



## Developing Applications using Cisco Core Platforms and APIs v1.0(350-901)

**試験概要:** Developing Applications using Cisco Core Platforms and APIs v1.0(DEVCOR 350- 901)は、DevNet Professional 認定に関する試験であり、試験時間は 120 分です。この試験では、APIの使用、シスコ プラットフォーム、アプリケーションの展開とセキュリティ、およびインフラストラクチャと自動化など、ソフトウェア開発に関する受験者の知識が問われます。本試験の受験対策として、Developing Applications using Cisco Core Platforms and APIs コースの受講をお勧めします。

次に、この試験の一般的な出題内容を示します。ただし、試験によっては、ここに示されていない関連項目も出題される場合があります。試験内容をより適切に反映し、明確にするために、次のガイドラインは予告なく変更されることがあります。

- 20%    1.0    **ソフトウェアの開発と設計**
  - 1.1    分散アプリケーションについての説明(フロントエンド、バックエンド、負荷分散の概念との関連において)
  - 1.2    スケーラビリティおよびモジュラリティを考慮したアプリケーション設計の評価
  - 1.3    ハイアベイラビリティおよび復元力を考慮したアプリケーション設計の評価(オンプレミス、ハイブリッド、クラウドを含む)
  - 1.4    遅延およびレート制限を考慮したアプリケーション設計の評価
  - 1.5    保守性を考慮したアプリケーション設計および実装の評価
  - 1.6    可観測性を考慮したアプリケーション設計および実装の評価
  - 1.7    イベントに関連するログに基づいたアプリケーションの問題の診断
  - 1.8    アプリケーション要件を考慮したデータベース タイプの選択の評価(リレーショナル、ドキュメント、グラフ、カラムナー、Time Series など)
  - 1.9    アーキテクチャ パターンの説明(モノリシック、サービス指向、マイクロサービス、イベント駆動型)
  - 1.10   Git でよく使用される高度なバージョン管理手法の活用
    - 1.10.a   ブランチのマージ
    - 1.10.b   コンフリクトの解消
    - 1.10.c   git reset
    - 1.10.d   git checkout
    - 1.10.e   git revert
  - 1.11   リリース パッケージングおよび依存関係管理の説明
  - 1.12   API コールを含むシーケンス図の作成

- 20% 2.0 **API の使用**
  - 2.1 タイムアウトやレート制限のための堅牢な REST API エラー処理の実装
  - 2.2 コンシューマコードの回復不能な REST API エラーに対する制御フローの実装
  - 2.3 HTTP のキャッシュコントロールを通じて API 使用を最適化する方法の特定
  - 2.4 ページネーションに対応した REST API を消費するアプリケーションの作成
  - 2.5 OAuth2 許可コード (three-legged) を付与するフローの各ステップの説明
  
- 20% 3.0 **シスコプラットフォーム**
  - 3.1 Webex Teams API を使用して Chatops を実装する API 要求の作成
  - 3.2 Firepower デバイス管理 (FDM) を使用してオブジェクトを作成、削除する API 要求の作成
  - 3.3 Meraki プラットフォームを使用して以下のタスクを遂行する API 要求の作成
    - 3.3.a Meraki ダッシュボード API を使用した SSID の有効化
    - 3.3.b Meraki ロケーション API を使用した位置データの取得
  - 3.4 Intersight からデータを取得する API コールの実装
  - 3.5 UCS API を使用して、テンプレートで指定された新しい UCS サーバのプロビジョニングを行う Python スクリプトの実装
  - 3.6 Cisco DNA Center API を使用してワイヤレスのヘルス情報を取得する Python スクリプトの実装
  - 3.7 アプリケーションのインストール時における AppDynamics の機能の説明
  - 3.8 Cisco API から収集したデータを表示するカスタム ダッシュボードを構築する手順の説明
  
- 20% 4.0 **アプリケーションの展開とセキュリティ**
  - 4.1 CI/CD パイプライン機能の診断 (依存関係の喪失、バージョン互換性のないコンポーネント、テストエラーなど)
  - 4.2 Docker および Kubernetes を活用するためにプレビルドされた CD 環境へのアプリケーションの統合
  - 4.3 CI パイプラインでの連続性テストおよび静的コード解析のメリットの説明
  - 4.4 Docker を活用したアプリケーションのコンテナ化
  - 4.5 Twelve Factor App の原理の説明
  - 4.6 アプリケーションにとって効果的なロギング戦略についての説明
  - 4.7 データのストレージおよび伝送に関連するデータの機密性に関する懸念についての説明
  - 4.8 指定されたシナリオに適合する機密情報保管のアプローチの特定
  - 4.9 アプリケーション固有の SSL 証明書の構成
  - 4.10 OWASP (XSS、CSRF、SQL インジェクションなど) の脅威に対する軽減戦略の実装
  - 4.11 エンドツーエンド暗号化原理の API への適用方法の説明
  
- 20% 5.0 **インフラストラクチャと自動化**
  - 5.1 モデル駆動型テレメトリにおける検討事項の説明 (データ消費およびデータ保管を含む)
  - 5.2 RESTCONF の活用によるネットワーク デバイス (インターフェイス、スタティック ルータ、VLAN (IOS XE のみ)) の構成
  - 5.3 以下の手段を使用してネットワーク パラメータを構成するための作業フローの実装:
    - 5.3.a Ansible プレイブック
    - 5.3.b Puppet マニフェスト

- 5.4 技術的およびビジネス的要件を達成するためのコンフィグレーション マネジメント ソリューションの特定
- 5.5 ネットワーク デバイス (Catalyst 9000 および Cisco IOX 対応デバイスを含む) 上にアプリケーションをホストする方法の説明