A multi-tenant application is designed to serve multiple customers (tenants) from a single instance of the software. Each tenant's data is isolated and typically cannot be accessed by other tenants. This architecture is common in SaaS (Software as a Service) applications.

**Key Concepts**

1. **Tenants**: Different customers or clients using the application.
2. **Data Isolation**: Ensuring each tenant's data is secure and separate from others.
3. **Shared Resources**: The application and infrastructure are shared among tenants to optimize resource usage.

**Scenario**

Imagine a multi-tenant blog platform where multiple companies can manage their blogs independently. Each company (tenant) can have its own set of users, posts, and settings.

**Steps to Create a Multi-Tenant Application**

1. **Database Design**:
   * Use a shared database with tenant-specific tables.
   * Add a TenantId column to each table to segregate data.
2. **Tenant Identification**:
   * Identify the tenant based on the request (e.g., subdomain, URL parameter, or HTTP header).
3. **Middleware**:
   * Create middleware to identify and set the tenant context for each request.
4. **Services and Repositories**:
   * Modify services and repositories to filter data based on the tenant context.