Orion Content - Phase 3 Usage Guide

Quick reference for using the Python pipeline tools with examples.

Setup

1. Install Dependencies

```
cd python
make install
```

2. Environment Configuration

Create a `.env` file or export variables:

```
# Required
export ORION_BASE_URL="http://localhost:3000"
export ORION_CONSOLE_TOKEN="your-48-char-api-token-from-setup-wizard"

# Optional (WordPress integration)
export WP_BASE_URL="https://your-site.com"
export WP_USERNAME="admin"
export WP_APP_PASSWORD="abcd efgh ijkl mnop"
```

3. Verify Setup

```
make smoke-test
```

Command Examples

Generate Topics

Create 30 trending topics for the current week:

```
ORION_CONSOLE_TOKEN=abc123... python -m orion.gather_trends \
  --site-key my-site \
  --count 30
```

Output:

```
2024-01-15 10:30:00 - orion.api_client - INFO - Checking API health...
2024-01-15 10:30:00 - orion.api_client - INFO - Finding site: my-site
2024-01-15 10:30:00 - orion.api_client - INFO - Found site: My Content Site (ID: cltx8.
..)
2024-01-15 10:30:00 - orion.api_client - INFO - Ensuring current week exists...
2024-01-15 10:30:00 - orion.api_client - INFO - Week: 2024-W03 (ID: cltx9..., Status: P
ENDING)
2024-01-15 10:30:00 - orion.api_client - INFO - Generated 30 topics across 3 categories
2024-01-15 10:30:01 - orion.api_client - INFO - Successfully created 30 topics
SUMMARY:
  Week ID: cltx9...
  ISO Week: 2024-W03
  Topics Created: 30
  First 5 topics:
    1. Tech Trend #07 — Neural Networks
    2. AI Update: Large Language Models
    3. Latest in Business: Digital Transformation
    4. Edge Computing — A Technology Deep Dive
    5. Breaking: Quantum Algorithms in AI
```

Dry run (generate but don’t post):

```
python -m orion.gather_trends --site-key my-site --count 10 --dry-run
```

WordPress Publishing

Create a draft post (dry-run if WordPress not configured):

```
python -m orion.publish.publisher_wp \
  --title "AI Revolution in 2024" \
  --content "<p>The AI landscape is rapidly evolving...</p>"
```

Dry-run output (no WordPress config):

```
{
  "id": "dry-run",
  "link": "https://example.com/dry-run-post",
  "status": "draft",
  "title": {
    "rendered": "AI Revolution in 2024"
  }
}
```

Real WordPress output:

```
{
  "id": 123,
  "link": "https://mysite.com/ai-revolution-2024",
  "status": "draft",
  "title": {
    "rendered": "AI Revolution in 2024"
  },
  "date": "2024-01-15T10:30:00",
  "modified": "2024-01-15T10:30:00"
}
```

Publish with categories:

```
python -m orion.publish.publisher_wp \
  --title "Breaking Tech News" \
  --content "Content here..." \
  --status publish \
  --categories "Technology,AI,News"
```

WordPress Audit

List recent draft posts:

```
python -m orion.audit.audit_wp --limit 5
```

Output:

```
ID         Title                                              Status
Date
---------------------------------------------------------------------------------------
---
123        AI Revolution in 2024                             draft
2024-01-15T10:30:00
122        Breaking Tech News                                 publish
2024-01-15T09:15:00
121        The Future of Edge Computing                       draft
2024-01-14T16:45:00
120        Machine Learning Best Practices                    draft
2024-01-14T14:20:00
119        Quantum Computing Advances                         draft
2024-01-14T11:10:00
```

List all recent posts:

```
python -m orion.audit.audit_wp --limit 10 --status any
```

Integration with Orion Console

1. Get Console API Token

From the Orion setup wizard or by checking your Connection table:

```
# From setup wizard - generates and shows token
curl -X POST http://localhost:3000/api/setup/secrets \
  -H "Authorization: Bearer YOUR_SESSION_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"kind":"console_api_token","data":{"token":"auto-generated-token"}}'
```

2. Verify API Connection

```
ORION_CONSOLE_TOKEN=your-token python -c "
from orion.api_client import client
print('Health:', client.health_check())
print('Sites:', [s['key'] for s in client.get_sites()])
"
```

3. Create a Site (if needed)

Via Orion web UI or API:

```
curl -X POST http://localhost:3000/api/sites \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"key":"my-blog","name":"My Tech Blog","timezone":"UTC"}'
```

Workflow Examples

Daily Content Pipeline

```
#!/bin/bash
# scripts/daily-pipeline.sh

export ORION_BASE_URL="https://your-orion.vercel.app"
export ORION_CONSOLE_TOKEN="your-console-token"
export WP_BASE_URL="https://your-blog.com"
export WP_USERNAME="admin"
export WP_APP_PASSWORD="your-app-password"

# 1. Generate topics for the week (Monday only)
if [ $(date +%u) -eq 1 ]; then
  echo "Monday: Generating weekly topics..."
  python -m orion.gather_trends --site-key main-site --count 30
fi

# 2. Create daily WordPress draft
echo "Creating daily draft..."
TITLE="Daily Tech Update - $(date +%Y-%m-%d)"
CONTENT="<p>Today's tech highlights and trends...</p>"

python -m orion.publish.publisher_wp \
  --title "$TITLE" \
  --content "$CONTENT" \
  --categories "Daily,Technology"

# 3. Audit recent posts
echo "Recent drafts:"
python -m orion.audit.audit_wp --limit 5
```

Weekly Topic Generation

```bash
#!/bin/bash
# Generate topics for multiple sites

SITES=("tech-blog" "ai-news" "business-insights")

for site in "${SITES[@]}"; do
  echo "Generating topics for $site..."
  python -m orion.gather_trends \
    --site-key "$site" \
    --count 20
done
```

Troubleshooting

Common Issues

1. **"API Error: 401 Unauthorized"**

   ```
   bash
      # Check token is set
      echo $ORION_CONSOLE_TOKEN
      # Verify token works
      curl -H "Authorization: Bearer $ORION_CONSOLE_TOKEN" \
           http://localhost:3000/api/health
   ```

2. **"Site not found: my-site"**

   ```
   bash
      # List available sites
      python -c "from orion.api_client import client; print([s['key'] for s in client.get_sites()])"
   ```

3. **"Topics must be an array" or Schema Errors**
   - Check that your site has categories defined
   - Verify API schemas match expected format

4. **WordPress "401 Unauthorized"**
   - Verify Application Password (not account password)
   - Check username spelling
   - Ensure WordPress REST API is enabled

5. **Network timeouts**

   ```
   bash
      # Test connectivity
      curl -v http://localhost:3000/api/health
      curl -v https://your-wp-site.com/wp-json/wp/v2/posts
   ```

Debug Mode

Enable detailed logging:

```python
import logging
logging.getLogger('orion').setLevel(logging.DEBUG)
```

Or set environment variable:

```bash
export ORION_LOG_LEVEL=DEBUG
```

Manual API Testing

Test API endpoints directly:

```bash
# Health check
curl http://localhost:3000/api/health

# Get sites (requires auth)
curl -H "Authorization: Bearer YOUR_TOKEN" \
     http://localhost:3000/api/sites

# Create week
curl -X POST http://localhost:3000/api/weeks \
     -H "Authorization: Bearer YOUR_TOKEN" \
     -H "Content-Type: application/json" \
     -d '{"isoWeek":"2024-W03"}'

# WordPress test
curl -u "username:app-password" \
     https://your-site.com/wp-json/wp/v2/posts?per_page=1
```

Performance Notes

- **Topic Generation**: ~1-2 seconds for 30 topics
- **WordPress API**: ~500ms per post creation
- **Batch Operations**: Topics are created in single API call
- **Rate Limits**: None currently enforced, but use reasonable request patterns
- **Timeouts**: 30 second default, configurable in `config.py`

Next Steps

After Phase 3 is working:

1. **GitHub Actions Integration**: Use these tools in automated workflows
2. **Advanced Content**: Add content research and AI writing
3. **Multi-site Management**: Parallel processing for multiple sites
4. **Analytics Integration**: Track topic performance
5. **Content Templates**: Customize topic generation patterns

---

For issues or questions, check the main README or create an issue in the repository.