DS3500
More on APIs

Rush Sanghrajka

Today's Goals

Secure API authentication

Handling errors gracefully

Work with rate limits

Fetch large datasets via pagination

Build an API pipeline

Authentication

OAuth 2.0

API key in parameters ⊗

API key in headers ✓

Bearer Tokens

Avoiding API key exposure

.env
↓
add the key in .env
↓
load the variable using
dotenv

Error Handling

```
if response.status_code == 200:
    return response.json()
elif                     == 404:
    # access alternate/simpler endpoints
```

Check for status codes!

Use try/except blocks (handle specific error types)

Log your errors for debugging

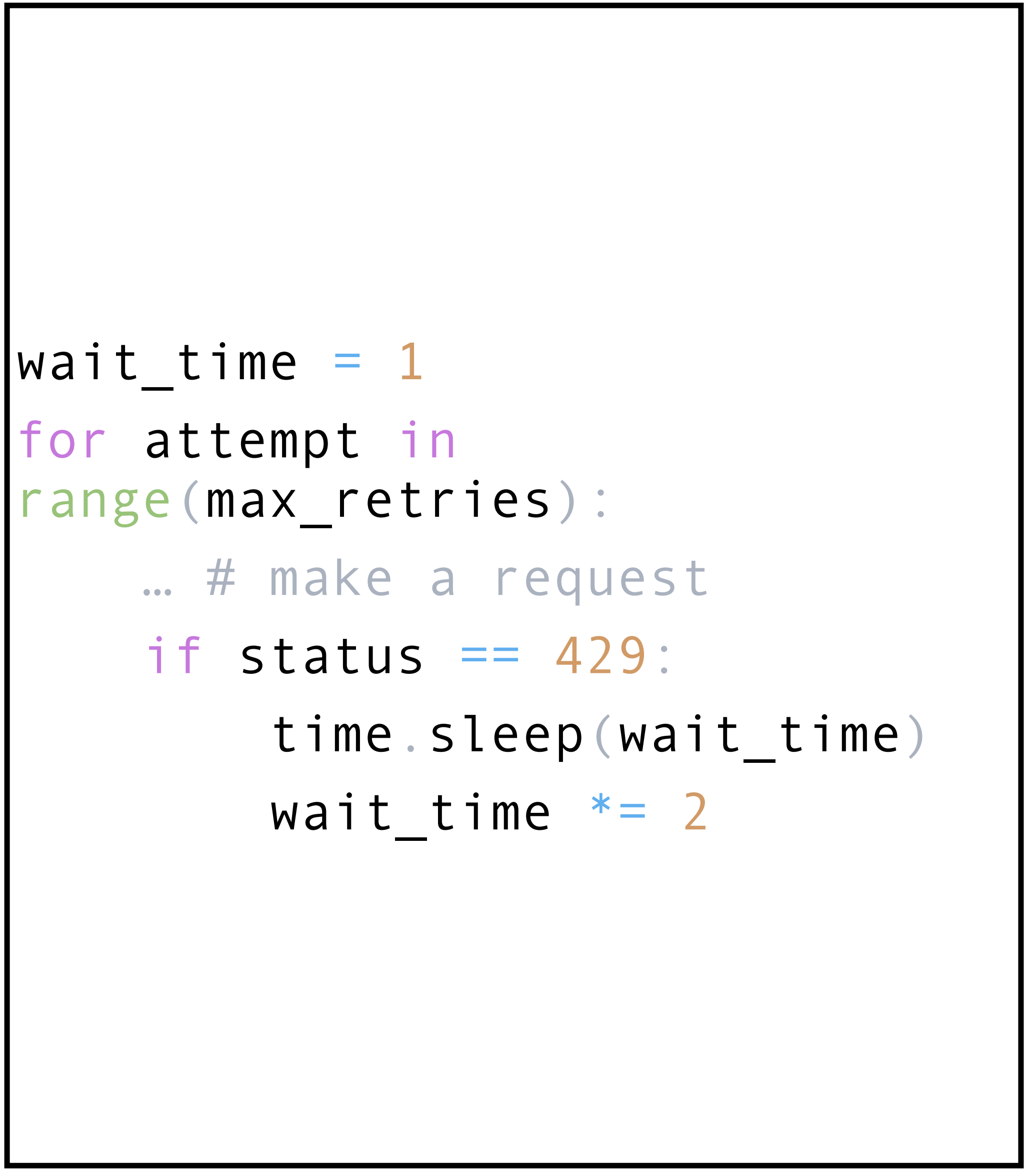
429

```
try:
    response = requests.get(url, timeout=10)

except requests.exceptions.Timeout:
    # Server too slow

except requests.exceptions.ConnectionError:
    # Network problem

except Exception as e:
    # Catch-all
```

Handling Rate Limits

Be polite and add wait time between requests

Exponentially back off



```
wait_time = 1
for attempt in range(max_retries):
    … # make a request
    if status == 429:
        time.sleep(wait_time)
        wait_time *= 2
```

Pagination

Say you want 10,000 records

API returns maximum 100 per request

Need to make 100 requests!

Make multiple requests and combine them together

Check metadata if there are more results

```
?page=1
```

```
?offset=100
```

Stitch it all together

Create a function that calls other functions- each function does one step in your pipeline. Use that function as your API pipeline.