

SEA-E Quick Start Guide

Get up and running with SEA-E in 5 minutes! This guide will help you set up the system and process your first product quickly.

⚡ 5-Minute Setup

Prerequisites

- Python 3.8+ installed
- Git installed
- Active Airtable, Etsy, and Printify accounts

Step 1: Install SEA-E (2 minutes)

```
# Clone and setup
git clone https://github.com/your-repo/sea-engine.git
cd sea-engine

# Create virtual environment and install
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
pip install -r requirements.txt

# Verify installation
python run_engine.py --help
```

Expected Output:

```
SEA-E: Scalable E-commerce Automation Engine

usage: run_engine.py [-h] [--manifest MANIFEST] [--dry-run] [--list-manifests]
                  [--log-level {DEBUG,INFO,WARNING,ERROR,CRITICAL}]
```

Step 2: Configure API Keys (2 minutes)

```
# Copy environment template
cp .env.example .env

# Edit with your API credentials
nano .env # or use your preferred editor
```

Required Configuration:

```
# Add your actual API keys
AIRTABLE_API_KEY=your_airtable_personal_access_token
AIRTABLE_BASE_ID=your_airtable_base_id
ETSY_API_KEY=your_etsy_api_key
ETSY_API_SECRET=your_etsy_api_secret
ETSY_ACCESS_TOKEN=your_etsy_access_token
ETSY_REFRESH_TOKEN=your_etsy_refresh_token
ETSY_SHOP_ID=your_etsy_shop_id
PRINTIFY_API_KEY=your_printify_api_key
PRINTIFY_SHOP_ID=your_printify_shop_id
```

Step 3: Test Connection (1 minute)

```
# Test your setup
python run_engine.py --list-manifests
```

Success Indicators:

- No error messages
- Shows available manifests
- Displays SEA-E version info

First Run

Quick Test with Sample Data

```
# Run a safe test (no actual API calls)
python run_engine.py --manifest tshirts_q3_launch --dry-run

# Expected output:
#   Configuration loaded
#   API connections verified
#   Found X products to process
#   DRY RUN: Would generate mockups
#   DRY RUN: Would create products
#   DRY RUN: Would create listings
```

Process Your First Product

1. **Set up Airtable base** with the required tables (see [Tutorial](#) (TUTORIAL.md) for details)
2. **Add a test product** to your Products table
3. **Run the workflow:**

```
# Process one product
python run_engine.py --manifest first_product_test --log-level INFO
```

What happens:

1. Connects to Airtable and retrieves your product
2. Generates mockup images via Printify
3. Creates print-on-demand product

- 4. Creates Etsy listing draft
- 5. Updates status in Airtable

Essential Commands

Most Common Operations

```
# List all available workflows
python run_engine.py --list-manifests

# Test run without making changes
python run_engine.py --manifest MANIFEST_NAME --dry-run

# Process products with detailed logging
python run_engine.py --manifest MANIFEST_NAME --log-level DEBUG

# Process specific collection
python run_engine.py --manifest collection_summer_2024

# Monitor progress in real-time
tail -f logs/sea-engine.log
```

Quick Reference

Command	Purpose	Example
--list-manifests	Show available workflows	python run_engine.py --list-manifests
--dry-run	Test without API calls	python run_engine.py --manifest test --dry-run
--log-level DEBUG	Detailed logging	python run_engine.py --manifest test --log-level DEBUG
--help	Show all options	python run_engine.py --help

Troubleshooting Shortcuts

Common Issues & Quick Fixes

“Authentication failed”

```
# Check your .env file
cat .env | grep API_KEY
# Verify tokens haven't expired
```

“No products found”

```
# Check Airtable base structure
# Verify collection names match manifest filters
```

“Rate limit exceeded”

```
# Wait 1 hour or reduce batch size
# Check API usage in your provider dashboards
```

“Module not found”

```
# Reinstall dependencies
pip install -r requirements.txt
```

Quick Diagnostics

```
# Test individual components
python -c "from src.api.airtable_client import AirtableClient; print('Airtable: OK')"
python -c "from src.api.etsy import EtsyAPIClient; print('Etsy: OK')"
python -c "from src.api.printify import PrintifyAPIClient; print('Printify: OK')"

# Check log files
tail -20 logs/sea-engine.log

# Verify environment
python -c "import os; from dotenv import load_dotenv; load_dotenv(); print('Config loaded')"
```

Next Steps

After Your First Success

1. Read the [Complete Tutorial](#) (TUTORIAL.md) for detailed workflows
2. Set up your Airtable base with proper table structure
3. Create custom manifests for your specific needs
4. Scale up with batch processing

Key Resources

- [README.md](#) (README.md) - Complete system documentation
- [TUTORIAL.md](#) (TUTORIAL.md) - Step-by-step detailed guide
- **Configuration Examples** - In `config/` directory
- **Test Suite** - Run `pytest` for validation

Getting Help

- **Check logs first:** `tail -f logs/sea-engine.log`
- **Use dry-run mode** to test safely
- **Start small** with 1-2 products
- **Read error messages** carefully - they're usually specific

Success Checklist

- ☐ Python 3.8+ installed
- ☐ SEA-E repository cloned
- ☐ Virtual environment activated
- ☐ Dependencies installed
- ☐ API credentials configured
- ☐ Test run completed successfully
- ☐ First product processed
- ☐ Airtable updated with results
- ☐ Ready to scale up!

Congratulations! You're now ready to automate your e-commerce product workflow with SEA-E.

For advanced features and detailed explanations, continue with the [Complete Tutorial](#) (TUTORIAL.md).