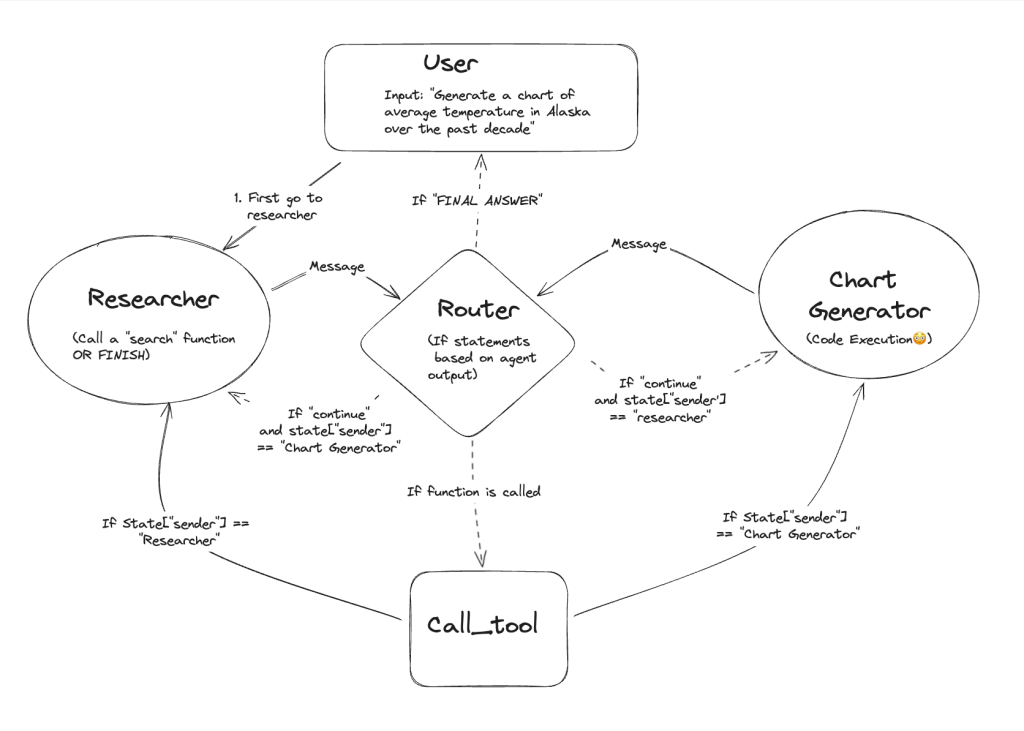
忙しい人のためのLangGraphまとめ

2024/9/10



目次

本紙の位置付け

- 以下の公式docの副読本として利用を想定します
    - https://langchain-ai.github.io/langgraph/

想定読者

- LLMを用いた複雑なアプリケーション開発に興味がある開発者
- LangChainユーザーで、より高度な制御と状態管理を求める方
- マルチエージェントシステムの設計・実装に携わる研究者や技術者
- AIによるワークフロー自動化に取り組む企業のエンジニア

用語辞書

- **LangGraph**: 状態管理可能なマルチエージェントアプリケーション構築ライブラリ
- **LangChain**: LLMアプリケーション開発のためのフレームワーク
- **エージェント**: 自律的に行動し、環境と相互作用するAIシステム
- **ヒューマンインザループ**: AI処理の途中で人間が介入する仕組み
- **ストリーミング**: データをリアルタイムで連続的に処理・送信する方式
- **マルチエージェントシステム**: 複数のエージェントが協調して動作するシステム
- **永続性**: データや状態を長期的に保存し、復元可能にする機能

基本概念の解説

- グラフ
  - ノードとエッジの高度な構造化と最適化技術
  - 動的グラフ生成とランタイムでの構造変更
  - グラフ理論に基づく最適化とパフォーマンス分析
- ステート
  - 分散システムにおける一貫性のある状態管理手法
  - 複雑な状態空間の効率的な表現と操作
  - 状態の永続化、バージョニング、ロールバック戦略
- ノード
  - 様々な処理単位の種類と特性
  - カスタムノードの設計原則とベストプラクティス
  - ノード間の依存関係管理と実行順序の最適化

1. イントロダクション

LangGraphの概要

- 状態管理可能なマルチエージェントアプリケーション構築ライブラリ
  - 複雑な対話型AIシステムの開発を大幅に簡素化
  - 非線形的な処理フローを直感的に設計可能
- 柔軟な状態管理機能
  - 長期的な文脈保持や複雑なタスク管理を実現
- 高度な並列処理と条件分岐
  - 効率的なリソース利用と複雑なロジックの実装をサポート

LangChainとの関係

- LangChainの発展版だが、独立して使用可能
  - LangChainのコア機能のみに依存し、軽量な実装
- LangChainの機能を拡張
  - より複雑なワークフローとエージェントの相互作用をサポート
- 完全な互換性
  - 既存のLangChainプロジェクトに容易に統合可能

主な特徴

- 1.サイクリックな分岐処理
  - 複雑な決定木や反復的なタスクの実装が容易
  - 動的なワークフロー調整が可能
- 2.永続性
  - 各ステップ後の状態を詳細に保存
  - エラーからの復旧やセッションの再開が容易
  - 長期的なタスク管理や複雑な対話履歴の維持が可能
- 3.ヒューマンインザループ機能
  - 実行中のプロセスに人間が介入するポイントを柔軟に設定
  - 半自動化されたワークフローの構築が可能
  - AIの判断に対する人間の監督と修正を容易に実装
- 4.ストリーミングサポート
  - リアルタイムでの出力生成と処理。生成過程を逐次表示
  - ユーザーエクスペリエンスの向上と即時フィードバックの実現

LangChainとの統合

- LangChainのツールやエージェントを簡単に組み込み可能
  - 既存のLangChainベースのプロジェクトを容易に拡張
- 独自の関数やクラスも容易に統合可能
  - カスタム機能の追加が柔軟に行える

基本的な使用例

- 複雑な検索と意思決定を行うインテリジェントなチャットボット
- ユーザー入力に基づき動的に処理フローを変更する適応型AIアシスタント
- 複数のAIモデルやツールを組み合わせたマルチステージのデータ分析パイプライン
- 長期的なゴール達成のための計画立案と実行を行う自律エージェントシステム

2. チュートリアル

クイックスタート

1. 基本的なチャットボットの作成 (7ステップ)
    - グラフの初期化とノードの定義
    - 基本的な対話ロジックの実装
    - エッジの接続とフロー制御
    - 初期状態の設定と実行
2. ツールの使用方法
    - 外部APIや検索エンジンなどのツールの統合
    - ツール使用結果に基づく条件分岐の実装
    - エラーハンドリングとリトライロジックの追加
3. メモリ機能の追加
    - チェックポインター機能を用いた状態保存
    - 長期記憶と短期記憶の実装
    - コンテキストに応じた記憶の取り出しと更新

クイックスタート（続き）

4. ヒューマンインザループの実装
    - 特定条件下で人間の介入を要求する仕組み
    - ユーザー入力に基づくグラフの動的な変更
    - 人間の判断を AI の処理フローに統合する方法
5. マニュアルステート更新
    - 実行状況の手動更新方法
    - デバッグや特殊なケース処理への応用
    - 状態の一貫性を保ちつつ手動介入を行う技術

クイックスタート（続き）

6. カスタムステート
    - メッセージ履歴以外の情報追加（ユーザープロファイル、タスク進捗など）
    - 複雑な状態管理の実装例
    - カスタムオブジェクトのシリアライズと永続化
7. タイムトラベル機能
    - 過去の実行状態への巻き戻し方法
    - エラー発生時の状態復元技術
    - A/Bテストや「what-if」分析への応用

応用例

- カスタマーサポートボット
  - 旅行予約システムの例を用いた複雑な対話シナリオの実装
  - ユーザー要求の理解、適切なツールの選択、応答生成のフロー
  - 多段階の予約プロセスと例外処理の管理
- プロンプト生成
  - ユーザーリクエストからの動的なプロンプト自動生成
  - コンテキストに応じたプロンプトの最適化技術
  - メタプロンプティング技術の活用と自己改善メカニズム
- コードアシスタント
  - コード生成と自己修正機能の実装
  - 静的解析ツールとの連携によるコード品質向上
  - 複数言語対応と言語固有の最適化テクニック

応用例（続き）

- 適応型エージェント
  - 入力に応じて処理を分岐させる仕組み
  - ユーザーの意図理解と適切なサブグラフの選択方法
  - 学習と適応のメカニズムの実装
- マルチエージェントシステム
  - 複数エージェントの協調動作の実装
  - エージェント間の通信プロトコルと役割分担の設計
  - 競合解決と合意形成アルゴリズムの統合
- プランニングエージェント
  - タスク分解と実行計画の立案プロセス
  - 動的な計画修正と再実行の仕組み
  - 長期目標と短期行動のバランシング技術

応用例（続き）

- LLMコンパイラー
  - 依存関係解決と並列実行の最適化
  - 大規模タスクの効率的な処理方法
  - 動的コード生成と実行の安全性確保
- 自己発見エージェント
  - 新しいスキルの自動学習メカニズム
  - 継続的な性能向上と適応能力の実現方法
  - メタ学習アルゴリズムの実装と評価

3. How-toガイド

コントロール機能

- 並列ノード実行
  - 複数ノードの同時実行とデータ集約の詳細な方法
  - スレッド管理と同期処理の実装テクニック
  - 並列処理のパフォーマンスチューニングとボトルネック解消
- 条件分岐
  - 複数の条件に基づく複雑な分岐処理の実装
  - 動的な条件評価と分岐先の決定方法
  - 確率的分岐と強化学習を用いた最適経路選択

コントロール機能

- マップリデュース処理
  - 大規模データの並列処理と結果の効率的な集約方法
  - カスタムマップ関数とリデュース関数の設計ガイドライン
  - 分散処理環境での効率的なデータ分割と結果マージ
- 再帰制限
  - グラフの無限ループ防止メカニズムの詳細
  - 再帰深度の動的調整と安全性確保の方法
  - エッジケースの検出と自動回復メカニズム

永続性

- メモリサーバーの実装
  - インメモリ、SQLite、PostgreSQLの詳細な設定と使用法
  - パフォーマンスとスケーラビリティの考慮点
  - データの一貫性と耐障害性の確保
- カスタムメモリサーバーの作成
  - MongoDB、Redisなどの実装例と最適化テクニック
  - 分散システムにおける状態管理の課題と解決策
  - カスタムシリアライゼーションと圧縮アルゴリズムの実装

永続性

- 会話履歴の管理
  - 高度なフィルタリングアルゴリズムの実装
  - 効率的な履歴削除と要約生成のテクニック
  - プライバシーを考慮した履歴の匿名化と暗号化
- スレッド間での状態共有
  - ユーザーIDに基づく安全な情報共有メカニズム
  - 競合状態の回避と一貫性維持の方法
  - 分散ロックと楽観的並行制御の実装

ヒューマンインザループ

- ブレークポイントの設定
  - 静的および動的な中断ポイントの高度な使用法
  - 条件付きブレークポイントの実装テクニック
  - リモートデバッギングとライブ監視の統合
- グラフ状態の確認と編集
  - 実行状況の詳細な可視化方法
  - 安全な手動更新プロセスとロールバック機能
  - 差分ベースの状態更新と変更の追跡

ヒューマンインザループ

- エラー呼び出しのレビュー
  - 詳細なエラーログと診断情報の生成
  - 自動エラー分析と修正提案システムの実装
  - エラーパターンの学習と予防的措置の実装

ストリーミング

- LLMトークン出力のストリーミング
  - 低遅延でのリアルタイムテキスト生成表示技術
  - トークン単位の処理と表示の最適化
  - 進捗表示とキャンセル機能の実装

ストリーミング（続き）

- 複数ストリーミングモードの設定
  - 全体状態と更新差分の効率的な取得方法
  - カスタムストリーミングプロトコルの設計と実装
  - マルチモーダルストリーミング（テキスト、音声、画像）の統合
- サブグラフのストリーミング
  - 部分的な実行状況の高精度モニタリング技術
  - 大規模グラフにおける効率的な状態追跡方法
  - リアルタイムグラフ可視化と対話的な調整機能

ツール呼び出し

- 組み込みツールノードの使用
  - 外部ツールの簡単な統合と拡張方法
  - カスタムツールノードの作成ガイドライン
  - ツール間の連携とパイプライン構築テクニック
- エラー処理
  - ツール呼び出し時の高度な例外処理技術
  - 自動リトライとフォールバックメカニズムの実装
  - グレースフルデグラデーションと部分的成功の処理
- 複数ツールの管理
  - 様々なツールの効率的な利用と優先順位付け
  - ツール間の相互作用と結果の統合方法
  - 動的なツール選択とコンテキストに応じた最適化

4. コンセプチュアルガイド

エージェント的アプリケーションの定義

- LLMを使用した制御フローの決定メカニズム
  - 言語モデルによる動的な判断と行動選択
  - コンテキスト理解と多段階推論の実現方法
- エージェント性の連続的な尺度と評価方法
  - 自律性、適応性、目標指向性の定量化
  - マルチエージェントシステムにおける協調と競争の評価指標

LangGraphの利点

- 制御性の向上
  - 複雑なワークフローの簡潔な表現方法
  - 動的なグラフ構造の変更とフロー制御
  - 細粒度の実行制御とモニタリング機能
- ヒューマンインザループのファーストクラスサポート
  - 柔軟な介入ポイントの設計と実装
  - 人間の知識とAIの能力を効果的に組み合わせる手法
  - インタラクティブな学習と改善のサイクル
- ストリーミングのネイティブサポート
  - 大規模処理における効率性と応答性の向上
  - リアルタイムフィードバックと進捗モニタリング
  - 非同期処理と並行実行の最適化技術

一般的なエージェントパターン

- リアクティブエージェント
  - 即時応答型システムの設計と実装
  - 環境変化への迅速な適応メカニズム
  - イベント駆動型アーキテクチャとの統合
- プラン＆実行エージェント
  - 長期的目標達成のための戦略立案と実行
  - 階層的タスク分解と動的計画法の応用
  - 不確実性下での意思決定と計画修正
- 反省と批評を行うエージェント
  - 自己評価と継続的改善メカニズムの実装
  - メタ認知能力の模倣と学習転移の促進
  - フィードバックループと強化学習の統合

5. リファレンス

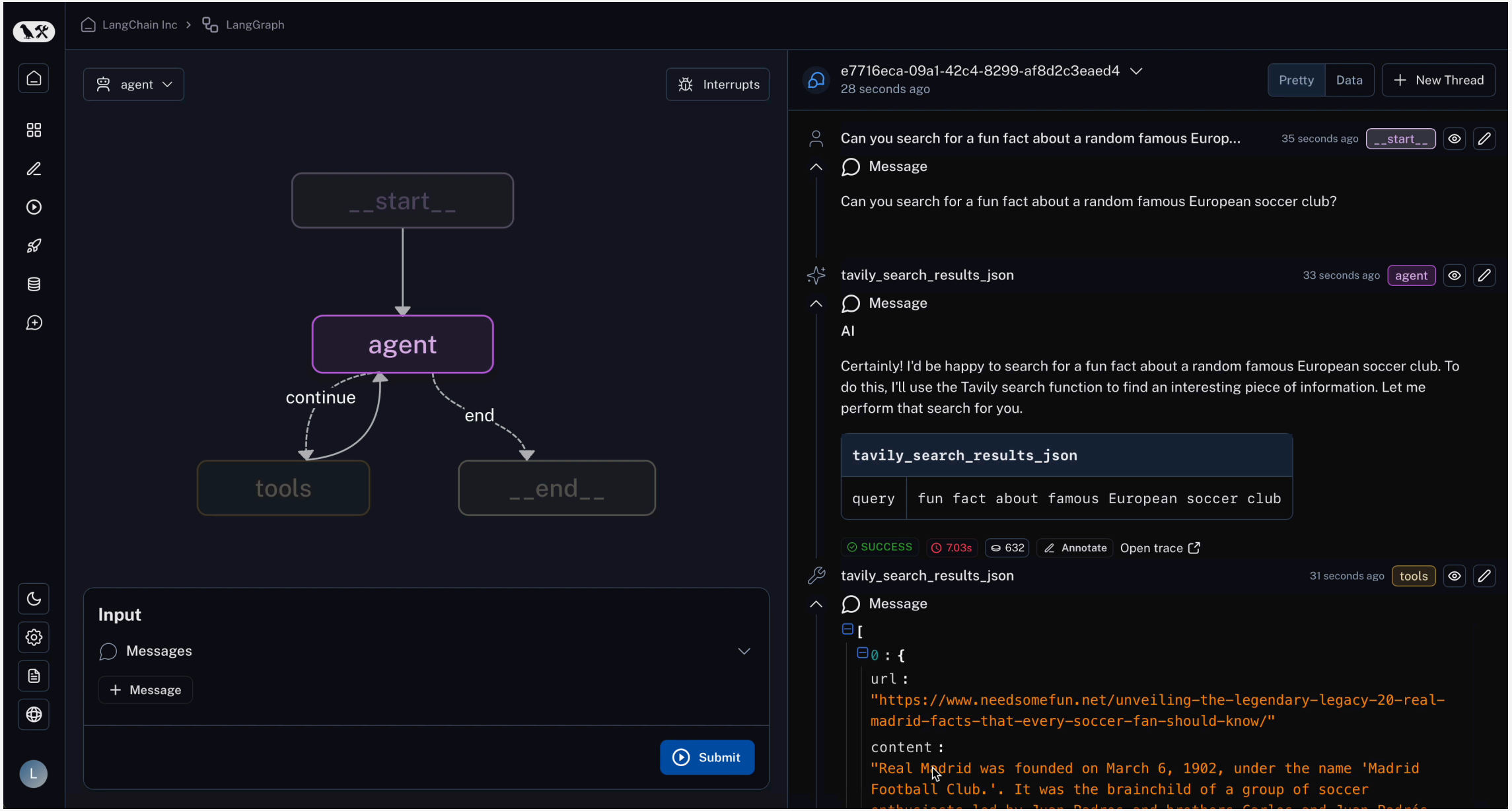
各クラスの詳細な説明

- メソッド、プロパティの完全な一覧と使用例
  - 主要クラスの全メソッドとプロパティの詳細解説
  - 複雑な使用シナリオとエッジケースの対処法
- 内部実装の詳細と最適化のヒント
  - パフォーマンスクリティカルな部分の実装詳細
  - メモリ管理と計算効率化のテクニック
- 拡張ポイントとカスタマイズ可能な箇所の解説
  - プラグインアーキテクチャとフックポイントの説明
  - カスタム拡張の設計と実装ガイドライン

APIドキュメント

- 全パラメータの詳細な説明と有効な値の範囲
  - 各APIエンドポイントの完全な仕様
  - 入力バリデーションと型チェックのルール
- 戻り値の構造と意味の解説
  - 全ての可能な戻り値パターンとその解釈
  - エラーコードと例外の包括的なリスト
- 包括的なエラー処理ガイドラインと全例外クラスの説明
  - 一般的なエラーシナリオとその対処法
  - カスタム例外の作成と使用方法
- バージョン間の互換性情報と移行ガイド
  - メジャーバージョン間の変更点と移行手順
  - 非推奨機能と将来の削除予定機能の一覧

6. LangGraph Cloud (ベータ版)


LangChain Inc
LangGraph
agent
Interrupts
__start__
agent
continue
end
tools
__end__
Input
Messages
Message
Submit
e7716eca-09a1-42c4-8299-af8d2c3eaed4
28 seconds ago
Pretty
Data
New Thread
Can you search for a fun fact about a random famous Europ...
35 seconds ago
__start__
Message
Can you search for a fun fact about a random famous European soccer club?
tavily_search_results_json
33 seconds ago
agent
Message
AI
Certainly! I'd be happy to search for a fun fact about a random famous European soccer club. To do this, I'll use the Tavily search function to find an interesting piece of information. Let me perform that search for you.
tavily_search_results_json
query
fun fact about famous European soccer club
SUCCESS
7.03s
632
Annotate
Open trace
tavily_search_results_json
31 seconds ago
tools
Message
[
0 : {
url :
"https://www.needsomefun.net/unveiling-the-legendary-legacy-20-real-madrid-facts-that-every-soccer-fan-should-know/"
content :
"Real Madrid was founded on March 6, 1902, under the name 'Madrid Football Club.'. It was the brainchild of a group of soccer

デプロイメント機能

- グラフのクラウドへの簡単なデプロイプロセス
  - ワンクリックデプロイメントとCI/CD統合
  - 環境変数とシークレット管理の最適プラクティス
- 環境設定とスケーリングオプションの高度な調整方法
  - オートスケーリングルールとリソース最適化
  - カスタムランタイムとコンテナ化のサポート
- セキュリティ設定と認証メカニズムの実装ガイド
  - 多層防御戦略とデータ暗号化の実装
  - IDおよびアクセス管理（IAM）の詳細設定

管理ツール

- 実行状況のリアルタイムモニタリングダッシボード
  - カスタマイズ可能な監視メトリクスとアラート設定
  - 異常検知と自動障害回復メカニズム
- パフォーマンス最適化のための詳細な分析ツール
  - ボトルネック特定とリソース使用効率の分析
  - A/Bテストとパフォーマンスベンチマーク機能
- 自動スケーリングとロードバランシングの設定
  - 予測型スケーリングアルゴリズムの実装
  - マルチリージョン展開と地理的負荷分散

統合機能

- 他のクラウドサービスとの高度な連携方法
  - 主要クラウドプロバイダーのAIサービスとの統合
  - サーバーレスアーキテクチャとの融合
- カスタムプラグインの開発とデプロイプロセス
  - プラグインSDKとAPI仕様の詳細
  - セキュリティ審査とパフォーマンス認証プロセス
- APIゲートウェイとの統合によるマイクロサービスアーキテクチャの実現
  - サービスメッシュとAPIバージョニング戦略
  - トラフィック管理と高度なルーティング機能

まとめ

まとめ

- LangGraphは、複雑なAIアプリケーション開発を大幅に簡素化する強力なツール
- 主要な特徴：
    - サイクリックな処理能力
    - 高度な永続性機能
    - シームレスなヒューマンインザループ統合
    - 効率的なストリーミングサポート
- 豊富なチュートリアルと詳細なガイドで、段階的な学習と高度な実装が可能
- 高度なコントロール機能と柔軟な拡張性で、多様なユースケースに対応
- クラウド版（ベータ）でスケーラブルなデプロイメントと運用管理を実現