

Quick Start Guide

Get up and running with UtilityFog-Fractal-TreeOpen in minutes!

Installation

Prerequisites

- Python 3.9 or higher
- pip or pipx package manager

Option 1: Install with pip

```
# Install from PyPI (when available)
pip install utilityfog-fractal-tree

# Or install from source
git clone https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen.git
cd UtilityFog-Fractal-TreeOpen
pip install -r testing_requirements.txt
```

Option 2: Install with pipx (Recommended)

```
# Install with pipx for isolated environment
pipx install utilityfog-fractal-tree

# Or from source
pipx install git+https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen.git
```

Platform-Specific Instructions

Linux

```
# Ubuntu/Debian
sudo apt update
sudo apt install python3 python3-pip
pip3 install utilityfog-fractal-tree

# Fedora/RHEL
sudo dnf install python3 python3-pip
pip3 install utilityfog-fractal-tree

# Arch Linux
sudo pacman -S python python-pip
pip install utilityfog-fractal-tree
```

macOS

```
# Using Homebrew
brew install python3
pip3 install utilityfog-fractal-tree

# Using MacPorts
sudo port install python311
pip3.11 install utilityfog-fractal-tree
```

Windows

```
# Using Python from python.org
python -m pip install utilityfog-fractal-tree

# Using Chocolatey
choco install python3
pip install utilityfog-fractal-tree

# Using Scoop
scoop install python
pip install utilityfog-fractal-tree
```

Quick Test

Verify your installation:

```
# Run diagnostic check
ufog-diagnose --json

# Start visualization demo
ufog-visualize --demo

# Run basic simulation
python -c "
from UtilityFog_Agent_Package.agent.main_simulation import main
main()
"
```

Basic Usage

1. Create a Simple Tree

```
from UtilityFog_Agent_Package.agent.foglet_agent import FogletAgent
from UtilityFog_Agent_Package.agent.observability import StructuredLogger

# Initialize logger
logger = StructuredLogger("quickstart")

# Create root agent
root = FogletAgent("root", logger=logger)

# Create child agents
child1 = FogletAgent("child1", parent=root, logger=logger)
child2 = FogletAgent("child2", parent=root, logger=logger)

print(f"Created tree with root '{root.node_id}' and children: {[c.node_id for c in root.children]}")
```

2. Send Messages

```
# Send a message from child to parent
child1.send_message(root.node_id, {"type": "greeting", "data": "Hello parent!"})

# Process messages
root.process_messages()
```

3. Visualize the Tree

```
from utilityfog_frontend.cli_viz.cli import VisualizationCLI

# Create CLI instance
cli = VisualizationCLI()

# Add your tree data
cli.add_tree_data(root, child1, child2)




# Generate visualization
cli.export_html("my_tree.html")
print("Visualization saved to my_tree.html")
```

Next Steps

- 📖 Read the [full documentation](https://goldislops.github.io/UtilityFog-Fractal-TreeOpen/) (https://goldislops.github.io/UtilityFog-Fractal-TreeOpen/)
- 🔍 Explore [examples and tutorials](#) (examples/)
- 🐛 Check the [troubleshooting guide](#) (troubleshooting.md)
- 💬 Join our [community discussions](https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/discussions) (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/discussions)

Need Help?

- 📄 [Report issues](https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues) (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues)

-  [Request features](https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues/new?template=feature_request.md) (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues/new?template=feature_request.md)
-  [Browse documentation](#) (docs/)
-  [View troubleshooting](#) (troubleshooting.md)