

Spring ApplicationEvents

Justin van Leuvenum

Wat zijn Spring Application Events?

Het zijn events die worden getriggerd door het spring framework. Aan deze events zijn Publishers en Listeners verbonden die met elkaar kunnen communiceren.

Hoe toepassen?

Wanneer een service meerdere services aanroept, ontstaat er een strakke koppeling tussen deze componenten (tight coupling). In mijn geval gebeurt dit bij de CreateDepartureService. Deze service roept niet alleen de DepartureService aan, maar ook andere services zoals de WBTService en TerrainService.

Door de CreateDepartureService als publisher te gebruiken, kan deze een event publiceren in plaats van direct andere services aan te roepen. De andere services – die eerder afhankelijk waren van de CreateDepartureService – worden nu listeners voor dit event. Zij reageren op het event door hun specifieke taken uit te voeren die nodig zijn na een truckDeparture. Op deze manier wordt de onderlinge afhankelijkheid tussen de services verminderd en blijft de structuur van de applicatie flexibeler en beter onderhoudbaar.

Dit kan ook toegepast worden op de CreateArrivalService, CreateFIFOArrivalService...

Voordelen

Lage koppeling: Grote services voeren vaak veel taken uit, maar door events te gebruiken, wordt de verantwoordelijkheid van de service verminderd. Hierdoor hebben aanpassingen in de grote service niet noodzakelijk invloed op andere services die ermee verbonden zijn.

Mogelijkheid tot asynchrone verwerking: Events kunnen asynchroon worden gemaakt, waardoor taken op de achtergrond uitgevoerd kunnen worden zonder de hoofdworkflow te vertragen.

Nadelen

Moeilijk te debuggen: Door de asynchrone verwerking kan het lastig zijn om problemen op te sporen en de logica te doorgronden, wat de complexiteit en leesbaarheid van de code kan vergroten.

Onvoorspelbare volgorde: De volgorde waarin listeners reageren is willekeurig, wat kan leiden tot onvoorspelbaar gedrag, vooral wanneer de volgorde belangrijk is.