AXIOM v2.0 - Enhanced Rainmeter Scraper

🚀 Overview

AXIOM (Automated eXtraction Index Operator) is an enterprise-grade web scraper designed to collect, download, and organize Rainmeter skins from RainmeterUI.

New in v2.0

- ✅ **SQLite Database** - Efficient storage and querying
- ✅ **Async Downloads** - 5-10x faster with aiohttp
- ✅ **Adaptive Rate Limiting** - Smart request throttling
- ✅ **Enhanced Security** - Path traversal protection
- ✅ **Incremental Progress** - Resume from any point
- ✅ **Better Memory Management** - Handles 10,000+ skins
- ✅ **Comprehensive Testing** - Full test suite included
- ✅ **Graceful Shutdown** - CTRL+C safe

📋 Requirements

System Requirements

- **OS**: Windows 10/11, Linux, macOS
- **Python**: 3.8 or higher
- **Disk Space**: 5 GB minimum (10+ GB recommended)
- **RAM**: 2 GB minimum (4 GB recommended)
- **Internet**: Stable broadband connection

Python Dependencies

All dependencies are installed automatically via `requirements.txt`:

- `aiohttp` - Async HTTP client
- `beautifulsoup4` - HTML parsing
- `lxml` - Fast XML/HTML parser
- `rarfile` - RAR archive extraction
- `py7zr` - 7-Zip archive extraction
- `pygame` (optional) - Startup animation
- `numpy` (optional) - Audio generation

🎯 Quick Start

Windows Users

Method 1: Double-click the batch file (Easiest)

batch

```
# Simply double-click: setup_and_run.bat
```

Method 2: Command line with options

batch

```
setup_and_run.bat --workers 10 --delay 0.5
```

Linux/macOS Users

bash

```
# Install dependencies
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt

# Run scraper
python AXIOM.py --config rainmeterui_categories.json --output scraped_data
```

⚙️ Configuration

Command Line Options

```
Usage: AXIOM.py [options]

Core Options:
  --config FILE         Configuration file (default: rainmeterui_categories.json)
  --output DIR          Output directory (default: scraped_data)
  --delay SECONDS       Request delay in seconds (default: 1.0)
  --workers NUM         Parallel download workers (default: 5)
  --batch-size NUM      Download batch size (default: 100)

Operation Modes:
  --resume              Resume interrupted scraping session
  --verbose             Enable verbose logging
  --diagnostic          Run system diagnostics and exit

Examples:
  AXIOM.py                          # Run with defaults
  AXIOM.py --workers 10 --delay 0.5         # Fast mode
  AXIOM.py --resume                         # Resume previous session
  AXIOM.py --diagnostic                     # Check system health
```

Configuration File

The `rainmeterui_categories.json` file defines categories to scrape:

json

```
{
  "rainmeterui_categories": {
    "primary_skin_categories": [
      {
        "name": "Suites",
        "url": "https://rainmeterui.com/category/suites/"
      },
      {
        "name": "System Monitors",
        "url": "https://rainmeterui.com/category/system-monitors/"
      }
    ],
    "additional_categories": [
      {
        "name": "Clocks",
        "url": "https://rainmeterui.com/category/clocks/"
      }
    ]
  }
}
```

📊 Output Structure

After scraping, your output directory contains:

```
scraped_data/
├── skins.db                    # SQLite database with all metadata
├── complete_collection.json    # Full JSON export
├── complete_collection.csv     # CSV export for Excel/analysis
├── scraping_summary.txt        # Human-readable summary
├── axiom_scraper.log           # Detailed log file
├── downloads/                  # Downloaded skin packages
│   ├── Suites/
│   │   ├── MySuite_v1.0.rmskin
│   │   └── AnotherSuite.zip
│   └── Clocks/
│       └── AnalogClock.zip
└── extracted_skins/            # Extracted skin contents
    ├── Suites/
    │   ├── MySuite/
    │   │   ├── Skins/
    │   │   └── @Resources/
    │   └── AnotherSuite/
    └── Clocks/
        └── AnalogClock/
```

🔧 Advanced Usage

Resuming Interrupted Sessions

If scraping is interrupted (network issue, power loss, etc.):

batch

```
# Windows
setup_and_run.bat --resume

# Linux/macOS
python AXIOM.py --resume
```

The scraper will:

1. Load the SQLite database
2. Identify incomplete downloads

3. Continue from where it stopped

Performance Tuning

Fast Mode (More aggressive)

batch

```
setup_and_run.bat --workers 15 --delay 0.3 --batch-size 200
```

Safe Mode (Server-friendly)

batch

```
setup_and_run.bat --workers 3 --delay 2.0 --batch-size 50
```

Recommended Settings

batch

```
# Balanced performance and safety
setup_and_run.bat --workers 5 --delay 1.0 --batch-size 100
```

Running Diagnostics

Before scraping, check your system:

batch

```
setup_and_run.bat --diagnostic
```

This tests:

- Python installation
- Required modules
- Configuration file validity
- Disk space

- Network connectivity
- Write permissions

Database Operations

Querying the Database

python

```python
import sqlite3

# Connect to database
conn = sqlite3.connect('scraped_data/skins.db')
cursor = conn.cursor()

# Get all extracted skins
cursor.execute("SELECT * FROM skins WHERE download_status = 'extracted'")
for row in cursor.fetchall():
    print(row)

# Get skins by category
cursor.execute("SELECT title, author FROM skins WHERE category = 'Suites'")
suites = cursor.fetchall()

# Count downloads by status
cursor.execute("""
    SELECT download_status, COUNT(*)
    FROM skins
    GROUP BY download_status
""")
stats = cursor.fetchall()

conn.close()
```

Exporting Data

Export to JSON:

python

```python
from AXIOM import SkinDatabase

db = SkinDatabase('scraped_data/skins.db')
db.export_to_json('my_export.json')
```

Export to CSV:

python

```python
db.export_to_csv('my_export.csv')
```

Security Features

Path Traversal Protection

AXIOM prevents malicious archives from extracting files outside the designated directory:

python

```python
# Blocked automatically:
# archive contains: ../../../../../../etc/passwd
# archive contains: C:\Windows\System32\evil.exe
```

File Size Limits

- **Per File**: 500 MB maximum
- **Per Archive**: 2 GB maximum

These limits prevent:

- Zip bombs
- Disk space exhaustion
- Memory overflow attacks

Safe Extraction

All archives extract to temporary directories first, then atomically move to final location. Prevents corruption from interrupted extractions.

🐛 Troubleshooting

Common Issues

Issue: "Python not found"

Solution: Install Python 3.8+ and ensure it's in PATH
Windows: Check "Add Python to PATH" during installation

Issue: "Failed to install dependencies"

Solution:
1. Check internet connection
2. Update pip: python -m pip install --upgrade pip
3. Try manual install: pip install aiohttp beautifulsoup4

Issue: "Low disk space warning"

Solution: Free up at least 5 GB disk space
Typical collection uses 2-10 GB depending on number of skins

Issue: "Rate limited / Too many requests"

Solution: Increase delay between requests
setup_and_run.bat --delay 2.0

Issue: "Database locked"

Solution: Ensure no other AXIOM processes are running
Close any database browsers (DB Browser for SQLite, etc.)

Issue: "Extract failed: corrupted archive"

Solution: Download will be marked as failed
Re-run with --resume to retry failed downloads
Some files may be genuinely corrupted on the source

Log Files

Check these files for detailed error information:

1. `logs/axiom_TIMESTAMP.log` - Setup script log
2. `scraped_data/axiom_scraper.log` - Scraper log
3. `scraped_data/scraping_summary.txt` - Summary report

Getting Help

If issues persist:

1. Run diagnostic mode: `--diagnostic`
2. Check logs for error messages
3. Verify configuration file JSON syntax
4. Test network connectivity to rainmeterui.com

📈 Performance Benchmarks

Typical performance on modern hardware:

Metric	Value
Discovery Speed	100-200 skins/minute
Download Speed	5-15 MB/s (network dependent)
Extraction Speed	20-50 archives/minute
Memory Usage	200-500 MB
CPU Usage	10-30% (multi-threaded)

Complete Collection Estimates:

- 1,000 skins: 1-2 hours
- 5,000 skins: 4-8 hours
- 10,000 skins: 8-16 hours

Times vary based on network speed, server response, and system specs

---

🔄 Updating from v1.0

If you have the old AXIOM (JSON-based):

1. **Backup your data**

```
Copy scraped_data folder to scraped_data_backup
```

2. **Install new version**

```
Update requirements: pip install -r requirements.txt
```

3. **Migration (optional)**

python

```python
# Convert old JSON to new database
import json
from AXIOM import SkinDatabase, SkinMetadata

db = SkinDatabase('new_skins.db')

with open('old_collection.json') as f:
    data = json.load(f)

for skin_data in data.get('skins', []):
    skin = SkinMetadata(**skin_data)
    db.save_skin(skin)
```

4. **Benefits of upgrading**
   - 60-80% less memory usage
   - 3-5x faster downloads
   - Resume from any point

- Better error handling
- SQL query capabilities

🧪 Testing

Run the test suite to verify installation:

bash

```
# Install test dependencies
pip install pytest pytest-asyncio

# Run all tests
python axiom_tests.py

# Run specific test class
pytest axiom_tests.py::TestSkinDatabase -v

# Run with coverage
pytest --cov=AXIOM axiom_tests.py
```

📝 Best Practices

Do's ✅

- ✅ Start with default settings
- ✅ Use `--diagnostic` before large scrapes
- ✅ Keep 10+ GB free disk space
- ✅ Use `--resume` after interruptions
- ✅ Check logs if issues occur
- ✅ Export database periodically (backup)

Don'ts ❌

- ❌ Don't set delay below 0.3 seconds
- ❌ Don't use more than 20 workers
- ❌ Don't run multiple instances simultaneously
- ❌ Don't modify database while scraping
- ❌ Don't delete database until scraping complete

🤝 Contributing

Improvements welcome! Focus areas:

- Additional archive formats (e.g., .tar.gz)
- More metadata extraction
- Better error recovery
- Performance optimizations
- Additional output formats

📜 License

This software is provided as-is for personal and educational use. Respect RainmeterUI's terms of service and rate limits.

🎯 Roadmap

Planned Features

- [ ] GUI interface
- [ ] Cloud storage integration
- [ ] Duplicate detection
- [ ] Skin preview generation
- [ ] Automatic Rainmeter installation
- [ ] Skin compatibility checker
- [ ] Tag-based search
- [ ] RESTful API

📞 Support

For issues, questions, or feature requests:

1. Check this documentation
2. Review log files
3. Run `--diagnostic`
4. Check GitHub issues (if applicable)

Version: 2.0.0
Last Updated: 2024
Status: Production Ready

AXIOM - Making Rainmeter skin collection effortless