

# AI Orchestrator - Quick Reference

---

Copy-paste commands for common tasks. **Always activate venv first!**

---

## First Step (Every Terminal Session)

---

```
cd ~/ai-orchestrator && source venv/bin/activate
```

Or use the helper script:

```
./quick-start.sh <command>
```

---

## Check Setup

---

```
ai-orchestrator status  
ai-orchestrator test-api
```

---

## Test Each AI Model

---

### OpenAI (ChatGPT)

```
ai-orchestrator ask -m openai "Hello"
```

### Anthropic (Claude)

```
ai-orchestrator ask -m anthropic "Hello"
```

### Google (Gemini)

```
ai-orchestrator ask -m gemini "Hello"
```

### Moonshot (Kimi)

```
ai-orchestrator ask -m moonshot "Hello"
```

## Test All at Once

```
ai-orchestrator test-api
```

---



## Run Tasks

---

### Auto-Route (Best Model Selected Automatically)

```
ai-orchestrator run "Your task description"
```

### Force Specific Model

```
ai-orchestrator run -m openai "Architecture task"  
ai-orchestrator run -m anthropic "Coding task"  
ai-orchestrator run -m gemini "Reasoning task"  
ai-orchestrator run -m moonshot "Review task"
```

### Save Output to File

```
ai-orchestrator run "Your task" -o output.md
```

### Quiet Mode (Response Only)

```
ai-orchestrator run -q "Your task"
```

---



## Quick Questions (ask command)

---

```
# OpenAI - Architecture & Design  
ai-orchestrator ask -m openai "Design a REST API"  
  
# Anthropic - Coding  
ai-orchestrator ask -m anthropic "Write a Python sort function"  
  
# Gemini - Reasoning  
ai-orchestrator ask -m gemini "Explain Big O notation"  
  
# Moonshot - Code Review  
ai-orchestrator ask -m moonshot "Review this code"
```

---



## Analyze Task Routing

---

```
ai-orchestrator analyze "Build a web application with authentication"
```

---

## List Available Models

```
ai-orchestrator list-models gemini
```

---

## Debug Mode

```
ai-orchestrator ask -m anthropic -d "Your prompt"  
ai-orchestrator run -m openai -d "Your task"
```

---

## Initialize New Project

```
ai-orchestrator init
```

---

## Common Workflows

### 1. Design → Implement

```
ai-orchestrator run -m openai "Design a user auth system" -o design.md  
ai-orchestrator run -m anthropic "Implement the user auth system from design.md"
```

### 2. Code → Review

```
ai-orchestrator ask -m anthropic "Write a login function" > login.py  
ai-orchestrator run -m moonshot "Review this code: $(cat login.py)"
```

### 3. Debug Session

```
ai-orchestrator test-api  
ai-orchestrator status  
ai-orchestrator ask -m anthropic -d "Test message"
```

---

## Common Errors

Error	Solution
<code>ModuleNotFoundError</code>	<code>source venv/bin/activate</code>
<code>command not found</code>	<code>source venv/bin/activate</code>
No API key configured	Check <code>.env</code> file
API call failed	Run <code>ai-orchestrator test-api</code>

## Full Documentation

- [USAGE\\_GUIDE.md](#) (USAGE\_GUIDE.md) - Complete CLI reference
- [README.md](#) (README.md) - Project overview
- [TROUBLESHOOTING.md](#) (TROUBLESHOOTING.md) - Error solutions

**Tip:** Bookmark this page for quick access! 