shows the Postman application interface. On the left, a sidebar lists various API endpoints with their methods and URLs. A specific endpoint for a DELETE method is highlighted. The main workspace shows a DELETE request being sent to the URL `https://api.airtable.com/v0/appM38HXIEVhxmnqx/flaskdemo`. The 'Body' tab is selected, showing the content type as `x-www-form-urlencoded`. The response body is displayed as a JSON object:

```
1 {  
2   "deleted": true,  
3   "id": "recb1ZFR8ZL0HB9ti"  
4 }
```

REST API Test Using JupyterNotebook

In [33]:

```
1 #  
2 # List Records  
3 #  
4 import requests  
5 import pandas as pd  
6  
7 headers = {  
8     'Authorization': 'Bearer ' ,  
9 }  
10  
11 r = requests.get('https://api.airtable.com/v0/appM38HX1EVhxmnqx/flaskdemo?sortField=_createdTime&sortDirection=desc')  
12 print("Status Code:",r.status_code)  
13 print(r.text)  
14 dict = r.json()  
15 a_list = dict['records']  
16 df = pd.DataFrame.from_dict(a_list)  
17 df.head()
```

Status Code: 200

```
{"records":[{"id":"recb1ZFR8ZL0HB9ti","fields":{"full_name":"William Lo","lname":"Lo","student_id":"5555","date_of_birth":"2002-02-21","record_id":"recb1ZFR8ZL0HB9ti","fname":"William"},"createdTime":"2019-09-21T02:11:26.000Z"}, {"id":"reczlouWve6Oh1JPv","fields":{"full_name":"Winifred Wong","lname":"Wong","student_id":"010585","date_of_birth":"1958-01-28","record_id":"reczlouWve6Oh1JPv","fname":"Winifred"},"createdTime":"2019-09-20T16:37:52.000Z"}, {"id":"rect6erpDSPNty1TN","fields":{"full_name":"Bernard Suen","lname":"Suen","student_id":"11111","date_of_birth":"1958-05-23","record_id":"rect6erpDSPNty1TN","fname":"Bernard"},"createdTime":"2019-09-19T15:18:41.000Z"}, {"id":"rec8vF2VbIgOTdx3F","fields":{"full_name":"Mary Lu","lname":"Lu","student_id":"99999","date_of_birth":"2001-08-22","record_id":"rec8vF2VbIgOTdx3F","fname":"Mary"},"createdTime":"2019-09-19T14:32:08.000Z"}, {"id":"recASxs4aPaHLeJj3","fields":{"full_name":"Emily Chan","lname":"Chan","student_id":"67891","date_of_birth":"2008-04-03","record_id":"recASxs4aPaHLeJj3","fname":"Emily"},"createdTime":"2019-09-19T14:04:24.000Z"}]}
```

Out[33]:

	createdTime	fields	id
0	2019-09-21T02:11:26.000Z	{'full_name': 'William Lo', 'lname': 'Lo', 'stu...}	recb1ZFR8ZL0HB9ti
1	2019-09-20T16:37:52.000Z	{'full_name': 'Winifred Wong', 'lname': 'Wong'...}	reczlouWve6Oh1JPv
2	2019-09-19T15:18:41.000Z	{'full_name': 'Bernard Suen', 'lname': 'Suen',...}	rect6erpDSPNty1TN
3	2019-09-19T14:32:08.000Z	{'full_name': 'Mary Lu', 'lname': 'Lu', 'stude...}	rec8vF2VbIgOTdx3F
4	2019-09-19T14:04:24.000Z	{'full_name': 'Emily Chan', 'lname': 'Chan', '...}	recASxs4aPaHLeJj3

In [38]:

```
1 #  
2 # Add Record  
3 #  
4  
5 import requests  
6 import pandas as pd  
7  
8 data = {  
9     "records": [  
10         {  
11             "fields": {  
12                 "fname": "Emily",  
13                 "lname": "Lau",  
14                 "student_id": "686868",  
15                 "date_of_birth": "2002-02-21"  
16             }  
17         }  
18     ]  
19 }  
20 headers = {'Authorization': 'Bearer - - - - -', 'Content-Type': 'application/json; charset=utf-8'}  
21 r = requests.post('https://api.airtable.com/v0/appM38HX1EVhxmnqx/flaskdemo', json=data, headers=headers)  
22 print("Status Code:", r.status_code)  
23 print(r.text)  
24 dict = r.json()  
25 a_list = dict['records']  
26 df = pd.DataFrame.from_dict(a_list)  
27 df.head()  
  
Status Code: 200  
{ "records": [{ "id": "recggGoKr4zQhqj3l", "fields": { "lname": "Lau", "student_id": "686868", "date_of_birth": "2002-02-21", "fname": "Emily", "full_name": "Emily Lau", "record_id": "recggGoKr4zQhqj3l" }, "createdTime": "2019-09-21T02:53:23.000Z" } ]}
```

Out[38]:

	createdTime	fields	id
0	2019-09-21T02:53:23.000Z	{'lname': 'Lau', 'student_id': '686868', 'date... recggGoKr4zQhqj3l	

In [40]:

```
1 #  
2 # Update Records  
3 #  
4 import requests  
5 import pandas as pd  
6  
7 data = {  
8     "records": [  
9         {  
10             "id": "recggGoKr4zQhqj3l",  
11             "fields": {  
12                 "fname": "Emily",  
13                 "lname": "Chan",  
14                 "student_id": "67891",  
15                 "date_of_birth": "2008-04-03"  
16             }  
17         }  
18     ]  
19 }  
20 headers = {'Authorization': 'Bearer', 'Content-Type': 'application/json; charset=utf-8'}  
21 r = requests.put('https://api.airtable.com/v0/appM38HX1EVhxmnqx/flaskdemo', json=data, headers=headers)  
22 print("Status Code:", r.status_code)  
23 print(r.text)  
24 dict = r.json()  
25 a_list = dict['records']  
26 df = pd.DataFrame.from_dict(a_list)  
27 df.head()  
  
Status Code: 200  
{"records": [{"id": "recggGoKr4zQhqj3l", "fields": {"lname": "Chan", "student_id": "67891", "date_of_birth": "2008-04-03", "fname": "Emily", "full_name": "Emily Chan", "record_id": "recggGoKr4zQhqj3l"}, "createdTime": "2019-09-21T02:53:23.000Z"}]}
```

Out[40]:

	createdTime	fields	id
0	2019-09-21T02:53:23.000Z	{"lname": "Chan", "student_id": "67891", "date... recggGoKr4zQhqj3l	

In [41]:

```
1 #  
2 # Delete Record  
3 #  
4 import requests  
5 import pandas as pd  
6  
7 headers = {'Authorization': 'Bearer', 'Content-Type': 'application/x-www-form-urlencoded'}  
8 r = requests.delete('https://api.airtable.com/v0/appM38HX1EVhxmnqx/flaskdemo/recggGoKr4zQhqqj31',headers=headers)  
9 print("Status Code:",r.status_code)  
10 print(r.text)  
11  
12 headers = {  
13     'Authorization': 'Bearer',  
14 }  
15 r = requests.get('https://api.airtable.com/v0/appM38HX1EVhxmnqx/flaskdemo?  
16                 sortField=_createdTime&sortDirection=desc', headers=headers)  
17 dict = r.json()  
18 a_list = dict['records']  
19 df = pd.DataFrame.from_dict(a_list)  
20 df.head()  
  
Status Code: 200  
{"deleted":true,"id":"recggGoKr4zQhqqj31"}
```

Out[41]:

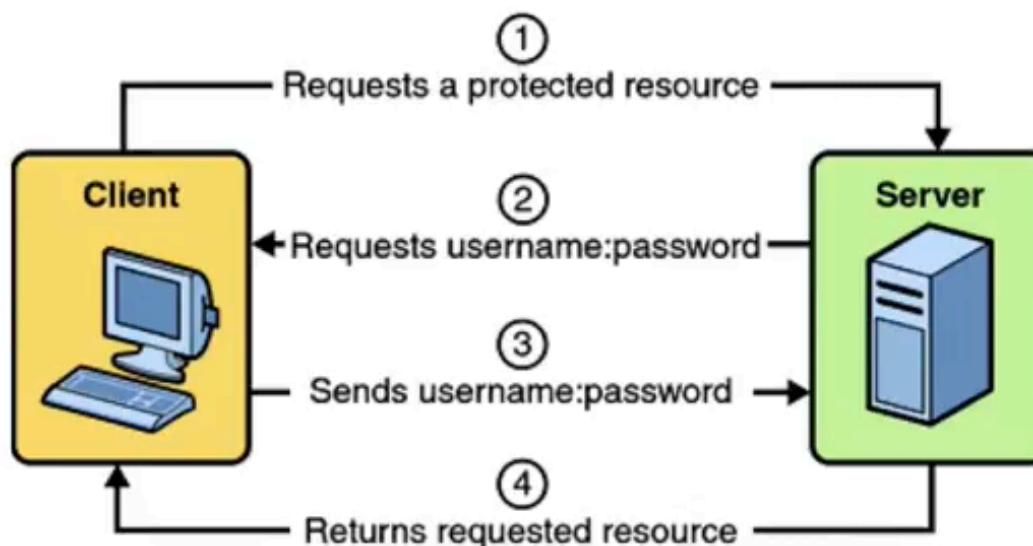
	createdTime	fields	id
0	2019-09-21T02:11:26.000Z	{'full_name': 'William Lo', 'lname': 'Lo', 'stu...}	recb1ZFR8ZL0HB9ti
1	2019-09-20T16:37:52.000Z	{'full_name': 'Winifred Wong', 'lname': 'Wong'...}	reczlouWve6Oh1JPv
2	2019-09-19T15:18:41.000Z	{'full_name': 'Bernard Suen', 'lname': 'Suen',...}	rect6erpDSPNty1TN
3	2019-09-19T14:32:08.000Z	{'full_name': 'Mary Lu', 'lname': 'Lu', 'stude...}	rec8vF2VblgOTdx3F

Web Security in Data Exchange Through API

TO GRAB DATA THROUGH WP REST API, YOU NEED:

- URI (Resources Identifier)
- HTTP method
- Headers (Authorization, Content-Type)
- Body (JSON, form-urlencoded)
- Authentication method (Basic, OAuth 2)

Basic Authentication



Source: <http://docs.oracle.com/cd/E19226-01/820-7627/bnc1>



https://en.wikipedia.org/wiki/Basic_access_authentication

The screenshot shows a web browser window with the URL https://en.wikipedia.org/wiki/Basic_access_authentication. The page title is "Basic access authentication". The main content area describes the concept of basic access authentication in the context of an HTTP transaction, mentioning its use by user agents like web browsers, and how it involves sending credentials in base64 encoding. It also notes its specification in RFC 7617 (2015) and RFC 2617 (1999). A sidebar on the right contains a summary of HTTP-related topics.

Article Talk Read Edit View history Search Wikipedia

Basic access authentication

From Wikipedia, the free encyclopedia

In the context of an [HTTP](#) transaction, **basic access authentication** is a method for an [HTTP user agent](#) (e.g. a web browser) to provide a [user name](#) and [password](#) when making a request. In basic HTTP authentication, a request contains a header field of the form `Authorization: Basic <credentials>`, where `credentials` is the base64 encoding of id and password joined by a colon.

It is specified in [RFC 7617](#) from 2015, which obsoletes [RFC 2617](#) from 1999.

Contents [hide]

- [1 Features](#)
- [2 Security](#)
- [3 Protocol](#)
 - [3.1 Server side](#)
 - [3.2 Client side](#)
 - [3.3 URL encoding](#)
- [4 See also](#)
- [5 References and notes](#)
- [6 External links](#)

Features [edit]

HTTP

Persistence · Compression · HTTPS · QUIC

Request methods

OPTIONS · GET · HEAD · POST · PUT · DELETE · TRACE · CONNECT · PATCH

Header fields

Cookie · ETag · Location · HTTP referer · DNT · X-Forwarded-For

Status codes

301 Moved Permanently · 302 Found · 303 See Other · 403 Forbidden · 404 Not Found · 451Unavailable For Legal Reasons

Security access control methods

[Basic access authentication](#) · [Digest access authentication](#)

V · T · E

The background of the image is a clear blue sky with a few wispy white clouds. The clouds are thin and wispy, scattered across the upper half of the frame.

<https://www.youtube.com/watch?v=zEysfglbqlg>

OAuth 2.0 [edit]

OAuth 2.0 is not backwards compatible with OAuth 1.0. OAuth 2.0 provides specific authorization flows for web applications, desktop applications, mobile phones, and [smart devices](#). The specification and associated RFCs are developed by the IETF OAuth WG;^[7] the main framework was published in October 2012.

[Facebook's Graph API](#) only supports OAuth 2.0.^[8] [Google](#) supports OAuth 2.0 as the recommended authorization mechanism for all of its APIs.^[9] [Microsoft](#)^[10] also supports OAuth 2.0 for various [APIs](#) and its Azure Active Directory service, which is used to secure many Microsoft and third party [APIs](#).

The OAuth 2.0 Framework^[11] and Bearer Token Usage^[12] were published in October 2012.

Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
Related changes
Upload file

OAuth

From Wikipedia, the free encyclopedia

For MediaWiki's (the software used by Wikipedia) OAuth support, see [mw:Help:OAuth](#)

OAuth is an [open standard](#) for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.^[1] This mechanism is used by companies such as Amazon,^[2] Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites. Generally, OAuth provides to clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with [Hypertext Transfer Protocol](#) (HTTP), OAuth essentially allows [access tokens](#) to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.^[3]

OAuth is a service that is complementary to and distinct from [OpenID](#). OAuth is also distinct from [OATH](#).



The OAuth logo, designed by [Chris Messina](#)

- ▶ **Web template development in Python: Introducing Flask Jinja2 (i.e. render_template library)**

Login-Logout in Flask

Bernard's Login Page

Resize this responsive page to see the effect!

Please login in first.

Please sign in

bernard

Remember me

Sign in

Home Page

Resize this responsive page to see the effect!

Dear bernard, welcome to Bernard's pages. Your login has been granted.

Column 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...

Column 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...

Column 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit...

Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris...

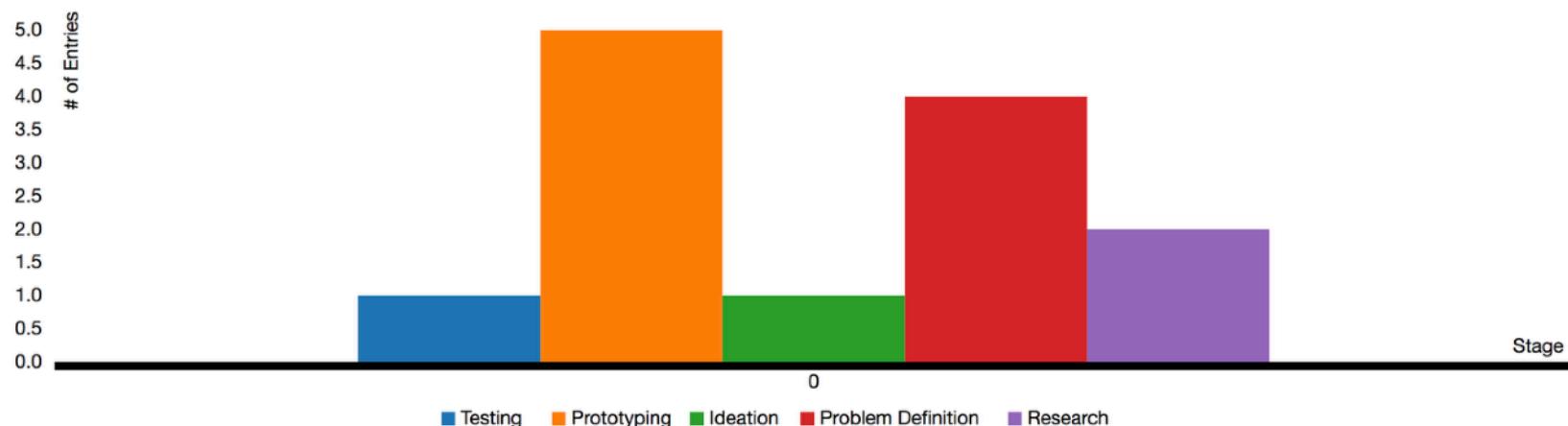
Bernard's Login Page

Resize this responsive page to see the effect!

You're already login.

Bernard's C3 Chart Page

Resize this responsive page to see the effect!



Bernard's Login Page

Resize this responsive page to see the effect!

Thanks for logging out.

Please sign in

Remember me

Search...

Getting started

Layout

Content

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Modal

Navs

Navbar

Pagination

Popovers

Forms

Examples and usage guidelines for form control styles, layout options, and custom components for creating a wide variety of forms.



Swift & straightforward
bug fixes for your web &
mobile apps. Try Bugsnag
free.
ads via Carbon

Overview

Bootstrap's form controls expand on [our Rebooted form styles](#) with classes. Use these classes to opt into their customized displays for a more consistent rendering across browsers and devices.

Be sure to use an appropriate `type` attribute on all inputs (e.g., `email` for email address or `number` for numerical information) to take advantage of newer input controls like email verification, number selection, and more.

Here's a quick example to demonstrate Bootstrap's form styles. Keep reading for documentation on required classes, form layout, and more.

Email address

Enter email

We'll never share your email with anyone else.

Overview

Form controls

Sizing

Readonly

Readonly plain text

Checkboxes and radios

Default (stacked)

Inline

Without labels

Layout

Form groups

Form grid

Inline forms

Help text

Disabled forms

Cross-browser compatibility

Validation

How it works

Custom styles

Browser defaults

Server side

Supported elements

Tooltips

Custom forms

Checkboxes and radios

Select menu

File browser

Search...

Getting started

Layout

Content

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Modal

Navs

Navbar

Pagination

Popovers

Email address

Enter email

We'll never share your email with anyone else.

Password

Password

Check me out

Submit

Copy

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Email address" value="johndoe@example.com" />
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password" value="password" />
  </div>
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1" checked="" />
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Overview

Form controls

Sizing

Readonly

Readonly plain text

Checkboxes and radios

Default (stacked)

Inline

Without labels

Layout

Form groups

Form grid

Inline forms

Help text

Disabled forms

Cross-browser compatibility

Validation

How it works

Custom styles

Browser defaults

Server side

Supported elements

Tooltips

Custom forms

Checkboxes and radios

Select menu

File browser

Search...

Getting started

Layout

Content

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Modal

Navs

Navbar

Pagination

Popovers

Form grid

More complex forms can be built using our grid classes. Use these for form layouts that require multiple columns, varied widths, and additional alignment options.

First name

Last name

Copy

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="First name">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Last name">
    </div>
  </div>
</form>
```

Form row

You may also swap `.row` for `.form-row`, a variation of our standard grid row that overrides the default column gutters for tighter and more compact layouts.

First name

Last name

Copy

```
<form>
  <div class="form-row">
```

Overview

Form controls

Sizing

Readonly

Readonly plain text

Checkboxes and radios

Default (stacked)

Inline

Without labels

Layout

Form groups

Form grid

Inline forms

Help text

Disabled forms

Cross-browser compatibility

Validation

How it works

Custom styles

Browser defaults

Server side

Supported elements

Toolips

Custom forms

Checkboxes and radios

Select menu

File browser

```
from flask import Flask, render_template, request, json, redirect, session
from flask import Markup
from flask_login import LoginManager, login_user, logout_user, login_required, UserMixin
import requests

app = Flask(__name__)
app.config["DEBUG"] = False
app.config['SECRET_KEY'] = "JutzX21J0Bq0dxlCV8xqqnx0"
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    return User(user_id)

class User(UserMixin):
    def __init__(self,id):
        self.id = id

@app.route("/")
@login_required
def home():
    return render_template('home.html')

@app.route("/result")
@login_required
def result():
    headers = {
        'Authorization': 'Bearer [REDACTED]', 
    }

    params = (
        ('maxRecords', '25'),
        ('view', 'Main View'),
    )

    r = requests.get('https://api.airtable.com/v0/appM38HX1EVhxmnqx/Tasks?api_key=keyTcsTzckqyBTlk8&sortField=_createdTime&sortDirection=desc', headers=headers, params=params)
    dict = r.json()
    datacont = dict
```

```
params = (
    ('view', 'Grid view'),
)
r = requests.get('https://api.airtable.com/v0/appM38HX1EVhxmnx/venues?api_key=keyTcsTzckqyBTlk&sortField=_createdTime&sortDirection=desc', headers=headers, params=params)
dict = r.json()
dataset = []
data = []
items = {}
total_entries_list = []
for i in dict['records']:
    dict = i['fields']
    dataset.append(dict)
return render_template('map.html', entries = dataset)

@app.route("/login")
def login():
    message = 'Please login in first.'
    return render_template('login.html', message=message)

@app.route("/process",methods=['POST'])
def process():
    username = request.form['username']
    password = request.form['password']
    if password == 'password':
        login_user(User(1))
        message = "Dear " + username + ", welcome to Bernard's pages. Your login has been granted."
        return render_template('home.html', message=message)
    message = 'wrong password!'
    return render_template('login.html',message=message)

@app.route('/logout/')
@login_required
def logout():
    logout_user()
    message = 'Thanks for logging out.'
    return render_template('login.html',message=message)

if __name__ == '__main__':
    app.run(debug = True)
```

```
</div>
<div class="collapse navbar-collapse" id="myNavbar">
  <ul class="nav navbar-nav">
    <li><a href="/">Home</a></li>
    <li class="active"><a href="/result">Result</a></li>
    <li><a href="/table">Table</a></li>
    <li><a href="/chart">Chart</a></li>
    <li><a href="/map">Map</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="/logout"><span class="glyphicon glyphicon-log-out"></span>Logout</a></li>
    <li><a href="/login"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
  </ul>
</div>
</div>
</nav>
<div class="jumbotron text-center">
  <h1>Bernard's Login Page</h1>
  <p>Resize this responsive page to see the effect!</p>
  {% if current_user.is_authenticated %}
    <p>You're already login.</p>
  {% else %}
    <p>{{message}}</p>
  {% endif %}
</div>

<div class="container">
  {% if not current_user.is_authenticated %}
    <center><form class="form-signin" style="width:30%;" method="post" action="/process">
      <h2 class="form-signin-heading">Please sign in</h2>
      <label for="inputEmail" class="sr-only">User name</label>
      <input type="text" name="username" id="username" class="form-control" placeholder="User name" required autofocus>
      <label for="inputPassword" class="sr-only">Password</label>
      <input type="password" name="password" id="inputPassword" class="form-control" placeholder="Password" required>
      <div class="checkbox">
        <label>
          <input type="checkbox" value="remember-me"> Remember me
        </label>
      </div>
      <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
    </form></center>
  {% endif %}
</div>
```

Add advanced interaction controls to your HTML tables *the free & easy way*

1 - Include these two files ↓

css //cdn.datatables.net/1.10.19/css/

js //cdn.datatables.net/1.10.19/js/

Show 10 entries				Search:
Name	Position	Office	Age	
Airi Satou	Accountant	Tokyo	33	
Angelica Ramos	Chief Executive Officer (CEO)	London	47	

<https://datatables.net/>

THANK YOU FOR YOUR TIME!