Cosa e Saga Pattern

Gestire le transazioni su più microservizi è un problema difficile da risolvere. Il modello Saga è stato introdotto per risolvere questo problema di gestione delle transazioni distribuite. Le transazioni distribuite non sono un problema nuovo. In passato avevamo bisogno di una soluzione per gestire le transazioni distribuite su database, code di messaggi del file system ecc. in un sistema monolitico.

Con la nascita dei microservizi, la necessità di una soluzione di transazione distribuita è ora più evidente che mai. Nel modello Saga, la transazione distribuita è gestita tramite più transazioni locali nei microservizi. Se una transazione all'interno di un microservizio fallisce, in base al modello Saga, il sistema dovrebbe eseguire dei rollback (sotto forma di operazioni di compensazione) per i microservizi che hanno già completato le loro operazioni. Questo assicurerà che tutte le operazioni prima della transazione fallita siano state annullate. Quindi, mantenendo intatta l'integrità del sistema.

3 tipi di transazioni

Nel modello Saga ci sono tre tipi di transazioni. Il primo tipo di transazioni sono transazioni che possono essere potenzialmente invertite con un'operazione. Ad esempio, una voce di database. Il secondo tipo di transazioni sono le transazioni nel modello Saga che decidono se la transazione distribuita deve procedere in avanti o fermarsi. Il terzo tipo di tran

Implementazione Esistono

due tipi di possibili implementazioni del pattern saga. Il primo è chiamato Coreografia. Nell'implementazione della coreografia, ogni microservizio comunica

tra loro tramite eventi. E non necessita di alcun coordinatore centralizzato.

Pro/Contro

Dato che non necessita di alcun coordinatore, diventa molto complesso da gestire in caso di molto semplice da avviare. un gran numero di microservizi. Dato che non c'è un coordinatore, ci sono test di integrazione complessi, poiché tutti nessun singolo punto di errore. i microservizi devono essere attivi e funzionanti.

Orchestrazione Come suggerisce il nome, nell'implementazione dell'orchestrazione c'è un orchestratore centralizzato

L'orchestratore comunica a ogni microservizio quale operazione eseguire. E in caso di problemi, invia anche messaggi a diversi microservizi per il rollback della transazione. Gestisce lo stato di ogni tack tramite una macchina a stati.