# TokenFlow - Multi-Service OIDC Testing Tool (Updated)

## Project Overview

TokenFlow is a Python-based CLI tool to automate testing of multiple OpenID Connect (OIDC) services with **one login session**. It fetches OIDC metadata, launches a real browser via Playwright to capture authorization codes, exchanges them for tokens, and retrieves user information.

## Getting Started

1. **Clone the repository:**
   git clone https://your-org/tokenflow.git   after that type: cd tokenflow
2. **Create and activate a virtual environment:**
   python3 -m venv myenv
3. source myenv/bin/activate  # (Linux/macOS) myenv\Scripts\activate    # (Windows)
4. **Install dependencies:**
   pip install -r requirements.txt
5. **Install Playwright browsers:**
   playwright install

## Usage

### 1. Add services from metadata

Add an OIDC service dynamically by specifying a name and metadata URL:

```
python tokenflow.py --add-service unpd0002 "https://git.unl.edu/../..../-/../../.../edu-unl-unpd0002.xml"
```

### 2. Set environment variables for secrets

Each service must have its client secret set as an environment variable:

```
export UNPD0002_SECRET="your-client-secret"
```

### 3. Run tests

Run tests for one or multiple services:

```
python tokenflow.py --run unpd0002
```

Or multiple services:

```
python tokenflow.py --run unpd0002 unpd0016
```

Optional flags:

- --json → output userinfo in JSON format
- --output <filename> → specify custom CSV output file

Example:

```
python tokenflow.py --run unpd0002 unpd0016 --json --output myresults.csv
```

### 4. List all saved services

```
python tokenflow.py --list-services
```

### CSV Output

Each run generates a CSV file (tokenflow_results.csv by default) with the following columns:

| Column | Description |
|---|---|
| Service Name | Name of the service tested |
| Auth Code Captured | Yes/No |
| Token Exchange | Success/Failed |
| UserInfo | Email/sub retrieved or status |
| Timestamp | Date and time of test |
| Error | Error message if any |

### Features

- Test multiple OIDC services in one login session
- Dynamically pull metadata from GitLab or other sources
- Auto-generate authorization URLs based on metadata
- Hybrid flow support (code id_token)
- CSV reporting for audit trail

### Troubleshooting

- **invalid_grant** → Ensure redirect_uri and client_id match metadata and environment secret.
- **Browser not launching** → Run playwright install to install necessary browsers.
- **No auth code extracted** → Make sure login fully completes (including MFA if applicable).
- **Environment variable missing** → Ensure you exported <SERVICE_NAME>_SECRET.

### Notes

- Services must support authorization code or hybrid flow.
- Client secrets are never stored inside the project, only pulled securely from environment variables.
- Metadata should include grant_types, response_types, and redirect URIs.

### Developed by

**Samay Bhojwani – 2025**<br>
🔗 Connect with me on LinkedIn