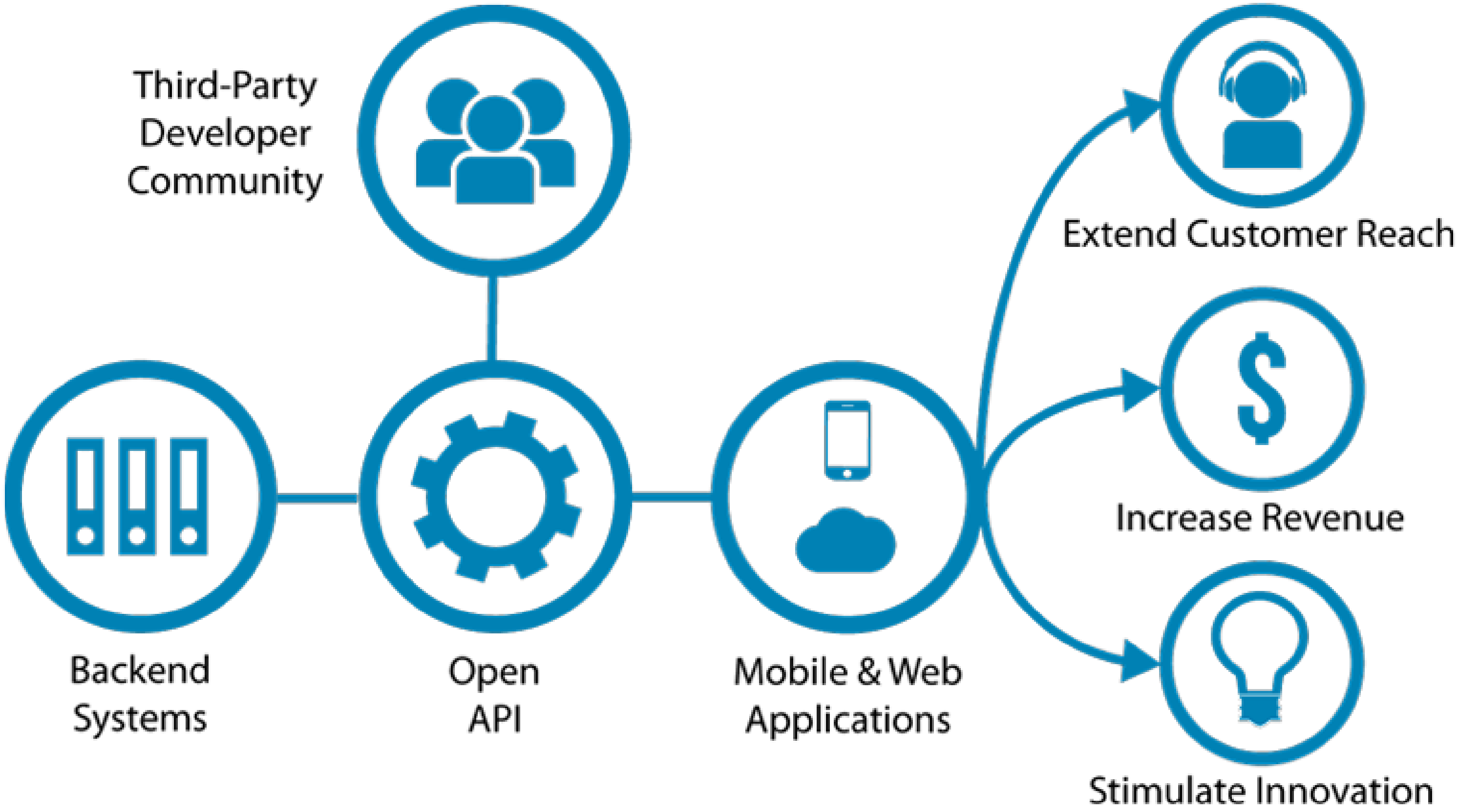
REST

AARHUS UNIVERSITY

AARHUS UNIVERSITY SCHOOL OF ENGINEERING

MICHAEL LOFT
ML@ECE.AU.DK

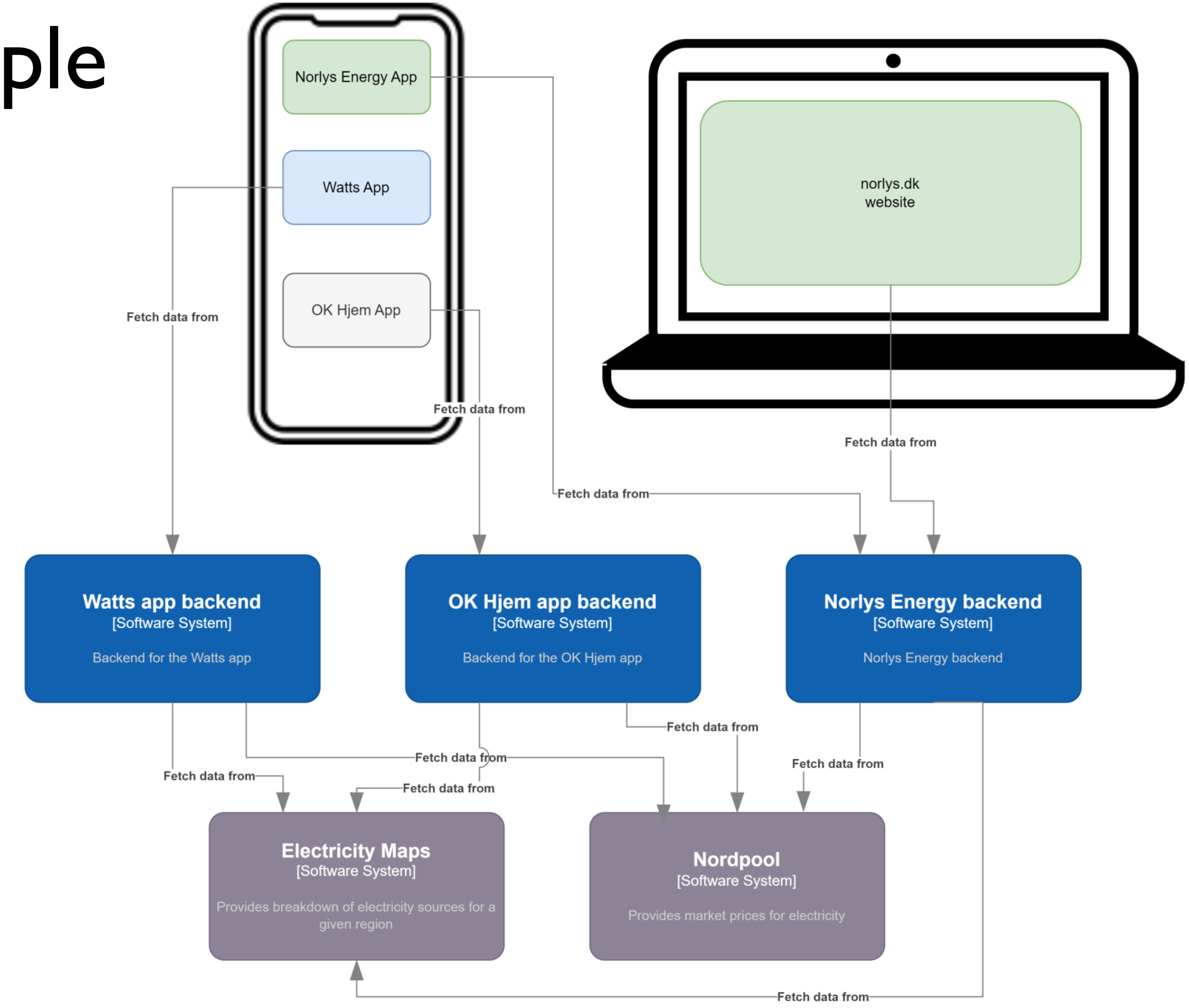
Systems depending on other systems



APIs lets applications interface to backend systems.

If the API is public, it can be used from other systems if you give the system access.

Example


Norlys Energy App
Watts App
OK Hjem App
norlys.dk
website
Fetch data from
Fetch data from
Fetch data from
Fetch data from
Watts app backend
[Software System]
Backend for the Watts app
OK Hjem app backend
[Software System]
Backend for the OK Hjem app
Norlys Energy backend
[Software System]
Norlys Energy backend
Fetch data from
Fetch data from
Fetch data from
Fetch data from
Fetch data from
Electricity Maps
[Software System]
Provides breakdown of electricity sources for a given region
Nordpool
[Software System]
Provides market prices for electricity
Fetch data from

WHAT IS REST?

A set of guiding principles that a Web API should adhere to, to be a REST API

Architectural style

REpresentational State Transfer

The ‘State’ is the data managed by the API.

Data is (almost) always stored in a database.

The REST API itself is stateless, it merely translates the data (state) stored in the database to and from JSON objects.

RESTFul principles

1. Uniform interface
    1. Resource identification
    2. Initial URL
2. Client-server
    1. Separation of concern
3. Stateless
    1. The server doesn’t hold any state
4. Cacheable
    1. The client can save the response
5. Layered System
    1. Only depended on the next immediate layer
6. Code on Demand (optional and not often used in practice)
    1. Possibility to extend clients

DESIGNING REST API

1. Identify resources
2. Create URI
3. Determining representation
4. Assigning HTTP Verbs

Domain model

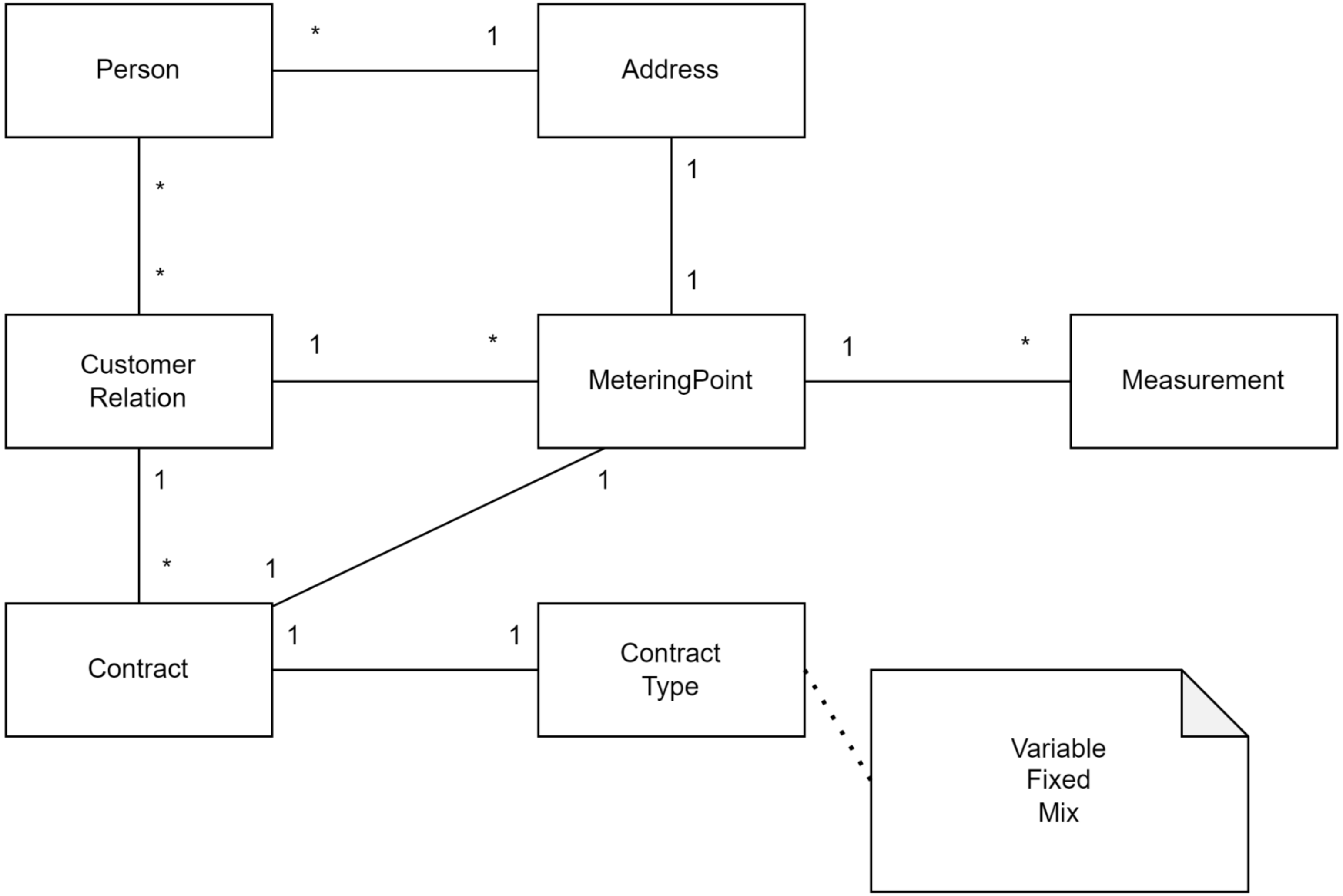
Before defining a service and the REST API, you should define the domain model.

The domain model describes the domain and the REST API defines how to navigate and query that model.

Domain model example

Before defining a service and the REST API, you should define the domain model.

The domain model describes the domain and the REST API defines how to navigate and query that model.



CREATING URIS

- Should be **nouns** only
- Resource normally in plural
  - And {id} to select specific
  - `/patients` vs `/patients/{cpr}`
- Sub-collections can be specified in URL by specifying collection URL
  - `/patients/{id}`/observations

CREATING URIS

- Should be **nouns** only
- Resource normally in plural
  - And {id} to select specific
  - `/patients` vs `/patients/{cpr}`
- Sub-collections can be specified in URL by specifying collection URL
  - `/patients/{id}`/observations

Think carefully about, how API clients should be able to navigate your data model as structured trees. The API reflects that.

CREATING URIS

- Should be **nouns** only
- Resource normally in plural
  - And {id} to select specific
  - `/patients` vs `/patients/{cpr}`
- Sub-collections can be specified in URL by specifying collection URL
  - `/patients/{id}`/observations

Think carefully about, how API clients should be able to navigate your data model as structured trees. The API reflects that.

You probably need to redesign your API 2 - 3 times before you are satisfied. Don’t worry – this is normal.

RESOURCE REPRESENTATION

- Either XML or JSON
- Returning the most important data - especially when returning a collection
- Returning a single resource
    - Include all data
    - Include relevant links
    - E.g. link to patients observations
  - /patients/1234567890/oberservations/123
  - /patients/1234567890/oberservations/180
  - /patients/1234567890/oberservations/231

HTTP VERBS

- Use GET to browse data
  - Big collections should have pagination
  - /patients/123456789/observations
- Use POST to create
- Use PUT to update
- Use DELETE to delete

Very similar to basic database operations:

Create - POST
Read - GET
Update - PUT
Delete - DELETE

HATEOAS

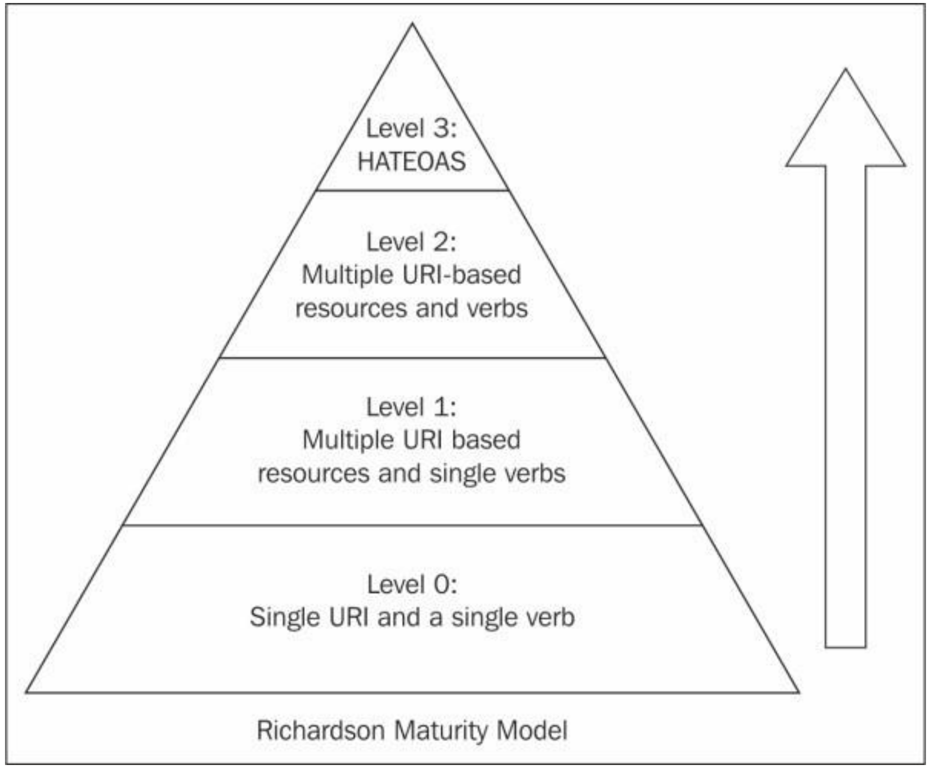
Hypermedia **A** **T**he **E**ngine **O**f **A**pplication **S**tate

```
{
 "departmentId": 10, "departmentName": "Administration", "locationId": 1700, "managerId": 200,
        "links": [ {
                  "href": "10/employees",
                  "rel": "employees",
                  "type" : "GET”
         } ]
}
```

- A dynamic way of navigating to related resources
- Clients don’t need to be hard-coded
- Don’t need to be in the body – could also be in the header

REST MATURITY LEVELS

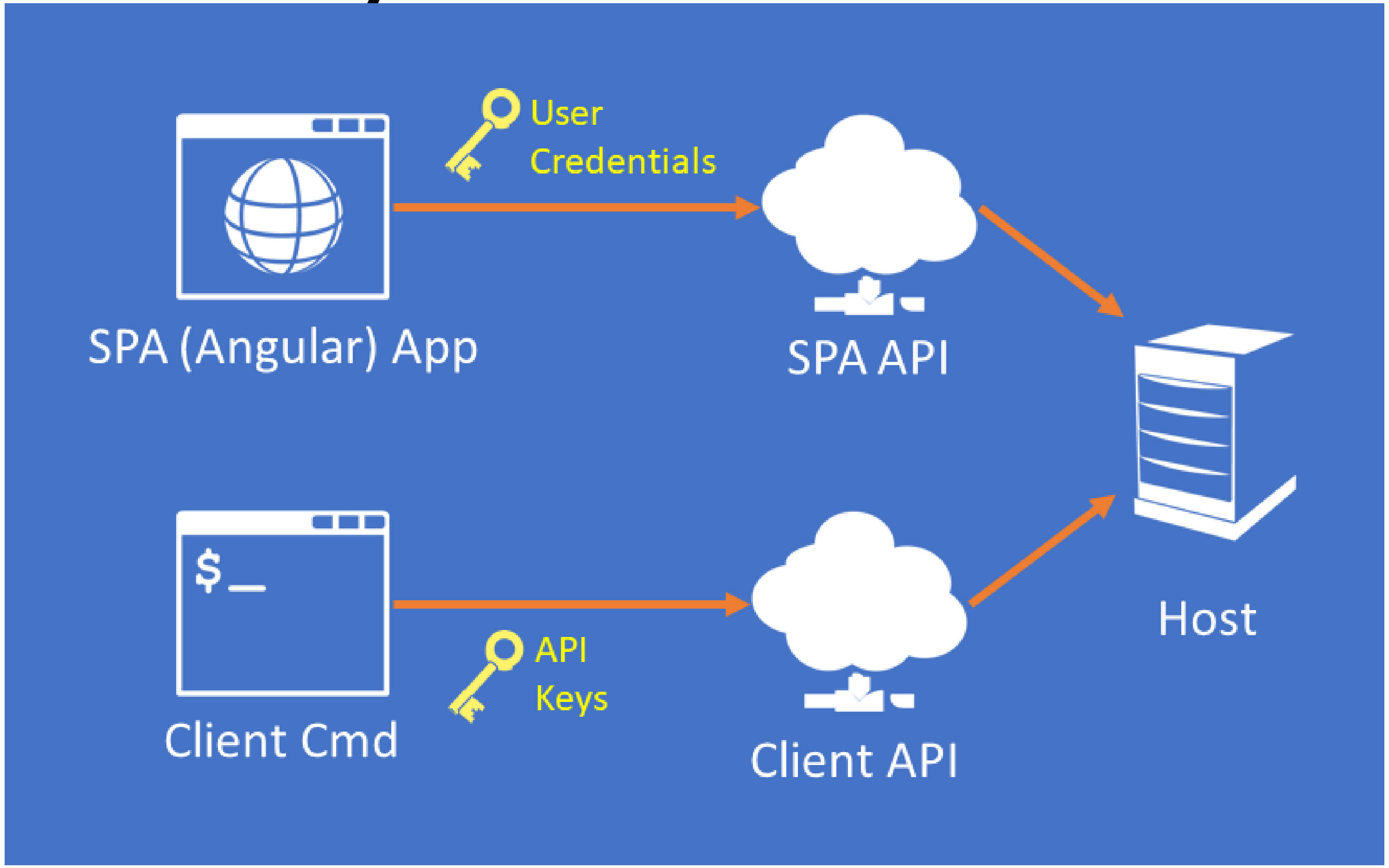
- Level 0:
  - Single word and single VERB
  - E.g. only a single http endpoint using post or get
- Level 1:
  - Level 0, but with multiple resources
- Level 2:
  - Multiple resources (endpoints) and uses different VERBs for CRUD operations
  - This you get from WebAPI if you use it like in the tutorial
- Level 3:
  - Level 2 + HATEOAS

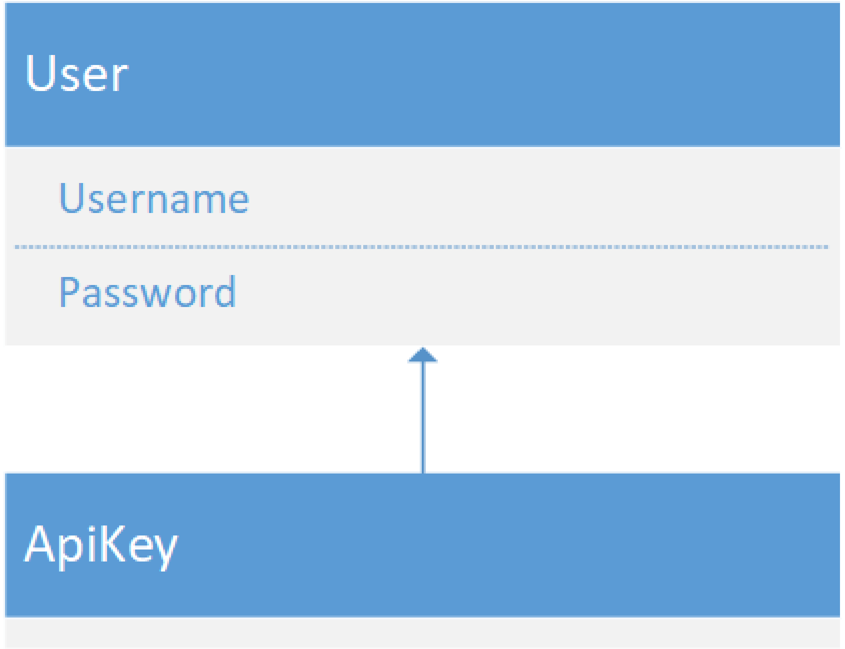

Level 3:
HATEOAS
Level 2:
Multiple URI-based resources and verbs
Level 1:
Multiple URI based resources and single verbs
Level 0:
Single URI and a single verb
Richardson Maturity Model

SECURITY

- HTTPS
  - Always use HTTPs when creating Web API (And web pages in general)
- API keys
  - Hide this key
- OAUTH

API Keys



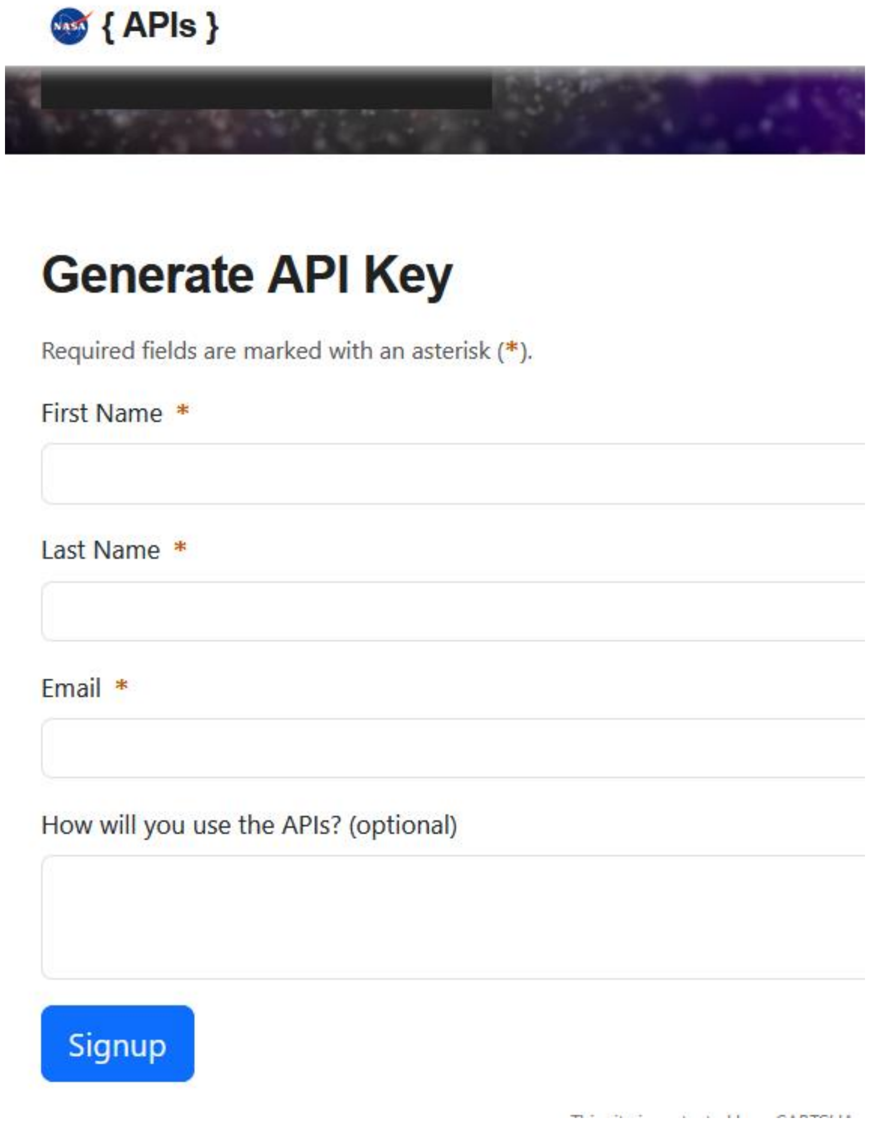


A way to connect to an API with credentials that are separate and more limited than login credentials (e.g. username+password)

https://blog.infernored.com/securing-web-apis-with-api-keys/

Example of using API keys

Request key on
https://api.nasa.gov/



{ APIs }

[redacted]

Generate API Key

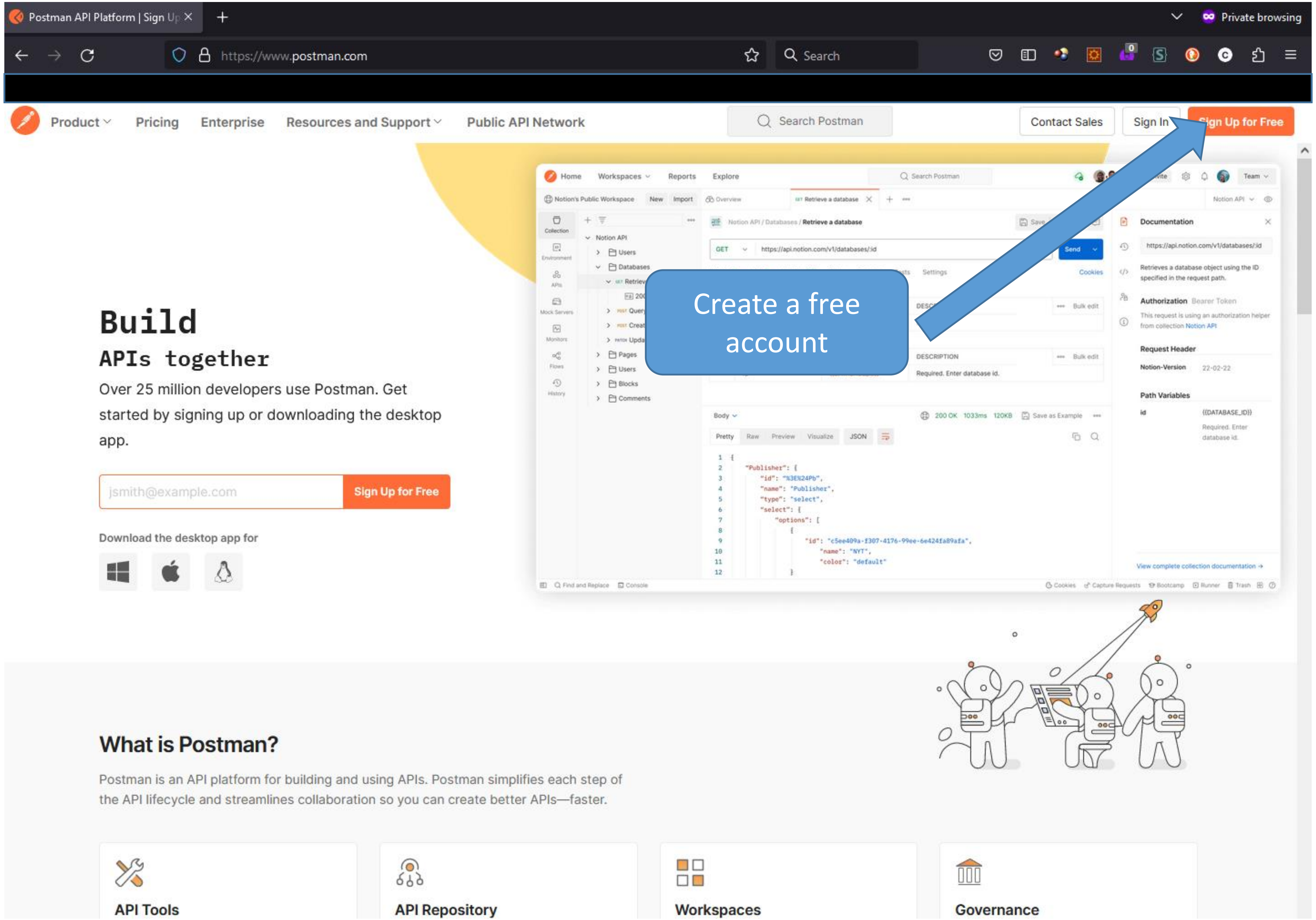
Required fields are marked with an asterisk (*).

First Name *

Last Name *

Email *

How will you use the APIs? (optional)



Create a free account

Build
APIs together

Over 25 million developers use Postman. Get started by signing up or downloading the desktop app.

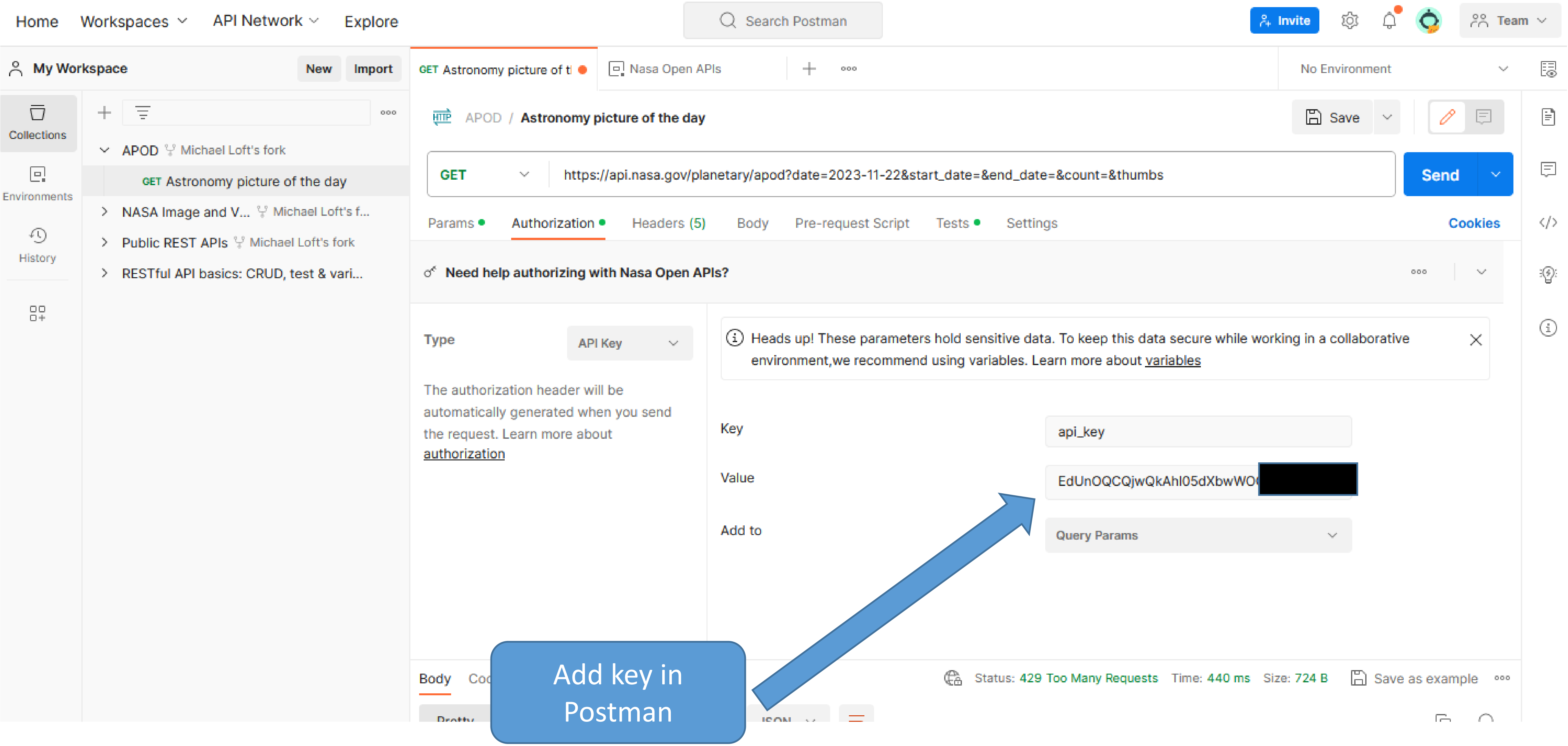
jsmith@example.com

Sign Up for Free

Download the desktop app for

What is Postman?

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.



GET https://api.nasa.gov/planetary/apod?date=2023-11-22&start_date=&end_date=&count=&thumbs

Need help authorizing with Nasa Open APIs?

Type: API Key

The authorization header will be automatically generated when you send the request. Learn more about authorization

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recommend using variables. Learn more about variables

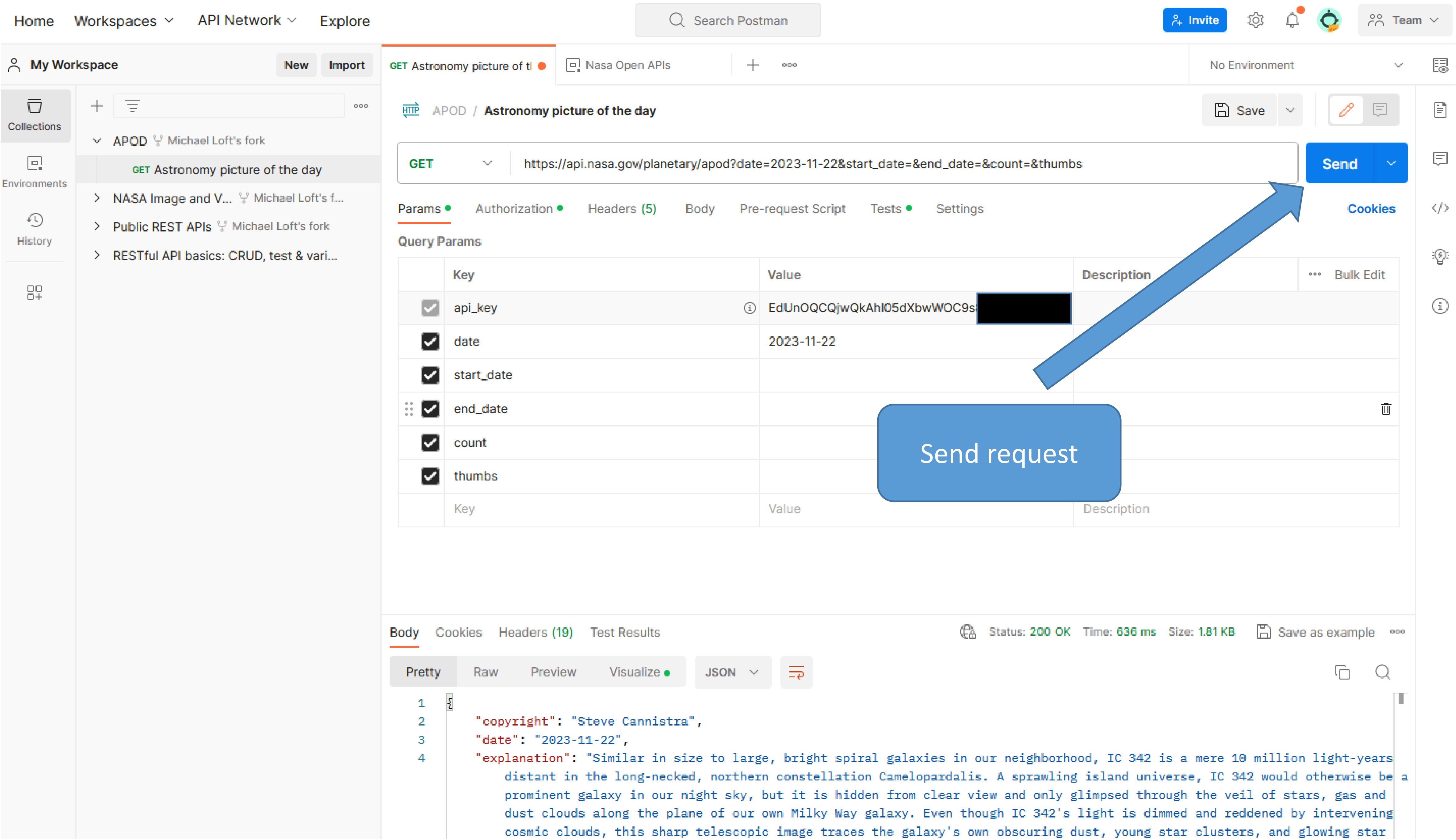
Key: api_key

Value: EdUnOQCQjwQkAhl05dXbwWO[redacted]

Add to: Query Params

Status: 429 Too Many Requests Time: 440 ms Size: 724 B

Add key in Postman



GET https://api.nasa.gov/planetary/apod?date=2023-11-22&start_date=&end_date=&count=&thumbs

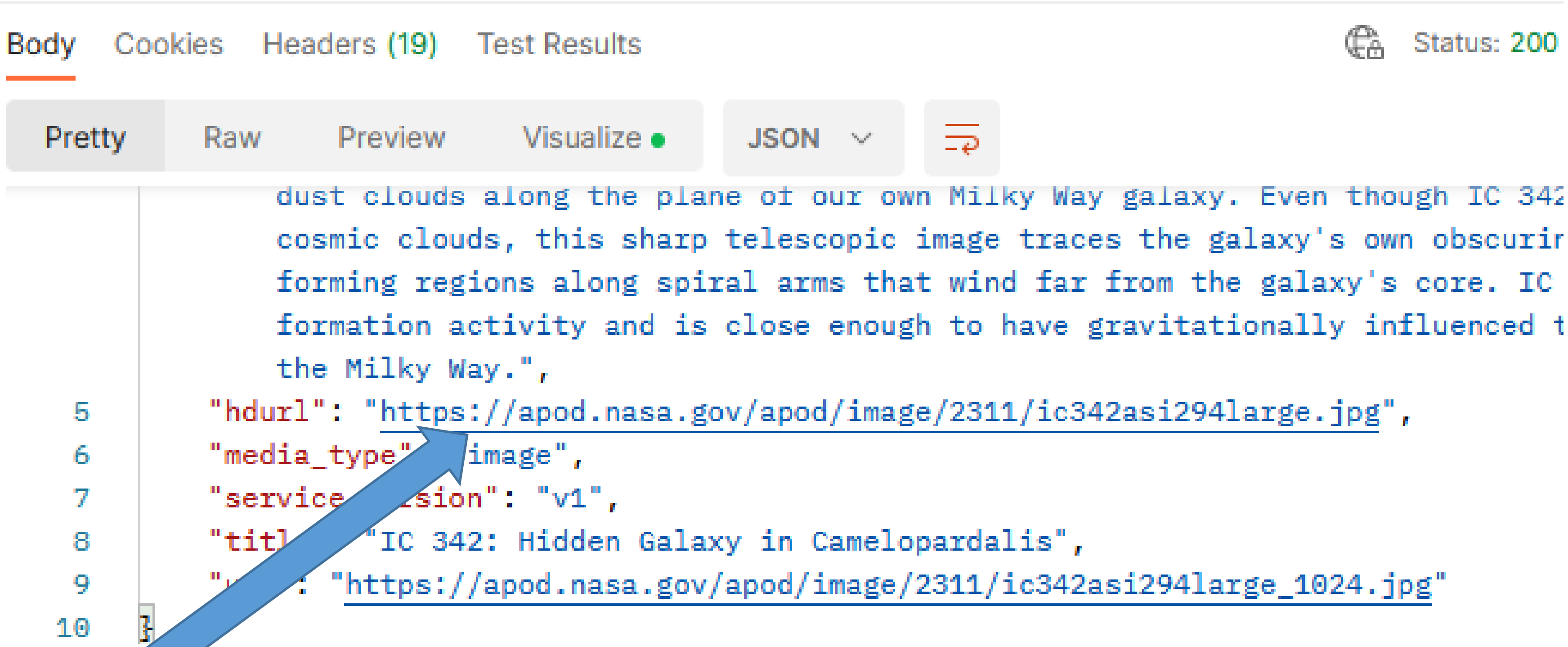
Query Params

Key	Value	Description
☑ api_key	EdUnOQCQjwQkAhl05dXbwWOC9s[redacted]	
☑ date	2023-11-22	
☑ start_date		
☑ end_date		
☑ count		
☑ thumbs		
Key	Value	Description

Send request

Status: 200 OK Time: 636 ms Size: 1.81 KB

```
{
    "copyright": "Steve Cannistra",
    "date": "2023-11-22",
    "explanation": "Similar in size to large, bright spiral galaxies in our neighborhood, IC 342 is a mere 10 million light-years
        distant in the long-necked, northern constellation Camelopardalis. A sprawling island universe, IC 342 would otherwise be a
        prominent galaxy in our night sky, but it is hidden from clear view and only glimpsed through the veil of stars, gas and
        dust clouds along the plane of our own Milky Way galaxy. Even though IC 342's light is dimmed and reddened by intervening
        cosmic clouds, this sharp telescopic image traces the galaxy's own obscuring dust, young star clusters, and glowing star
```



```
        dust clouds along the plane of our own Milky Way galaxy. Even though IC 342
        cosmic clouds, this sharp telescopic image traces the galaxy's own obscurir
        forming regions along spiral arms that wind far from the galaxy's core. IC
        formation activity and is close enough to have gravitationally influenced t
        the Milky Way.",
    "hdurl": "https://apod.nasa.gov/apod/image/2311/ic342asi294large.jpg",
    "media_type": "image",
    "service_version": "v1",
    "title": "IC 342: Hidden Galaxy in Camelopardalis",
    "url": "https://apod.nasa.gov/apod/image/2311/ic342asi294large_1024.jpg"
}
```

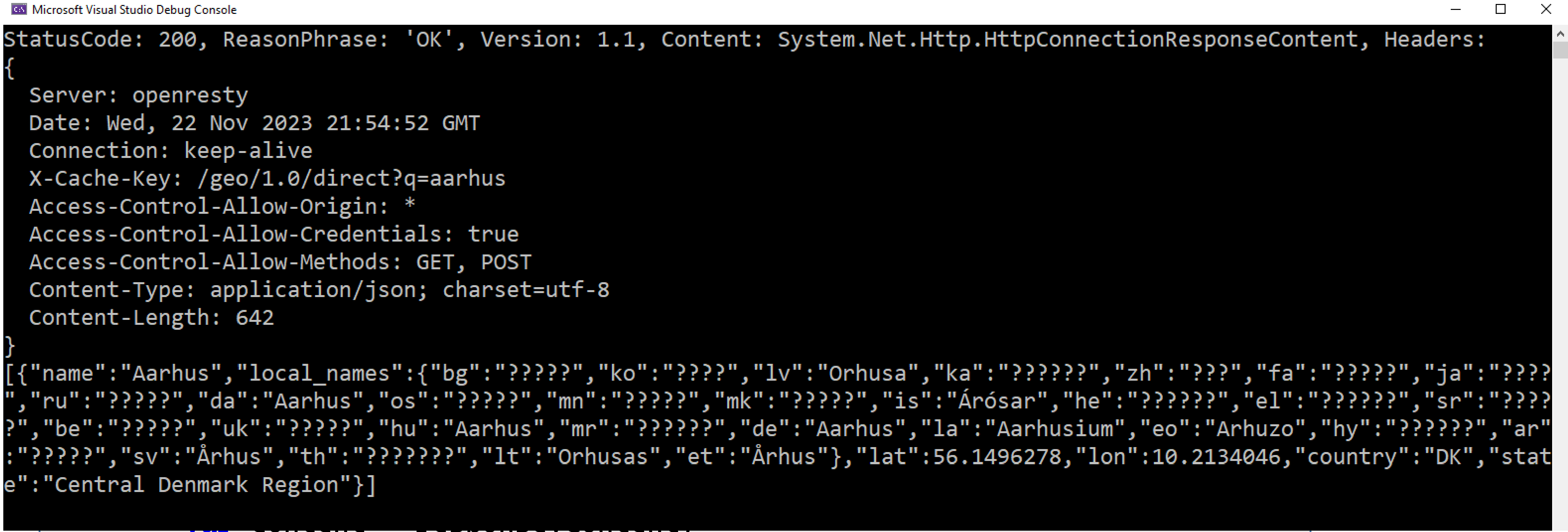
HATEOAS

Get API data with HttpClient

```
internal class Program
{
    private static string apikey = "41bc7[redacted]0";

    private static HttpClient sharedClient = new()
    {
        BaseAddress = new Uri("http://api.openweathermap.org")
    };

    static void Main(string[] args)
    {
        using HttpResponseMessage response =

sharedClient.GetAsync($"/geo/1.0/direct?q=Aarhus&appid={apikey}").Result;

        Console.WriteLine(response);

        var content = response.Content;
        var contentResponse = content.ReadAsStringAsync().Result;
        Console.WriteLine($"{contentResponse}\n");
    }
}
```

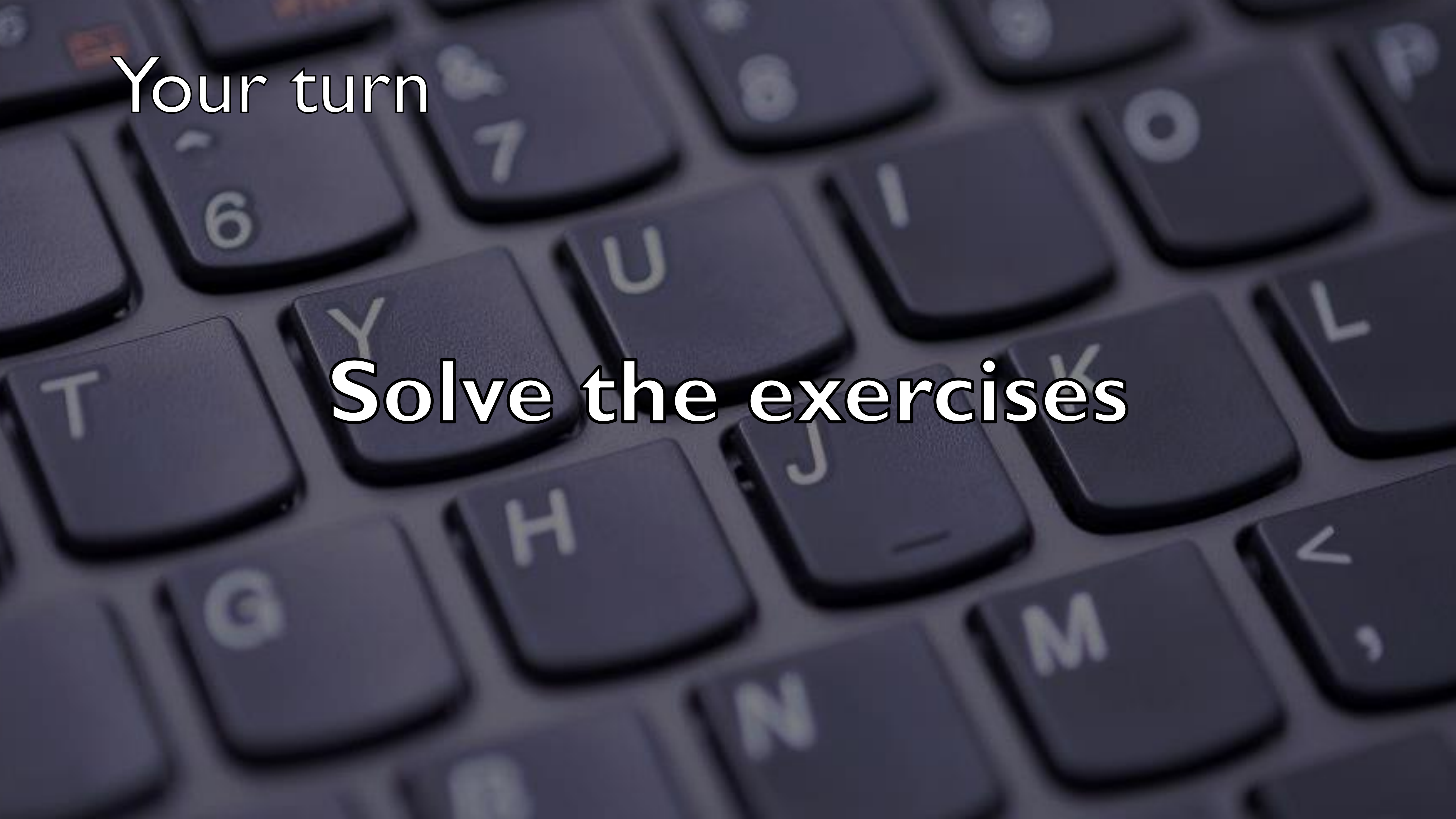
Get API data with HttpClient

```
internal class Program
{
```



```
StatusCode: 200, ReasonPhrase: 'OK', Version: 1.1, Content: System.Net.Http.HttpConnectionResponseContent, Headers:
{
  Server: openresty
  Date: Wed, 22 Nov 2023 21:54:52 GMT
  Connection: keep-alive
  X-Cache-Key: /geo/1.0/direct?q=aarhus
  Access-Control-Allow-Origin: *
  Access-Control-Allow-Credentials: true
  Access-Control-Allow-Methods: GET, POST
  Content-Type: application/json; charset=utf-8
  Content-Length: 642
}
[{"name":"Aarhus","local_names":{"bg":"?????","ko":"????","lv":"Orhusa","ka":"??????","zh":"???","fa":"?????","ja":"????
","ru":"?????","da":"Aarhus","os":"?????","mn":"?????","mk":"?????","is":"Árósar","he":"??????","el":"??????","sr":"????
?","be":"?????","uk":"?????","hu":"Aarhus","mr":"??????","de":"Aarhus","la":"Aarhusium","eo":"Arhuzo","hy":"??????","ar"
:"?????","sv":"Århus","th":"???????","lt":"Orhusas","et":"Århus"},"lat":56.1496278,"lon":10.2134046,"country":"DK","stat
e":"Central Denmark Region"}]
```

```
            var contentResponse = content.ReadAsStringAsync().Result;
            Console.WriteLine($"{contentResponse}\n");
        }
}
```



Your turn

Solve the exercises

References and image sources

Images:

Computer keyboard: http://stockmedia.cc/computing_technology/slides/DSD_8790.jpg

Bonus: http://wjreviews.com/reviews-cta/bonus.png

Cellphone: https://vectorified.com/download-image#iphone-icon-vector-2.png

Laptop: https://clipartix.com/laptop-clipart-image-19319/

AARHUS UNIVERSITY