

# Dynamic Integration of Real-Time Risk-Free Rate via FRED API

## 1 Overview

To ensure consistency and accuracy in financial modeling, the system retrieves the most recent 10-Year U.S. Treasury yield from the Federal Reserve Economic Data (FRED) service. This yield is treated as the risk-free rate and is used across valuation and forecasting models. The implementation supports a fallback constant in the event of connectivity or data access failure, ensuring robustness.

## 2 Environment Variable Configuration

The model requires a valid FRED API key, which can be stored either as a system-level environment variable or in a local `.env` configuration file. For Windows systems, the key is registered persistently as follows:

```
setx FRED_API_KEY "your_actual_fred_api_key_here"
```

## 3 Local Development Configuration

For development environments, the key may optionally be stored in a `.env` file in the project root:

```
FRED_API_KEY=your_actual_fred_api_key_here
```

This design supports flexibility across deployment contexts while ensuring separation between code and credentials.

## 4 System Constants

A default fallback risk-free rate is defined in the constants module:

```
# constants.py  
RISK_FREE_RATE = 0.04 # Used if FRED API is unavailable
```

## 5 Dynamic Retrieval from FRED

The system attempts to access the real-time 10-Year Treasury yield (DGS10) from FRED. If the API key is missing or the request fails, it reverts to the predefined constant.

```
# market_data.py
import os
from dotenv import load_dotenv
from fredapi import Fred
from constants import RISK_FREE_RATE

load_dotenv() # Load from .env if available

def get_10yr_treasury_yield():
    try:
        api_key = os.getenv("FRED_API_KEY")
        if not api_key:
            raise EnvironmentError("FRED_API_KEY not found.")
        fred = Fred(api_key=api_key)
        data = fred.get_series_latest_release("DGS10")
        return float(data.dropna().iloc[-1]) / 100
    except Exception as e:
        print(f"[WARN] Fallback rate used. Error: {e}")
        return RISK_FREE_RATE
```

## 6 Application Example

The retrieved yield can be used as a global constant across valuation functions:

```
from market_data import get_10yr_treasury_yield

rf = get_10yr_treasury_yield()
print(f"10-Year Treasury Yield: {rf:.2%}")
```

## 7 Security and Version Control

Sensitive credentials such as API keys are excluded from version control. The `.env` file is explicitly added to the `.gitignore`:

```
echo ".env" >> .gitignore
```

This practice ensures that production environments remain secure and credentials are not exposed through repository history.

## 8 Conclusion

The integration of the FRED API enables the system to adapt dynamically to changes in macroeconomic conditions while retaining operational integrity under degraded network conditions. This architecture combines real-time responsiveness with deterministic fallback behavior, which is critical in financial modeling applications where time consistency and numerical stability are paramount.