

Advanced Drupal 10 Interview Questions & Answers

1. How would you debug a Symfony event subscriber that isn't firing?

A Symfony event subscriber is a class that listens to events in Drupal. If it's not firing, make sure:

- It implements `EventSubscriberInterface`
- It is correctly registered in your module's `services.yml` with the `'event_subscriber'` tag
- The event name matches what Drupal emits.

Example:

`services:`

```
my_module.event_subscriber:
  class: Drupal\my_module\EventSubscriber\MySubscriber
  tags:
    - { name: event_subscriber }
```

2. What are cache contexts, tags, and max-age in Drupal?

Drupal's caching system relies on three components:

- **Cache Contexts**: Vary cache by dynamic elements like user, language, or URL.
- **Cache Tags**: Attach identifiers (e.g., `node:5`) to allow selective invalidation.
- **Max-age**: Time-based expiry (e.g., 3600 seconds).

Example:

```
return [
  '#markup' => 'Hello',
  '#cache' => [
    'contexts' => ['user.roles'],
    'tags' => ['node:5'],
    'max-age' => 600,
  ],
];
```

3. What are the common challenges in migrating from Drupal 7 to Drupal 10?

Migrating from Drupal 7 to 10 involves many layers:

- Code changes (no more `hook_menu`, use YAML and controllers).
- Theme changes (Twig replaces `PHPTemplate`).
- Module changes (many contrib modules deprecated).

Solution:

- Use Migrate API, Migrate Plus, and Migrate Tools.
- Validate source data.
- Back up before each batch.

Advanced Drupal 10 Interview Questions & Answers

4. How do you create and use a custom Drush command?

Drush lets you automate CLI operations like rebuilding cache or triggering actions.

To write a custom Drush command:

1. Create a class extending `DrushCommands`
2. Use `@command` annotation.

Example:

```
```php
class MyCommands extends DrushCommands {
 /**
 * @command mymodule:say-hi
 */
 public function sayHi() {
 $this->output()->writeln('Hello!');
 }
}
```
```

5. What is the difference between Config and State API in Drupal?

Use **Config API** for values that go in Git (e.g., site name), and **State API** for runtime, env-specific values (e.g., last cron time).

Example Config:

```
$site_name = \Drupal::config('system.site')->get('name');
```

Example State:

```
\Drupal::state()->set('my_module.last_run', time());
```

6. How do you make a custom entity translatable and revisionable?

To make a custom content entity translatable and revisionable:

- Use `translatable = TRUE` and `revisionable = TRUE` in the annotation.
- Enable field-level translation.

Example:

```
@ContentEntityType(
    id = 'employee',
    label = @Translation('Employee'),
    translatable = TRUE,
    revisionable = TRUE,
    base_table = 'employee',
    ...
)
```

Advanced Drupal 10 Interview Questions & Answers

)

7. How would you migrate file and media fields from Drupal 7 to Drupal 10?

Migrating media fields from D7 to D10 involves:

- Migrating files with the file migration
- Creating media entities that wrap those files
- Referencing media in content migrations via entity_lookup process plugin.

8. Explain the use of services.yml and the service container in Drupal.

Drupal uses the Symfony service container. Define services in yourmodule.services.yml.

Example:

```
services:
  my_module.mailer:
    class: Drupal\my_module\Mailer
    arguments: ['@logger.factory']
```

9. How do you prevent cron batch jobs from timing out in large data sets?

To avoid timeouts in cron batch jobs:

- Use sandbox values to track progress
- Set small batch sizes
- Ensure finished = 1 only after all items are processed.

Example:

```
$context['sandbox']['progress']++;
if ($context['sandbox']['progress'] >= $max) {
  $context['finished'] = 1;
}
```

10. What are render arrays in Drupal and how are they useful?

Render arrays are structured arrays that Drupal converts into HTML. They support theming, caching, and AJAX.

Example:

```
return [
  '#markup' => '<p>Hello World</p>',
  '#cache' => ['max-age' => 0],
];
```

11. How do you override a form alter from a third-party module?

Use hook_form_FORM_ID_alter() or hook_form_alter() to override third-party forms. Check

Advanced Drupal 10 Interview Questions & Answers

the form ID from HTML source or devel.

Example:

```
function my_module_form_user_login_form_alter(&$form, &$form_state, $form_id) {  
  $form['name']['#title'] = t('Custom Username');  
}
```

12. How can you dynamically generate routes using RouteSubscriber?

Use RouteSubscriberBase to dynamically alter existing routes or paths in your module.

Example:

```
class MyRouteSubscriber extends RouteSubscriberBase {  
  protected function alterRoutes(RouteCollection $collection) {  
    if ($route = $collection->get('user.login')) {  
      $route->setPath('/custom-login');  
    }  
  }  
}
```

13. What are Traits in PHP and how are they used in Drupal code?

Traits allow sharing common methods across classes. In Drupal, they're used for translation, string building, etc.

Example:

```
use StringTranslationTrait;  
class MyClass {  
  use StringTranslationTrait;  
  public function sayHello() {  
    return $this->t('Hello!');  
  }  
}
```

14. How do you optimize performance for authenticated users?

Authenticated users bypass the page cache, so their experience depends heavily on render performance. To optimize:

- Use render caching: Add proper #cache metadata to render arrays.
- Lazy builders: Delay building parts of the page until requested.
- BigPipe: Sends static page parts first, then dynamic parts after.

These techniques improve perceived and actual performance for logged-in users.

15. Explain the Drupal 10 theming workflow and key files involved.

.info.yml for metadata, .libraries.yml for CSS/JS, .html.twig for templates, and

Advanced Drupal 10 Interview Questions & Answers

theme.theme file for preprocess hooks.

Use Twig Debugging and Devel module to inspect variables in templates.

16. What steps do you follow to create a secure file upload field?

Use managed_file field with #upload_validators and store uploads in private:// to restrict access.

Example:

```
$form['upload'] = [
  '#type' => 'managed_file',
  '#upload_location' => 'private://resumes/',
  '#upload_validators' => [
    'file_validate_extensions' => ['pdf doc'],
  ],
];
```

17. What is a computed field, and how do you implement it in a custom entity?

Computed fields derive values dynamically. Use DataDefinition, setComputed(TRUE), and define a computed class.

Common in scenarios like full name = first + last name.

18. Explain how Layout Builder integrates with field formatters.

Layout Builder allows custom placement of fields and uses field formatters to control their display. You can create your own by extending FormatterBase.

19. What is a plugin manager, and how would you create a custom plugin type?

A plugin manager in Drupal allows you to define and manage pluggable components such as blocks, field types, and views. To create a custom plugin type:

- Create an interface (e.g., MyPluginInterface).
- Create a plugin annotation (e.g., @MyPlugin).
- Create a plugin manager class extending DefaultPluginManager.
- Implement discovery and instantiation logic.

This enables extensibility and modularity in your application.

20. How would you expose a REST endpoint for a custom entity type?

Extend ResourceBase, implement GET/POST logic. Add permissions and path via @RestResource annotation.

Protect endpoint with user permissions and CSRF token if needed.

Advanced Drupal 10 Interview Questions & Answers

21. Describe the difference between functional, kernel, and unit testing in Drupal.

Testing types in Drupal:

- Unit tests: test PHP logic without bootstrapping Drupal.
- Kernel tests: load service container and database schema, test configuration and entity APIs.
- Functional tests: simulate full web interactions using browser-like tools.

Example (Kernel):

```
class MyKernelTest extends KernelTestBase { ... }
```

22. How do you safely manage configuration deployment across environments?

Safe config deployment includes:

- Using 'drush config:export' (cex) in dev.
- Using 'drush config:import' (cim) in staging/prod.
- Handling environment-specific config with 'config_split'.
- Using Git to track changes.

This ensures consistency across environments.

23. What are hook_entity_presave() and hook_entity_insert(), and when do you use them?

hook_entity_presave(): Called before an entity is saved. Use for validation or setting default values.

hook_entity_insert(): Called after an entity is saved. Use for logging or triggering additional logic.

Example:

```
function mymodule_entity_insert($entity) {  
  \Drupal::logger('custom')->notice('Saved new entity: ' . $entity->id());  
}
```

24. Explain how Drupal handles multilingual content and interface translation.

Drupal handles multilingual capabilities through:

- Interface Translation: Translates system UI strings using the Locale module and .po files.
- Content Translation: Enabled per entity type or field using the content_translation module.

Use t() or StringTranslationTrait in code for interface translation.

Example: \$this->t('Welcome user');

25. How do you handle conditional Webform submissions or branching logic?

Webform supports dynamic behavior like showing/hiding fields or conditional validation.

Advanced Drupal 10 Interview Questions & Answers

- Use States API for UI behavior.
- Use 'Conditions' tab in UI for logic branching.
- Add custom logic in `hook_webform_submission_presave()`.

Example:

```
function mymodule_webform_submission_presave(WebformSubmissionInterface $sub) {
    $data = $sub->getData();
    if ($data['email'] && !str_contains($data['email'], '@company.com')) {
        \Drupal::messenger()->addWarning('Please use company email.');
```

26. What is the purpose of schema.yml files in Drupal modules?

.schema.yml defines the structure and types for config values.

- Used by ConfigFormBase and config validation tools.
- Required for translatability and clarity.

Example:

```
my_module.settings:
  type: config_object
  label: 'Module Settings'
  mapping:
    message:
      type: string
      label: 'Custom Message'
```

27. How do you write a custom Views filter plugin?

Create custom Views filters by extending FilterPluginBase.

- Use annotations to register your filter plugin.
- Implement query() to add custom conditions.

Example:

```
class MyCustomFilter extends FilterPluginBase {
    public function query() {
        $this->query->addWhere(0, 'my_field', $this->value);
    }
}
```

28. What's the best way to implement Single Sign-On (SSO) in Drupal 10?

For SSO in Drupal 10:

- Use SimpleSAMLphp for SAML.
- Use OAuth / OpenID Connect for modern identity providers.

Advanced Drupal 10 Interview Questions & Answers

- Configure secure endpoints and test thoroughly.

Use roles mapping to control permissions per user group.

29. How do you apply role-based access control to custom routes?

To apply access control to custom routes:

- Define `_role` or `_permission` in `routing.yml`.
- Optionally use a custom access service returning `AccessResult`.

Example:

requirements:

```
_permission: 'access secure content'
```

30. What are deprecated APIs in Drupal 7 and how to replace them in Drupal 10?

Drupal 7 APIs like `hook_menu` and `db_query` are deprecated.

Use `routing.yml`, `controllers`, and `database services` instead.

Examples:

- `drupal_set_message` -> `\Drupal::messenger()->addStatus()`
- `db_query()` -> `Database::getConnection()->select()`

31. What are the strategies to reduce downtime during a large-scale migration?

To reduce downtime during large-scale migrations:

- Use `'--update'` flag to perform incremental migrations.
- Freeze content entry before final sync.
- Validate migration results in staging.
- Use queues or batch processing to distribute migration steps.

32. How do you debug a failing Drupal migration process?

To debug a failing migration:

- Use `'drush migrate:messages'` to check logged errors.
- Add debug logging in `prepareRow()/transform()` methods.
- Run migrations with `'--debug'` flag.
- Validate source data integrity.

33. How do you use Drupal's queue system for background processing?

Drupal's queue system enables background task processing. Use `@QueueWorker` annotation and implement `processItem()`.

Example:

```
@QueueWorker(id="my_queue")
```


Advanced Drupal 10 Interview Questions & Answers

```
class MyQueueWorker extends QueueWorkerBase {  
    public function processItem($data) {  
        // Process logic  
    }  
}
```

34. How do you prevent duplicate content during migration from D7?

Prevent duplicate content by setting unique IDs using sourceid1 in migration config. Use '--update' to update existing content.

Example:

destination:

```
plugin: entity:node  
default_bundle: article
```

source:

```
plugin: d7_node  
node_type: article
```

35. What is dependency injection and how is it used in Drupal services?

Dependency Injection is a design pattern that allows services to receive dependencies via their constructor.

Example:

```
class MyService {  
    public function __construct(LoggerInterface $logger) {  
        $this->logger = $logger;  
    }  
}
```

36. How do you override a base theme's libraries without altering its code?

To override a base theme's libraries:

- Use 'libraries-override' in your theme's .info.yml
- Avoid editing the base theme directly

Example:

libraries-override:

```
classy/global-styling:  
  css/component/links.css: false
```

37. What is Typed Data API in Drupal, and where is it used?

Typed Data API provides metadata and validation for entity fields and config. Used heavily in Field API.

Advanced Drupal 10 Interview Questions & Answers

Example:

```
$value = $node->get('field_age')->first()->getValue();
```

38. How do you create a custom token for use in email templates?

To create a custom token:

- Implement `hook_token_info()` and `hook_tokens()`.
- Return your custom values as replacements.

Example:

```
function mymodule_tokens(...) {  
  $replacements['[custom:greeting]'] = 'Hello!';  
}
```

39. How would you design a multisite architecture for shared content types?

Multisite setup shares codebase but has different databases. To share content types:

- Use installation profiles or modules to sync content types.
- Use `config_split/config_ignore` to handle differences.
- Use APIs or entity replication to share content.

40. What's the proper way to create and attach a library to a specific Twig template?

Attach a library to a specific Twig template using preprocess hook.

Example:

```
function mytheme_preprocess_node(&$variables) {  
  $variables['#attached']['library'][] = 'mytheme/my_custom_library';  
}
```

41. How would you expose entity data to a React frontend securely?

Use JSON:API or a custom REST resource to expose entity data. Use permissions and OAuth for secure access.

Use serialization to restrict fields and secure access with roles or scopes.

42. How do you avoid configuration conflicts when multiple teams are involved?

Avoid config conflicts:

- Use `'config_split'` for environment-specific settings.
- Pull latest config with `'drush cex && drush cim'` regularly.
- Avoid editing YAML files manually; use Config UI or Features module.

43. How would you implement access control for a complex content workflow?

Advanced Drupal 10 Interview Questions & Answers

Use the Workflow and Content Moderation modules to define transitions like Draft -> Review -> Published.

- Assign permissions per transition.
- Use access control hooks or workflows for complex rules.

44. What is the difference between #theme and #markup in render arrays?

#markup outputs raw HTML while #theme renders via a Twig template with structured variables.

Use #theme for complex rendering and separation of logic from presentation.

45. How do you troubleshoot slow Views queries in Drupal?

Slow Views can be optimized by:

- Indexing custom fields.
- Using exposed filters carefully.
- Reviewing joins via Devel module's query log.

46. What is a preprocess hook and when should you use it over Twig logic?

Preprocess hooks allow you to set or transform variables before Twig rendering.

- Use them for logic.
- Keep Twig templates clean.

Example:

```
$vars['my_title'] = strtoupper($vars['label']);
```

47. How do you write and register a custom field widget?

Custom widgets define how a field appears in a form.

Use @FieldWidget, extend WidgetBase, and implement formElement().

48. How do you extend the Migrate API to support custom source plugins?

To extend the Migrate API:

- Create a class extending SourcePluginBase.
- Implement initialize(), getIds(), and prepareRow().
- Add via @MigrateSource annotation.

49. How do you set up a private file system and use it in image fields?

Private file storage protects files from direct access.

- Configure private path in settings.php.
- Use 'private://' in file fields.
- Secure access via custom hooks if needed.

Advanced Drupal 10 Interview Questions & Answers

50. What are security best practices for writing custom modules?

Security tips for custom modules:

- Sanitize output with `Html::escape()`
- Use Form API for CSRF protection
- Validate all input and use permissions properly