**PATHWAY OUTPUT SCRIPT (PowerShell)**

---

**Overview**

This demo showcases **Pathway's live indexing** capability.
You'll use **two terminals** for this demonstration.

---

**INSTRUCTIONS**

1. **Open Terminal 1:** Run the Docker build and run commands.
2. **Open Terminal 2:** Executed Steps 1-4

---

**TERMINAL 1 - RUN THE SERVER**

*(Ran these commands first and left this terminal open)*

---

**Step 1 - Build the Docker image**

docker build -t pathway-financial-agent .

This builds the containerized Pathway application image.

---

**Step 2 - Run the Docker container**

docker run -p 8000:8000 -v ${PWD}/feed.jsonl:/app/feed.jsonl --rm --name pathway-app pathway-financial-agent

logs similar to:

Starting DocumentStoreServer on 0.0.0.0:8000

Starting Pathway engine (pw.run())

---

**TERMINAL 2 - LIVE DEMONSTRATION**

*(Run these commands in a **new** PowerShell terminal)*

---

**STEP 1 - Query the Initial Data**

We'll query **"Tesla"**.
The initial feed.jsonl file contains one article about Tesla.

echo "STEP 1: Querying for 'Tesla' with initial data..."

Invoke-RestMethod -Method Post -Uri "http://localhost:8000/v1/retrieve" -ContentType "application/json" -Body '{"query": "What is the news about Tesla?", "k": 1}' | ConvertTo-Json -Depth 5

**Output (Step 1)**


{

"documents": [

{

"text": "Tesla shares rose 8% after announcing breakthrough in battery technology that could reduce costs by 30%.",

"metadata": {

"headline": "Tesla Stock Jumps"

}

}

]

}


---

**STEP 2 - Simulate New Streaming Data**

Now we'll **add a new article** about Tesla to the same feed.jsonl file — *while the server is still running*. This simulates a **live news feed** being ingested.

echo "STEP 2: Adding new article to feed.jsonl..."

$newArticle = '{"headline": "Tesla Faces New Competition", "body": "A new report suggests competition in the EV market is heating up, with Tesla facing new challenges from rival automakers."}'

Add-Content -Path ./feed.jsonl -Value $newArticle
autocommit_duration_ms=500 in app.py ensures near-instant refresh.

---

**STEP 3 - Re-query to Show Real-Time Update**

Re-run the same query.
Pathway should now return the **newest** Tesla article — proving live indexing works.

echo "STEP 3: Re-querying for 'Tesla'..."

Invoke-RestMethod -Method Post -Uri "http://localhost:8000/v1/retrieve" -ContentType "application/json" -Body '{"query": "What is the news about Tesla?", "k": 1}' | ConvertTo-Json -Depth 5

**Output (Step 3)**

```
{
  "documents": [
    {
      "text": "A new report suggests competition in the EV market is heating up, with Tesla facing new challenges from rival automakers.",
      "metadata": {
        "headline": "Tesla Faces New Competition"
      }
    }
  ]
}
```

---

**Step 4: Looping over many batches (script for larger datasets)**

```
# Assume we have N batches as files named batch-0.jsonl, batch-1.jsonl, ...
$batchFiles = Get-ChildItem -Path ./batches -Filter "batch-*.jsonl" | Sort-Object Name
foreach ($f in $batchFiles) {
Get-Content $f | Add-Content -Path ./feed.jsonl
Write-Output "Appended $($f.Name)"
Start-Sleep -Milliseconds 500 # control ingestion rate
}
```

---

To stop the server (Terminal 1):

CTRL + C

---