アクイア認定
バックエンドスペシャリスト - Drupal 9
学習ガイド

本資料は「アクイア認定バックエンドスペシャリスト - Drupal 9」試験の学習ガイドです。このガイドでは、試験の構成とトピックの解説、試験で出題される問題の性質を理解することに役立ちます。

---

1. 試験概要

試験名　　：アクイア認定バックエンドスペシャリスト - Drupal 9
試験時間　：90分
受験料　　：350ドル
問題数　　：60問
合格ライン：70%

アクイア認定バックエンドスペシャリストは、バックエンド開発（コーディング）の分野におけるDrupal開発者のスキルおよび知識の検証を目的としています。 この試験はDrupal 9をベースとしています。

この試験は、受験者の次の能力を検証します。

- Drupal 9をベースとしたソリューションの設計、開発、デプロイ
- バックエンド開発に関するDrupalコアの基本的なアーキテクチャに関するベストプラクティスの理解
- 新規のDrupal 9モジュールの開発、実装
- 既存のDrupal 9モジュールのカスタマイズと拡張

このレベルで必要となる基本的な知識およびスキルには、次のエリアおよび対象コンポーネントがすべて含まれます。

- Drupalテクノロジーを用いた専門的な経験
- Drupalサイトのセットアップと構成
- 新規モジュールの開発や既存モジュールのカスタマイズ
- HTML、CSS、JavaScript/jQuery、オブジェクト指向PHPに関する知識

2. 認定取得までの道のり

Drupal入門コースを受講する

試験を受験予定の全ての方は、Acquia AcademyのDrupal入門コースに無料でアクセスできます。Drupal開発者向けに次のコースを提供しています。Acquia Academyにログインしてご活用ください。

- Drupal 9 Site Building
- Drupal Layout/Theming Training
- Drupal 9 Module Development

コースを受講した後は、コースで学んだ概念を補強するために、自分で小さなウェブサイトを構築してみましょう。

実務経験を積む

コースを修了したからといって、認定試験を受ける準備ができているとは限りません。試験を受験するためには、いくつかの実務経験が必要です。少なくとも2〜3のプロジェクトに取り組み、実務経験を積むことをお勧めします。

自己評価を行う

試験を受けることを決めたら、試験でカバーされている領域とトピック、およびそれらの相対的な重みを勉強することをお勧めします。また、試験の構成でカバーされているドメイン/トピックに対して、自分の得意分野と不得意分野の自己評価を行う必要があります。これは経験豊富なDrupal開発者にお勧めの出発点です。

学習ガイドを活用する

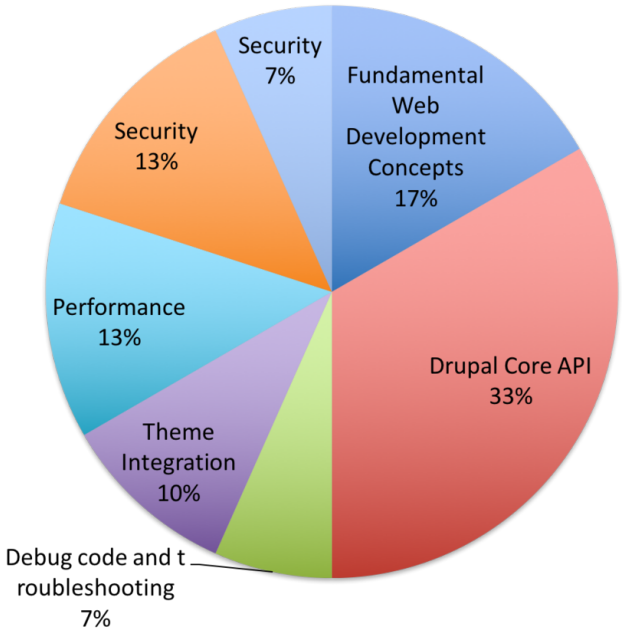
自己評価に基づいて、いくつかのドメインやトピックの知識や経験を得る必要があるかもしれません。Drupalを学ぶ最善の方法は、実際に作業したり、実験したりすることです。本ガイドの試験の準備やリソース集が参考になります。

試験を受ける

この段階で、あなたは試験を受ける準備ができているはずです。アクイア認定 受験方法のページにアクセスして受験方法を確認します。

3. 試験ドメイン

ドメイン	配分
1.0 基本的なウェブ開発概念	17%
2.0 Drupal Core API	33%
3.0 コードのデバッグおよびトラブルシューティング	7%
4.0 テーマのインテグレーション	10%
5.0 パフォーマンス	13%
6.0 セキュリティ	13%
7.0 コミュニティの活用	7%
合計	100%



4. 自己評価

試験のドメインとトピックに関して、スキルと専門知識の自己評価を行うことをお勧めします。各トピックに対して、あなたのスキルを0〜10段階でランク付けします。知識や経験が全くない場合は0を、そのトピックを完全にマスターしている場合は10とします。

ドメイン	トピック	スコア
1	**基本的なウェブ開発概念**	
1.1	HTML、CSS およびJavaScriptに関する知識を実証	
1.2	PHPオブジェクト指向プログラミングの概念に関する知識を実証	
1.3	Composerを用いた依存関係管理に関する知識を実証	
1.4	バージョン管理におけるGitの使用を実証	
1.5	自動テスト概念に関する知識を実証	
2	**Drupal Core API**	
2.1	ルーティングシステムやMenu APIを使用して、DrupalでのURLリクエストの処理方法を定義するためのパスを登録する能力を実証	
2.2	Form APIを使用してフォームを構築、変更、検証、送信する能力を実証	
2.3	Entity APIを使用してエンティティシステムとの対話能力を実証	
2.4	Drupalの機能を構築・拡張するためのコアAPIを使用する能力を実証	
3	**コードのデバッグとトラブルシューティング**	
3.1	コードのデバッグ能力を実証	
3.2	サイトに関する問題のトラブルシューティング能力を実証	
4	**テーマのインテグレーション**	
4.1	Render APIやJavaScript APIを使用してDrupalのテーマシステムと連携する能力を実証	
4.2	Twigのテンプレートおよび構文を使用する能力を実証	
5	**パフォーマンス**	
5.1	サイト構成により発生するサイトパフォーマンスの問題の分析、解決能力を実証	

5.2	カスタムコードにより発生するサイトパフォーマンスの問題の分析、解決能力を実証	
5.3	Cache APIを用いたDrupalキャッシュ戦略の実装	
6	**セキュリティ**	
6.1	サイト構成により発生するセキュリティの問題の分析、解決能力を実証	
6.2	カスタムコードにより発生するセキュリティの問題の分析、解決能力を実証	
6.3	Drupalコアのセキュリティメカニズムの実装能力を実証	
7	**コミュニティの活用**	
7.1	コミュニティに貢献する能力を実証	
7.2	Drupalコーディング規約を用いたコーディング能力を実証	

5. 試験の準備

準備が必要なトピックのリストができたら、次の2つの課題に直面することになります。

1. トピックのための文書や知識のリソースを探して勉強する
2. 学ぶべきトピックを網羅した十分な実践的なケーススタディを見つける

知識リソース

入門的なDrupalトレーニングを、ドキュメントとオンラインリソースで補うのは良い考え方です。本ガイドのリソース集に、各学習領域のオンラインリソースとドキュメントのリストがあります。

実践的なケーススタディ

2つ目の課題は、トピックをカバーする十分な実用的な事例を見つけることです。ここでは、準備のための事例を見つけるためのいくつかのアイデアと提案を紹介します。

バックエンド開発

バックエンド開発やモジュール開発に携わったことがない場合、この試験の準備は困難な作業になる可能性があります。非常に複雑なモジュールや、要件やユースケースに固有である可能性があるため、あなたのチームが開発したモジュールのコードを調査することは役に立つ場合と役に立たない場合があります。この試験では、DrupalのAPI、コアプラグインシステム、およびそれらのアプリケーションを総合的に理解している必要があります。

このドメインの準備のために、モジュール開発の有償トレーニングに参加したり、シンプルなモジュールを自作することをお勧めします。

独学で学習している場合は、モジュールのサンプルコードが詰まっているExamples for Developersモジュールも参考になると思います。このモジュールでは、4.1項と4.2項でカバーされているDrupal APIやコンセプトについて、高度にドキュメント化された作業コードのスニペットを提供しています。

6. リソース集

Drupal開発のために知っておくべきツール

Drupalコーディング規約

Drupalコミュニティで適用されているコーディングのベストプラクティスと規約の概要です。

これらのコーディング規約は、Drupalの開発中にコードが生成される可能性のあるすべての領域について詳細に記述されており、コミュニティで広く確立されたベストプラクティスを表しています。これらのベストプラクティスに従うことで、開発がよりスムーズに進み、Drupalコアやモジュールへの貢献がより容易に受け入れられるようになります。アクイアのテクニカルサポートを利用してプロジェクトに取り組んでいる場合、これらはプルリクエストが遵守しなければならないコーディング基準です。

Drush

DrushはDrupalサイト開発の基本的なスキルと考えられています。

トピック別 参照資料

1.0 基本的なウェブ開発概念

1.1 HTML、CSS およびJavaScriptに関する知識を実証

- 開発者のためのWebテクノロジー (Mozilla)
- HTML5 Doctor
- Webplatform.org
- CSS-Tricks
- *.libraries.ymlを介したアセットの追加
- JavaScript API

1.2 PHPオブジェクト指向プログラミングの概念に関する知識を実証

- CodePen
- PHP.net
- JavaScript (Mozilla)
- jQuery API

1.3 Composerを用いた依存関係管理に関する知識を実証

- Composerを使用する

1.4 バージョン管理におけるGitの使用を実証

- チュートリアル：GitでBitbucketを理解する
- Gitワークフロー

1.5 自動テスト概念に関する知識を実証

- 自動テスト
- 自動テスト (Drupal API)

2.0 Drupal Core API

2.1 ルーティングシステムやMenu APIを使用して、DrupalでのURLリクエストの処理方法を定義するためのパスを登録する能力を実証

- カスタムモジュールの作成
- 背景と前提知識 - カスタムモジュールの作成
- 名前空間
- Drupal 8におけるPSR-4の名前空間とオートローダー
- 名前空間 概要 (PHP.net)

2.2 Form APIを使用してフォームを構築、変更、検証、送信する能力を実証

- Form API

2.3 Entity APIを使用してエンティティシステムとの対話能力を実証

- Entity API

2.4 Drupalの機能を構築・拡張するためのコアAPIを使用する能力を実証

- public static function Drupal::service
- core.services.yml
- Block API 概要
- Cache API
- サービスと依存性注入
- サービスと依存性注入コンテナ
- Plugin API
- views.api.php
- 全てのDrupal API一覧

3.0 コードのデバッグとトラブルシューティング

3.1 コードのデバッグ能力を実証

- 開発ツール
- Xdebugによるデバッグ

3.2 サイトに関する問題のトラブルシューティング能力を実証

- Develモジュール
- トラブルシューティング
- Drupalのトラブルシューティング

4.0 テーマのインテグレーション

4.1 Render APIやJavaScript APIを使用してDrupalのテーマシステムと連携する能力を実証

- Drupalのテーマ開発
- *.libraries.ymlを介したアセットの追加
- JavaScript API 概要

4.2 Twigのテンプレートおよび構文を使用する能力を実証

- DrupalのTwig
- Twigテンプレートのデバッグ

5.0 パフォーマンス

5.1 サイト構成により発生するサイトパフォーマンスの問題の分析、解決能力を実証

- Drupal 8を飛ばす

5.2 カスタムコードにより発生するサイトパフォーマンスの問題の分析、解決能力を実証

- Drupalのプロファイリング

5.3 Cache APIを用いたDrupalキャッシュ戦略の実装

- Cache API
- Cache API (Drupal API)

6.0 セキュリティ

6.1 サイト構成により発生するセキュリティの問題の分析、解決能力を実証

- Filterモジュール 概要

6.2 カスタムコードにより発生するセキュリティの問題の分析、解決能力を実証

- Drupalの安全なコードを書く

6.3 Drupalコアのセキュリティメカニズムの実装能力を実証

- Drupalのセキュリティ

7.0 コミュニティの活用

7.1 コミュニティに貢献する能力を実証

- 貢献エリア

7.2 Drupalコーディング規約を用いたコーディング能力を実証

- コーディング規約

本ガイドは以上になります。