

Creazione di un nuovo Controller:

Introduce una nuova classe di Manager, "ManagerData", che sarà responsabile della gestione delle operazioni di comunicazione con il backend e la gestione dei modelli. Questa classe avrà metodi per accedere ai dati dal backend, effettuare operazioni di creazione/modifica/eliminazione e fornire i dati ai Manager.

Ruolo dei Manager:

anziché svolgere direttamente le operazioni di comunicazione con il backend e la gestione dei modelli, richiamino i metodi appropriati nel nuovo ManagerData per eseguire tali operazioni. I Manager si concentreranno principalmente sulla gestione delle interazioni con le View e sulla coordinazione delle operazioni con il ManagerData.

Flusso di dati:

Il flusso dei dati avverrà attraverso il ManagerData. Quando una View richiede dati o richiede operazioni di creazione/modifica/eliminazione, il Manager corrispondente richiamerà i metodi appropriati nel ManagerData. Il ManagerData si occuperà di comunicare con il backend, eseguire le operazioni richieste e fornire i dati ai Manager che richiameranno l'opportuna Funzione di modifica della Schermata definita appunto nella View.

Separazione delle responsabilità:

Questa modifica consentirà una maggiore separazione delle responsabilità tra i Manager e il ManagerData.

I Manager si concentreranno sulla gestione delle interazioni con le View e la logica specifica dell'applicazione, mentre il ManagerData si occuperà della comunicazione con il backend e della gestione dei modelli.

Questa separazione delle responsabilità può rendere il codice più modulare, facile da testare e consentire una maggiore flessibilità nel caso in cui si desideri apportare modifiche o aggiornamenti al sistema.

Il passaggio di dati da Manager a View deve avvenire con Dati Primitivi. Se passassimo un Model alla View, essa diventerebbe dipendente dalla Struttura e dalla logica dietro ai Model. Questo renderebbe ardua la manutenzione. Una modifica al Model comporterebbe una modifica anche alla View.



