Top 50 Drupal Microservices Interview Questions & Answers

Q1: What are microservices in the context of Drupal?

A: Microservices in Drupal involve breaking down functionalities into independent services that communicate via APIs.

```
$response = \Drupal::httpClient()->get('https://api.example.com/data');
$data = json_decode($response->getBody(), TRUE);
```

Q2: How does a microservices architecture differ from a monolithic Drupal setup?

A: Microservices allow modularization, making scaling easier compared to a monolithic Drupal structure where all functionalities are tightly coupled.

Q3: How does Drupal expose APIs for microservices?

A: Drupal provides REST, JSON:API, and GraphQL modules for exposing APIs.

```
$settings['jsonapi_exposed'] = TRUE;
```

Q4: How can you consume external APIs in Drupal?

A: Using GuzzleHttp for API calls.

```
use GuzzleHttp\Client;
$client = new Client();
$response = $client->request('GET', 'https://api.example.com/data');
$data = json_decode($response->getBody(), TRUE);
```

Q5: What is decoupled (headless) Drupal?

A: Decoupled Drupal separates the backend (Drupal) from the frontend (React, Vue, etc.) via APIs.

Q6: How do you optimize API performance in Drupal?

A: Caching, database indexing, and optimizing API responses.

```
$cache = \Drupal::cache();
$cache->set('custom_cache_key', $data, strtotime('+1 hour'));
```

Q7: How do you secure API endpoints in Drupal?

A: Use OAuth2, API keys, and permissions.

```
if (!\Drupal::currentUser()->hasPermission('access api')) {
  throw new AccessDeniedHttpException();
}
```

Q8: How do you optimize database queries in a microservices setup?

A: Using Entity Query API instead of raw SQL.

```
$query = \Drupal::entityQuery('node')->condition('status', 1)->execute();
```

```
$nodes = \Drupal\node\Entity\Node::loadMultiple($query);
```

Q9: What tools can be used for CI/CD in Drupal?

A: GitHub Actions, Jenkins, CircleCI, and Docker.

```
name: Drupal CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Install dependencies
        run: composer install
```

Q10: How do you implement event subscribers in Drupal?

A: Using EventSubscriberInterface.

```
namespace Drupal\custom_module\EventSubscriber;
use Symfony\Component\EventDispatcher\EventSubscriberInterface;

class CustomEventSubscriber implements EventSubscriberInterface {
  public static function getSubscribedEvents() {
    return ['kernel.request' => 'onRequest'];
  }

  public function onRequest() {
    \Drupal::logger('custom_module')->notice('Request event triggered.');
  }
}
```