LiteLLM Basic Completion Example

Your First LLM API Call with LiteLLM

Complete Working Example

```python
# basic_completion.py
from litellm import completion
import json

def basic_llm_call():
    try:
        # Make a simple completion request
        response = completion(
            model="gpt-3.5-turbo",
            messages=[{
                "role": "system",
                "content": "You are a helpful assistant."
            }, {
                "role": "user",
                "content": "Explain machine learning in one sentence."
            }],
            temperature=0.7,
            max_tokens=100
        )
        return response
    except Exception as e:
        print(f"Error: {e}")
        return None
```

Expected Output

```
AI Response:
Machine learning is a subset of artificial
intelligence where computers learn to make
predictions or decisions by finding patterns in
data without being explicitly programmed for each
specific task.

Token Usage:
Prompt tokens: 32
Completion tokens: 28
Total tokens: 60
```

Error Handling Demonstration

```python
# Robust error handling
def safe_llm_call(model="gpt-3.5-turbo"):
    try:
        response = completion(
            model=model,
            messages=[ ... ],
            timeout=30
        )
        return response.choices[0].message.content
    except litellm.RateLimitError:
        return "Rate limit exceeded."
    except litellm.AuthenticationError:
        return "Authentication failed."
    except Exception as e:
        return f"An error occurred: {str(e)}"
```

Key Insights & Tips

- 💡 **Response Structure**: Access content via `response.choices[0].message.content`
- 💡 **Token Counting**: Monitor usage with `response.usage` for cost tracking.
- 💡 **Error Handling**: Always wrap API calls in try-catch blocks for robustness.
- 💡 **Timeout**: Set reasonable timeouts to prevent hanging requests.