

Singleton Design Pattern

Vad, varför och hur i C#

Vad är Singleton?

- Singleton-mönstret säkerställer att en klass endast har en enda instans.
- Tillhandahåller en global åtkomstpunkt till denna instans.
- Används för att hantera delade resurser, skapa centraliserad kontroll och minska overhead.

Varför använda Singleton?

- Hantera en delad resurs (t.ex. logger, databasanslutningar).
- Centraliserad kontroll för gemensamma logiska funktioner.
- Reducerat minnesutrymme genom att undvika duplicerade objekt.
- Enkel global åtkomst till en gemensam instans.

Fördelar och nackdelar

Fördelar:

- Enkel och global åtkomst.
- Reducerat minnesutrymme.
- Kontrollerad instanshantering.

Nackdelar:

- Bryter Single Responsibility Principle.
- Kan orsaka globalt tillståndsberoende.
- Svår att testa och mocka.

Grundläggande implementation i C#

Trådsäker Singleton i C#

Sammanfattning

- Singleton-mönstret säkerställer en enda instans av en klass.
- Används för delade resurser som logger, konfigurationer och cache.
- Var försiktig med globalt tillstånd och testbarhet.