Microsoft

Reactor

Microsoft Reactor
Explore the Art World Using RESTFUL APIs

Event Code: #13081

FAQ's

What are the Reactors?

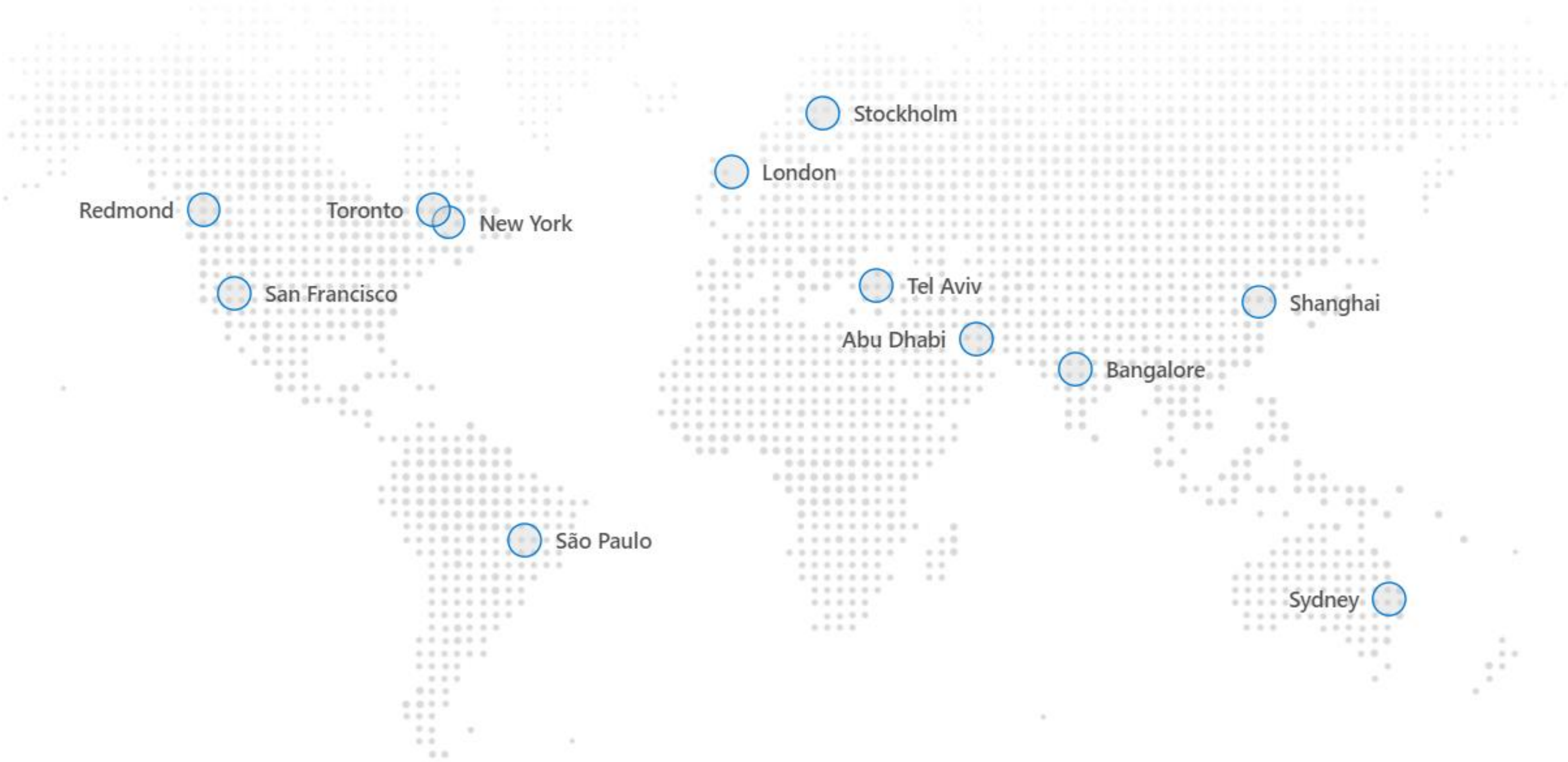
Reactors are community spaces where technology professionals meet, learn, and connect—to both their local peers as well as industry-leading ideas and technology from Microsoft, partners, and the open source community.

With a diverse mix of hands-on workshops, expert panels, and community events, there's something for everyone—whether you're just getting started or working on complex projects.

Reactor programming is always free and inclusive of a broad set of products, tools, and technologies. You will find:

- First party content designed to help developers learn new skills in high-demand fields such as Machine Learning, AI, and Data Science
- New content around breakthrough technologies and concepts such as quantum computing and blockchain
- Resources, talks and classes from Microsoft for Startups to deliver what startups need to succeed
- Cloud Advocates who deep-dive into their technical specialties and the Industry Experiences team to provide tailored content around applications for Manufacturing, Healthcare, Retail, and other verticals
- Reactors always welcome community groups, Meetups, partners and MVPs to use our space

Map


Stockholm
London
Redmond
Toronto
New York
San Francisco
Tel Aviv
Abu Dhabi
Shanghai
Bangalore
São Paulo
Sydney

Our Code of Conduct

Microsoft is dedicated to empowering every person and every organization on the planet to achieve more.

This includes Microsoft Reactor events where we seek to provide a respectful, friendly, professional experience for everyone, regardless of gender, sexual orientation, physical appearance, disability, age, race or religion.

We do not tolerate any behaviour that is harassing or degrading to any individual, in any form. Individuals are responsible for knowing and abiding by these standards. We encourage everyone to assist in creating a welcoming and safe environment.

- Be aware of others
- Be friendly and patient
- Be welcoming and respectful
- Be open to all questions and viewpoints
- Be understanding of differences
- Be kind and considerate to others

Our Code of Conduct

Microsoft is dedicated to empowering every person and every organization on the planet to achieve more.

This includes Microsoft Reactor events where we seek to provide a respectful, friendly, professional experience for everyone, regardless of gender, sexual orientation, physical appearance, disability, age, race or religion.

We do not tolerate any behaviour that is harassing or degrading to any individual, in any form. Individuals are responsible for knowing and abiding by these standards. We encourage everyone to assist in creating a welcoming and safe environment.

Our Ask of You

Be aware of others

Be friendly and patient

Be welcoming and respectful

Be open to all questions and viewpoints

Be understanding of differences

Be kind and considerate to others

Explore the Art World Using RESTFUL APIs

Allen Sanders

Principal Cloud Architect

Speaker Bio: Allen Sanders is a Principal Cloud Architect with over 25 years of experience in software engineering, architecture and design delivering technology strategy and business solutions in multiple verticals. As a Microsoft Certified Professional, he has taught, led and mentored teams of varying sizes in multiple application transformation efforts. He has a passion for learning – both his own and for others.

Agenda

1 Introduction

2 What is an API?

3 Build a Local API

4 Query the Metropolitan API

5 Authentication Strategies

6 Query the Cooper Hewitt API

7 Handle Responses

8 Use Libraries

9 Knowledge Check

10 Summary

Introduction

https://aka.ms/RESTfulAPIArt-8

What are we covering?

- RESTful APIs
- Building a simple API
- Strategies and approaches to connecting to APIs

Prerequisites

- Internet-connected browser
- Node.js, installed locally
- npm, installed locally

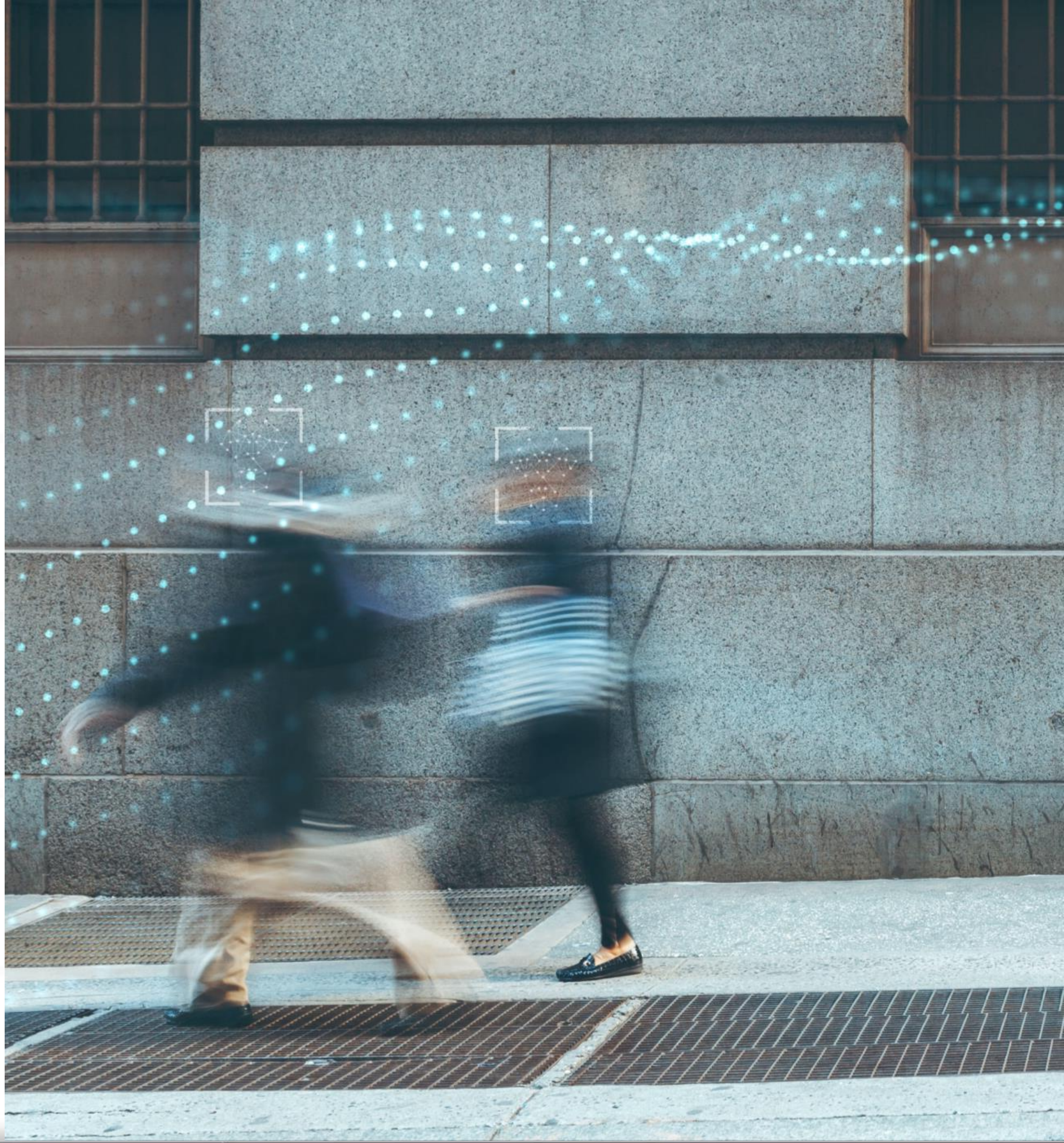
What is an API?

What is an API?

- API stands for Application Programming Interface
- Provides reusable functionality through a well-defined interface
- Only have to know about interface, not inner workings

Types of APIs

- Includes
  - SOAP – Simple Object Access Protocol + XML
  - REST – Representational State Transfer + JSON
  - RPC – Remote Procedure Call
  - In process or embedded libraries



Build a Local API

Build a Local API

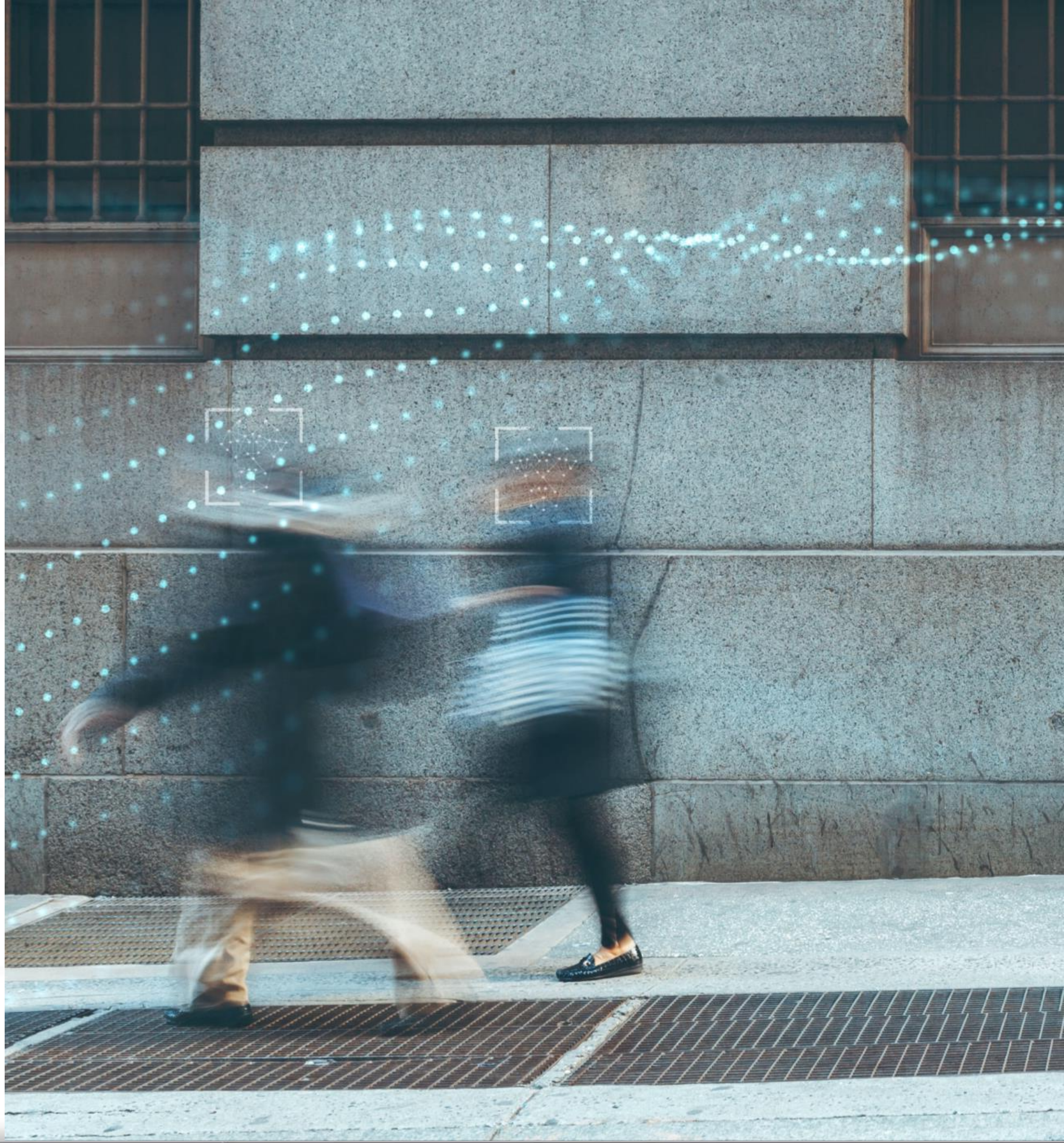
- Uses a package called JSON Server via npm
- Can expose a local JSON file over REST for simulation
- Install using `npm install -g json-server`

Build a Local API

- Start using `json-server --watch <filename>.json`
- API accessible using http://localhost:3000/objects (for list)
- Use http://localhost:3000/objects/<id> (for specific)

Anatomy of a REST API Call

- Includes 4 major parts:
  - ➢ Endpoint – Route that provides access to data; URL
  - ➢ Method – HTTP-based action to be taken against the endpoint
  - ➢ Headers – Additional info about the request; e.g., authentication or authorization, format, etc.
  - ➢ Data – Information sent with the request to define what’s requested



HTTP Methods

- Common ones include:
  - GET
  - POST
  - PUT
  - PATCH
  - DELETE

Query the Metropolitan API

Authentication Strategies

AuthN vs. AuthZ

- Authentication = how do I know you are who you say you are
- Authorization = once I know who you are, how do I determine what you are allowed todo

Methodologies

- Can include 1 of multiple strategies – e.g., Basic vs. Bearer
- Can include use of an API key
- Might include a combination

OAuth 2.0

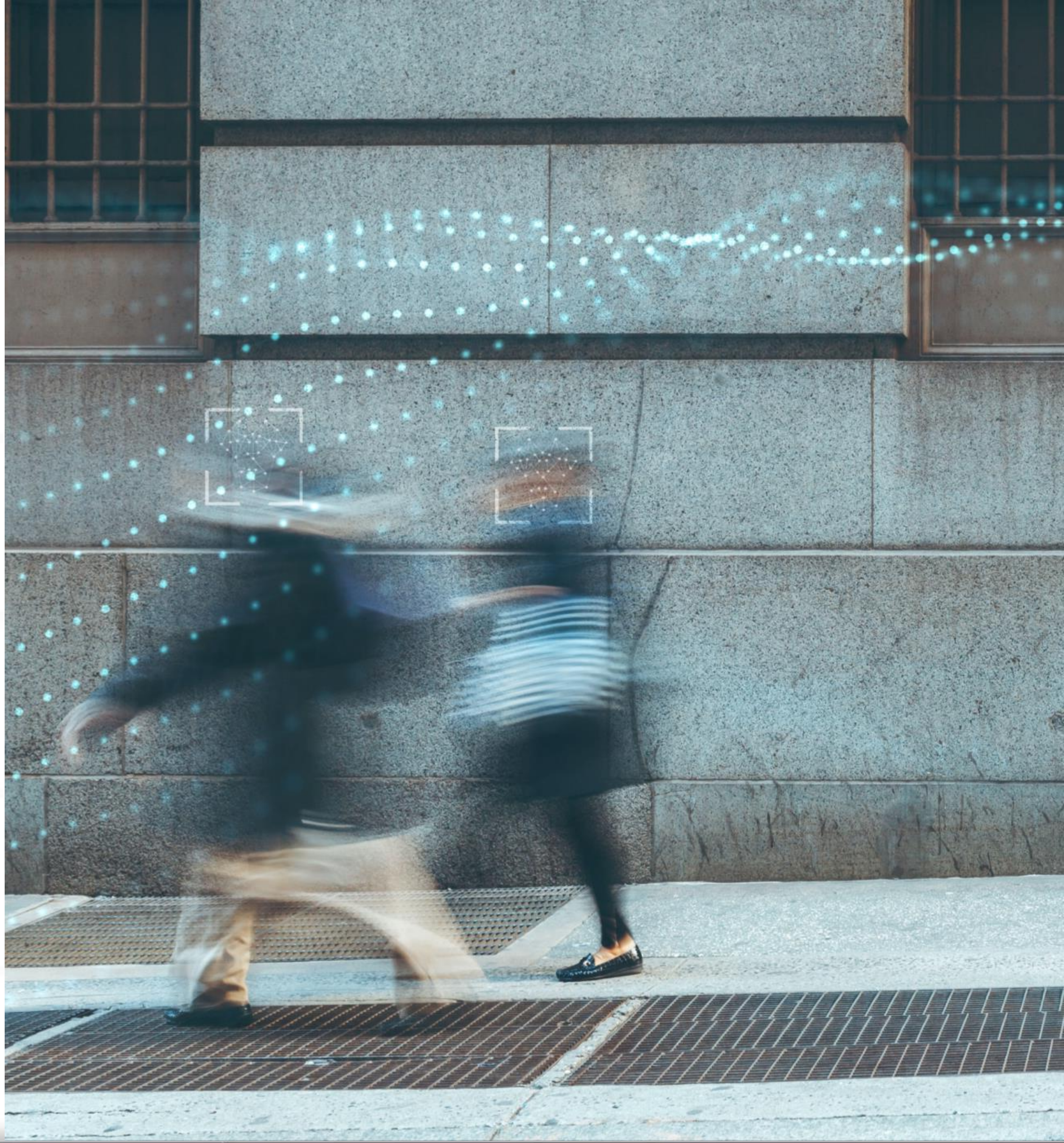
- Provides a more fine-grained AuthN strategy
- Supports multiple flows (e.g., username/password, Facebook, etc.)
- Results in an access token (JWT – JSON Web Token, pronounced “jot”)
- Can include additional detail within the token (e.g., expiration)

Query the Cooper Hewitt API

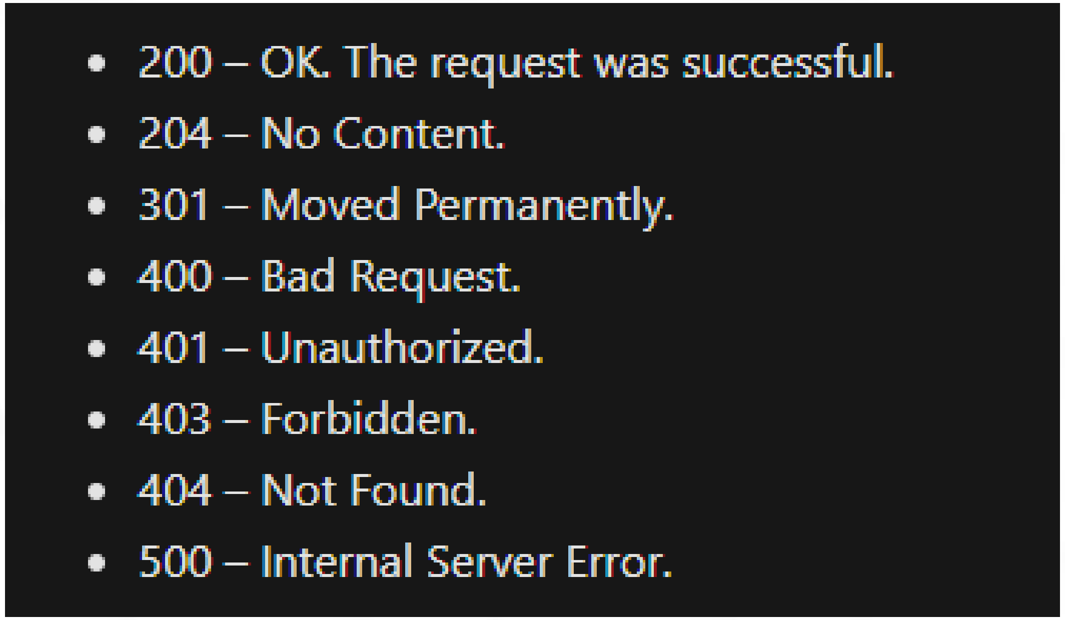
Handle Responses

Request vs. Response

- Request detail can include detail on URL or in body of request
- Response might include status code + data
- Response might only include a status code (based on request type)



Common HTTP Status Codes



- 200 – OK. The request was successful.
- 204 – No Content.
- 301 – Moved Permanently.
- 400 – Bad Request.
- 401 – Unauthorized.
- 403 – Forbidden.
- 404 – Not Found.
- 500 – Internal Server Error.

Use Libraries

Multiple Options

- Send request through browser (if GET)
- Submit request via tool like POSTMan or Insomnia
- Programmatically using a library

Multiple Libraries

- Axios with JavaScript
- `Invoke-RestMethod` with PowerShell
- Requests with Python
- `HttpClient` with C#

Knowledge Check

Summary

What Did We Cover?

- What APIs are
- Used REST to interact with multiple types of APIs
- Talked about specifics of a REST request
- Using libraries to interact with APIs through code

Next Steps / Further Study

- https://aka.ms/RESTfulAPIArt-8
- https://sec.ch9.ms/ch9/5a10/f5eeb830-8cb5-49af-bbad-b3db8d785a10/StartDevChangeHolland_high.mp4
- https://jwt.io
- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Microsoft

Reactor

Thank You!

Q&A

https://www.meetup.com/pro/microsoft-reactor

@MSFTReactor

http://www.youtube.com/c/MicrosoftReactor

aka.ms/ReactorEmailSignUp

Microsoft

Reactor

We are constantly striving to create excellent content and would appreciate if you could take this brief survey.

Survey Link: https://aka.ms/Reactor/Survey

Please enter the event code 13081 at the start of survey

Microsoft