TermAI Quick Start Guide

What is TermAI?

TermAI is a production-ready, cross-platform CLI tool that brings AI assistance directly to your terminal. It supports multiple AI providers through OpenAI-compatible APIs with beautiful terminal UI, streaming responses, and conversation context.

Installation

Building from Source

```
cd /home/ubuntu/termai
make build
```

The binary `termai` will be created in the project directory.

Installing Globally

```
make install
# Or manually:
sudo cp termai /usr/local/bin/
```

First Time Setup

1. **Check the version:**
   `bash`
   `./termai --version`
   `# Output: termai version 1.0.0`
2. **View help:**
   `bash`
   `./termai --help`
3. **View default configuration:**
   `bash`
   `./termai config show`
4. **Edit configuration to add your API keys:**
   `bash`
   `./termai config edit`

Or manually edit `~/.termai/config.yaml`:
`yaml`
`default_profile: "abacus"`
`profiles:`
`- name: "abacus"`
`provider: "abacus"`

```
    endpoint: "https://api.abacus.ai/v1"
    api_key: "your-actual-api-key-here"  # ← Replace this
    model: "gpt-4"
    temperature: 0.7
    max_tokens: 2000
```

Usage Examples

1. One-Line Prompts

Ask quick questions directly from the command line:

```
# Using default profile
./termai "Explain quantum computing in simple terms"

# Using specific profile
./termai --profile ollama "What is the capital of France?"
./termai -p openai "Write a Python function to sort a list"
```

2. Interactive Chat Mode

Start a conversation with full context:

```
# Start chat with default profile
./termai chat

# Start chat with specific profile
./termai chat --profile ollama
```

Chat Commands:

- Type your message and press Enter
- `/exit` or `/quit` - Exit chat
- `/clear` - Clear conversation context
- `/help` - Show help
- `Ctrl+C` - Exit immediately

3. Profile Management

```
# List all profiles
./termai profiles list

# Add a new profile (interactive)
./termai profiles add

# Show profile details
./termai profiles show abacus

# Set default profile
./termai profiles set-default openai

# Remove a profile
./termai profiles remove old-profile
```

4. Configuration Management

```
# Show current configuration
./termai config show

# Show config file path
./termai config path

# Edit config in default editor
./termai config edit
```

Supported AI Providers

TermAI works with any OpenAI-compatible API:

1. **OpenAI** - https://api.openai.com/v1
   - Models: gpt-4, gpt-3.5-turbo, etc.

2. **Abacus.AI** - https://api.abacus.ai/v1
   - OpenAI-compatible API with various models

3. **Ollama** (Local) - http://localhost:11434/v1
   - Run models locally like llama3.1, mistral, etc.
   - No API key needed (use “ollama” as placeholder)

4. **Claude** (via OpenAI-compatible endpoint)
   - If available through OpenAI-compatible wrapper

5. **Custom Providers**
   - Any service that implements OpenAI-compatible API

Getting API Keys

Abacus.AI

1. Visit Abacus.AI (https://abacus.ai/)
2. Sign up/Login → Settings → API Keys
3. Generate new key

OpenAI

1. Visit OpenAI Platform (https://platform.openai.com/)
2. Sign up/Login → API Keys
3. Create new key

Ollama (Local)

1. Install from ollama.ai (https://ollama.ai/)
2. Pull model: `ollama pull llama3.1`
3. Start: `ollama serve`
4. No API key needed

Features

✨ **Multi-Provider Support** - Works with OpenAI, Abacus.AI, Ollama, and any OpenAI-compatible API

🎨 **Rich Terminal UI** - Beautiful markdown rendering, syntax highlighting, and styled output

⚡ **Streaming Responses** - Real-time streaming with smooth rendering

💬 **Interactive Chat** - Full conversation context in chat mode

🔧 **Profile Management** - Easy switching between different models and providers

🔒 **Secure** - API keys stored with proper file permissions (0600)

🌍 **Cross-Platform** - Works on Linux, macOS, and Windows

Project Structure

```
termai/
├── main.go                      # Entry point
├── cmd/                         # CLI commands
│   ├── root.go                  # Root command
│   ├── chat.go                  # Interactive chat
│   ├── config.go                # Config management
│   └── profiles.go              # Profile management
├── internal/
│   ├── config/                  # Configuration handling
│   │   └── config.go
│   ├── provider/                # AI provider implementations
│   │   ├── provider.go
│   │   └── openai_compatible.go
│   ├── ui/                      # Terminal UI components
│   │   ├── chat.go
│   │   ├── stream.go
│   │   └── formatter.go
│   └── context/                 # Conversation management
│       └── manager.go
├── README.md                    # Full documentation
├── QUICK_START.md               # This file
├── Makefile                     # Build automation
└── LICENSE                      # MIT License
```

Building for Different Platforms

```
# Build for Linux
make build-linux

# Build for macOS
make build-darwin

# Build for Windows
make build-windows

# Build for all platforms
make build-all
```

Shell Integration

Add convenient aliases to your `~/.bashrc` or `~/.zshrc`:

```
alias ai='termai'
alias chat='termai chat'
alias ai-local='termai -p ollama'
alias ai-gpt4='termai -p openai'
```

Then use:

```
ai "What is Rust?"
chat
ai-local "Tell me a joke"
```

Troubleshooting

“API key not found” Error

Make sure you’ve edited the config and replaced placeholder API keys:

```
termai config edit
```

Ollama Connection Error

Ensure Ollama is running:

```
ollama serve
```

Chat UI Issues

Try resizing your terminal window. The UI adapts to terminal size.

What’s Next?

- Try different AI models by creating new profiles
- Experiment with different temperature settings
- Use chat mode for longer conversations
- Integrate into your development workflow

Support

For issues or questions:
- Check the README.md (README.md) for detailed documentation
- Review the Troubleshooting section (README.md#troubleshooting)
- Open an issue on GitHub (if project is published)

Made with ❤️ by the TermAI team

Version: 1.0.0 | License: MIT