GABRIELE ROMANATO

Implementazione del Design Pattern Singleton in TypeScript

Il design pattern Singleton garantisce che una classe abbia una sola istanza e fornisce un punto di accesso globale a essa. Questo pattern è particolarmente utile quando si desidera controllare l'accesso a risorse condivise, come connessioni a database o file di configurazione.

Caratteristiche principali del Singleton

- Un'unica istanza condivisa
- Accesso controllato tramite un metodo statico
- Costruttore privato per prevenire l'istanziazione esterna

Implementazione in TypeScript

Di seguito un esempio di implementazione di una classe Singleton in TypeScript:

```
class Singleton {
  private static instance: Singleton;

  // Costruttore privato per evitare la creazione diretta
  private constructor() {
    console.log('Istanza Singleton creata');
  }

  // Metodo statico per ottenere l'istanza
  public static getInstance(): Singleton {
    if (!Singleton.instance) {
      Singleton.instance = new Singleton();
    }
    return Singleton.instance;
  }

  public doSomething(): void {
    console.log('Metodo del Singleton chiamato');
  }
}

// Utilizzo
const singleton1 = Singleton.getInstance();
singleton1.doSomething();

const singleton2 = Singleton.getInstance();
console.log(singleton1 === singleton2); // true
```

Spiegazione del codice

- `private static instance`: contiene l'unica istanza della classe.
- `private constructor`: impedisce la creazione di nuove istanze dall'esterno.
- `getInstance()`: metodo statico che crea o restituisce l'istanza esistente.

Vantaggi del Singleton

- Controllo centralizzato dell'accesso all'istanza

- Risparmio di memoria evitando la creazione di oggetti duplicati
- Utile per la gestione di risorse condivise

Considerazioni

Anche se il Singleton può sembrare conveniente, un uso eccessivo può introdurre dipendenze globali non desiderate, rendendo più difficile testare e mantenere il codice. Usalo con giudizio, soprattutto in applicazioni di grandi dimensioni.

Applicazioni Correlate

- ### JavaScript Password Mask

  Un esempio in JavaScript di mascheramento di una password con l'aggiunta della funzionalità di copia negli appunti.
  DockerDocker ComposeJavaScript

- **Python Placeholder Image**

  Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.
  DockerDocker ComposePythonFlaskJavaScript

-

  Go Placeholder Image

  Applicazione in Go per la creazione di immagini segnaposto.
  DockerDocker ComposeGoJavaScript