

Salesforce.Slack-Dev-201.v2026-08-04.q86

試験コード：	Slack-Dev-201
試験名称：	Salesforce Certified Slack Developer
認証ベンダー：	Salesforce
無料問題の数：	86
バージョン：	v2026-08-04
ページの閲覧量：	104
問題集の閲覧量：	903

<https://www.jpnsiken.com/shiken/Salesforce.Slack-Dev-201.v2026-08-04.q86.html>

質問: 1

Slackアプリの配布先を単一のワークスペースに限定するのが最も適切なシナリオはどれですか？

- A. アプリが国際的なデータセキュリティおよびプライバシー規制に準拠していることを保証するため
- B. より広範な市場分析とフィードバックのために、アプリの機能を複数の組織で活用する。
- C. アプリに、標準的なSlackユーザーであれば誰でも適用できる汎用的な機能が含まれている場合
- D. アプリを広くリリースする前に、管理された環境でテストすることが目的の場合

正解: ([正解を表示します](#))

アプリをより広範囲にリリースする前に、管理された環境でテストすることが目的の場合、Slack アプリの配布先を単一のワークスペースに限定することが最も適切な場合があります。これにより、開発者はアプリのパフォーマンスを監視し、フィードバックを収集し、より広範囲にリリースする前に必要な調整を行うことができます。

他の選択肢ではダメなのか？

配布先を単一のワークスペースに限定すると、複数の組織にわたるより広範な市場分析が促進されません。

*標準的なSlackユーザーであれば誰でも利用できる汎用的な機能は、単一のワークスペースに限定するのではなく、より広範囲に配布することでメリットが得られるでしょう。データセキュリティとプライバシーは重要ですが、通常、アプリを単一のワークスペースに限定する主な理由ではありません。

質問: 2

チームコラボレーションのためにSlackアプリを統合する場合、より生産性の高い職場環境を実現するというSlackの目標に最も合致するアプローチはどれでしょうか？

- A. アプリはチームの進捗状況を1時間ごとにメインチャンネルに自動的に投稿します
- B. アプリは、チームメンバーの進捗状況を1時間ごとにマネージャーに直接メッセージで送信します。

C. チームメンバー全員が、毎日終業時にメインチャンネルに個別の最新情報を投稿することが義務付けられています。

D. チームメンバーが専用のプロジェクトチャンネルで進捗状況を自主的に共有できる機能

正解: [\(正解を表示します\)](#)

チームメンバーが専用のプロジェクトチャンネルで進捗状況の更新を自主的に共有できる機能。チームメンバーが専用のプロジェクトチャンネルで更新を自主的に共有できるようにすることで、過剰な通知や強制的な投稿でメンバーを圧倒することなく、コラボレーション環境を促進できます。

他の選択肢ではダメなのか？

このアプローチは、快適な作業環境を維持しながら生産性を向上させるというSlackの目標に合致しています。

*1時間ごとの自動更新はメインチャンネルを煩雑にし、注意散漫の原因となる可能性があります。

*メインチャンネルでの個別の投稿は繰り返しが多く、進捗状況を効率的に伝えられない可能性があります。マネージャーに毎時間ダイレクトメッセージを送ることは負担が大きく、チーム全体の進捗状況を正確に反映しない可能性があります。

質問: 3

あなたは、ユーザーに代わってチャンネルにメッセージを投稿する必要があるSlackアプリを開発しています。

この機能を安全かつ適切に有効化するには、どのOAuthスコープを要求すればよいでしょうか？

A. ユーザー:読む

B. チャンネル:読み取り

C. チャット:書き込み

D. ファイル:書き込み

正解: C ([コメントを発表する](#))

chat:write スコープは、チャンネルにメッセージを投稿する必要がある Slack アプリに必要です。このスコープにより、アプリはユーザーとしてメッセージを送信できるようになり、権限を逸脱することなくアプリの機能に沿った操作が可能になります。

他の選択肢ではダメなのか？

*Users:read は、ユーザーの情報を閲覧するためのアクセス権を与えますが、これはメッセージを投稿するためには必要ありません。

*Channels:read only は、アプリがチャンネルを表示することを許可しますが、メッセージを投稿することは許可しません。

*Files:writeはファイルのアップロードと管理のためのものであり、チャンネルでのメッセージ送信とは関係ありません。

質問: 4

あなたは、セキュリティポリシーにより外部からのインターネット接続を許可していない組織向けに、Slackアプリを開発しています。

このアプリにとって、Slackからイベントを受信する最も適切な方法はどれでしょうか？

- A. 公開アクセス可能なエンドポイントを備えたHTTPベースのイベント配信
- B. WebSocketを使用したソケットモード
- C. VPNを使用したHTTPベースのイベント配信
- D. IPホワイトリストによるHTTPベースのイベント配信

正解: ([正解を表示します](#))

WebSocket を使用したソケット モード WebSocket を使用したソケット モードは、受信接続に厳しい制限がある環境に最適です。このモードでは、アプリが組織のネットワーク内から開始された WebSocket 接続を介して Slack からイベントを受信できます。

他の選択肢ではダメなのか？

*HTTPベースのイベント配信には、一般にアクセス可能なエンドポイントが必要ですが、これは組織のセキュリティポリシーと矛盾します。

HTTPベースのイベント配信にVPNを使用する場合でも、受信接続の管理が必要となり、組織のセキュリティ上の制約によっては実現不可能な場合があります。

*IPホワイトリストを使用したHTTPベースのイベント配信では、受信接続のためにネットワークを開放する必要がありますが、組織のポリシーによってはそれが許可されない場合があります。

質問: 5

Slackアプリを構築する際に、Bolt開発フレームワークを使用するのが最も適切なのはどのような場合ですか？

- A. 既存の機能に頼らず、ゼロから高度にカスタマイズされたSlackクライアントを開発する必要がある場合
- B. 迅速かつ効率的な開発が求められるアプリを構築する場合、一般的なSlack機能のための事前構築済みの抽象化とミドルウェアを活用します。
- C. Slack API呼び出しの詳細な手動処理を必要とする低レベルプロトコルおよび通信レイヤーを作成する場合
- D. アプリ開発において、SlackのWeb APIまたはイベントAPIとの連携が不要な場合

正解: B ([コメントを発表する](#))

迅速かつ効率的な開発が求められるアプリを構築する場合、一般的な Slack 機能のための事前構築済みの抽象化とミドルウェアを活用するには、Bolt フレームワークが最適です。Bolt フレームワークは、事前構築済みの抽象化とミドルウェアを提供するため、迅速かつ効率的な開発が求められる Slack アプリの構築に理想的です。これにより、開発者は低レベルの API 呼び出しを処理するのではなく、アプリ独自の機能の構築に集中できるため、開発プロセスが高速化されます。

他の選択肢ではダメなのか？

*Boltは、Slackの既存の機能を活用するように設計されており、クライアントをゼロから構築するものではありません。

*Boltは低レベルのプロトコルや通信レイヤーの処理を不要にすることで、より効率的な開発体験を提供します。

*Boltは、SlackのWeb APIおよびイベントAPIとの連携を容易にするために特別に設計されています。

質問: 6

あるチームが、従業員の業績評価のためのSlackアプリを開発している。このアプリは、複数のチームメンバーから詳細なフィードバックを収集する必要がある。

この目的に最も適したSlackのサーフェスはどれでしょうか？

- A. モーダルを使用して、構造化されたフォームを通じて詳細なフィードバックを収集します。
- B. 共通チャンネルにフィードバックを求めるメッセージを投稿する
- C. 各チームメンバーにフィードバックを求める自動ダイレクトメッセージを送信する
- D. ボタン付きのインタラクティブなメッセージを作成し、迅速なフィードバック応答を実現します。

正解: ([正解を表示します](#))

モーダルウィンドウを使用して、構造化されたフォームを通じて詳細なフィードバックを収集しましょう。モーダルウィンドウは、ユーザーが構造化された整理された方法で情報を入力できる集中的なインターフェースを提供するため、詳細なフィードバックを収集するのに最適です。これは、詳細かつ構造化された入力が求められる業績評価において特に役立ちます。

他の選択肢ではダメなのか？

共通チャンネルでのメッセージは、プライバシーが確保されておらず、詳細なフィードバックを行うための構造も欠けている。

自動生成されたダイレクトメッセージは、モーダルメッセージに比べて、魅力や構成が劣る場合があります。

*ボタン付きのインタラクティブメッセージは、よりシンプルで詳細度の低い回答に適しています。

質問: 7

貴社のSlackアプリは、チーム間のプロジェクト管理を効率化するように設計されています。どのような場合に、Slack Connectを使用して外部と共有されるチャンネルのサポートを追加することを検討すべきでしょうか？

- A. プロジェクト管理アプリが単一組織内での内部利用のみを目的としている場合
- B. アプリが共有チャンネルで外部パートナー、クライアント、またはベンダーが関わるプロジェクトでのコラボレーションを促進する場合
- C. アプリに、社内チームメンバーのみを対象とした非常に機密性の高い情報が含まれている場合
- D. アプリが、無関係な多数の視聴者に向けて一般的な告知を放送するために使用される場合

正解: ([正解を表示します](#))

アプリが外部パートナー、クライアント、またはベンダーが関わるプロジェクトで共有チャンネルでのコラボレーションを促進する場合、Slack Connect で外部共有チャンネルのサポートを追加することは適切です。これにより、異なる組織間でのシームレスなコミュニケーションと調整が促進され、共有チャンネルでのコラボレーション体験が向上します。他の選択肢ではダメなのか？

*アプリが社内利用のみを目的としている場合は、外部共有チャンネルのサポートは不要です。

*Slack Connectを通じて機密性の高い社内情報を外部の関係者と共有することは、通常は推奨されません。

*大規模な、関連性のない聴衆に向けて一般的な告知を配信する場合、通常、Slack Connect が提供するコラボレーション機能は必要ありません。

質問: 8

あなたのチームは、リモートチーム向けの毎日のスタンドアップミーティングを支援する Slack アプリを開発しています。このアプリでは、チームメンバーが日々の進捗状況を簡単に報告できる必要があります。

これらの更新情報を送信するのに最も適した機能はどれでしょうか？

- A. チームメンバーがダイレクトメッセージで最新情報を送信できるボットを作成する
- B. チームメンバーが更新情報を入力できるように、Slack チャンネルにリンクされた共有ドキュメントを設定します。
- C. 各チームメンバーが最新情報を投稿できる共通チャンネルにインタラクティブなメッセージを実装する
- D. チームメンバーが毎日のスタンドアップミーティングの進捗状況を送信するために実行できるスラッシュコマンドを使用します。

正解: ([正解を表示します](#))

チームメンバーが毎日のスタンドアップミーティングの進捗状況を送信するために実行できるスラッシュコマンドを使用してください。スラッシュコマンドを使用すると、チームメンバーはスタンドアップミーティングの進捗状況を迅速かつ簡単に送信できます。この方法により、チームメンバーは効率的に構造化された形式で進捗状況を入力できるため、プロセスが効率化され、リモートチームにとって管理しやすくなります。

他の選択肢ではダメなのか？

ボットへの直接メッセージは、チーム全体の可視性を損なう可能性があります。

個々の更新にインタラクティブなメッセージを使用すると、チャンネルが煩雑になり、全員の意見を追跡するのが難しくなる可能性があります。

*外部ツールに依存しており、Slack のインタラクティブな機能は活用していません。

質問: 9

Slack アプリでは、認証後に OAuth トークンを保存する必要があります。

これらのトークンを安全に保管するための最善の方法は何ですか？

- A. トークンをサーバー上のローカルファイルに保存して、簡単にアクセスできるようにします。
- B. トークンはアクセス制限のある安全な暗号化データベースに保管してください。
- C. 簡便化のため、トークンをアプリのソースコードに直接埋め込む
- D. 利便性のために、ユーザーのデバイスにトークンを保存するためにクッキーを使用します。

正解: ([正解を表示します](#))

トークンは、アクセスが制限された安全な暗号化データベースに保管してください。OAuth トークンをアクセスが制限された安全な暗号化データベースに保存することは、ベストプラクティスです。この方法により、トークンが不正アクセスから保護され、トークンの漏洩や盗難のリスクが軽減されます。

他の選択肢ではダメなのか？

*サーバーが侵害された場合、トークンをローカルファイルに保存することは安全ではない可能性があります。

ソースコードにトークンを埋め込むことは、特にコードベースがアクセス可能または共有されている場合、非常に危険です。

*機密性の高いトークンをCookieに保存すると、クライアント側の脆弱性によりセキュリティ上のリスクが生じる可能性があります。

質問: 10

Slackアプリの開発プロセスにおいて、Slack開発者ツール (SDT) を活用するのが適切なのはどのような場合ですか？

- A. メッセージ形式のデバッグや、Slack内で直接APIリクエストとレスポンスをテストする場合
- B. 広範なバックエンドデータベース管理およびサーバー構成調整用
- C. Slackアプリが連携する可能性のある、Slackとは関係のない外部APIを作成および管理するためのもの
- D. モバイルまたはデスクトップアプリケーション向けの高忠実度グラフィカルユーザーインターフェースを設計する場合

正解: ([正解を表示します](#))

メッセージ形式のデバッグや、Slack環境内でのAPIリクエストとレスポンスのテストを行う場合、Slack開発者ツールは特に役立ちます。Slackアプリの動作をリアルタイムで検査・改善できるため、シームレスな統合と機能性を確保する上で非常に重要です。

他の選択肢ではダメなのか？

*SDTは、バックエンドのデータベース管理やサーバー構成には特化していません。

*SDTはSlackのAPIでを使用することを想定して設計されており、外部APIの管理を目的としたものではありません。

*SDTは、モバイルまたはデスクトップアプリケーション向けの高精度GUI設計を目的としたものではなく、Slack固有の開発タスクを対象としています。

質問: 11

Enterprise Grid Sandboxの主要機能の中で、アプリケーションの開発とテストにとって非常に重要なものは何ですか？

- A. 開発者がリアルタイムテストのためにライブユーザーデータにアクセスして変更することを可能にします
- B. テストされていないコードを本番環境のエンタープライズグリッド環境に直接デプロイして、即座にフィードバックを得ることができます。
- C. サンドボックス環境では、テスト対象が基本的なSlack機能に限定され、エンタープライズ固有の機能は除外されます。
- D. エンタープライズグリッド環境のデータと構造を含むレプリカを提供し、隔離されたテストを可能にします。

正解: ([正解を表示します](#))

Enterprise Grid Sandboxは、実際のEnterprise Grid環境（データと構造を含む）のレプリカを提供し、隔離されたテストを可能にします。これは、アプリの隔離された安全なテストに不可欠です。これにより、開発者は実際のユーザーデータや運用に影響を与えるリスクを負うことなく、本番環境を忠実に再現した環境でテストを実行できます。

他の選択肢ではダメなのか？

テスト目的で実際のユーザーデータにアクセスしたり変更したりすることは危険であり、データ漏洩やシステム障害につながる可能性があります。

サンドボックス環境は、包括的なテストを行うための企業固有の機能を含むように設計されています。

*テストされていないコードを本番環境に直接デプロイすることは危険であり、推奨される方法ではありません。

質問: 12

貴社は、従業員からのフィードバックプロセスを効率化するために、Slackアプリを開発しています。このアプリは、フィードバックの送信と確認を容易にする機能を備えている必要があります。

この目的において、Slackのアーキテクチャに最も適した機能はどれでしょうか？

- A. 透明性を確保するため、アプリを設定してフィードバックの送信先を社内チャンネルに振り分けてください。
- B. 従業員がどのチャンネルでも直接フィードバックを投稿できるスラッシュコマンドを実装する
- C. 外部のアンケートツールを使用し、フィードバック送信用のリンクをSlackに投稿してください。
- D. モーダルを使用して非公開のフィードバック送信フォームを作成し、結果を指定された人事チャンネルに送信する。

正解: ([正解を表示します](#))

モーダルウィンドウを使用して非公開のフィードバック送信フォームを作成し、結果を人事部門指定のチャンネルに送信します。モーダルウィンドウを使用して非公開のフィードバックを送信することで、機密性と使いやすさが確保されます。結果を人事部門指定のチャンネルに送信することで、プライバシーを尊重しながら、フィードバックを体系的に確認・処理できます。

他の選択肢ではダメなのか？

社内全体に共有するチャンネルにフィードバックを投稿すると、プライバシーの問題が発生する可能性があります、率直なフィードバックを阻害する恐れがあります。

*外部ツールを使用するため、Slack内でのシームレスな統合と相互作用が損なわれます。

*どのチャンネルにフィードバックを投稿しても、機密性が損なわれたり、フィードバックの収集が混乱したりする可能性があります。

質問: 13

あなたのチームは、会議室の予約ができるSlackアプリを開発しています。

ユーザーがSlack経由で会議室を予約できるようにする最も効率的な方法は何ですか？

- A. ユーザーが予約リクエストを投稿できる専用チャンネルを作成する
- B. ユーザーが部屋を予約するために記入できるモーダルフォームを実装する
- C. スラッシュコマンドを使用して部屋の予約プロセスを開始します
- D. 空室オプションのあるユーザーにダイレクトメッセージを送信します。

正解: C ([コメントを发表する](#))

スラッシュコマンドを使用して会議室予約プロセスを開始します。スラッシュコマンドは、ユーザーが操作して会議室予約プロセスを迅速に開始できる方法です。現在の画面から離れることなく会議室を予約したいユーザーにとって効率的です。スラッシュコマンドは、モーダルフォームやその他のインタラクティブな要素をトリガーして予約を完了させることができます。

他の選択肢ではダメなのか？

*これにより、構造化されていないリクエストが発生し、予約管理が困難になる可能性があります。

*モーダルフォームから始める場合、それを開くための追加の手順が必要になりますが、スラッシュコマンドを使用するとこの手順を簡略化できます。

*これは積極的なものではなく、ユーザーが選択肢付きのダイレクトメッセージを待つことに依存しています。

質問: 14

お使いのSlackアプリは、メッセージの投稿やチャンネル情報へのアクセスなど、幅広い機能を実現するためにSlackのWeb APIと連携するように設計されています。

安全かつ適切なアクセスを確保するために、アプリはどのタイプのトークンを要求すべきでしょうか？

- A. ボットユーザーOAuthトークン
- B. ユーザーOAuthトークン

C. ワークスペースアクセストークン

D. クラシックボットトークン

正解: ([正解を表示します](#))

ボットユーザーOAuthトークン ボットユーザーOAuthトークンは、SlackのWeb APIの様々な機能と連携する必要のあるアプリに最適です。このトークンタイプを使用すると、セキュリティのベストプラクティスを維持しながら、メッセージの投稿やチャンネル情報へのアクセスなど、ボットの範囲内でアクションを実行できます。

他の選択肢ではダメなのか？

*ユーザーOAuthトークンは、特定のユーザーの権限でアクセスを提供しますが、このユースケースでは不要であり、セキュリティも低下する可能性があります。

*ワークスペースアクセストークンは、Slackアプリの標準トークンタイプではありません。

*クラシックボットトークンは、従来のカスタム統合向けであり、新しいSlackアプリには推奨されません。

質問: 15

従業員の休暇申請を処理するためのSlackアプリが開発されています。このアプリを使えば、従業員は休暇の日付と理由を簡単に提出できるようになります。

休暇申請送信インターフェースを作成するには、どのブロックキットコンポーネントを使用すればよいですか？

A. 従業員が休暇の日付と理由を入力できるように、入力ブロックを使用します。

B. 送信フォームの上部にヘッダーブロックを配置して指示を記載する

C. 視覚的な分離のために、各フィールド間に区切りブロックを実装します。

D. 休暇日を選択するためのインタラクティブなボタンを複数作成する

正解: ([正解を表示します](#))

従業員が休暇の日付と理由を入力するには、入力ブロックを使用してください。入力ブロックは、ユーザー入力を収集するために特別に設計されており、フォームのようなインターフェースに最適です。従業員が休暇の日付と理由を直接的かつ構造的に入力できるため、明確さと使いやすさが確保されます。

他の選択肢ではダメなのか？

*ヘッダーブロックはタイトルや簡単な紹介文に使用され、実際のデータ入力には使用されません。

※区切りブロックは視覚的な区切りのために使用され、入力手段を提供するものではありません。

インタラクティブボタンは、特定の日付や詳細な理由を入力するのには適していません。これらの情報は入力フィールドで処理する方が適切です。

質問: 16

Slackワークスペースの管理において、SCIM APIの利用が最も有益となるのはどのようなシナリオでしょうか？

A. Slackチャンネル内のメッセージをリアルタイムで翻訳する機能を実装する

B. ワークスペース内で外部アプリの統合とプラグインを管理するためのツールを作成する場合

C. チャンネルのアクティビティとユーザーの参加状況に関する詳細なレポートを提供するSlackアプリを構築する。

D. 企業のHRデータベースの変更に応じてユーザーの追加、更新、削除を自動化する場合

正解: ([正解を表示します](#))

企業のHRデータベースの変更に応じてユーザーの追加、更新、削除を自動化する場合、SCIM APIは、特に企業のHRデータベースの変更に対応して、Slackワークスペースでのユーザーの追加、更新、削除といったユーザー管理タスクを自動化するのに非常に役立ちます。これにより、Slack内のユーザー情報が企業の記録と常に一致することが保証され、管理効率と正確性が向上します。

他の選択肢ではダメなのか？

*SCIM APIはリアルタイムのメッセージ変換には使用されません。

*SCIM APIは、活動状況や参加状況に関するレポートを生成することを目的としたものではありません。

外部アプリの統合やプラグインの管理は、SCIM API の主要な機能ではありません。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-Dev-201**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> **242問、30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 17

Slackアプリは、プロジェクトにおけるチームコラボレーションを促進するように設計されています。

Slackのユーザーエクスペリエンス目標に沿うためには、あなたのアプリはどのような機能を優先すべきでしょうか？

A. チームメンバーがプロジェクト関連の最新情報やファイルを共有できる、Slack内のコラボレーションワークスペースの作成

B. チームメンバーの活動ログを2時間ごとに一般チャンネルに自動投稿します。

C. 各チームメンバーによる終業時の報告書は必須であり、指定されたSlackチャンネルに投稿される。

D. チームメンバーにプロジェクトの進捗状況を求める自動ダイレクトメッセージを頻繁に送信する

正解: ([正解を表示します](#))

Slack内に共同作業スペースを作成し、チームメンバーがプロジェクト関連の最新情報やファイルを共有できるようにします。Slack内に共同作業スペースを作成し、最新情報やファイルを共有することで、チームコラボレーションのための生産的な環境が促進されます。このアプローチは、Slackが掲げる、仕事の効率化と整理整頓、そしてプロジェクト関連のコミュニケーションのための集中スペース提供という理念に合致しています。

他の選択肢ではダメなのか？

頻繁な自動投稿は、一般的なチャンネルを煩雑にし、有意義なプロジェクトの進捗状況を効果的に伝えることができない場合があります。

義務的な報告書は、必ずしもユーザーエクスペリエンスの向上や効果的なコラボレーションに貢献するとは限らない、反復的な作業になりかねません。

頻繁に送信される自動ダイレクトメッセージは、邪魔に感じられ、ワークフローを妨げる可能性があり、快適な作業環境というSlackの原則に反します。

質問: 18

チャンネルにメッセージが投稿されたときに特定のワークフローを開始するSlackアプリを開発する場合、Slack Events APIを使用してどのイベントを購読すればよいでしょうか？

- A. メッセージチャンネル
- B. メッセージが投稿されました
- C. チャンネルメッセージ
- D. app_mention

正解: **A** ([コメントを発表する](#))

message.channels は、チャンネルにメッセージが投稿されるたびにワークフローをトリガーする必要があるアプリにとって、購読すべき適切なイベントです。このイベントは、アプリがメンバーとなっているすべてのチャンネルに投稿されたメッセージに対して発生します。

他の選択肢ではダメなのか？

*SlackのイベントAPIにはmessage.postedイベントはありません。

*Channel.message は Slack Events API において有効なイベントタイプではありません。

*App_mentionは、アプリを直接言及しているメッセージにのみ適用されます。チャンネルに投稿されたすべてのメッセージを対象とするわけではありません。

質問: 19

あなたは、ユーザーが自分の仕事のタスクを追跡・管理できるSlackアプリを開発しています。

Slackのデザイン原則に沿ってユーザーエクスペリエンスを向上させるために、どのような機能を実装すべきでしょうか？

- A. アプリは、ユーザーに保留中のタスクに関するリマインダーを1時間ごとに送信します。
- B. このアプリは、ユーザーがタスクを追加したりチェックしたりできるカスタマイズ可能なToDoリストをSlack内に統合しています。

C. アプリでは、ユーザーが共通のチャンネルに日々のタスクの進捗状況を手動で入力する必要があります。

D. アプリは、ユーザーに割り当てられた新しいタスクごとにポップアップウィンドウを生成します

正解: B ([コメントを发表する](#))

このアプリは、ユーザーがタスクを追加したり完了済みとしてチェックを入れたりできるカスタマイズ可能なToDoリストをSlackに統合します。カスタマイズ可能なToDoリストをSlackに統合することで、ユーザーはワークスペースを離れることなく効率的にタスクを管理でき、仕事の簡素化というSlackの目標に合致しています。この機能は、シームレスなタスク管理ツールを提供することで、生産性とユーザーエクスペリエンスを向上させます。他の選択肢ではダメなのか？

*1時間ごとのリマインダーは煩わしく、ワークフローを阻害する可能性があり、Slackが目指す快適な職場環境とは相反する。

共通チャンネルでタスクの進捗状況を手動で入力する必要があると、情報が散乱したり非効率になったりする可能性があり、これはSlackの合理化された設計理念に合致しません。新しいタスクごとにポップアップウィンドウが表示されると、煩わしく、ユーザーエクスペリエンスを向上させるどころか、むしろ阻害する可能性があります。

質問: 20

Slackアプリは、公開チャンネル内の特定のキーワードを含むメッセージに応答する必要があります。

これを実現するには、アプリはイベントAPIのどのイベントを購読すればよいですか？

A. ファイル共有

B. app_mention

C. message.im

D. メッセージチャンネル

正解: D ([コメントを发表する](#))

message.channels は、公開チャンネルのメッセージを監視および応答するために購読するのに最適なイベントです。これにより、アプリはメッセージをリアルタイムで処理し、特定のキーワードに基づいてアクションを実行できます。

他の選択肢ではダメなのか？

*App_mentionは、アプリがメッセージ内で直接言及された場合にのみトリガーされ、一般的なキーワード監視には使用されません。

*Message.imは、公開チャンネルメッセージではなく、ダイレクトメッセージ（プライベートメッセージ）専用です。

*File_sharedはファイル共有イベントを監視するためのものであり、チャンネル内のテキストメッセージを監視するためのものではありません。

質問: 21

Slackワークスペースにとって、SCIM APIの利用が最も有益となるのはどのようなシナリオでしょうか？

- A. Slackチャンネル内で高度なデータ暗号化とセキュリティプロトコルを有効にする
- B. 異なる言語を話すユーザー間のメッセージのリアルタイム翻訳を容易にする
- C. ワークスペースの使用状況とユーザーエンゲージメント指標を示す詳細な分析ダッシュボードを作成する
- D. 外部人事管理システムの変更に基づいてユーザーの役割と権限を更新するプロセスを自動化する場合

正解: ([正解を表示します](#))

外部人事管理システムの変更に基づいてユーザーの役割と権限を更新するプロセスを自動化する場合、SCIM APIは、外部人事管理システムの変更に合わせてSlackのユーザーの役割と権限の更新を自動化するのに特に役立ちます。これにより、Slackのユーザーの役割が組織内での現在のステータスや役職を正確に反映し、ワークスペース管理の効率性とセキュリティが向上します。

他の選択肢ではダメなのか？

- *SCIM APIは、データ暗号化やセキュリティプロトコルには対応していません。
- *SCIM APIは、分析ダッシュボードを作成するために設計されたものではありません。
- *SCIM APIはリアルタイムのメッセージ変換をサポートしていません。

質問: 22

あなたはプロジェクト管理用のSlackアプリを作成しています。

Slackが掲げる快適で生産的な職場環境という理念に沿うために、あなたのアプリはプロジェクトの最新情報や通知をどのように表示すべきでしょうか？

- A. メインチャンネルでプロジェクトの最新情報を毎時間表示します。
- B. 各チームメンバーに個別のダイレクトメッセージを送信し、日々の最新情報を伝える。
- C. 主要な更新情報を集約し、指定されたチャンネルで週次ダイジェスト形式で配信する。
- D. Slack内のポップアップウィンドウを使用して、プロジェクトの変更に關するリアルタイムの更新情報を取得します。

正解: ([正解を表示します](#))

重要な更新情報を集約し、指定されたチャンネルで週次ダイジェスト形式で表示します。重要な更新情報を集約し、指定されたチャンネルで週次ダイジェスト形式で表示することで、情報の必要性和頻繁な中断の回避とのバランスが取れ、より快適で生産的な作業環境を維持できます。

他の選択肢ではダメなのか？

毎時間包括的なアップデートを提供すると、ユーザーを圧倒し、メインチャンネルが煩雑になり、生産性を阻害する可能性があります。

*毎日個別に状況報告のダイレクトメッセージを送ることは、各チームメンバーにとって煩わしく、負担になる可能性がある。

プロジェクトの変更点ごとにリアルタイムでポップアップウィンドウを表示すると、気が散りやすく、作業の流れを妨げる可能性があります。

質問: 23

あなたは、企業のプロジェクト管理システムと連携するSlackアプリを開発しています。Slackが掲げる、快適で生産性の高いユーザーエクスペリエンスの創造という理念に最も合致するデザインはどれでしょうか？

- A. プロジェクト管理システムの軽微な更新ごとにSlackで自動通知
- B. すべての更新情報については、ユーザーをプロジェクト管理システムのウェブサイトへリダイレクトする
- C. チームメンバーに未完了のタスクについて1時間ごとにリマインダーを送信するボット
- D. Slackで毎日配信される、主要なプロジェクトの更新情報とタスクをまとめたダイジェストメッセージ

正解: **D** ([コメントを发表する](#))

Slackで配信される、主要なプロジェクトの更新情報とタスクをまとめた日次ダイジェストメッセージ。主要な更新情報とタスクをまとめた日次ダイジェストは、チームメンバーが通知に圧倒されることなく、情報を簡潔かつ効率的に把握できる方法です。これは、煩雑さや情報過多を軽減することで、作業体験を簡素化し向上させるというSlackの目標に合致しています。

他の選択肢ではダメなのか？

些細なアップデートのたびに通知が届くと、煩わしく、生産性を阻害する可能性があります。

1時間ごとのリマインダーは、邪魔でイライラさせられる可能性があり、快適な職場環境を損なう恐れがあります。

*すべての更新情報を外部ウェブサイトへリダイレクトすると、Slack内のワークフローが阻害され、プロセスの効率と統合性が低下する可能性があります。

質問: 24

Slackアプリの主な目的が、ユーザー固有の権限を必要とせずに、チャンネル内のイベントを監視してそれに応じて応答することである場合、どのトークンが最も適していますか？

- A. ユーザーOAuthトークン
- B. ワークスペースアクセストークン
- C. クラシックボットトークン
- D. ボットユーザーOAuthトークン

正解: ([正解を表示します](#))

ボットユーザーOAuthトークン チャンネル内のイベントを監視し、それに応じて応答するアプリには、ボットユーザーOAuthトークンが最適です。このトークンタイプを使用すると、アプリはイベントに基づいてチャンネル内でやり取りできるため、ユーザー固有の権限へのアクセスを必要とせずにチャンネルのアクティビティに応答するボットに最適です。

他の選択肢ではダメなのか？

*ユーザーOAuthトークンは、特定のユーザーの権限を必要とするアクション向けに設計されており、一般的なチャンネルイベントのリスニングや応答には必要ありません。

*ワークスペースアクセストークンは、Slack APIの標準トークンタイプではありません。
*クラシックボットトークンは、従来の統合向けであり、ボットユーザーOAuthトークンなどのより安全で機能豊富なトークンタイプを使用できる新しいアプリには推奨されません。

質問: 25

Slackワークスペース内のユーザーリストにアクセスする必要があるSlackアプリを開発している場合、セキュリティのベストプラクティスに準拠するために、どのOAuthスコープを要求すべきでしょうか？

- A. チャット:書き込み
- B. ファイル:読み込み
- C. ユーザー:読み取り
- D. チャンネル:管理

正解: ([正解を表示します](#))

users:read スコープは、ワークスペース内のユーザー一覧へのアクセスを必要とするアプリに最適な選択肢です。このスコープでは、アプリがユーザーに関する情報を閲覧できるようにすることで、必要な権限を超えずにアプリの機能に適合させることができます。他の選択肢ではダメなのか？

*チャット書き込みはメッセージの送信に使用され、ユーザー情報へのアクセスには使用されません。

*Channels:manage は、チャンネル設定を管理するための権限を提供します。これは、ユーザーリストの表示とは関係ありません。

*Files:read は、Slack で共有されているファイル情報にアクセスするためのものであり、ユーザー情報にアクセスするためのものではありません。

質問: 26

従業員からのフィードバックを容易にするためのSlackアプリが開発されています。このアプリには、ユーザーが簡単にフィードバックを入力して送信できるユーザーインターフェースが必要です。

フィードバック送信インターフェースを作成するには、どのBlock Kitコンポーネントを使用すべきですか？

- A. フィードバックフォームの異なるセクションを区切るための区切りブロックを実装する
- B. 入力ブロックを使用して、ユーザーが直接フィードバックを入力できるようにします。
- C. ヘッダーブロックを使用して、フィードバックの送信方法に関する説明を表示します。
- D. テキストフィールドを含むセクションブロックを使用して、カテゴリ別のフィードバックを収集する

正解: ([正解を表示します](#))

入力ブロックを使用すると、ユーザーが直接フィードバックを入力できます。入力ブロックはユーザー入力専用設計されているため、フィードバック送信インターフェースとし

て最適です。ユーザーは簡単にフィードバックを入力して送信できるため、シンプルで使いやすい操作性を実現します。

他の選択肢ではダメなのか？

*区切りブロックはコンテンツを視覚的に区切るために使用され、ユーザーによる直接的な入力を受け付けません。

ヘッダーブロックは説明を提供するのに便利ですが、ユーザーによる入力できません。テキストフィールドを含むセクションブロックは、コンテンツの整理には便利ですが、入力ブロックのようなインタラクティブな入力機能は提供しません。

質問: 27

Slack監査ログAPIを利用することが最も有益なのは、どのようなシナリオでしょうか？

- A. Slackチャンネルのユーザーエンゲージメントを向上させるための強化されたユーザーインターフェースを設計する場合
- B. セキュリティとガバナンスのために、ワークスペース内のチャンネル、アプリ、ファイルの作成、変更、削除を監視および監査する。
- C. 様々なSlackチャンネルへのメッセージやアナウンスの投稿を自動化する
- D. 個々のユーザーメッセージの内容を追跡し、感情分析とフィードバック収集を行う。

正解: B ([コメントを発表する](#))

セキュリティとガバナンスのために、ワークスペース内のチャンネル、アプリ、ファイルの作成、変更、削除を監視および監査するには、Slack Audit Logs API が非常に役立ちます。この機能は、ワークスペースのセキュリティとガバナンスを維持し、ワークスペース内のすべての変更がコンプライアンスと監視のために追跡および記録されることを保証するために不可欠です。

他の選択肢ではダメなのか？

*監査ログAPIはユーザーインターフェースの設計には関与しません。

※このAPIはメッセージの自動投稿を目的としたものではありません。

*監査ログAPIは、感情分析などの目的で個々のユーザーメッセージの内容を追跡しません。

質問: 28

Slack Admin API を利用するのが最も効果的なのはどのような場合でしょうか？

- A. 生産性分析のために、各ユーザーがさまざまなSlackチャンネルで費やした時間を追跡する機能を開発する。
- B. 組織内の複数のワークスペースにわたるチャンネル作成ポリシーを管理および適用するための集中システムを構築する場合
- C. ワークスペース内で最も頻繁に使用される絵文字を集約して表示するアプリ機能
- D. Slackアプリが外部の非Slackデータベースと連携する機能を強化する

正解: ([正解を表示します](#))

組織内の複数のワークスペースにわたるチャンネル作成ポリシーを管理・適用するための集中管理システムを構築する場合、Slack Admin API は、チャンネル作成などのポリシーを

複数のワークスペースにわたって管理 適用するシステムの構築に特に役立ちます。この機能は、複雑な管理ニーズを持つ大規模組織において、一貫性とガバナンスを維持するために不可欠です。

他の選択肢ではダメなのか？

*チャンネルにおける個々のユーザーのアクティビティを追跡することは、Admin API の一般的な使用例ではありません。

絵文字の使用状況データの集計は、Admin API が処理するように設計されている管理タスクには含まれません。

*外部データベースとの連携は、Slack Admin API の主要機能ではありません。

質問: 29

SlackアプリのOAuthフローでリダイレクトURIを処理する際に、必ず実装すべき重要なセキュリティ対策は何ですか？

- A. 公開されているリダイレクトURIを利用して、容易なアクセスを確保する
- B. 透明性を確保するため、OAuthアクセストークンをリダイレクトURIに追加する
- C. リダイレクトURIがHTTPSを使用してデータ送信を安全に保護していることを確認してください。
- D. OAuthトークンを受け取った後、ユーザーをサードパーティの認証サービスにリダイレクトする

正解: ([正解を表示します](#))

リダイレクト URI が HTTPS を使用していることを確認し、データ送信を保護してください。リダイレクト URI が HTTPS を使用していることを確認することは、OAuth フロー中のデータ送信を保護するために不可欠です。この暗号化された接続は、機密性の高いトークンを含む、交換されるデータの完全性と機密性を保護するのに役立ちます。

他の選択肢ではダメなのか？

*公開されているリダイレクトURIを使用すると、権限のない第三者が容易にアクセスできる可能性があるため、セキュリティ上のリスクが生じる可能性があります。

*リダイレクトURIにOAuthアクセストークンを追加すると、トークンが漏洩したり、悪用される可能性があります。

OAuthトークンを受け取った後にユーザーをサードパーティの認証サービスにリダイレクトすると、フローが複雑になり、追加のセキュリティリスクが発生する可能性があります。

質問: 30

Slackアプリでワークスペース内のすべての公開チャンネルを一覧表示する必要がある場合、セキュリティのベストプラクティスに準拠するために、どのOAuthスコープを含めるべきですか？

- A. チャンネル:書き込み
- B. チャンネル:読み取り
- C. チャット:書き込み
- D. users:read.email

正解: ([正解を表示します](#))

channels:read スコープは、ワークスペース内のすべての公開チャンネルを一覧表示するために必要です。このスコープにより、アプリはチャンネル情報を表示する特定の機能を利用できます。これは、アプリの意図された機能に合致し、必要な権限のみを要求するという原則に準拠しています。

他の選択肢ではダメなのか？

*Channels:write は、チャンネルの作成やユーザーをチャンネルに招待するために使用され、チャンネル情報の読み取りには使用されません。

*チャットの書き込み機能はメッセージを送信するためのものであり、チャンネルに関する情報にアクセスするためのものではありません。

*Users:read.email は、ユーザーのメールアドレスにアクセスするためのものであり、チャンネルの一覧表示とは関係ありません。

質問: 31

ある企業は、新入社員に必要な情報を提供し、よくある質問への回答を提供する、従業員オンボーディング用のSlackアプリを導入したいと考えている。

この目的には、どの方法が最も効果的でしょうか？

- A. 新入社員が情報を見つけられるよう、さまざまなオンボーディングトピックに関する一連のチャンネルを作成する
- B. 新入社員にオンボーディング情報を含む自動ダイレクトメッセージを送信する
- C. オンボーディングリソースの一覧を表示するスラッシュコマンドを使用します
- D. 新入社員がオンボーディング情報を入手したり質問したりできるボットを開発する

正解: ([正解を表示します](#))

新入社員がオンボーディング情報を入手したり、質問をしたりできるボットを開発しましょう。ボットは、新入社員が個別のオンボーディング情報を受け取り、質問への回答をオンデマンドで得られる、インタラクティブで使いやすい方法を提供します。このアプローチにより、オンボーディングプロセスがより魅力的で効率的になります。

他の選択肢ではダメなのか？

チャンネルは情報を整理することはできますが、ボットが提供できるような双方向性やパーソナライズされたサポートは欠けています。

スラッシュコマンドは便利ですが、ユーザーが何を探しているのかを知っている必要があります。ボットが提供するような会話的なやり取りがありません。

自動送信されたダイレクトメッセージは双方向性がなく、新入社員が抱える可能性のある具体的な質問には対応していません。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-**

Dev-201問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> 242問、30%ディスカ

ウント、特別な割引コード: **JPNshiken**」

質問: 32

Slackアプリで使用するために機密性の高いユーザーデータを取得する場合、データセキュリティを確保するための最善策は何ですか？

- A. パフォーマンス向上のため、データを暗号化せずに送信および保存します。
- B. 暗号化されたデータにアクセスするには、チーム全体で共通のパスワードを使用する。
- C. データを共有クラウドストレージサービスに保存して、簡単にアクセスできるようにする
- D. 送信中にデータを暗号化し、安全な暗号化形式で保存する

正解: D ([コメントを发表する](#))

送信中にデータを暗号化し、安全な暗号化形式で保存する 機密性の高いユーザーデータを送信中に暗号化し、安全な暗号化形式で保存することは、機密性と完全性を維持するために不可欠です。この方法により、データは転送中も保存中も保護されます。

他の選択肢ではダメなのか？

暗号化せずにデータを送信および保存すると、潜在的な情報漏洩や不正アクセスにさらされることになります。

適切なセキュリティ対策を講じずに機密データを共有クラウドストレージサービスに保存すると、脆弱性や不正アクセスにつながる可能性があります。

チーム全体で共通のパスワードを使用すると、パスワードの取り扱いミスや漏洩が発生した場合にセキュリティリスクが生じる可能性があります。

質問: 33

開発者がSlackアプリを単一のワークスペースに限定する理由は何でしょうか？

- A. アプリが可能な限り幅広いユーザーに届き、複数のワークスペースでユーザーエンゲージメントを最大化することを保証するため
- B. アプリが特定の企業の内部プロセス向けに設計されており、その特定の組織にのみ関連するカスタム機能が含まれている場合
- C. アプリをより複雑でやりがいのあるものにすることで、ユーザーのエンゲージメントと関心を高める。
- D. アプリを様々なSlackワークスペースに配布することで、アプリの収益化を容易にする。

正解: ([正解を表示します](#))

アプリが特定の企業の内部プロセス向けに設計され、その組織にのみ関連するカスタム機能が含まれている場合、Slack アプリを単一のワークスペースにまとめることがよく選択されます。これにより、アプリがその組織の特定のニーズとワークフローに合わせて調整され、より焦点を絞った効果的なツールとして機能します。

他の選択肢ではダメなのか？

アプリを単一のワークスペースに限定しても、より幅広いユーザーにリーチすることには役立ちません。

アプリを複雑で難解なものにすることが、必ずしもユーザーの利用頻度や関心を高めるとは限りません。

*アプリを単一のワークスペースに収めることは、一般的に収益化を容易にするための戦略ではありません。

質問: 34

貴社のマーケティングチームは、ソーシャルメディア上で会社について言及があった際に、Slackで即座に通知を受け取りたいと考えています。

この要件を満たすために、どのSlack機能を実装すべきでしょうか？

- A.** ソーシャルメディアでの言及に関する定期的な更新情報を送信するボットを作成する
- B.** スラッシュコマンドを使用して、チームメンバーがメンションを手動で確認できるようにします。
- C.** 受信ウェブフックを実装して、ソーシャルメディアのメンションの通知を自動的に投稿します。
- D.** ソーシャルメディアの言及を表示するアプリのホームタブを設定します。

正解: ([正解を表示します](#))

受信ウェブフックを実装して、ソーシャルメディアでの言及通知をマーケティングチャンネルに自動的に投稿します。受信ウェブフックは、Slack内でリアルタイムの外部通知を効率的に受信する方法を提供します。ソーシャルメディアでの言及を監視する場合、ウェブフックを使用してこれらの通知を特定のマーケティングチャンネルに自動的に投稿することで、チームが最新情報を即座に把握できるようになります。

他の選択肢ではダメなのか？

定期的な更新では、ソーシャルメディアでの言及を監視するのに十分な速さではない場合があります。

*手動チェック用のスラッシュコマンドはリアルタイム通知を提供せず、チームメンバーによる積極的な作業が必要です。

*アプリのホームタブには情報が表示されますが、チャンネル通知ほど即時性や視認性は高くありません。

質問: 35

Slackのエンタープライズグリッド向けアプリを設計する際、複数のワークスペースでアプリが効果的に動作することを保証する上で、どのような点が重要となるでしょうか？

- A.** アプリは、個々のワークスペースの設定や好みに関係なく、すべてのワークスペースで同じように機能するように設計してください。
- B.** 個々のワークスペースの設定や好みによって左右されず、すべてのワークスペースでアプリが同じように動作するように設計する。

C. アプリの機能を基本的なSlack機能に限定し、グリッド上のすべてのワークスペースとの互換性を確保する。

D. ワークスペースレベルの設定ではなく、個々のユーザー向けの幅広いカスタマイズオプションに重点を置く

正解: ([正解を表示します](#))

Grid 内の各ワークスペースの特定の設定、権限、カスタマイズを処理し、尊重できる機能を備えたアプリを開発します。Enterprise Grid 用のアプリを設計するには、各ワークスペースの特定の設定、権限、カスタマイズに適応できる機能が必要です。このアプローチにより、アプリはさまざまなワークスペースに存在する多様な環境内で効果的に動作し、ワークスペース固有の構成を尊重しながらシームレスなエクスペリエンスを提供できます。

他の選択肢ではダメなのか？

個々のワークスペースの設定や好みを無視すると、画一的なアプローチになり、各ワークスペース固有のニーズに合わない可能性があります。

*機能を基本的な機能に限定すると、エンタープライズグリッド上のワークスペースの特定のニーズを満たせない可能性があります。

個々のユーザーのカスタマイズのみには焦点を当てると、ワークスペースレベルの設定の重要性や、それらがグリッド全体でのアプリの動作にどのように影響するかを見落としてしまいます。

質問: 36

ある企業が、社内ヘルプデスクの問い合わせ対応を円滑にするためのSlackアプリを開発したいと考えている。このアプリでは、従業員が問題を報告し、解決状況を追跡できるようにする必要がある。

このアプリにとって最も効果的な機能はどれでしょうか？

A. 従業員が問題を報告し、解決状況の最新情報を受け取ることができるボットを導入する

B. 従業員がヘルプデスクへのリクエストを送信するには、スラッシュコマンドを使用します。

C. 従業員が問題を投稿し、その解決状況を追跡できる専用チャンネルを作成する。

D. 問題提出用のモーダルフォームと更新通知用のメール通知を設定する

正解: A ([コメントを发表する](#))

従業員が問題を報告し、解決状況の最新情報を受け取ることができるボットを導入しましょう。ボットは、従業員が簡単に問題を報告し、リアルタイムで最新情報を受け取ることができる、インタラクティブで魅力的な体験を実現するのに最適です。このアプローチは、会話型のインターフェースを提供し、報告プロセスを案内し、継続的なステータス更新を提供することで、ユーザーのエンゲージメントと効率性を向上させます。

他の選択肢ではダメなのか？

スラッシュコマンドは、アクションを開始するのに便利ですが、ボットと同じレベルのインタラクティブ性や継続的なやり取りは提供しません。

専用チャンネルは情報が多すぎて煩雑になり、個別の更新追跡ができなくなる可能性があります。

モーダルフォームとメール通知は、問題報告と更新プロセスをSlack環境から分離するため、効率性とリアルタイムのやり取りが低下します。

質問: 37

貴社のカスタマーサポートチームは、社内コミュニケーションにSlackを使用しています。顧客からの問い合わせに関する最新情報はメールで頻繁に届き、その後Slackで議論を行います。

このプロセスを自動化する最も効果的な方法は何ですか？

- A. 特定のチャンネルにメールを転送するSlackボットを開発する
- B. Slackのメール連携機能を使って、顧客ごとにチャンネルを作成する
- C. メールを手動で指定のSlackチャンネルに転送する
- D. RSSフィードを設定して、新しいメールが届いた際にチャンネルに通知する。

正解: ([正解を表示します](#))

メールを特定のチャンネルに転送するSlackボットを開発する。メールを指定されたSlackチャンネルに直接転送できるSlackボットを使用してプロセスを自動化するのが最も効率的な方法です。これにより、手作業が削減され、チームが最新情報を迅速に受け取れるようになります。

他の選択肢ではダメなのか？

顧客ごとに個別のチャンネルを作成すると、チャンネルの数が管理不能になり、作業スペースが煩雑になる可能性があります。

*このシステムはプロセスを自動化するものではなく、手作業に依存しています。

*RSSフィードは通常、メールには使用されないため、この特定の要件には適していません。

質問: 38

Slackアプリでサードパーティサービスと連携する機能を開発する場合、どのような状況でサンプルコードや外部リソースを使用することが推奨されますか？

- A. 統合に貴社独自の知的財産に特有のアルゴリズムが含まれる場合
- B. 十分に文書化されたAPI統合を実装する場合で、信頼性が高く安全なサンプルコードが利用可能です。
- C. 市場の競合他社との差別化を図るためのコア機能の開発
- D. アプリのデータを保護するための独自の暗号化方式を作成する場合

正解: B ([コメントを發表する](#))

十分に文書化されたAPI統合を実装する場合、信頼性とセキュリティの高いサンプルコードが利用可能な場合は、サンプルコードと外部リソースの使用が推奨されます。このアプローチにより、開発プロセスが効率化され、統合が確立されたベストプラクティスに準拠することが保証されます。

他の選択肢ではダメなのか？

独自のアルゴリズムや知的財産は、機密性と管理権を維持するために、一般的に社内で開発されるべきである。

*市場においてアプリを差別化する中核機能を実現するには、特定のイノベーションや差別化の目標を達成するために、特注の開発が必要となる場合があります。

暗号化方式の作成は機密性の高い作業であり、最大限のセキュリティと有効性を確保するためには、通常、カスタム開発が必要となります。

質問: 39

Webhookリクエストを受信するSlackアプリを開発する際、これらのリクエストを処理する上で重要なセキュリティ対策は何ですか？

- A. 受信するすべてのウェブフック要求を自動的に受け入れ、データの損失を防ぎます
- B. 事前に定義されたIPアドレスリストからのリクエストのみを処理する
- C. Slackの署名シークレットを使用して、各Webhookリクエストの正当性を検証する
- D. 簡単なクエリパラメータをウェブフックURLに使用して、迅速な検証を行う

正解: ([正解を表示します](#))

Slackの署名シークレットを使用して各Webhookリクエストの正当性を検証する Slackの署名シークレットを使用して受信する各Webhookリクエストを検証することは、重要なセキュリティ対策です。これには、各リクエストにSlackが提供する署名を検証して正当性を確認することが含まれており、なりすましや悪意のある攻撃を防ぐことができます。

他の選択肢ではダメなのか？

*すべてのリクエストを自動的に受け入れると、不正なデータアクセスなどのセキュリティ上の脆弱性につながる可能性があります。

*検証にIPアドレスのみに依存すると、IPアドレスは偽装される可能性があるため、リクエストの正当性が保証されません。

検証に単純なクエリパラメータを使用するのは、簡単に操作または模倣される可能性があるため、十分なセキュリティとは言えません。

質問: 40

あなたのチームは、営業部門が週ごとの売上目標を追跡するためのSlackアプリを開発しています。このアプリは、営業チームが進捗状況を迅速かつ簡単に更新できる機能を提供する必要があります。

この機能を実現するには、どのインタラクティブコンポーネントを使用すべきでしょうか？

- A. 営業担当者が営業状況の最新情報をメッセージで送信できるボットを作成する
- B. 営業担当者が共通チャンネルに更新情報を投稿するために使用できるスラッシュコマンドを使用する
- C. 営業担当者が週ごとの進捗状況を選択できるドロップダウンメニュー付きのインタラクティブなメッセージを実装する
- D. 営業担当者が販売データを入力するためのモーダルフォームを毎週開くように設定する

正解: **C** ([コメントを发表する](#))

営業担当者が週ごとの進捗状況を選択できるドロップダウンメニュー付きのインタラクティブメッセージを実装しましょう。ドロップダウンメニュー付きのインタラクティブメッセージは、営業担当者が進捗状況を迅速かつ効率的に更新できる方法を提供します。この方法により、データの入力簡単かつ一貫して行えるため、週ごとの目標追跡に最適です。

他の選択肢ではダメなのか？

ボットへのメッセージ送信は、煩雑になりがちで、一貫したデータ追跡のための構造化が難しくなる可能性がある。

共通チャンネルでスラッシュコマンドを使用すると、更新が煩雑になり、整理されない可能性があります。

モーダルフォームを毎週開くのは、インタラクティブなメッセージに比べて時間がかかり、即時性も低い可能性があります。

質問: **41**

Slack Discovery APIを使用するのに最も適切なタイミングはいつですか？

- A. Slackチャンネル内でのユーザーの生産性とエンゲージメントをリアルタイムで監視します。
- B. Slackアプリでユーザーメッセージのカスタム分析を必要とする機能を実装する場合
- C. チャンネルで言及されたキーワードに自動的に応答するSlackボットの開発
- D. Slackワークスペース内のすべての通信をアーカイブおよび検索することにより、法的または規制上の要件を遵守する。

正解: ([正解を表示します](#))

Slackワークスペース内のすべての通信をアーカイブおよび検索することで、法的または規制上の要件を遵守できます。Discovery APIは、コンプライアンスアーカイブとeDiscoveryに特に適しており、組織はSlack内の通信をアーカイブ、検索、およびエクスポートできます。これは、デジタル通信の保存とレビューを義務付ける法的および規制上の要件を遵守するために不可欠です。

他の選択肢ではダメなのか？

*Discovery APIは、生産性のリアルタイム監視を目的として設計されていません。

※このAPIは、Slackアプリ内でカスタム分析を開発することを目的としたものではありません。

*Discovery APIは、キーワードに反応するボットの構築には使用されません。

質問: **42**

ある企業が、外部のプロジェクト管理システムと連携するSlackアプリの開発を計画しています。このアプリは、プロジェクトの更新を同期し、Slack内でタスクの割り当てを可能にする必要があります。このような開発のために、いつ共同作業チームを編成すべきでしょうか？

- A. 展開後にユーザーからのフィードバックを受け取ったら

- B. アプリのコーディング段階のみ
- C. アプリの最初のバージョンが完成した後
- D. プロジェクトが承認され次第

正解: ([正解を表示します](#))

プロジェクトが承認されたらすぐに、共同作業チームを編成することで、統合要件の理解からユーザーインターフェースの設計まで、アプリのあらゆる側面が効果的に検討され、対処されます。早期の連携は、アプリ開発をビジネスニーズと技術的な制約に合致させるのに役立ちます。

他の選択肢ではダメなのか？

コーディング段階まで待つと、設計や統合に関する考慮事項を見落とす可能性があります。

共同作業なしに最初のバージョンを完成させると、重要な機能や統合ニーズが欠ける可能性があります。

導入後にユーザーからのフィードバックを待つと、開発段階における重要な共同作業によるインプットを逃してしまう。

質問: 43

Slackアプリはユーザー認証を行う必要があります。

アプリにユーザー認証を実装するための安全な方法は何ですか？

- A. アプリ内でカスタムユーザー名とパスワードシステムを実装する
- B. 利便性のためにユーザーが認証をスキップできるようにする
- C. 簡単なセキュリティ質問を用いたユーザー認証
- D. Slackの組み込みユーザー認証システムを利用する

正解: D ([コメントを发表する](#))

Slackの組み込みユーザー認証システムを利用する Slackの組み込みユーザー認証システムを利用することは、安全かつ効率的な方法です。このシステムは既にセキュリティ基準を満たすように設計されており、追加の認証メカニズムを必要とせずに、スムーズなユーザーエクスペリエンスを提供します。

他の選択肢ではダメなのか？

*カスタムユーザー名とパスワードシステムを実装する場合、正しく行わないとセキュリティ上の脆弱性が生じる可能性があります。

*セキュリティに関する質問だけに頼ることは、堅牢な認証方法ではなく、容易に侵害される可能性があります。

ユーザーが認証を完全に回避できるようにすると、アプリのセキュリティが著しく損なわれます。

質問: 44

開発者はどのような場合に、自分のSlackアプリを他のワークスペースと共有することを検討すべきでしょうか？

- A. アプリに、開発者組織の内部プロセスに特化した高度にカスタマイズされた機能が含まれている場合
- B. アプリが、さまざまな組織の幅広いSlackユーザーにとって有益な一般的な生産性ツールや機能を提供している場合
- C. アプリが開発初期段階にあり、開発者の作業スペース内での広範なテストとフィードバックが必要な場合
- D. アプリが開発会社の専有データのみを使用して動作するように設計されている場合

正解: ([正解を表示します](#))

アプリが、さまざまな組織の幅広いSlackユーザーにとって有益な一般的な生産性ツールや機能を提供している場合、他のワークスペースとアプリを共有することは適切です。このアプローチにより、より幅広い利用が可能になり、アプリの影響力と関連性が最大限に高まります。

他の選択肢ではダメなのか？

*特定の組織向けに高度にカスタマイズされた機能を持つアプリは、他の組織にとっては関連性や有用性に欠ける場合があります。

初期開発段階では、通常、広範な展開よりも内部テストに重点が置かれます。

*独自データを使用するように設計されたアプリは、通常、他のワークスペースとの共有には適していません。

質問: 45

人事チームが従業員の職場満足度に関するフィードバックを収集するためのSlackアプリを開発中です。フィードバックは詳細かつ体系的に収集される必要があります。

この目的に最も適したインタラクティブコンポーネントはどれでしょうか？

- A. ボットを利用してダイレクトメッセージで質問したり、フィードバックを収集したりする。
- B. 従業員が満足度を選択できるボタン付きのインタラクティブなメッセージを使用する
- C. 従業員がフィードバックを投稿できる専用チャンネルを作成する
- D. 詳細なフィードバックのための様々なフィールドを含むフォームを備えたモーダルを実装する

正解: ([正解を表示します](#))

詳細なフィードバックのための様々なフィールドを含むフォームを備えたモーダルを実装します。フォームを備えたモーダルは、構造化された詳細なフィードバックを収集するのに最適です。人事担当者は、従業員が記入できる具体的な質問とフィールドを設計できるため、包括的で整理されたフィードバックを得ることができます。

他の選択肢ではダメなのか？

ダイレクトメッセージは構造化されていないため、詳細なフィードバックに必要な形式やスペースが不足する可能性があります。

専用のチャンネルを設けると、フィードバックが整理されず、従業員のプライバシーが侵害される可能性がある。

*ボタン付きのインタラクティブメッセージは、詳細なフィードバックよりも、迅速で簡単な応答に適しています。

質問: 46

Slack Admin API を使用するのに最も適したシナリオはどのようなものですか？

- A. Slackワークスペースまたは組織内で多数のユーザーを追加、削除、または管理するプロセスを自動化するツールを作成する場合
- B. Slackアプリ内で、すべてのユーザーが個々のSlackインターフェーステーマをカスタマイズできる機能を開発する場合
- C. 組織内のどのユーザーにも直接メッセージを送信できるようにするため、
- D. 特定のSlackチャンネル内でよくある質問に回答するチャットボットを構築する場合

正解: ([正解を表示します](#))

Slackワークスペースや組織内で多数のユーザーを追加、削除、管理するプロセスを自動化するツールを作成する場合、Admin APIは多数のユーザーの管理などの管理タスクを自動化するのに最適です。ユーザーのオンボーディング、オフボーディング、役割や権限の変更を効率的にプログラムで制御できるため、大規模なワークスペースや組織の管理に不可欠です。

他の選択肢ではダメなのか？

*個別のインターフェースのカスタマイズは、通常、Admin API で処理される管理タスクには含まれません。

ダイレクトメッセージ機能の有効化は、管理者APIの機能ではありません。

*Admin APIは、チャットボット、特にチャンネル内のFAQへの回答に特化したチャットボットの構築には適していません。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-Dev-201**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> **242問、30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 47

Slack アプリの古いバージョンでは、API エンドポイント channels.list に遭遇します。

アプリのアップデートの一環として、この従来のエンドポイントを置き換えるのに最も適切な最新のAPIエンドポイントはどれですか？

- A. ユーザーの会話
- B. グループリスト
- C. im.list

D. conversations.list

正解: **D** ([コメントを发表する](#))

conversations.list は、従来の channels.list エンドポイントに代わる、現在利用可能な適切なエンドポイントです。Slack の新しい API の一部であり、公開チャンネル、プライベートチャンネル、ダイレクトメッセージ、複数人によるダイレクトメッセージなど、あらゆる種類のチャンネルを網羅しています。

他の選択肢ではダメなのか？

*Groups.list は旧式の用語であり、最新の API 構造を表すものではありません。

*Im.listは、すべての会話タイプの包括的なリストではなく、特にダイレクトメッセージチャンネルを指します。

*Users.conversations は有効な、または存在する Slack API エンドポイントではありません。

質問: 48

分散型Slackアプリを最もよく表しているのは次のうちどれですか？

- A. 単一のワークスペース内でのみ利用可能で、他のワークスペースにはインストールできないアプリ
- B. Slackアプリディレクトリで入手可能で、どのSlackユーザーでもインストールできるアプリ
- C. 組織内での内部利用を目的として設計され、Slackアプリディレクトリに掲載されていないアプリ
- D. 各ワークスペース間で手動での共有とインストールが必要なアプリ

正解: ([正解を表示します](#))

Slackアプリディレクトリで入手可能で、Slackユーザーなら誰でもインストールできるアプリ。Slackにおける分散型アプリとは、Slackアプリディレクトリで入手可能なアプリのことで、Slackを使用しているユーザーや組織なら誰でもアプリを見つけてインストールできます。これにより、幅広いユーザー層がアプリにアクセスしやすくなり、単一ワークスペース用アプリや内部アプリとは区別されます。

他の選択肢ではダメなのか？

*これは、特定のワークスペースに限定された、単一のワークスペースアプリについて説明します。

※これは、特定の組織内での使用を目的として設計された内部アプリであり、一般公開はされていません。

※これはアプリのカテゴリではなく、共有プロセスを説明するものです。

質問: 49

公開されたアプリディレクトリアプリと、Slackで配布されるアプリの違いは何ですか？

- A. 公開型アプリディレクトリアプリは特定の業界のみで使用されますが、分散型アプリはあらゆる種類のビジネスで一般的に使用されます。

- B.** 公開されたアプリディレクトリアプリは複数のワークスペースにインストールできますが、分散アプリは単一のワークスペースに制限されます。
- C.** 分散型アプリは社内コミュニケーション専用ですが、公開アプリディレクトリアプリは外部関係者向けです。
- D.** 公開されたアプリディレクトリアプリはSlackによって承認され、Slackアプリディレクトリに掲載されていますが、配布されたアプリは必ずしもそこに掲載されているとは限りません。

正解: ([正解を表示します](#))

公開されたアプリディレクトリアプリはSlackによって承認され、Slackアプリディレクトリに掲載されますが、配布されたアプリは必ずしもそこに掲載されるとは限りません。公開されたアプリディレクトリアプリと配布されたアプリの主な違いは、掲載状況です。公開されたアプリディレクトリアプリはSlackの承認プロセスを経てSlackアプリディレクトリに掲載されるため、誰でもSlackアプリを見つけてインストールできます。一方、配布されたアプリは複数のワークスペースで共有できますが、必ずしもアプリディレクトリに掲載されるとは限りません。

他の選択肢ではダメなのか？

*アプリの分類は、特定の業界に基づいているわけではありません。

※これらのアプリのカテゴリは、内部コミュニケーション用か外部コミュニケーション用かという用途によって定義されるものではありません。

*公開されたアプリディレクトリアプリと配布アプリの両方を、複数のワークスペースにインストールできます。

質問: 50

SlackアプリのSlackディレクトリへの登録が完了し、承認されました。

アプリの認知度とリーチに関して、どのようなことが期待できますか？

- A.** あなたのアプリは自動的にSlackディレクトリの上位リストに表示されます
- B.** 可視性とリーチは、ユーザーレビュー、評価、ユーザーエンゲージメントなどの要因によって左右されます。
- C.** Slackは広告キャンペーンを通じて、あなたのアプリを外部に宣伝します。
- D.** あなたのアプリはプレミアムSlackユーザー限定で表示されます

正解: ([正解を表示します](#))

アプリの可視性とリーチは、ユーザーレビュー、評価、ユーザーエンゲージメントなどの要素によって左右されます。承認後、アプリはSlackディレクトリに掲載されますが、ディレクトリ内での可視性とリーチは、ユーザーレビュー、評価、ユーザーエンゲージメントなど、さまざまな要素によって左右されます。肯定的なレビューと高い評価を受けたアプリは、ディレクトリ内で目立つように掲載され、可視性が高まる可能性が高くなります。Slackは、外部の広告キャンペーンではなく、アプリの品質と有用性に基づいてアプリを宣伝します。

他の選択肢ではダメなのか？

*アプリの認知度は、自動的な上位表示のみに基づいているわけではなく、ユーザーのインタラクションやフィードバックにも左右されます。

*Slackは通常、外部の広告キャンペーンではなく、プラットフォーム内でアプリを宣伝します。

*Slackディレクトリに掲載されているアプリは、プレミアムユーザーだけでなく、幅広いSlackユーザーがアクセスできます。

質問: 51

顧客サービス対応を目的としたSlackアプリは、どのような場合にSlack Connectで共有されるチャンネルのサポートを含めるべきでしょうか？

- A. アプリの主な機能が社内チームのワークフロー管理と顧客サービス研修である場合
- B. アプリが共有外部チャンネルを通じて顧客と直接的なサポートやコミュニケーションを提供する必要がある場合
- C. アプリが、社内チーム内に留めておくべき機密性の高い顧客情報を扱う場合
- D. アプリが社内顧客サービスリクエストの自動化を目的としており、外部クライアントとのやり取りを行わない場合

正解: B ([コメントを发表する](#))

アプリが顧客との直接的なサポートやコミュニケーションを外部共有チャンネルを通じて提供する必要がある場合、Slack Connect を介して外部で共有されるチャンネルのサポートは不可欠です。これにより、共有チャンネル環境で顧客と効果的かつ直接的なやり取りが可能になり、顧客サポート体験が向上します。

他の選択肢ではダメなのか？

*アプリが主に社内ワークフローやトレーニングに使用される場合、外部チャンネルとの統合は不要です。

機密性の高い顧客情報を扱う場合、通常は外部チャンネルには適さないプライバシーとセキュリティ対策が求められます。

内部の自動リクエストのみに特化したアプリは、外部チャンネルのサポートによるメリットを受けません。

質問: 52

貴社は、社内での勤怠管理と報告のためのSlackアプリを開発する予定です。このアプリは、既存の人事管理ソフトウェアとの連携と、使いやすいインターフェースを備えている必要があります。

このSlackアプリの開発で協力するチームは、どの段階で結成すべきでしょうか？

- A. インターフェースデザインが確定した後のみ
- B. 初期の構想・計画段階から
- C. 人事ソフトウェアとの統合が完了したら
- D. 選ばれたユーザーグループによる最初のテストラウンドの後

正解: B ([コメントを发表する](#))

構想 計画の初期段階から協力的なチームを編成することが非常に重要です。これにより、アプリのデザイン、機能、既存の人事ソフトウェアとの統合について、すべての関係者からの意見を取り入れて検討することができ、より効果的でユーザー中心のソリューションにつながります。

他の選択肢ではダメなのか？

インターフェース設計が最終決定されるまで待つと、設計が統合要件や機能要件を満たさない場合、手戻りが発生する可能性があります。

*先に統合を完了すると、アプリの機能とユーザーインターフェースのニーズとの間に乖離が生じる可能性があります。

最初のテストラウンドが終わるまで待つと、開発プロセスの後半で大幅な変更が必要となり、時間とリソースの消費が増加する可能性があります。

質問: 53

プロジェクト管理用のSlackアプリを開発する際、仕事生活をより快適かつ効率的にするというSlackの目標に合致するデザインアプローチとはどのようなものでしょうか？

A. アプリはプロジェクトのあらゆるマイナーアップデートに対してリアルタイム通知を送信します

B. 詳細なプロジェクト追跡のために、ユーザーを別のプロジェクト管理ツールにリダイレクトする

C. すべてのチームメンバーは、プロジェクトの状況を更新するために、アプリを介して1時間ごとにチェックインすることが義務付けられています。

D. Slack内のダッシュボードで、プロジェクトのステータスの概要が表示され、さらに詳細を表示するためのオプションも用意されています。

正解: **D** ([コメントを发表する](#))

Slack 内のダッシュボードは、プロジェクトのステータスの概要を提供し、詳細情報へのドリルダウンオプションも備えています。Slack 内のダッシュボードは、プロジェクトのステータスの概要と詳細なドリルダウンオプションを提供し、効率性と合理化されたユーザーエクスペリエンスに対する Slack の取り組みをサポートします。一目で重要な情報を確認できるだけでなく、必要に応じて詳細情報を柔軟に探索することもできます。

他の選択肢ではダメなのか？

些細なアップデートのたびに通知が届くのは、煩わしく、逆効果になる可能性がある。

1時間ごとの強制的なチェックインは負担となり、日々の業務を妨げ、生産性を低下させる可能性がある。

*ユーザーを別のツールにリダイレクトすると、Slackの統合されたエクスペリエンスが損なわれ、効率が低下する可能性があります。

質問: 54

ある企業が、インシデント管理プロセスを効率化するためにSlackアプリを開発したいと考えている。このアプリは、報告者からインシデントの詳細を迅速に収集し、関係チームに通知する必要がある。

このアプリにはどのようなデザインアプローチを採用すべきでしょうか？

- A. ユーザーが事件を投稿し、他のユーザーが絵文字で反応して認識を示すことができるチャンネルを作成する。
- B. モーダルフォームを使用してインシデントの詳細を収集し、その詳細を専用のインシデント管理チャンネルに自動的に投稿します。
- C. 新しいインシデントが報告された際に、チームメンバーに自動的にダイレクトメッセージを送信する
- D. 追加の操作や通知なしにインシデントをログに記録するためのカスタムスラッシュコマンドを開発する

正解: ([正解を表示します](#))

モーダルフォームを使用してインシデントの詳細を収集し、その詳細を専用のインシデント管理チャンネルに自動的に投稿します。モーダルフォームは、構造化されたインシデントの詳細を収集する効果的な方法です。これらの詳細を専用チャンネルに自動的に投稿することで、適切なチームに迅速に通知され、迅速な対応と透明性が確保されます。

他の選択肢ではダメなのか？

ユーザーがチャンネルにインシデントを投稿することに依存すると、構造化されていないデータが生成され、重要な詳細を見落とす可能性があります。

*チームメンバーへの自動ダイレクトメッセージは、中央のインシデントチャンネルにリンクされていない場合、混乱を招いたり、文脈が欠落したりする可能性があります。

スラッシュコマンドだけでは、詳細な情報を収集したり、関係チームに効果的に通知したりするための体系的な方法を提供することはできません。

質問: 55

Enterprise Grid Sandbox の環境において、開発者はアプリのテストを行う際にどのような点に留意すべきでしょうか？

- A. サンドボックスで行われた変更と設定は永続的に保存され、デプロイ時に本番のエンタープライズグリッド環境に自動的に転送されます。
- B. サンドボックスは、エンタープライズグリッドの本番環境を忠実に再現した隔離されたテスト環境を提供し、実際の運用に影響を与えることなく正確なテストを可能にします。
- C. サンドボックスでのテストは、重要度の低い機能に限定する必要があります。サンドボックスは、本番環境のエンタープライズグリッドのセキュリティ機能を完全に再現するものではないためです。
- D. サンドボックス環境は、アプリのユーザーインターフェース面のみをテストすることを可能にし、バックエンドの統合やデータ処理はテストしません。

正解: ([正解を表示します](#))

サンドボックスは、エンタープライズ グリッドの本番環境を忠実に再現した隔離されたテスト環境を提供し、実際の運用に影響を与えることなく正確なテストを可能にします。エンタープライズ グリッド サンドボックスは、エンタープライズ グリッドの本番環境を再現した隔離された環境を提供するように設計されています。これにより、開発者は実際のユーザーや運用に影響を与えることなく、本番環境に非常に近い方法でアプリの正確なテストを実施できます。

他の選択肢ではダメなのか？

サンドボックスで行われた変更は永続的ではなく、デプロイ時に本番環境に自動的に反映されるわけではありません。

*サンドボックスは、エンタープライズグリッドの重要な機能やセキュリティ機能を含む、あらゆる機能を再現するように設計されています。

*サンドボックスは、統合やデータ処理など、アプリのユーザーインターフェースとバックエンドの両方の側面をテストすることをサポートします。

質問: 56

Slackアプリで、ワークスペースに新しく参加したメンバーにダイレクトメッセージを送信して自動的に歓迎メッセージを送るようするには、Slack Events APIのどのイベントを購読すればよいのでしょうか？

- A. user_change
- B. メッセージチャネル
- C. チーム参加
- D. reaction_added

正解: ([正解を表示します](#))

team_join は、アプリがワークスペースに新しいメンバーが参加したことを検出し、ウェルカムダイレクトメッセージを送信できるようにするために購読するのに適したイベントです。

他の選択肢ではダメなのか？

*User_changeは、ユーザープロファイルの更新を監視するために使用され、新規メンバーの参加には使用されません。

*Message.channelsはチャネル内のメッセージをリッスンするためのものであり、ワークスペースの新しいメンバーを追跡するためのものではありません。

*reaction_addedは、メッセージに追加されたリアクションを監視するために使用され、新規メンバーの歓迎とは関係ありません。

質問: 57

開発者がSlackアプリを単一のワークスペースに限定することを選択する理由は何でしょうか？

- A. 多様なワークスペース全体で最大限の露出とユーザーによる利用を確保するため
- B. アプリに、特定の組織のワークフローに特有の機能や情報が含まれている場合

C. 複数のワークスペースにわたる、より大規模で多様なオーディエンスにリーチすることで、収益化を容易にする。

D. 異なる組織の実際のユーザーを対象に、ライブ環境でアプリをテストする目的で使用します。

正解: ([正解を表示します](#))

アプリに特定の組織のワークフローに特化した機能や情報が含まれている場合、アプリを単一のワークスペースに限定することがよく選択されます。これは、アプリに特定の組織のワークフロー、プロセス、またはデータに合わせて調整された機能や情報が含まれている場合によく行われます。これにより、アプリは設計されたコンテキスト内で関連性と有用性を維持し、機密情報や秘密情報をより広いユーザーに公開することなく済みます。

他の選択肢ではダメなのか？

*アプリを単一のワークスペースに限定すると、その露出が制限され、広く普及させるには適していません。

*アプリを1つのワークスペースに限定しても、より広範な収益化の機会は得られません。

*異なる組織のユーザーとテストを行うには、単一のワークスペースを超えた分散環境が必要になります。

質問: 58

Slackワークスペース管理者にとって、Slack管理APIの利用が最も有益なのはどのような場合ですか？

A. 各ユーザー向けにSlackワークスペースのレイアウトとインターフェースをカスタマイズする

B. Slackアプリ内でメッセージをリアルタイムでさまざまな言語に翻訳する機能を実装する。

C. 各ユーザーが送信したメッセージ数を追跡し、ワークスペースへのより積極的な参加を促すため

D. 組織内の複数のワークスペースにわたって、チャンネルの作成やアプリのインストールなどのワークスペース設定を管理するための集中管理システムを構築する場合

正解: ([正解を表示します](#))

組織内の複数のワークスペースにわたるチャンネル作成やアプリのインストールなどのワークスペース設定を管理する集中管理システムを構築する場合、Slack Admin API は非常に役立ちます。これにより、管理者は組織内の複数のワークスペースにわたるチャンネル作成、アプリのインストール、ワークスペース全体の構成など、さまざまな側面を制御でき、一貫性と企業ポリシーへの準拠を確保できます。

他の選択肢ではダメなのか？

*管理者APIは、個々のユーザーレベルでのユーザーインターフェースのカスタマイズを目的として設計されていません。

*管理者APIは通常、個々のユーザーのメッセージ活動を追跡するために使用されるものではありません。

メッセージの翻訳は、Admin APIの機能ではなく、特定のSlackアプリまたは連携機能によって処理される機能です。

質問: 59

ある企業が、従業員のITサポートに関する問い合わせを支援するためのSlackアプリを開発したいと考えている。

これらの問い合わせの送信と追跡を迅速かつ効率的に行うために、Slackのどのインターフェースを主に利用すべきでしょうか？

- A. サポートに関する問い合わせを送信および追跡するためのアプリホームを実装する
- B. 各質問に対してサポートボットに直接メッセージを送信してください
- C. 専用のITサポートチャンネルに質問を投稿してください。
- D. Slackのメール連携機能を使って問い合わせを送信する

正解: **A** ([コメントを发表する](#))

サポートに関する問い合わせの送信と状況追跡のためのアプリホームを実装します。アプリホームは、ユーザー専用のパーソナライズされたスペースを提供し、ITサポートに関する問い合わせの送信と状況追跡に最適です。この集中管理方式により、関連するすべてのやり取りを1か所に集約できるため、ユーザーエクスペリエンスが向上します。

他の選択肢ではダメなのか？

ボットへの直接メッセージは最初の問い合わせには便利ですが、追跡のための集中管理機能がありません。

専用チャンネルは情報が多すぎて煩雑になりがちで、個々の問い合わせを追跡するためのパーソナライズ機能が不足する可能性がある。

*メール連携を使用すると、Slackの組み込み機能を活用して効率的なクエリ処理を行うという本来の目的から逸脱します。

質問: 60

チームメンバーが週ごとの進捗報告を提出できるSlackアプリを開発中です。このアプリは、アクセスしやすく、使いやすいものでなければなりません。

これらのレポートを送信するためのユーザー入力ポイント进行处理する最善の方法は何ですか？

- A. ユーザーが報告をメッセージとして投稿できるチャンネルを作成する
- B. ユーザーがアプリのホーム画面に移動してレポート提出フォームを見つけるようにする
- C. ボットからのダイレクトメッセージを使用して、ユーザーにレポートの提出を促す
- D. レポート送信用のモーダルを開くスラッシュコマンドを実装する

正解: ([正解を表示します](#))

レポート送信用のモーダルウィンドウを開くスラッシュコマンドを実装します。スラッシュコマンドは、ユーザーがレポート送信プロセスを開始するための簡単でアクセスしやすい入り口です。スラッシュコマンドの使用時にモーダルウィンドウを開くことで、レ

ポート送信のための構造化された集中型インターフェースが提供され、ユーザーエクスペリエンスが向上します。

他の選択肢ではダメなのか？

*レポート送信のためにアプリのホーム画面に移動する必要があるため、不要な手順が増え、アクセシビリティが低下します。

ボットからのダイレクトメッセージは、迷惑に感じられる可能性があり、ユーザー主導のメッセージとはみなされにくい。

チャンネルにレポートを投稿すると、プライバシーと構造が欠如します。

質問: 61

Slackアプリは、チャンネル内で言及された特定のキーワードに対して自動的に応答するように設計されています。

セキュリティのベストプラクティスに沿うためには、どのOAuthスコープを要求すべきでしょうか？

- A. チャット:書き込み
- B. ユーザー:閲覧
- C. ファイル:書き込み
- D. 管理者

正解: ([正解を表示します](#))

chat:write スコープは、チャンネル内のメッセージへの応答を自動化するアプリに適しています。このスコープにより、アプリはメッセージを投稿できるようになります。このスコープは、アプリが必要とする権限を超えず、最小権限の原則に沿って、アプリの機能に必要な権限を確実に取得できるようにします。

他の選択肢ではダメなのか？

*Users:read はユーザー情報にアクセスするためのものであり、メッセージを投稿するためのものではありません。

*Files:writeはファイルのアップロードと管理を目的としており、チャンネルメッセージへの応答とは関係ありません。

*管理者権限はワークスペースに対する広範な制御権限を与えるため、自動メッセージングに特化したアプリとしては過剰です。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-Dev-201**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> **242**問、**30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 62

SlackアプリにOAuthフローを実装する際、SlackからOAuthアクセストークンを受け取った後、取るべき重要なセキュリティ手順は何ですか？

- A. アプリが簡単にアクセスできるように、アクセストークンをクライアント側のJavaScriptに保存します。
- B. アクセストークンをメールでアプリ管理者に送信し、手動で処理してもらう。
- C. アクセストークンをサーバー上に安全に保管し、アプリケーションの必要な部分のみに公開するように制限します。
- D. ユーザー認証のために、アプリのユーザーインターフェースにアクセストークンを表示する

正解: ([正解を表示します](#))

アクセストークンをサーバーに安全に保管し、アプリケーションの必要な部分のみに公開するように制限することは、セキュリティにとって非常に重要です。これにより、トークンの盗難や漏洩のリスクを最小限に抑え、トークンが絶対に必要な場合にのみ使用されることが保証されます。

他の選択肢ではダメなのか？

クライアント側のJavaScriptにアクセストークンを保存すると、XSS攻撃による盗難の危険性が高まります。

トークンを電子メールで送信することは安全ではなく、傍受や不正アクセスにつながる可能性があります。

ユーザーインターフェースにアクセストークンを表示することは、UIにアクセスできるユーザーや攻撃者がトークンを閲覧して悪用する可能性があるため、重大なセキュリティリスクとなります。

質問: 63

Slack監査ログAPIを利用することが最も有益なのは、どのようなシナリオでしょうか？

- A. 特定のチャンネルに投稿されたメッセージのリアルタイム通知
- B. 機密性の高いチャンネルやデータへのユーザー権限やアクセス権限の変更を監視および確認する必要がある場合
- C. 個々のユーザー向けにSlackワークスペースのインターフェースとテーマ設定をカスタマイズする
- D. サポートチャンネルでよくある質問への回答を自動化するSlackボットを設計する場合

正解: B ([コメントを发表する](#))

ユーザーの権限や機密性の高いチャンネルまたはデータへのアクセス権限の変更を監視および確認する必要がある場合、Slack Audit Logs API は、特に機密性の高いチャンネルやデータへのアクセス権限の変更を監視および確認するのに非常に役立ちます。これにより、管理者はセキュリティ関連の重要な変更を追跡し、ワークスペース内でアクセス権限が適切に管理されていることを確認できます。

他の選択肢ではダメなのか？

*監査ログAPIは、リアルタイムのチャンネル通知には使用されません。

※このAPIは、ワークスペースのインターフェースやテーマのカスタマイズには対応していません。

*監査ログAPIは、Slackボットの構築や管理を目的としたものではありません。

質問: 64

あなたは従業員のオンボーディングのためのSlackアプリを設計しています。
新入社員の入社手続きをよりシンプルで快適なものにするために、どのような機能を含めるべきでしょうか？

- A. 新入社員の採用を告知する自動メッセージが全チャンネルに一斉送信される。
- B. 新入社員は一般チャンネルで毎日必ずチェックイン投稿を行う
- C. 新入社員がオンボーディングタスクを完了するよう、頻繁にポップアップ通知が表示される
- D. リソース、よくある質問、ウェルカムメッセージなどを掲載した専用のオンボーディングチャンネル

正解: ([正解を表示します](#))

リソース、よくある質問、ウェルカムメッセージを掲載した専用のオンボーディングチャンネルを作成する 新入社員にとって不可欠なリソース、よくある質問、パーソナライズされたウェルカムメッセージを掲載した専用のオンボーディングチャンネルを作成することで、一元化された、操作しやすいスペースを提供できます。このアプローチにより、オンボーディングプロセスが簡素化され、より使いやすくなります。

他の選択肢ではダメなのか？

*すべてのチャンネルへのメッセージ一斉送信は、新入社員と既存のチームメンバーの両方にとって、混乱を招き、負担となる可能性があります。

頻繁に表示されるポップアップ通知は、新入社員にとって煩わしく、ストレスの原因となる可能性があります。

公開チャンネルでの毎日のチェックインを義務付けると、新入社員に過度のプレッシャーを与えたり、一般チャンネルが混雑したりする可能性がある。

質問: 65

Slack環境でDiscovery APIを実装する上で、最も効果的なのはどのような場合でしょうか？

- A. Slack内のメッセージアクティビティに基づいて従業員の生産性を追跡する分析ダッシュボードを作成するため
- B. ワークスペース内のファイルアップロードやメッセージコンテンツを分析することで、潜在的なデータ漏洩を監視・防止する必要がある場合
- C. キーワードに基づいてSlackチャンネル内のユーザーメッセージに自動的に応答する機能を開発する。
- D. Slackワークスペース内のユーザーの通知設定と環境設定をカスタマイズするための機能

正解: ([正解を表示します](#))

ワークスペース内のファイルアップロードやメッセージコンテンツを分析することで、潜在的なデータ漏洩を監視・防止する必要がある場合、Discovery APIは特に有効です。組織はSlackワークスペース全体のファイルアップロードやメッセージコンテンツを分析できるため、データ損失の防止や機密情報のセキュリティ確保に不可欠です。

他の選択肢ではダメなのか？

※このAPIは、主に生産性追跡を目的としたものではありません。

*Discovery APIは、自動応答機能の開発を目的として設計されていません。

*ユーザー通知設定や環境設定のカスタマイズには対応していません。

質問: 66

Slackアプリは、パーソナライズされた設定やユーザープロフィール情報など、ユーザー固有のデータとやり取りするように設計されています。

セキュリティのベストプラクティスを遵守するために、アプリはどのような種類のトークンを要求すべきでしょうか？

- A. ユーザーOAuthトークン
- B. ボットユーザーOAuthトークン
- C. ワークスペースアクセストークン
- D. アプリレベルトークン

正解: A ([コメントを发表する](#))

ユーザーOAuthトークン アプリがユーザー固有のデータとやり取りする必要がある場合、ユーザーOAuthトークンが必要です。このトークンタイプは、アプリが個々のユーザーによって明示的に付与された権限で動作することを保証し、個人設定やプロフィール情報に安全にアクセスするために不可欠です。

他の選択肢ではダメなのか？

*ボットユーザーのOAuthトークンは通常、ワークスペース内でのより一般的な操作に使用され、ユーザー固有のデータへのアクセスには使用されません。

*ワークスペースアクセストークンは、Slack APIの標準トークンタイプではありません。

*アプリレベルトークンは、ワークスペースレベルのデータと機能へのより広範なアクセスを提供しますが、個々のユーザーレベルの情報へのアクセスは提供しません。

質問: 67

貴社は、従業員の経費精算申請と承認を円滑に行うためのSlackアプリを開発中です。

このアプリにはどのワークフローが最も適しているでしょうか？

- A. 従業員が経費を投稿し、管理者がそれを公開で承認する共有チャンネルを設定する。
- B. 従業員が経費を入力するためのスラッシュコマンドを開発し、管理者による確認なしに自動的に承認されるようにする。
- C. ボットを使用してダイレクトメッセージ経由で経費の詳細を収集し、共通の財務チャンネルに概要を投稿して承認を得る。

D. 従業員が経費を提出できるモーダルフォームを実装し、提出された経費はマネージャーがプライベートスレッドで確認・承認します。

正解: ([正解を表示します](#))

従業員が経費を提出できるモーダルフォームを実装し、提出された経費はマネージャーがプライベートスレッドで確認・承認します。従業員が経費を提出するためのモーダルを使用することで、詳細を入力する構造化された使いやすい方法が提供されます。マネージャーがプライベートスレッドでこれらの提出内容を確認・承認することで、プライバシーが保護され、承認プロセスが体系的に管理されます。

他の選択肢ではダメなのか？

経費の公開と承認は、プライバシーの問題やチャンネルの煩雑化につながる可能性があります。

ダイレクトメッセージにボットを使用すると、一元的な審査プロセスが欠如し、承認ワークフローが複雑になる可能性があります。

経費管理において、管理者による審査なしの自動承認は一般的に推奨されません。

質問: 68

時間管理と生産性向上を目的としたSlackアプリを設計する際、Slackのユーザーエクスペリエンス目標に沿うためには、どの機能を重視すべきでしょうか？

- A. Slack内のインタラクティブなツールで、ユーザーは作業セッションの開始、一時停止、ログ記録を行うことができます。
- B. マネージャーがSlack上でチームメンバーのリアルタイム活動ログを監視できる機能
- C. Slackでポップアップメッセージが表示され、ユーザーに1時間ごとに勤務時間を記録するよう促す。
- D. 毎日終業時に一般チャンネルで勤務時間ログを必ず報告すること

正解: A ([コメントを发表する](#))

Slack 内のインタラクティブなツールで、ユーザーは作業セッションの開始、一時停止、ログ記録を行うことができます。Slack 内の作業セッション管理のためのインタラクティブなツールは、生産的で快適な作業環境を実現するという Slack の取り組みに沿ったものです。ユーザーはプラットフォーム内での時間を簡単に追跡でき、効率性を高め、ワークフローの中断を最小限に抑えることができます。

他の選択肢ではダメなのか？

常時リアルタイム監視は、ストレスの多い職場環境を作り出し、プライバシーを侵害する可能性がある。

頻繁に表示されるポップアップメッセージは、集中力を妨げ、生産性に悪影響を与える可能性があります。

*義務的な公開報告は不必要なプレッシャーにつながり、効率的で良好なユーザーエクスペリエンスには貢献しない。

質問: 69

ワークスペース内のチャンネルを管理および作成する必要があるSlackアプリを作成する場合、どのOAuthスコープが最も適切でしょうか？

- A. チャンネル:読み取り
- B. チャット:書き込み
- C. チャンネル:書き込み
- D. users:read.email

正解: **C** ([コメントを发表する](#))

channels:write スコープは、チャンネルの管理と作成機能を必要とするアプリに最適です。このスコープでは、アプリがチャンネルを作成、変更、削除できるようになり、意図した機能を満たしつつ、必要以上に広範な権限を要求しないようにします。

他の選択肢ではダメなのか？

*チャンネル:読み取り専用では、チャンネル情報の表示のみが許可され、チャンネルの作成や管理はできません。

*Chat:write はアプリがメッセージを送信できるようにする機能であり、チャンネル管理とは関係ありません。

*Users:read.email はユーザーのメールアドレスにアクセスするためのものであり、チャンネル管理とは関係ありません。

質問: 70

Slackワークスペースの管理にSCIM APIを使用するのに最も適した状況はどのような場合ですか？

- A. Slackチャンネル内のメッセージとアクティビティをリアルタイムで監視します
- B. Slackを外部のID管理システムと統合して、ユーザーのプロビジョニングとデプロビジョニングを自動化する場合
- C. Slackチャンネル内で、投票やアンケートなどのカスタムインタラクティブ機能を開発する
- D. Slackワークスペースの使用状況に関する詳細な分析データを分析およびエクスポートします。

正解: **B** ([コメントを发表する](#))

Slackを外部ID管理システムと連携させてユーザーのプロビジョニングとデプロビジョニングを自動化する場合、SCIM APIは、Slackを外部ID管理システムと連携させてユーザーのプロビジョニングとデプロビジョニングのプロセスを自動化するのに最適です。これには、外部システムの変更に基づいてSlackのユーザーアカウントを作成、更新、削除し、プラットフォーム間でユーザー管理を効率化することが含まれます。

他の選択肢ではダメなのか？

*SCIM APIは、チャンネルアクティビティのリアルタイム監視を目的として設計されていません。

SCIM APIは、ユーザーとグループの管理に重点を置いており、インタラクティブなチャンネル機能の開発には対応していません。

*SCIM APIは、ワークスペースの使用状況データの分析およびエクスポートには使用されません。

質問: 71

主に様々なチャンネルにメッセージを投稿する機能を持ち、ユーザーのプライベートデータにアクセスする必要のないSlackアプリを開発している場合、どのトークンを要求すべきでしょうか？

- A. ボットユーザーOAuthトークン
- B. ユーザーOAuthトークン
- C. ワークスペースアクセストークン
- D. クラシックボットトークン

正解: **A** ([コメントを发表する](#))

ボットユーザーOAuthトークン プライベートなユーザーデータへのアクセスを必要とせず、にチャンネルにメッセージを投稿するアプリには、ボットユーザーOAuthトークンが最適です。このトークンタイプを使用すると、アプリはボットの範囲内でメッセージング機能を実行できます。これはこのユースケースには十分であり、最小権限の原則にも合致しています。

他の選択肢ではダメなのか？

*ユーザーOAuthトークンは、特定のユーザーの権限でアクセスを提供しますが、単にメッセージを投稿するだけでは不要です。

*ワークスペースアクセストークンは、Slackアプリの標準トークンタイプではありません。

*クラシックボットトークンは、従来のカスタム統合向けであり、新しいSlackアプリには通常推奨されません。

質問: 72

Slackアプリを開発する際、デバッグのためにユーザーのアクティビティをログに記録する必要があることに気づくでしょう。

このプロセスがセキュリティ原則に合致するようにするための最善策は何ですか？

- A. デバッグに必要な機密性の低い重要な情報のみをログに記録し、ログが安全に保管されていることを確認してください。
- B. 詳細なユーザーアクティビティ（機密データを含む）をログに記録し、包括的なデバッグ情報を取得する。
- C. ユーザーアクティビティのログ記録は常にセキュリティリスクとなるため、避けてください。
- D. ログは、すべての開発者が簡単にアクセスできるよう、共通の共有ドライブに保存します。

正解: **A** ([コメントを发表する](#))

デバッグに必要な機密性の低い重要な情報のみをログに記録し、ログが安全に保存されるようにする。デバッグに必要な機密性の低い重要な情報のみをログに記録し、ログが安全

に保存されるようにすることは、ベストプラクティスです。このアプローチは、ユーザーのプライバシーとセキュリティを尊重しつつ、デバッグに必要な情報を提供します。

他の選択肢ではダメなのか？

*詳細なユーザー活動、特に機密データを記録することは、重大なセキュリティおよびプライバシーリスクをもたらす可能性があります。

ログ記録を完全に避けると、効果的なデバッグが困難になる場合があります。重要なのは、責任を持ってログを記録することです。

*ログを多くの人アクセスする共有ドライブに保存すると、ログデータのセキュリティが損なわれる可能性があります。

質問: 73

ある企業が、ITサポートチケット管理のためにSlackアプリを導入したいと考えている。このアプリは、従業員が技術的な問題を簡単に報告できるようなものでなければならない。チケット作成プロセスを開始するには、どの機能を使用すべきですか？

- A. 従業員に報告すべきIT関連の問題があるかどうかを積極的に尋ねるボットを開発する
- B. 従業員がチケット作成プロセスを開始するために使用できるスラッシュコマンドを作成する
- C. 従業員がIT関連の問題を報告するためのフォームを開くグローバルショートカットを設定する
- D. 従業員が問題を報告できるように、共通のITチャンネルでインタラクティブなメッセージを使用する。

正解: ([正解を表示します](#))

従業員がIT関連の問題を報告するためのフォームを開くグローバルショートカットを設定します。グローバルショートカットは、従業員がITサポートチケットの手続きを開始するための最も使いやすく効率的な方法です。従業員が技術的な問題を詳細に記述できる構造化されたフォームにすぐにアクセスできるため、報告プロセスが簡素化されます。

他の選択肢ではダメなのか？

ボットによる積極的な質問は、プライバシーの侵害とみなされる可能性があります、必ずしも実際のIT問題の発生状況と一致するとは限りません。

*従業員は特定のスラッシュコマンドを覚えて使用する必要があり、グローバルショートカットよりも直感的ではない可能性があります。

共通チャンネルで対話型メッセージを使用すると、IT関連の問題を報告するために必要なプライバシーと構造化された入力確保されない場合があります。

質問: 74

ある企業が、顧客サポートチケットを管理するためのSlackアプリを開発したいと考えている。このアプリは、サポート担当者が効率的にチケットを受信し、対応できるようにするものでなければならない。

どのようなワークフローを導入すべきか？

- A. 顧客がサポートチケットを直接投稿できる公開チャンネルを作成する

- B. プライベートチャンネルでサポートスタッフにチケットを自動的に割り当てるボットを開発する
- C. 顧客がサポートスタッフにサポートチケットを送信できるように、ダイレクトメッセージを使用します。
- D. 顧客からのメールを専用のSlackチャンネルに転送するメール連携機能を実装する

正解: ([正解を表示します](#))

プライベートチャンネルでサポートスタッフにチケットを自動的に割り当てるボットを開発する。プライベートチャンネルでサポートスタッフにチケットを自動的に割り当てるボットは、顧客サポートチケットを管理する効果的な方法です。整理とプライバシーを確保しながら、各チケットの効率的な追跡と対応を可能にします。

他の選択肢ではダメなのか？

公開チャンネルを使用すると、チケットの提出が整理されずに済む可能性があり、顧客のプライバシーが侵害される恐れがあります。

ダイレクトメッセージは、サポートチケットの管理や追跡のための集中管理システムを提供していません。

*メール連携に依存すると、Slackのインタラクティブな機能を十分に活用できず、応答が遅れる可能性があります。

質問: 75

Slackにおける単一ワークスペースアプリと分散型アプリの違いは何ですか？

- A. 単一のワークスペースアプリはワークスペース内のどのユーザーでもカスタマイズできますが、分散アプリの設定はワークスペース管理者のみが制御できます。
- B. 単一ワークスペースアプリは特定のワークスペース内で機能するように設計されているのに対し、分散型アプリは複数のワークスペースにインストールできます。
- C. 分散型アプリは、単一ワークスペースと比較して、より高度なSlack API機能にアクセスできます。
- D. シングルワークスペースアプリは自動的にSlackアプリディレクトリに追加されますが、分散型アプリは承認プロセスを経る必要があります。

正解: ([正解を表示します](#))

単一ワークスペースアプリは、特定のワークスペース内で機能するように設計されていますが、分散アプリは複数のワークスペースにインストールできます。単一ワークスペースアプリと分散アプリの主な違いは、インストールと使用の範囲にあります。単一ワークスペースアプリは、特定のワークスペース内で作成および使用され、その固有のニーズに対応します。対照的に、分散アプリは、複数の異なるワークスペースにインストールして機能するように構築されているため、より幅広いユースケースに適しています。

他の選択肢ではダメなのか？

*アプリ設定をカスタマイズする機能は、これら2種類のアプリを区別しません。

*Slack APIの機能へのアクセスは、一般的に、単一ワークスペースか分散ワークスペースかといったアプリの種類によって制限されることはありません。

※単一ワークスペースアプリの場合、Slackアプリディレクトリへの掲載は自動的に許可されません。

質問: 76

プロジェクト管理チームは、Slackでのワークフローを改善したいと考えています。Slackから離れることなく、プロジェクトの進捗状況や締め切りを追跡できるようにしたいのです。

これを実現するための最善のアプローチは何でしょうか？

- A. プロジェクトのステータスを固定メッセージで定期的に更新する
- B. すべてのプロジェクトの締め切りにはSlackのリマインダーを使用する
- C. Slack内でプロジェクトを管理するためのカスタムアプリを作成する
- D. SlackをAsanaやTrelloなどのプロジェクト管理ツールと連携させる

正解: D ([コメントを发表する](#))

SlackをAsanaやTrelloなどのプロジェクト管理ツールと連携させる SlackをAsanaやTrelloなどのプロジェクト管理ツールと連携させることで、チームはSlack内で更新情報や締め切りを効率的に追跡できます。これにより、シームレスな操作が可能になり、アプリケーションを切り替える必要性が軽減されます。

他の選択肢ではダメなのか？

*Slackのリマインダーは、包括的なプロジェクト管理のニーズには適していません。

カスタムアプリの作成はリソースを大量に消費する可能性があり、既存のプロジェクト管理ツールと同じ機能を提供できない場合もあります。

*手動での更新が必要であり、リアルタイムの追跡や更新には効率的ではありません。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-Dev-201**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> **242**問、**30%**ディスカウント、特別な割引コード: **JPNshiken**」

質問: 77

あなたは、単一のワークスペース内でプロジェクト管理タスクに特化したリアルタイムの更新情報と通知を提供するSlackアプリを開発しました。

なぜこのアプリをそのワークスペースだけに限定したいのですか？

- A. ワークスペースごとにサブスクリプション料金を課金することで、アプリの収益化の可能性を高める
- B. アプリが複数のワークスペースから機密データにアクセスして分析できるようにする

C. アプリの複数のインスタンスによって引き起こされる可能性のある競合や混乱を避けるため

D. アプリのマルチワークスペース管理機能を活用する

正解: ([正解を表示します](#))

アプリの複数のインスタンスによって引き起こされる可能性のある競合や混乱を回避するため、このシナリオではアプリを単一のワークスペースに限定することで、アプリの複数のインスタンスが同じプロジェクト管理タスクを操作する際に発生する可能性のある競合や混乱を回避できます。これにより、通知や更新がそのワークスペース内のプロジェクト管理タスクに特化されるため、データの競合のリスクが軽減されます。

他の選択肢ではダメなのか？

*プレミアム機能の提供など、他の手段を通じて収益化の可能性を実現することも可能です。

複数のワークスペースから機密データにアクセスすることは、封じ込め措置の主な理由ではありません。適切な承認とセキュリティ対策を講じて行う必要があります。

このシナリオでは、アプリのマルチワークスペース管理機能は主な懸念事項ではありません。主な理由は、競合や混乱を回避することです。

質問: 78

あなたは、ワークスペース内のすべてのユーザーのリストとメールアドレスを取得する必要があるSlack連携機能を開発しています。

このタスクには、どのSlack Web APIメソッドを使用すべきですか？

A. `users.profile.get`

B. `include_locale` パラメータ付きの `users.list`

C. ユーザーのアイデンティティ

D. `include_email` パラメータ付きの `users.list`

正解: ([正解を表示します](#))

`include_email` パラメータ付きの `users.list` メソッド `include_email` パラメータ付きの `users.list` メソッドは、ユーザーのリストとメールアドレスを取得するために特別に設計されています。

他の選択肢ではダメなのか？

*`Users.profile.get` は、すべてのユーザーのリストではなく、単一ユーザーのプロファイル情報を取得するために使用されます。

*`users.list`に`include_locale`パラメータを含めると、メールアドレスではなくロケール情報が取得されます。

*`Users.identity`は、認証されたユーザーのID情報を提供するものであり、すべてのユーザーのメールアドレスのリストを提供するものではありません。

質問: 79

Slackアプリの開発にBoltフレームワークを選択すべきなのは、どのようなシナリオでしょうか？

- A. アプリが広範なカスタムサーバー構成と非標準の通信プロトコルを必要とする場合
- B. ボタン、モーダル、スラッシュコマンドなどの機能を備えたインタラクティブなSlackアプリを迅速に開発する場合、高レベルの抽象化を利用します。
- C. アプリ開発において、WebSocket接続の直接的な操作と細かな制御が求められる場合
- D. イベント処理やインタラクティブなコンポーネントを必要としない、Slackのブロックキットのみを使用するアプリを構築する場合

正解: [\(正解を表示します\)](#)

ボタン、モーダル、スラッシュコマンドなどの機能を備えたインタラクティブな Slack アプリを迅速に開発する場合、高レベルの抽象化を活用する Bolt フレームワークは、ボタン、モーダル、スラッシュコマンドなどの機能を備えたインタラクティブな Slack アプリを迅速に開発するのに特に適しています。Bolt は、これらのインタラクティブなコンポーネントの処理を簡素化する高レベルの抽象化を提供し、リッチでユーザーフレンドリーな Slack アプリを効率的に開発できるようにします。

他の選択肢ではダメなのか？

*Boltは、高度なカスタムサーバー構成や非標準プロトコルを必要とするアプリケーションを主な対象としていません。

*Boltは基盤となるWebSocket接続を抽象化するため、これらの接続を直接操作する必要があるシナリオにはあまり適していません。

*BoltはBlock Kitをサポートしていますが、その主な利点は、静的なUIコンポーネントだけでなく、インタラクティブな要素やイベントを処理できる点にあります。

質問: 80

Slackアプリは、ユーザーに代わって複数のチャンネルで操作を実行したり、メッセージを投稿したりするように設計されています。

機能性とセキュリティの両方を確保するには、どのような種類のトークンを要求すべきでしょうか？

- A. クラシックボットトークン
- B. ボットユーザーOAuthトークン
- C. ワークスペースアクセストークン
- D. ユーザーOAuthトークン

正解: [D \(コメントを發表する\)](#)

ユーザーOAuthトークンは、ユーザーに代わってアクションを実行したりメッセージを投稿したりする必要のあるアプリに最適です。このトークンタイプを使用すると、アプリは認証されたユーザーの権限で動作できるため、複数のチャンネルとパーソナライズされた方法でやり取りするのに不可欠です。

他の選択肢ではダメなのか？

*クラシックボットトークンは、従来の統合機能向けに設計されており、最新のSlackアプリでの使用は一般的に推奨されません。

*ワークスペースアクセストークンは、Slackアプリの標準トークンタイプではありません。

*ボットユーザーOAuthトークンはボットの権限で動作しますが、ボットは個々のユーザーに代わってアクションを実行するために必要な権限を持っていない場合があります。

質問: 81

アプリがアクセスできるSlackワークスペース内のすべてのチャンネルのリストを取得するには、どのSlack Web APIメソッドを使用すればよいですか？

- A. conversations.list
- B. チャンネルリスト
- C. ユーザーの会話
- D. channels.info

正解: ([正解を表示します](#))

conversations.list は、Slack ワークスペース内でアプリがアクセスできるすべてのチャンネル (およびその他の種類の会話) のリストを取得するための正しいメソッドです。

他の選択肢ではダメなのか？

*Channels.list は旧式のメソッドであり、現在の API では conversations.list に置き換えられています。

*Users.conversations は Slack の Web API には存在しないメソッドです。

Channels.infoは、アクセス可能なすべてのチャンネルのリストではなく、個々のチャンネルに関する詳細情報を提供します。

質問: 82

Slackのエンタープライズグリッド内でアプリを使用できるように適応させる際、ユーザーの役割と権限に関して考慮すべき重要な点は何ですか？

- A. アプリケーション設計の簡略化のため、エンタープライズグリッド内のすべてのユーザーは同じ役割と権限を持っていると仮定します。
- B. エンタープライズグリッド内のさまざまなワークスペースにおけるユーザーの多様な役割と権限を認識し、適応するようにアプリを設計する。
- C. アプリの使用をエンタープライズグリッド管理者のみに制限し、権限管理を簡素化します。
- D. Slackの組み込み権限システムをバイパスして、ユーザーアクセスをより直接的に制御できるようにアプリを設計する

正解: ([正解を表示します](#))

Enterprise Grid のさまざまなワークスペースにおけるユーザーの役割と権限を認識し、それに合わせてアプリを設計する Enterprise Grid 用のアプリを設計する際には、さまざまなワークスペースにおけるユーザーの役割と権限を考慮することが非常に重要です。これにより、アプリは各ワークスペースで設定された特定のアクセス制御と権限を尊重し、ユーザーの役割に応じてカスタマイズされた安全なエクスペリエンスを提供できます。

他の選択肢ではダメなのか？

グリッド全体で均一な役割と権限を想定すると、セキュリティリスクやカスタマイズ性の低下につながる可能性があります。

アプリの使用を管理者のみに制限すると、他のユーザーが必要とする機能やアクセス性が制限されます。

Slackに組み込まれている権限システムを迂回すると、セキュリティ上の脆弱性が生じ、ワークスペースレベルのアクセス制御の整合性が損なわれる可能性があります。

質問: 83

ある企業が、チームメンバーが同僚に自分の仕事に関するフィードバックを求めることができるように、Slackアプリを導入したいと考えている。

このフィードバック依頼プロセスを開始する最もユーザーフレンドリーな方法は何ですか？

- A. 定期的にフィードバックを求めるリマインダーを送信するボットを設定する
- B. 任意のメッセージからトリガーできるメッセージアクションショートカットを使用して、特定のアイテムに関するフィードバックを要求します。
- C. ユーザーがフィードバックのリクエストを投稿できる専用のフィードバックチャンネルを作成する
- D. フィードバックリクエストフォームを開くためのスラッシュコマンドを実装する

正解: ([正解を表示します](#))

どのメッセージからでも起動できるメッセージアクションショートカットを使用して、特定のアイテムに関するフィードバックをリクエストできます。メッセージアクションショートカットを使用すると、メッセージから直接フィードバックをリクエストできるため、プロセスが状況に応じて分かりやすくなります。これにより、チームメンバーは会話を中断することなく、特定のメッセージやコンテンツに関するフィードバックをすばやく求めることができます。

他の選択肢ではダメなのか？

定期的なリマインダーは、特定のフィードバック依頼に対しては、的を絞った即時的な対応とは言えません。

専用のフィードバックチャンネルは混乱を招く可能性があり、特定のメッセージに基づいて行動するよりも直接性に欠ける。

*コマンドを覚えて入力する必要があるため、メッセージから直接操作するよりも不便です。

質問: 84

ある企業が、社内求人情報を掲載するためのSlackアプリを開発している。このアプリを使えば、従業員は掲載された求人情報に簡単にアクセスして応募できるようになる。

求人情報にアクセスするための、最も効果的なユーザーエントリーポイントはどれでしょうか？

- A. ボットを設定して、全従業員にダイレクトメッセージで求人情報を送信する。
- B. 求人掲載専用のチャンネルを作成し、各求人情報にメッセージボタンを設定して応募フォームにアクセスできるようにする。
- C. 従業員が現在の求人情報を閲覧するために使用できるスラッシュコマンドを実装する

D. 従業員がアプリのホーム画面に移動して求人情報を閲覧し、応募することを義務付ける
正解: ([正解を表示します](#))

専用の求人掲載チャンネルを作成し、各求人情報にメッセージボタンを設定して応募フォームにアクセスできるようにします。各求人情報にメッセージボタンを備えた専用の求人掲載チャンネルは、従業員が求人に応募するための直感的で一元化された方法です。このアプローチにより、アクセスと視認性が向上し、従業員のエンゲージメントが促進されます。

他の選択肢ではダメなのか？

*全従業員にダイレクトメッセージで求人情報を送信することは、迷惑で負担が大きい場合があります。

スラッシュコマンドは便利ではあるものの、従業員が特定のコマンドを覚えておく必要があります。専用チャンネルほど高い可視性は得られない可能性がある。

*アプリのホーム画面へのナビゲーションが必要になると、不要な手順が増え、求人情報の視認性とアクセス性が低下します。

質問: 85

Slackアプリが短時間内に多数のAPI呼び出しを実行する必要がある場合、レート制限に引っかからないようにするにはどのような戦略を採用すべきでしょうか？

- A. 必要なAPI呼び出しをすべて遅延なく迅速に連続して実行します
- B. スロットリングメカニズムを使用して、API呼び出しを時間的に均等に分散する
- C. 複数のユーザートークンを使用してAPI呼び出しの量を分散します
- D. API呼び出しを圧縮して、より少ない、より大きなリクエストにまとめます。

正解: ([正解を表示します](#))

スロットリングメカニズムを使用して、API呼び出しを時間的に均等に分散する レート制限に達することを避けるため、スロットリングメカニズムを使用してAPI呼び出しを時間的に均等に分散することをお勧めします。このアプローチにより、呼び出しの間隔を空けることで、アプリがレート制限の範囲内に収まることが保証されます。

他の選択肢ではダメなのか？

短時間に多数の通話を行うと、通話制限が適用される可能性が高くなります。

*複数のトークンを使用してレート制限を回避することはベストプラクティスに反するため、SlackのAPIの悪用とみなされる可能性があります。

*呼び出しを圧縮してより少ない数の大きなリクエストにすることは、アプリが必要とするAPIインタラクションの性質によっては、常に実現可能または効果的とは限りません。

質問: 86

貴社は会議室の予約管理を行うためのSlackアプリを開発中です。このアプリを使えば、従業員は会議室の空き状況を素早く確認し、予約することができます。

これを最も効果的に実施する方法は何でしょうか？

- A. 従業員が会議室予約リクエストを投稿できるチャンネルを作成する
- B. 従業員がメッセージを送って部屋を予約できるボットを開発する

C. 空室状況を表示し、即時予約を可能にするスラッシュコマンドを使用します。

D. アプリのホーム画面からアクセスできる、部屋予約用のモーダルフォームを実装する

正解: ([正解を表示します](#))

空室状況を表示し、即座に予約できるスラッシュコマンドを使用してください。スラッシュコマンドは、従業員が部屋の空室状況を確認し、予約を行うための直接的かつ効率的な方法です。複数の手順やチャネルを経由することなく、部屋の状況と予約機能に即座にアクセスできます。

他の選択肢ではダメなのか？

チャンネルに投稿すると、混乱が生じる可能性があり、リアルタイムでの返信が保証されるわけではありません。

ボットへのメッセージ送信は、直接的なやり取りが少なく、追加の手順が必要になる場合があります。

*アプリのホーム画面からモーダルフォームにアクセスすると、予約プロセスに不要な手順が追加されますが、スラッシュコマンドを使用するとプロセスを効率化できます。

有効的な**Slack-Dev-201**問題集はJPNTTest.com提供され、**Slack-Dev-201**試験に合格することに役に立ちます！JPNTTest.comは今最新**Slack-Dev-201**試験問題集を提供します。JPNTTest.com Slack-Dev-201試験問題集はもう更新されました。ここで**Slack-Dev-201**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Slack-Dev-201-mondaishu> **242**問、**30%ディスカウント**、特別な割引コード: **JPNshiken**」