



# Model Reference Guide

This document provides detailed information about available models for each AI provider supported by AI Orchestrator.



## Quick Reference

Provider	Default Model	Recommended For	Access Level
OpenAI	gpt-4o-mini	Architecture, Planning	Standard API
Anthropic	claude-3-5-sonnet-20241022	Coding, Implementation	Standard API
Google	gemini-1.5-pro	Reasoning, Analysis	Standard API
Moonshot	moonshot-v1-8k	Code Review	Standard API



## OpenAI Models

### Available Models

Model	Context Window	Speed	Cost	Access
gpt-4o-mini 	128K	Fast	\$	Standard
gpt-4o	128K	Medium	\$\$\$	Standard
gpt-4-turbo	128K	Medium	\$\$\$	Standard
gpt-3.5-turbo	16K	Very Fast	\$	Standard

 = Default/Recommended

### Model Selection Guide

- **gpt-4o-mini** (Default): Best balance of speed, cost, and capability. Recommended for most users.
- **gpt-4o**: Latest flagship model with vision capabilities. Use when you need maximum capability.
- **gpt-4-turbo**: High capability with large context. Good for complex architecture tasks.
- **gpt-3.5-turbo**: Fastest and cheapest. Use for simple tasks or testing.

## Configuration

```
# In your .env file
OPENAI_MODEL=gpt-4o-mini
```

## Access Requirements

- Valid OpenAI API key
- No special access required for default models
- Usage limits based on API tier

## Common Errors

Error	Cause	Solution
404 model not found	Invalid model name	Use exact model name from list above
insufficient_quota	No credits	Add billing to OpenAI account
rate_limit_exceeded	Too many requests	Implement retry with backoff



## Anthropic (Claude) Models

### Available Models

Model	Context Window	Speed	Cost	Access
claude-3-5-sonnet-20241022 ★	200K	Fast	\$\$	Standard
claude-3-5-haiku-20241022	200K	Very Fast	\$	Standard
claude-3-opus-20240229	200K	Slower	\$\$\$\$	Standard

★ = Default/Recommended

## Model Selection Guide

- `claude-3-5-sonnet-20241022` (Default): Excellent coding capabilities, great balance of speed and quality.
- `claude-3-5-haiku-20241022`: Fastest Claude model. Good for quick coding tasks.
- `claude-3-opus-20240229` : Most capable but slower and more expensive. Use for complex implementations.

## Configuration

```
# In your .env file
ANTHROPIC_MODEL=claude-3-5-sonnet-20241022
```

## Access Requirements

- Valid Anthropic API key
- API access enabled in console
- Usage limits based on tier

## Common Errors

Error	Cause	Solution
invalid_api_key	Wrong or expired key	Regenerate key in Anthropic console
model_not_found	Invalid model name	Use exact model name with date suffix
overloaded_error	High demand	Retry with exponential back-off

## Google Gemini Models

### ⚠️ Important: Model Availability Varies

**Google Gemini model availability varies by region, account type, and API version.** Some models (like `gemini-2.0-flash`) may not be available to new users or in certain regions.

**Always check available models for your API key before configuring:**

```
# Using AI Orchestrator CLI
ai-orchestrator list-models gemini
```

## Available Models

Model	Context Window	Speed	Cost	Access
gemini-1.5-pro ⭐	2M	Medium	\$\$	Widely Available
gemini-1.5-flash	1M	Fast	\$	Widely Available
gemini-1.0-pro	32K	Fast	\$	Widely Available

★ = Default/Recommended (most stable and widely available)

## Model Selection Guide

- `gemini-1.5-pro` (Default): Most stable, widely available. Best for complex reasoning tasks.
- `gemini-1.5-flash`: Faster and cheaper. Good for simpler reasoning tasks.
- `gemini-1.0-pro`: Legacy model, but very stable. Good fallback option.

## Configuration

```
# In your .env file
GEMINI_MODEL=gemini-1.5-pro
```

## Access Requirements

- Valid Google AI API key (not GCP key)
- Get key from: <https://aistudio.google.com/apikey>
- Free tier available with rate limits

## 🔍 How to Check Available Models

Model availability varies by region and account. Use these methods to see what's available:

### Method 1: AI Orchestrator CLI (Recommended)

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

### Method 2: Python Script

```
#!/usr/bin/env python3
"""List available Gemini models for your API key."""

import google.generativeai as genai
import os

# Configure with your API key
api_key = os.getenv('GEMINI_API_KEY') or 'YOUR_API_KEY'
genai.configure(api_key=api_key)

print("Available Gemini Models for Text Generation:")
print("-" * 60)

for model in genai.list_models():
    if 'generateContent' in model.supported_generation_methods:
        name = model.name.replace('models/', '')
        print(f"\n📦 {name}")
        print(f"  Display Name: {model.display_name}")
        print(f"  Input Tokens: {getattr(model, 'input_token_limit', 'N/A')}")
        print(f"  Output Tokens: {getattr(model, 'output_token_limit', 'N/A')}")
```

### Method 3: Using the Helper Function

```
from ai_orchestrator.models.gemini_client import list_available_gemini_models

# List all available models
models = list_available_gemini_models('YOUR_API_KEY')
for m in models:
    print(f"{m['name']}: {m['description'][:50]}...")
```

### Common Errors

Error	Cause	Solution
404 model not found	Model not available for your account	Run <code>ai-orchestrator list-models gemini</code> to see available models
API_KEY_INVALID	Wrong key type	Use AI Studio key, not GCP key
RESOURCE_EXHAUSTED	Rate limit hit	Wait or upgrade to paid tier
not available to new users	Model restricted	Use <code>gemini-1.5-pro</code> instead

### ⚠️ Models with Limited Availability

Some newer models may not be available in all regions or to all accounts:

- `gemini-2.0-flash` - May not be available to new users
- `gemini-2.5-*` - Limited availability

If you encounter availability issues, use `gemini-1.5-pro` which is widely available.

### 💡 Model Name Format

Google Gemini models use simple names without prefixes:

- ✓ Correct: `gemini-1.5-pro`
- ✗ Wrong: `models/gemini-1.5-pro` (don't include the "models/" prefix)
- ✗ Wrong: `gemini/gemini-1.5-pro` (don't include redundant prefixes)

# Moonshot (Kimi) Models

## Available Models

Model	Context Window	Speed	Cost	Access
moonshot-v1-8k ★	8K	Fast	\$	Standard
moonshot-v1-32k	32K	Medium	\$\$	Standard
moonshot-v1-128k	128K	Slower	\$\$\$	Standard

★ = Default/Recommended

## Model Selection Guide

- moonshot-v1-8k (Default): Fast and efficient for code review tasks.
- moonshot-v1-32k : Larger context for reviewing bigger files.
- moonshot-v1-128k : Massive context for full codebase analysis.

## Configuration

```
# In your .env file
MOONSHOT_MODEL=moonshot-v1-8k
```

## Access Requirements

- Valid Moonshot API key
- Register at: <https://platform.moonshot.cn/>
- Free tier available

## How to Change Models

### Method 1: Environment Variables (Recommended)

Edit your .env file:

```
# Change OpenAI model
OPENAI_MODEL=gpt-4o

# Change Anthropic model
ANTHROPIC_MODEL=claude-3-opus-20240229

# Change Gemini model
GEMINI_MODEL=gemini-2.5-flash

# Change Moonshot model
MOONSHOT_MODEL=moonshot-v1-32k
```

## Method 2: Export in Shell

```
export OPENAI_MODEL=gpt-4o
export GEMINI_MODEL=gemini-2.0-flash
```

## Method 3: Programmatic

```
from ai_orchestrator.config import Config, ModelConfig

config = Config.load()
config.models.openai_model = "gpt-4o"
config.models.gemini_model = "gemini-2.0-flash"
```

## Pricing Considerations

### Cost Tiers

Tier	Models	Typical Use Case
\$ (Low)	gpt-4o-mini, claude-3-5-haiku, gemini-1.5-flash	Development, testing
\$\$ (Medium)	claude-3-5-sonnet, gemini-1.5-pro	Production workloads
\$\$\$ (High)	gpt-4o, gpt-4-turbo	High-value tasks
\$\$\$\$ (Premium)	claude-3-opus	Critical implementations

### Cost Optimization Tips

- Use defaults:** Default models are chosen for best cost/performance balance
- Task matching:** Let the router pick the right model for each task
- Context management:** Shorter prompts = lower costs
- Caching:** Avoid repeating identical requests

## Troubleshooting Model Errors

### “404 Model Not Found”

This usually means the model name is incorrect or not available for your account.

#### Common causes:

- Using model names not available in your region (e.g., `gemini-2.0-flash` may not be available to new users)
- Typos in model name

- Model was sunset by provider
- Using wrong model name format (e.g., `models/gemini-1.5-pro` instead of `gemini-1.5-pro`)

**Solution:**

1. Run `ai-orchestrator list-models gemini` to see available models
2. Check the exact model name in this document
3. Update your `.env` file with an available model
4. Restart your application

## Gemini-Specific 404 Errors

If you see an error like:

```
404 models/gemini-2.0-flash is not found
```

or:

```
gemini-2.0-flash is not available to new users
```

This means the model is not available for your account. **Use `gemini-1.5-pro` instead:**

```
# In your .env file
GEMINI_MODEL=gemini-1.5-pro
```

**Always check available models first:**

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

## “Access Denied” / “Insufficient Permissions”

**Common causes:**

- API key doesn't have access to the model
- Account needs billing setup
- Model requires special access

**Solution:**

1. Verify API key permissions in provider console
2. Add billing information if required
3. Request access for restricted models

## “Rate Limit Exceeded”

**Solution:**

1. Wait and retry with exponential backoff
2. Upgrade API tier for higher limits
3. Use a faster/cheaper model for testing



## Version History

Version	Date	Changes
2.2.0	Feb 2026	Changed Gemini default to gemini-1.5-pro (most stable/available). Added list-models command.
2.1.0	Feb 2026	Updated Gemini default to gemini-2.0-flash (gemini-1.5-flash deprecated)
2.0.0	Feb 2026	Updated defaults: gpt-4o-mini, gemini-1.5-flash
1.0.0	Initial	Original defaults: gpt-4, gemini-pro (now deprecated)