1:15 PM – 2:00 PM Astro Room

Mastering Salesforce Data at Scale: Bulk API, Python, Pandas

Jack McHugh

JULY 7-9, 2025

Buckeye Dreamin'

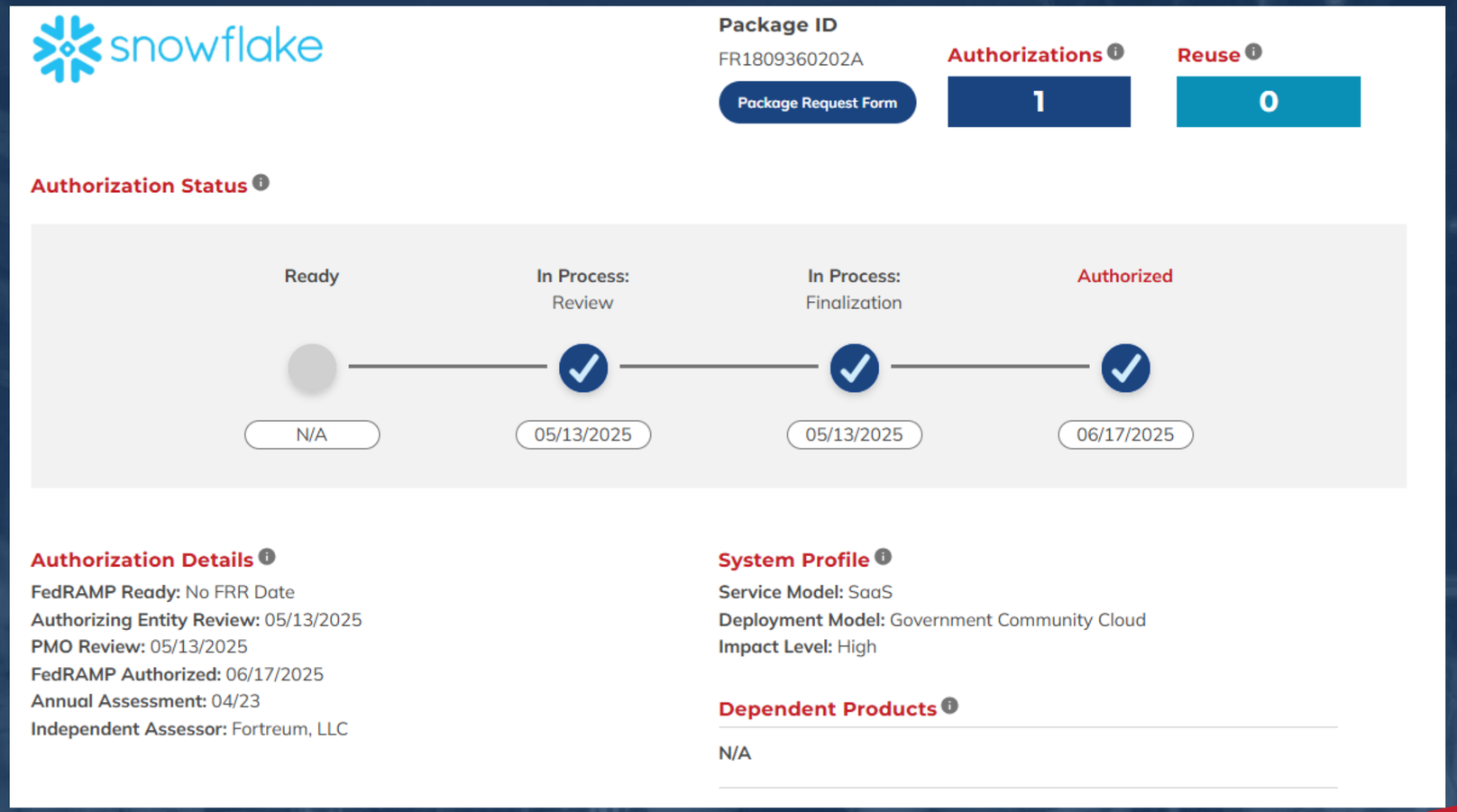
PRESENTER

Jack McHugh

Systems Analyst

THANK YOU to our Sponsors

Last Month News Recap: Snowflake Fedramp High



Last Month News Recap: Malicious Data Loader

Hackers abuse modified Salesforce app to steal data, extort companies, Google says

By **A.J. Vicens**

June 4, 2025 10:24 AM EDT · Updated June 4, 2025

The company logo for Salesforce.com is displayed on the Salesforce Tower in New York City, U.S., March 7, 2019. REUTERS/Brendan

AGENDA

01 What even is the Bulk API?

02 How it Actually Works

03 Real World Use Cases

04 Python & Pandas

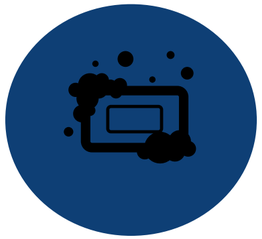
05 Landmines & Gotchas

06 Live Demo (Yay!)

Disclaimer: The Bulk API is not a toy. Used improperly, it can overwrite critical data, nuke production records, and summon angry admins from three time zones away. This talk includes live code, automation, and AI, all of which are powerful, risky, and probably not what your Salesforce admin had in mind. Think before you bulk. Test in a sandbox.

Internal

APIs at a Glance

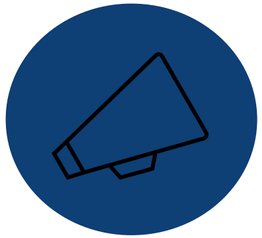


SOAP

REST

BULK

STREAMING

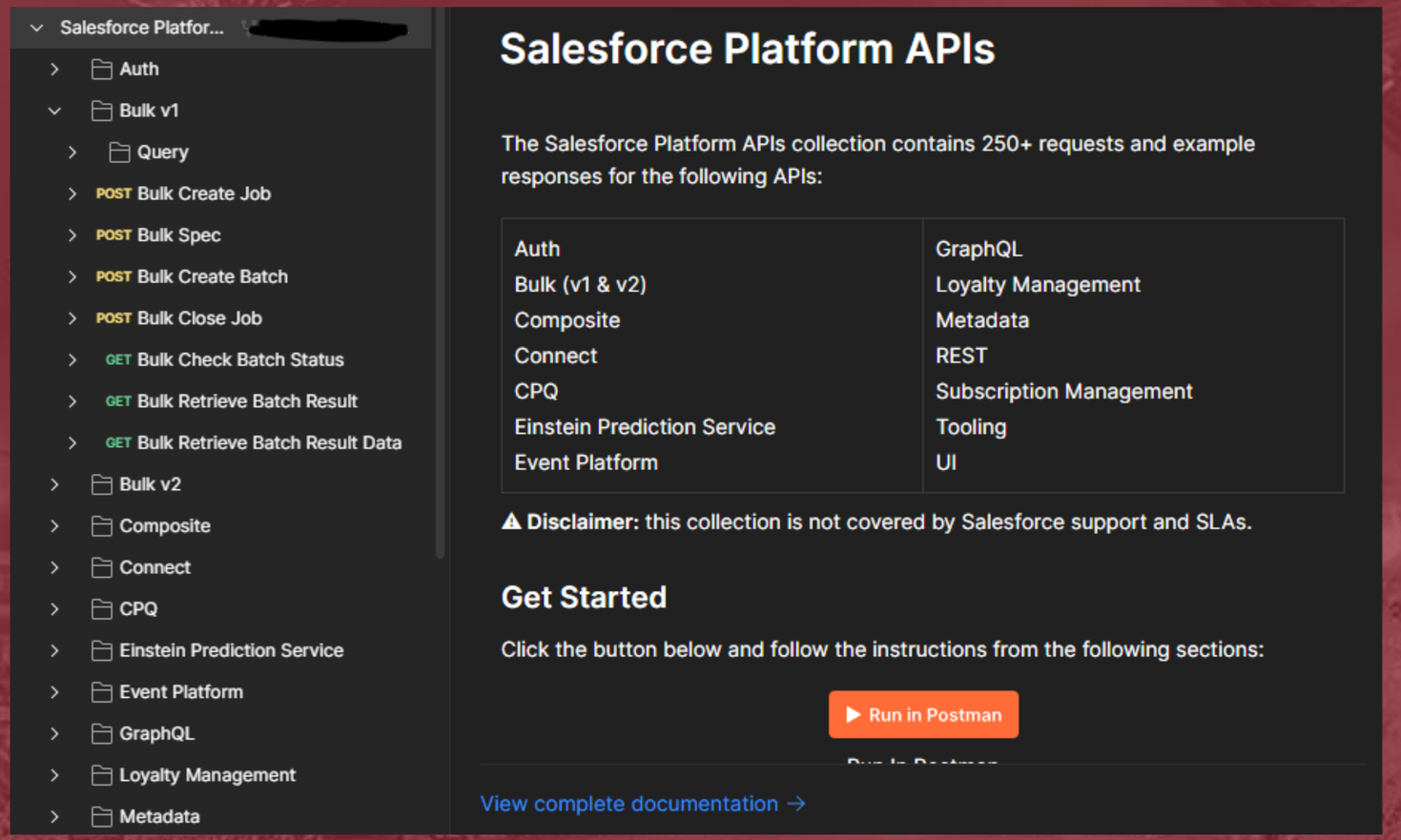


PUB/SUB

GRAPH QL

Buckeye Dreamin'

API Exploration
POSTMAN



Bulking Season: Salesforce Edition

01. Ideal Rows

If working with greater than 2000 rows, the Bulk API is a good candidate

02. When speed matters

Bulk API is built for high-throughput data loads. Perfect for nightly jobs, migrations, and ETL.

03. Async by Design

Jobs run in the background. No timeouts. No governor limits blowing up your script/notebook

04. Handles Massive Volume

Bulk API 2.0 supports 10k+ records per batch and auto-chunks uploads for you.

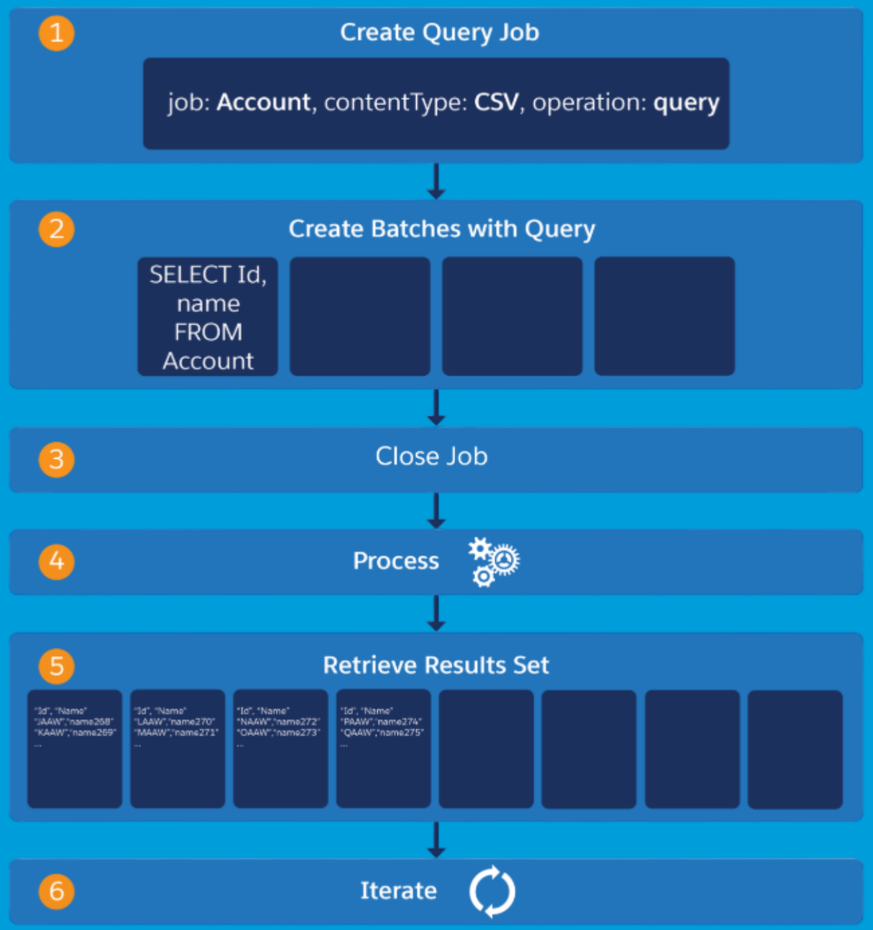
05. Stops the Retry Madness

Tired of scripts that break in the middle of a 5000-row import? Bulk API is built for failure and recovery.

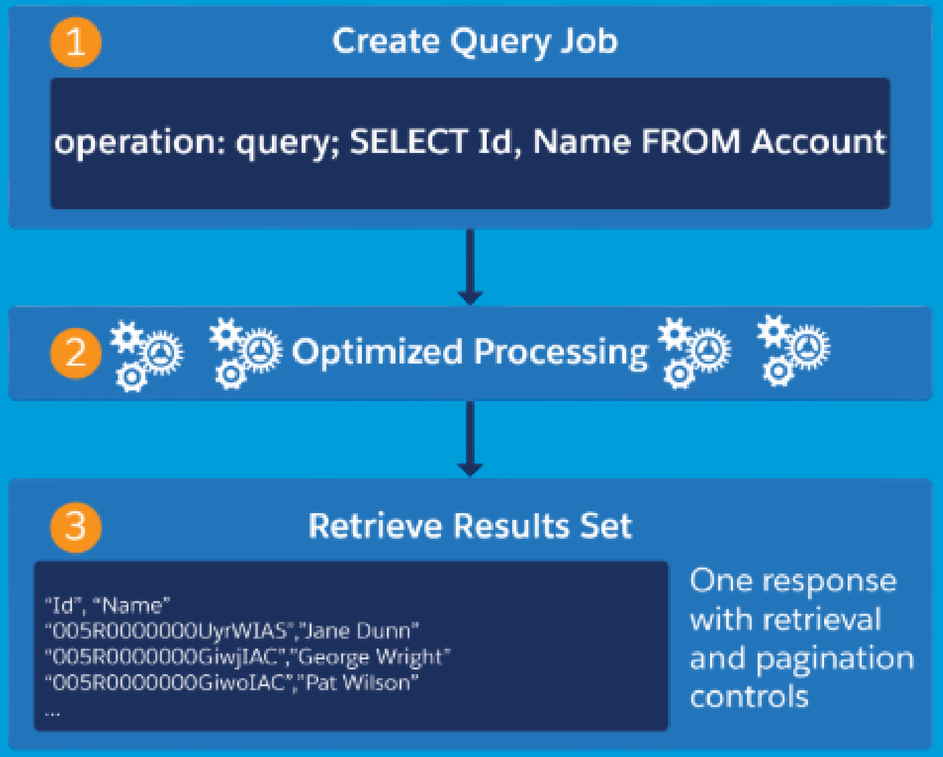
Internal

Choose Your Lane

Bulk API 1.0



Bulk API 2.0



INGEST

• Create Job

POST /services/data/v64.0/jobs/ingest
Content-Type: application/json

```
{
 "object": "Account",
 "operation": "insert",
 "lineEnding": "LF",
 "columnDelimiter": "COMMA",
 "contentType": "CSV"
}
```

1

• Upload Data

PUT
/services/data/v64.0/jobs/ingest/{jobId}
/batches
Content-Type: text/csv

Name,Phone
Buckeye Dreamin, 614-4321
Who Dey, 513-8585

2

• Close job

PATCH
/services/data/v64.0/jobs/ingest/{jobId}
Content-Type: application/json

```
{
 "state": "UploadComplete"
}
.
```

3

• Check Status

GET /services/data/v64.0/jobs/ingest/{jobId}

4

• Get Results

GET /services/data/v64.0/jobs/ingest/{jobId}/successfulResults
GET /services/data/v64.0/jobs/ingest/{jobId}/failedResults

5

QUERY

01. Create Job

POST /services/data/v60.0/jobs/query
Content-Type: application/json

```
{
  "operation": "query",
  "query": "SELECT Id, Name FROM Account",
  "contentType": "CSV"
}
```

02. Check Job Status

GET /services/data/v60.0/jobs/query/{jobId}

03. Get Results

GET /services/data/v60.0/jobs/query/{jobId}/results

Returns array of result ids

04. Download Results

GET /services/data/v60.0/jobs/query/{jobId}/results/{resultId}

Who is using this?
Use cases?

Large Enterprises

- Fortune 500s with millions of rows
- Regularly run mass updates, data deduplication, or backfills
- Sync Salesforce with ERP Systems

ETL / Data Integration

- Mulesoft
- Informatica
- Talend
- Most use Bulk or Bulk 2.0 Under the hood

Salesforce Admins

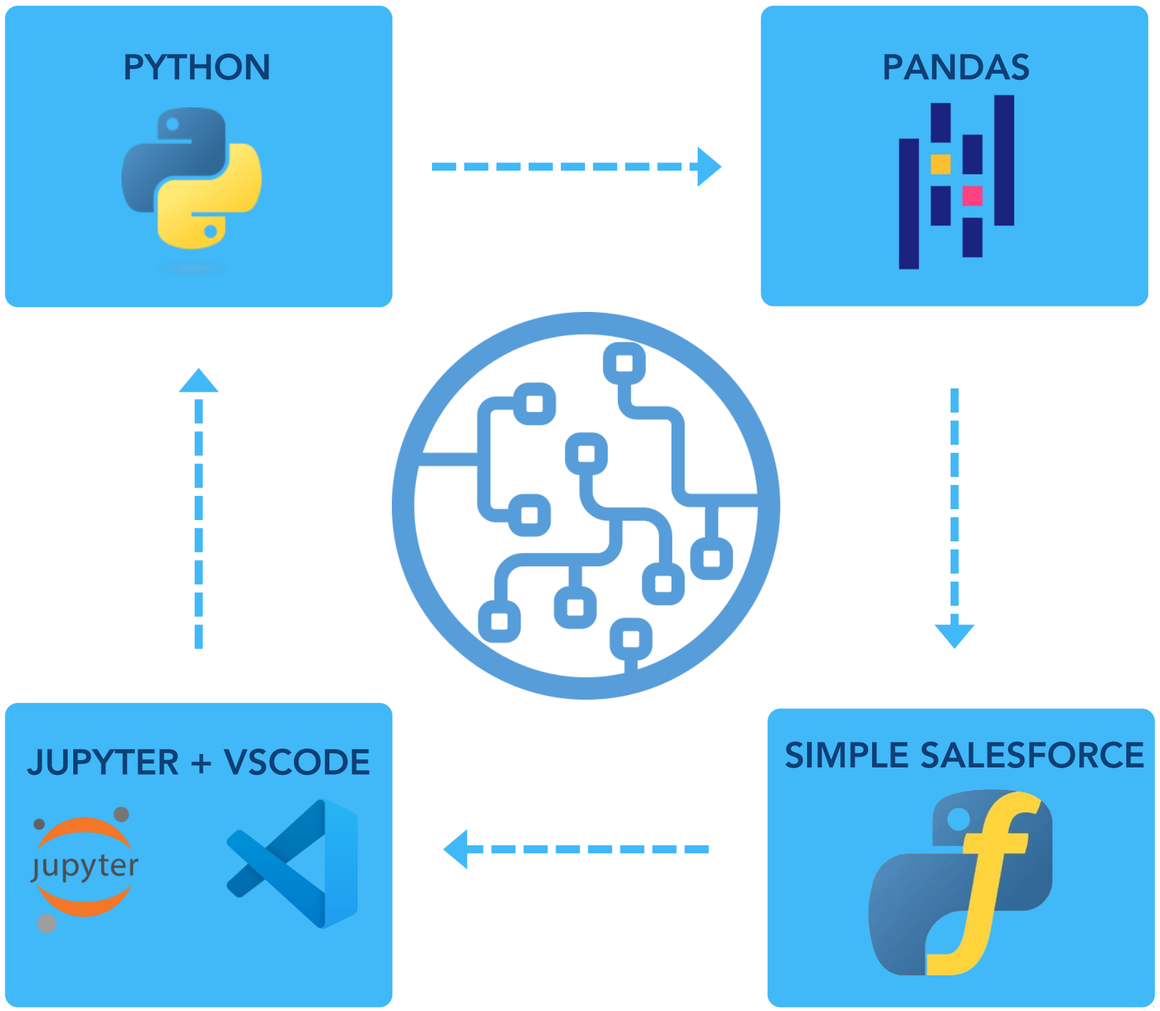
- Data Loader
- Workbench
- dataloader.io
- OwnBackup

Data Pipelines with Salesforce at the Core

- Insert, update, upsert, and hard delete millions of records without breaking a sweat
- Query millions of records and process them in memory using high-speed libraries
- Enrich Salesforce data using AI
- Feed records into machine learning models and write predictions back
- Automate large-scale data cleanup across fields, objects, and systems
- Build repeatable, testable, version-controlled data workflows
- Access and transform hard-to-reach sources like spreadsheets, APIs, or flat files
- Bring in legacy system data using clean, scriptable conversion pipelines
- Create data flows that are readable, documentable, and auditable
- Do whatever you can think of, *as long as you can write it in code*

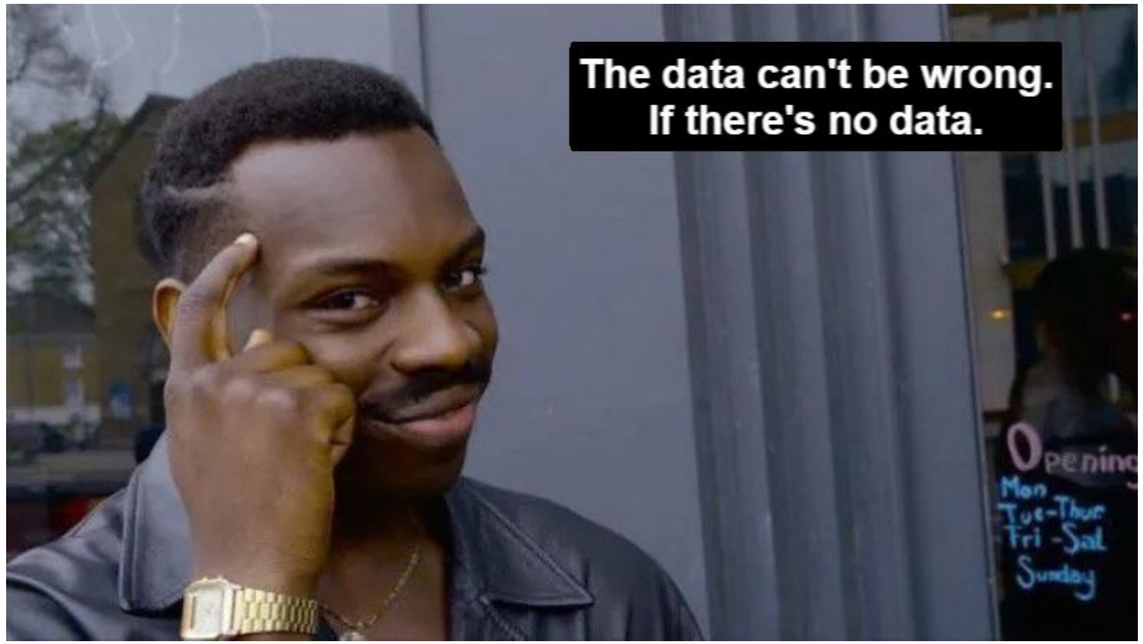
The Stack

This stack gives you programmatic control over Salesforce data using tools trusted across data engineering and AI development. Python is widely used for building data pipelines and automation. Pandas allows fast, in-memory processing for large datasets. Jupyter and VS Code provide a clear, testable environment for working with complex data. Simple Salesforce connects your logic directly to the Bulk API, making it possible to insert, update, upsert, and clean records efficiently. This setup supports advanced use cases like model scoring, data enrichment, and large-scale record transformation while staying aligned with the Salesforce platform.



Questions to Avoid Pitfalls

- Should I be inserting this many records, salesforce is not a data lake?
- Have I tested this fully in sandbox?
- Have I considered risks of running this in production?
- What is my Data Governance, will they let me do this?
- Do I have a backup plan should something go wrong?
- Am I taking business logic out of salesforce where it belongs?
- Is there a better solution?
- Architect hat: Is this solution scalable, maintainable, secure, compliant, and production grade (orchestrated)?

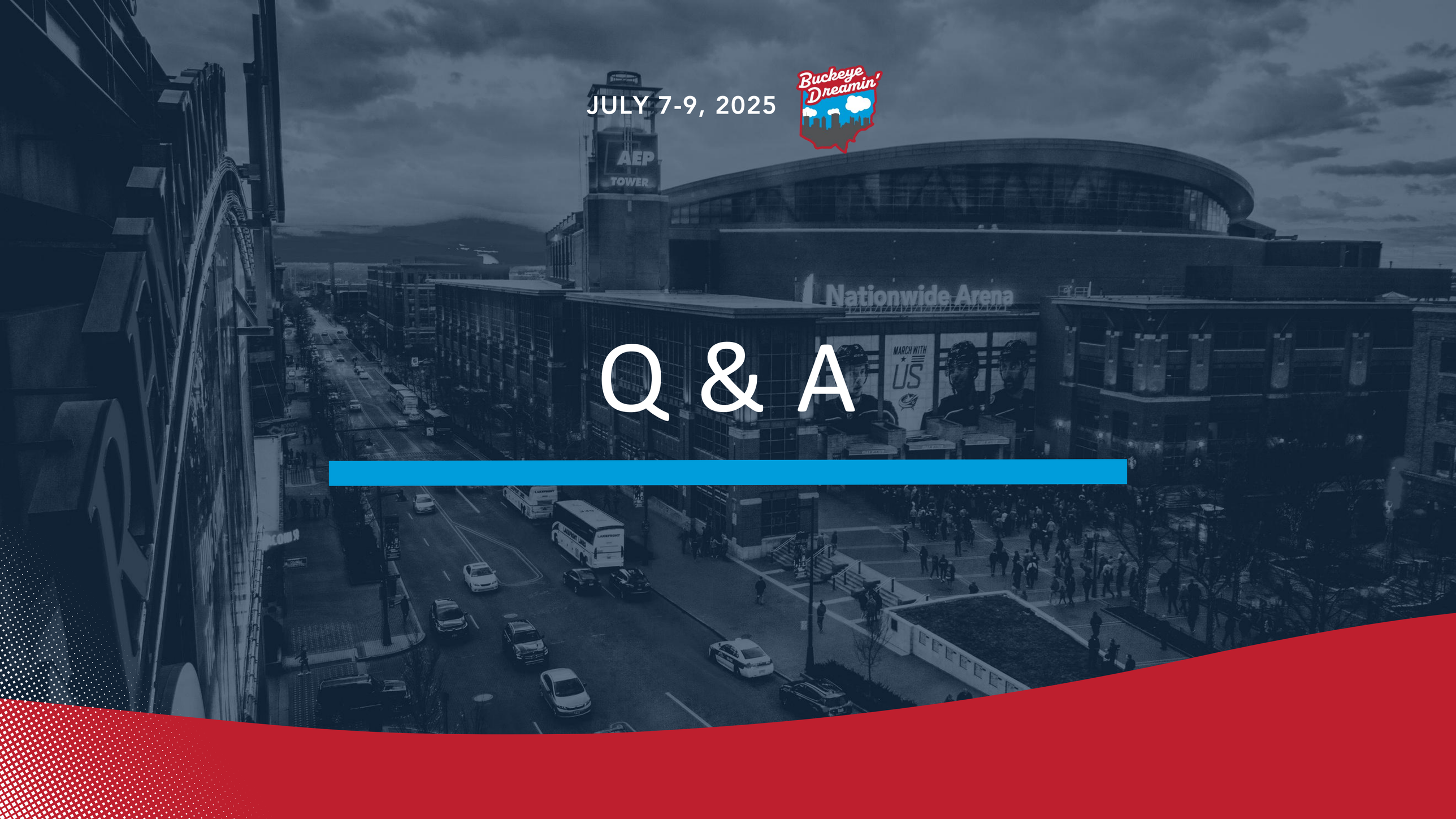


Internal


Own
Buckeye Dreamin'

JULY 7-9, 2025

Live Demo!



JULY 7-9, 2025

Buckeye Dreamin'

Q & A

CONTACT

Thank you!

LinkedIn

Github Repo

JULY 7-9, 2025