Apify Data Download Guide

Overview

This guide explains how to connect to Apify and download previous towing data runs.

Prerequisites

1. **Apify Account**: Sign up at https://apify.com
2. **API Token**: Get your token from https://console.apify.com/account/integrations
3. **Environment Setup**: Add `APIFY_TOKEN` to your `.env` file

Methods

1. Using the Python Script (Recommended)

The easiest way to download previous runs:

```
# List all previous towing runs
python scripts/download_apify_runs.py --list-only

# Download data from all previous runs (up to 10 runs)
python scripts/download_apify_runs.py --limit-runs 10

# Download data from a specific run
python scripts/download_apify_runs.py --run-id YOUR_RUN_ID

# Download with custom limits
python scripts/download_apify_runs.py --limit-runs 5 --limit-items 100 --
output my_data.json
```

Options:

- `--limit-runs N`: Maximum number of runs to process (default: 10)
- `--limit-items N`: Maximum items per run (default: all)
- `--output FILE`: Output file path (default: apify_towing_data.json)
- `--list-only`: Only list runs, don't download data
- `--run-id ID`: Download specific run by ID

2. Using the API Endpoints

List All Runs

```
curl -X GET "http://localhost:8000/api/v1/apify/runs?
limit=100&status=SUCCEEDED" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

List Towing Runs Only

```
curl -X GET "http://localhost:8000/api/v1/apify/runs/towing?limit=50" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Get Run Details

```
curl -X GET "http://localhost:8000/api/v1/apify/runs/{run_id}" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Download Run Data

```
curl -X GET "http://localhost:8000/api/v1/apify/runs/{run_id}/data?
limit=1000" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Download All Towing Data

```
curl -X POST "http://localhost:8000/api/v1/apify/runs/download-all?
limit_runs=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

API Endpoints

GET `/api/v1/apify/runs`

List all Apify actor runs

Query Parameters:

- `actor_id`: Apify actor ID (default: `apify/google-maps-scraper`)
- `limit`: Maximum runs to return (1-1000, default: 100)
- `offset`: Number of runs to skip (default: 0)
- `status`: Filter by status (`SUCCEEDED`, `FAILED`, `RUNNING`, etc.)

Response:

```
{
  "data": {
    "items": [
      {
        "id": "run_id",
        "status": "SUCCEEDED",
        "startedAt": "2024-01-01T00:00:00Z",
        "finishedAt": "2024-01-01T01:00:00Z",
        "input": {...},
        "stats": {...}
      }
    ],
    "total": 50
  }
}
```

GET `/api/v1/apify/runs/towing`

List all towing-related runs (filtered)

Query Parameters:

- `limit`: Maximum runs to return (default: 100)
- `status`: Filter by status (default: `SUCCEEDED`)

Response:

```
[
  {
    "run_id": "abc123",
    "status": "SUCCEEDED",
    "started_at": "2024-01-01T00:00:00Z",
    "finished_at": "2024-01-01T01:00:00Z",
    "search_query": "towing company Utah",
    "max_results": 100,
    "stats": {...}
  }
]
```

GET `/api/v1/apify/runs/{run_id}`

Get details of a specific run

Response:

```
{
  "data": {
    "id": "run_id",
    "status": "SUCCEEDED",
    "startedAt": "2024-01-01T00:00:00Z",
    "finishedAt": "2024-01-01T01:00:00Z",
    "defaultDatasetId": "dataset_id",
    "input": {...},
    "stats": {...}
  }
}
```

GET `/api/v1/apify/runs/{run_id}/data`

Download data from a completed run

Query Parameters:

- `limit`: Maximum items to return (optional)
- `offset`: Number of items to skip (default: 0)

Response:

```
{
  "run_id": "abc123",
  "companies_count": 45,
  "companies": [
    {
      "name": "ABC Towing",
      "phone_primary": "555-0100",
      "website": "https://...",
      "google_business_url": "https://...",
      "address_street": "123 Main St",
      "address_city": "Salt Lake City",
      "address_state": "UT",
      "address_zip": "84101",
      "rating": 4.5,
      "review_count": 120,
      "hours": {...},
      "source": "apify_google_maps"
```

```
    }
  ]
}
```

POST `/api/v1/apify/runs/download-all`

Download data from all previous towing runs

Query Parameters:

- `limit_runs`: Maximum runs to process (1-100, default: 10)
- `limit_items_per_run`: Maximum items per run (optional)

Response:

```
{
  "total_runs": 5,
  "total_companies": 250,
  "runs": [
    {
      "run_id": "abc123",
      "status": "SUCCEEDED",
      "companies_count": 50,
      "downloaded": true
    }
  ],
  "companies": [...]
}
```

Data Format

Downloaded company data follows this schema:

```
{
  name: string;                    // Company name
  phone_primary: string;           // Primary phone number
  website?: string;                // Company website
  google_business_url: string;     // Google Maps URL
  address_street: string;          // Street address
  address_city: string;            // City
  address_state: string;           // State code
  address_zip: string;             // ZIP code
  rating?: number;                 // Google rating (0-5)
  review_count?: number;           // Number of reviews
```

```
  hours?: object;                  // Hours of operation
  source: "apify_google_maps";     // Data source
}
```

Importing Downloaded Data

After downloading data, you can import it into your database:

Using the Bulk Import Endpoint

```
# Download data first
python scripts/download_apify_runs.py --output towing_data.json

# Import via API (you'll need to create a zone first)
curl -X POST "http://localhost:8000/api/v1/companies/bulk-import?zone_id=
{zone_id}" \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d @towing_data.json
```

Using Python Script

You can extend the download script to automatically import:

```
import asyncio
from app.services.apify_service import ApifyService
from app.services.company_service import CompanyService
from app.database import AsyncSessionLocal

async def download_and_import(zone_id):
    apify_service = ApifyService()
    company_service = CompanyService()

    # Download data
    result = await apify_service.download_all_towing_data(limit_runs=10)

    # Import to database
    async with AsyncSessionLocal() as db:
        for company_data in result['companies']:
            await company_service.create_or_update_company(
                db, company_data, zone_id
            )
        await db.commit()
```

```
    await apify_service.close()
```

Troubleshooting

Authentication Error

- Ensure `APIFY_TOKEN` is set in your `.env` file
- Verify token is valid at https://console.apify.com/account/integrations

No Runs Found

- Check if you have any completed runs in Apify console
- Verify the actor ID matches your runs
- Try without status filter to see all runs

Run Not Completed

- Only `SUCCEEDED` runs can be downloaded
- Check run status in Apify console
- Wait for running jobs to complete

Large Datasets

- Use `limit` parameter to paginate results
- Process runs individually for better control
- Consider using the script for batch operations

Best Practices

1. **Start Small**: Test with `--limit-runs 1` first
2. **Check Data Quality**: Review downloaded data before importing
3. **Deduplicate**: Use `google_business_url` to avoid duplicates

4. **Incremental Updates**: Download new runs periodically
5. **Backup**: Save downloaded JSON files before importing

Example Workflow

```
# 1. List available runs
python scripts/download_apify_runs.py --list-only

# 2. Download specific run
python scripts/download_apify_runs.py --run-id abc123 --output
run_abc123.json

# 3. Review the data
cat run_abc123.json | jq '.companies[0]'

# 4. Import to database (via API or script)
# ... import logic ...
```