# APIs

**Overview**

In this lesson, students will learn to access external sources of data using HTTP requests.

**Duration**
120 minutes

**Learning Objectives**

In this lesson, students will:

- Make HTTP requests for data to external API sources.
- Evaluate API documentation to determine data contents, accessibility, and request formatting.

# Pre-Class Materials and Preparation

**For remote classrooms**: Virtual breakout rooms and Slack may be needed to facilitate the partner exercise and discussions. As you plan for your lesson:

- Consider how you'll create pairs for the partner exercise (randomly, or with pre-assigned partners).
- Determine how (if at all) exercise timing may need to be adjusted.
- For helpful tips, keep an eye out for the **For remote classrooms** tag in the speaker notes.
- Prepare screenshots and answers to exercises in advance so that they can be easily shared in Slack during your lecture.

# Suggested Agenda

| Time | Activity |
| --- | --- |
| 0:00–0:15 | **Introduction + What Are APIs?** |
| 0:15–0:50 | **Requesting Data** |
| 0:50–1:00 | **Break** |
| 1:00–1:50 | **Evaluating APIs** |
| 1:50–2:00 | **Wrapping Up, Q&A, and Exit Ticket Completion** |

# Jupyter Notebook

The exercises referenced in this lesson can be found in the Python Workbooks + Data folder.

# APIs

# Today's Learning Objectives

In this lesson, you will:

- Make HTTP requests for data to external API sources.
- Evaluate API documentation to determine data contents, accessibility, and request formatting.

Guided Walk-Through:

# OpenWeatherMap API Key

5 minutes

Before starting, we'll need an API key to use a weather API for our first coding challenge, so let's walk through the process together as described in their documentation.

Don't worry: we'll learn what a key is, and why it's important, later on!

It may take their system a little time to respond with your unique API key, but they should be in place by the time we get to our code challenge.

APIs

# What Are APIs?

**Discussion:**

# External Data

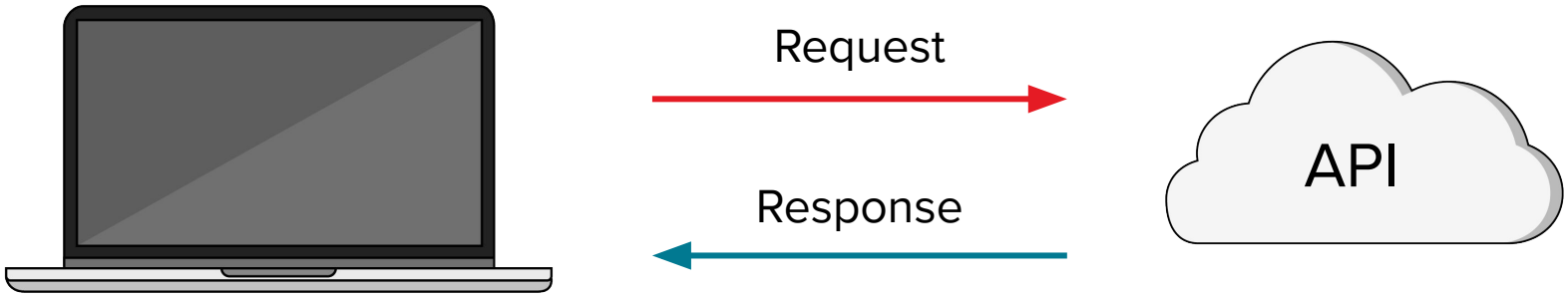
Have you ever needed information that wasn't ready to go in a file? Things like reference information, up-to-the-minute updated data, or data from an external vendor or another team within your organization?

**How did you go about sourcing that information, and what problems did you encounter?**

# Welcome to the Neighborhood!

APIs, or **application programming interfaces**, are data sources that can give you access to information from all sorts of places on the internet. Simply put, an API is a system that receives **requests** and returns **responses**.



# It’s All Interfaces?

We are used to numerous **interfaces** in real life. For example:

- **A door handle.** We don’t care how a door works, all that we expect is that pushing or pulling the handle will open the door.
- **File → Print.** We don’t care how the computer prints the document, all that we expect is that using the “Print” option in the “File” menu will result in the document being printed.

# What Kind of Data Is Served Up With an API?

It'd be difficult to find data that **doesn't** have an API. Common uses are:

- **Social media integration**: When you post something on Instagram, you can automatically have it post to Facebook as well.
- **Payment APIs**: When you want to send or receive money via Venmo or the Cash App, the app directly interfaces with your bank or financial institution.
- **Maps**: Many apps like Uber, Lyft, and Waze are built on top of existing map services like Google Maps or Mapbox.

An extensive list of public APIs can be found here:
https://github.com/public-apis/public-apis

APIs

# Requesting Data

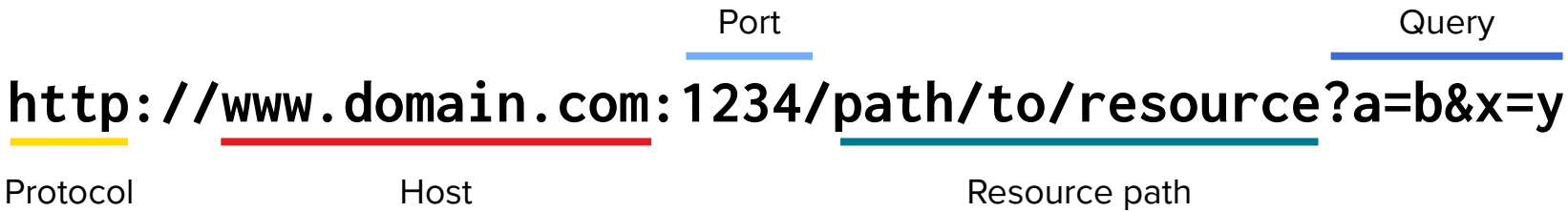
# APIs Sound Great! How Do I Get Started?

API requests use HTTP, so the way to request information from an API may look somewhat familiar to you!

```
http://www.domain.com:1234/path/to/resource?a=b&x=y
```

It's a URL address, like what you type when you want to go to any website. But there's more here than meets the eye. Let's talk about each part of this address individually.

# APIs Sound Great! How Do I Get Started?



# Request Methods

## GET

- Retrieve information.
- Used by your browser's address bar every time you visit a website.

## POST

- Send information.
- Used by most forms to submit data, such as logging in or creating a new post.

## PUT

- Update information.
- Requests to change information in a database.

## DELETE

- Delete information.

# Requesting Data With `requests`

Once you have identified an address and a method, you can send a request using Python's **`requests`** library.

```
import requests

data = requests.get(address)
```

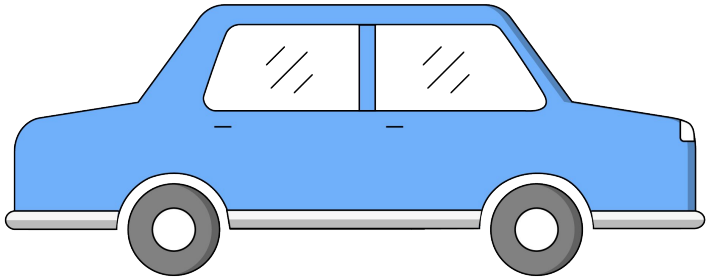
**What other methods might there be?**

# JSON

API endpoints will almost always return **JSON-formatted data**.

```
{
  “data”: “car”,
  “make”: “ford”,
  “model”: “focus”
  “details”: {
    “color” : “blue”,
    “mileage” : “54019”
  }
}
```



**Does this look like a Python data structure?**

# Dealing With JSON

It's **JSON** (JavaScript Object Notation).

However, the **`requests`** library has a built-in JSON decoder to turn JSON into a Python dictionary.

```
data = request.get(address)

parsed_data = data.json()
```

```
{
  “data”: “car”,
  “make”: “ford”,
  “model”: “focus”
  “details”: {
    “color” : “blue”,
    “mileage” : “54019”
  }
}
```

# Many-Layered Objects

API response objects are infamously complex in their formats. To dig down to the layer of information you actually want, carefully examine one layer at a time until you reach the data you're looking for.

```
response = {
  “data”: “car”,
  “make”: “ford”,
  “model”: “focus”
  “details”: {
    “color” : “blue”,
    “mileage” : “54019”
  }
};
```

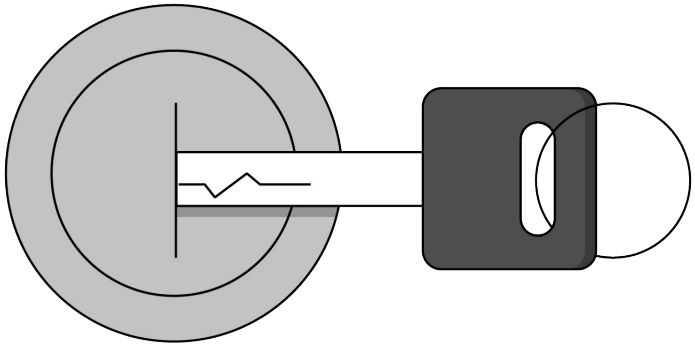
```
print(response)
# gives us the whole dictionary

print(response["details"])
# gives us the smaller dictionary

print(response["details"]["color"])
# finally, gives us "blue"
```

# API Keys

Many APIs require you to include a **key** with your request. You can think of this like a password that prevents someone from misusing the API.

Keys are also used to track your requests for rate limits and potential charges. Don't worry, most APIs have a free tier! You can register with the API's website to receive your unique key.



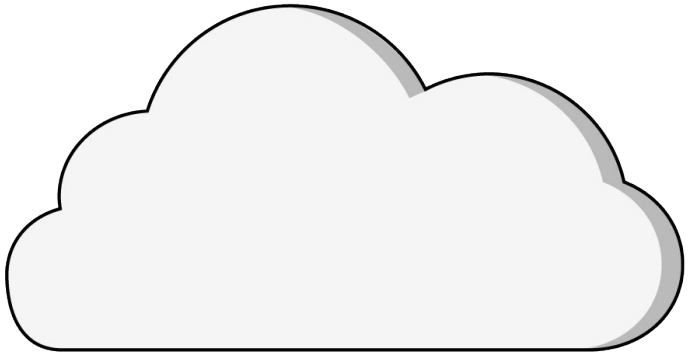
 

**Partner Exercise:**

# Weather API Application

**20 minutes**

Complete the functionality for a weather application in Section 7.1 of the workbook.



APIs

# Evaluating APIs

# Read the Docs

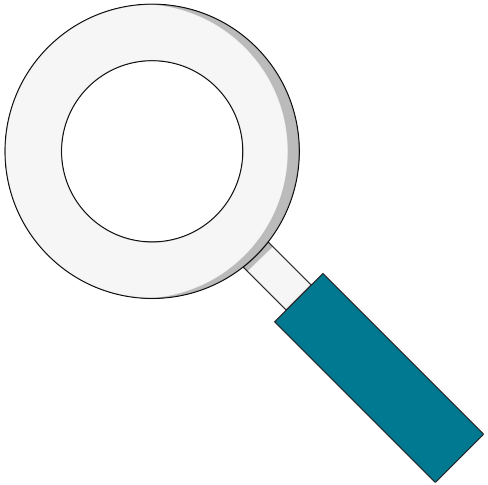
Every API is unique — you will have to explore its documentation to figure out what data you want to access and what URL to use in order to send requests. However, there are a few standard details to check for in any API:

1. **Do I need an API key?**
    a. If so, are there rate limits or charges for requests?
    b. How do I include the key in my requests?
2. **What are the available endpoints for data?**
    a. What URL format can search for specific data?
3. **What will the response data look like?**
    a. Typically, the response will be JSON.

## Group Exercise:
# API Scavenger Hunt

30 minutes

APIs open up a whole world of new information to our Python programs. Let's see just how much information you can find in Section 7.2 of the workbook.



## Guided Walk-Through:
# Let's Go API Shopping

15 minutes 

Using APIs listed in the [public-apis project](), let's see what to expect from documentation sources.

APIs

# Wrapping Up

# Recap

**In today's class, we...**

- Made HTTP requests for data to external API sources.
- Evaluated API documentation to determine data contents, accessibility, and request formatting.



# Looking Ahead

**On your own:**

- Work through the Python progress assessment on myGA (due at the end of the unit).
- Share your capstone project ideas with your instructor for review.
- Join someone else's project or invite others to join yours!

**Next Class:**

Exploratory Data Analysis With Pandas

# Don’t Forget: Exit Tickets!


GA