

Python Mock Services - Quick Start Guide

Quick Start (3 Commands)

```
# 1. Setup (one-time)
./scripts/setup-python-mocks.sh

# 2. Start all services
./scripts/start-python-mocks.sh

# 3. Test services
curl http://localhost:3001/health # CMS
curl http://localhost:3002/health # ROS
curl http://localhost:3003/health # WMS
```

Service URLs

Service	Port	Base URL	API Docs
CMS Mock	3001	http://localhost:3001	http://localhost:3001/docs
ROS Mock	3002	http://localhost:3002	http://localhost:3002/docs
WMS Mock	3003	http://localhost:3003	http://localhost:3003/docs

Common Commands

Start Services

```
# Start all services
./scripts/start-python-mocks.sh

# Start individual service
cd services/mocks/cms-mock
source venv/bin/activate
python app.py
```

Stop Services

```
# Stop all services
./scripts/stop-python-mocks.sh

# Stop individual service (Ctrl+C in terminal)
```

Test Services

```
# Run automated tests
./scripts/test-mock-services.sh

# Manual health checks
curl http://localhost:3001/health
curl http://localhost:3002/health
curl http://localhost:3003/health
```

API Endpoints

CMS Mock (Customer Management)

```
# Get all customers
curl http://localhost:3001/api/customers/

# Get customer by ID
curl http://localhost:3001/api/customers/{id}

# Create customer
curl -X POST http://localhost:3001/api/customers/ \
-H "Content-Type: application/json" \
-d '{"name": "John Doe", "email": "john@example.com", "phone": "+1-555-0100", "address": "123 Main St", "company": "Tech Corp", "status": "active"}'

# Update customer
curl -X PUT http://localhost:3001/api/customers/{id} \
-H "Content-Type: application/json" \
-d '{"name": "Updated Name"}'

# Delete customer
curl -X DELETE http://localhost:3001/api/customers/{id}
```

ROS Mock (Route Optimization)

```
# Get all routes
curl http://localhost:3002/api/routes/

# Optimize route
curl -X POST http://localhost:3002/api/routes/optimize \
-H "Content-Type: application/json" \
-d '{"origin": "New York", "destination": "Boston", "waypoints": [], "vehicle_type": "truck", "constraints": {"max_distance": 1000}}'

# Create route
```

```
curl -X POST http://localhost:3002/api/routes/ \
  -H "Content-Type: application/json" \
  -d '{"origin":"City A","destination":"City B","vehicle_id":"VEH-001","driver_id":"DRV-001","distance":100.0,"status":"planned"}'
```

WMS Mock (Warehouse Management)

```
# Get all inventory
curl http://localhost:3003/api/inventory/

# Get inventory by ID
curl http://localhost:3003/api/inventory/{id}

# Create inventory
curl -X POST http://localhost:3003/api/inventory/ \
  -H "Content-Type: application/json" \
  -d '{"sku":"PROD-100","name":"Widget","quantity":50,"location": {"warehouse_id":"WH-001","zone":"A","aisle":"1","rack":"1","bin":"A"}, "unit_price":9.99,"reorder_level":10}'

# Update inventory
curl -X PUT http://localhost:3003/api/inventory/{id} \
  -H "Content-Type: application/json" \
  -d '{"quantity":75}'

# Check stock
curl -X POST http://localhost:3003/api/inventory/check-stock \
  -H "Content-Type: application/json" \
  -d '{"items": [{"sku":"PROD-001","quantity":5}]}'
```

Data Storage

File Locations

```
services/mocks/cms-mock/data/customers.json
services/mocks/ros-mock/data/routes.json
services/mocks/wms-mock/data/inventory.json
```

View Data

```
# View customers
cat services/mocks/cms-mock/data/customers.json | python3 -m json.tool

# View routes
```

```
cat services/mocks/ros-mock/data/routes.json | python3 -m json.tool  
  
# View inventory  
cat services/mocks/wms-mock/data/inventory.json | python3 -m json.tool
```

Reset Data

```
# Delete data file (will be recreated with initial data on restart)  
rm services/mocks/cms-mock/data/customers.json  
rm services/mocks/ros-mock/data/routes.json  
rm services/mocks/wms-mock/data/inventory.json  
  
# Restart services  
../scripts/stop-python-mocks.sh  
../scripts/start-python-mocks.sh
```

Troubleshooting

Services Won't Start

```
# Check if port is already in use  
lsof -i :3001  
lsof -i :3002  
lsof -i :3003  
  
# Kill process on port  
kill -9 $(lsof -t -i:3001)
```

Dependencies Issues

```
# Remove virtual environments  
rm -rf services/mocks/*/venv  
  
# Reinstall  
../scripts/setup-python-mocks.sh
```

View Service Logs

```
# Services run in foreground, check terminal output  
# Or redirect to log file when starting:  
cd services/mocks/cms-mock  
source venv/bin/activate  
python app.py > cms.log 2>&1 &
```

Dependencies

All services use:

- **FastAPI** 0.115.6 - Web framework
 - **Uvicorn** 0.34.0 - ASGI server
 - **Pydantic** 2.10.6 - Data validation
 - **Python** 3.13+ - Runtime
-

Key Features

-  **File-Based Storage** - JSON files for data persistence
 -  **Thread-Safe** - Concurrent request handling
 -  **Auto-Generated Docs** - Interactive API documentation
 -  **CORS Enabled** - Cross-origin requests supported
 -  **Type Validation** - Pydantic models for type safety
 -  **Hot Reload** - Automatic restart on code changes
-

Additional Resources

- **Test Report:** [PYTHON_SERVICES_TEST_REPORT.md](#)
 - **Storage Guide:** [FILE_STORAGE_GUIDE.md](#)
 - **Migration Summary:** [FILE_STORAGE_SUMMARY.md](#)
 - **Verification Report:** [VERIFICATION_REPORT.md](#)
-

Support

For issues or questions:

1. Check service logs
 2. Review API documentation at [/docs](#) endpoint
 3. Verify data files exist in [data/](#) directories
 4. Ensure virtual environments are activated
 5. Check port availability
-

Last Updated: February 1, 2026

Status:  All Services Operational