
Alustan ja beanin hallitsevat transaktiot

J2EE arkkitehtuurissa on mahdollista toteuttaa transaktiot kahdella tavalla: alustan hallitsemina transaktioina (Container-Managed transactions) tai beanin hallitsemina transaktioina (Bean-Managed transactions).

Alustan hallitsevat transaktiot ovat ohjelmoijan kannalta erittäin helppoja: itse ohjelmakoodiin ei tarvitse tehdä mitään muutoksia. Ohjelmoijan täytyy ainoastaan määritellä konfiguraatiodostoihin, mitkä metodit ovat sellaisia, jotka on suoritettava transaktiona. Toisaalta alustan hallitsevat transaktiot ovat jossain määrin rajoittuneita: transaktio koskee aina yhtä metodia, eikä transaktiota ole mahdollista käynnistää tai jättää käynnistämättä tietyistä ohjelmoijan määrittelmistä säännöistä riippuen.

Beanin hallitsevat transaktiot tarjoavat alustan hallitsemia transaktioita joustavammat mahdollisuudet transaktioiden käsittelyyn. Huonona puolena beanin hallitsemisissa transaktioissa on kuitenkin se, että transaktioiden hallitseminen on täysin ohjelmoijan itsensä vastuulla. Transaktioiden aloittaminen ja päättäminen vaatii ylimääräistä ohjelmakoodia, jolloin ohjelmista tulee työläämpiä toteuttaa ja niiden virhealtuus kasvaa.

Alusta hallitsevat transaktiot

Alustaan voi määritellä automaattisen transaktion hallinnan

J2EE arkkitehtuurin mukaiset EJB-komponentit ajetaan sovelluspalvelimella EJB Container –alustalla. EJB Containeriin on mahdollista konfiguroida, että container hoitaa automaattisesti transaktioiden hallinnan. Tällöin ohjelmoijan vastuulla on kertoa containerille ainoastaan se, missä EJB:n metodeissa on otettava huomioon transaktiot.

Esimerkki

Ruokakauppa on EJB, jossa on metodi `poistaVanhentuneetTuotteet()`. Tämä metodi tarkistaa, mitkä tuotteet ovat vanhentuneita varastossa, poistaa nämä tuotteet varastosta ja lisää tuotteet roskakoriin. Haluamme, että tämä metodi on yksi transaktio: jos tuotteita ei voi lisätä roskakoriin, emme myöskään poista niitä varastosta. Toisaalta emme voi lisätä tuotteita roskakoriin, jos emme ole poistaneet niitä varastosta. Molempien operaatioiden on onnistuttava, tai kumpaakaan ei saa suorittaa. Pystymme hoitamaan yksinkertaisesti niin, että määrittelemme `RuokakauppaEJB:n` konfiguraatiodostoon, että `poistaVanhentuneetTuotteet()`-metodin suoritus on aina yksi transaktio. Meidän ei tarvitse kirjoittaa mitään ohjelmakoodia. EJB Container hoitaa puolestamme kaikki tarvittavat operaatiot, kunhan vain määrittelemme tarvittavan konfiguraation.

Erilaiset transaktiomääritykset

Transaktiot ovat suhteellisen helppoja siinä tapauksessa, että yhdessä metodissa tehdään kaikki tarvittavat toimenpiteet. Käytännössä kuitenkin aina erilaiset transaktion hallintaa vaativat toimenpiteet suoritetaan useissa eri metodeissa.

Esimerkki

Oletetaan, että meillä on EJB:t Ruokakauppa, Roskakori ja Kirjanpito. RuokakauppaEJB:ssä on metodi `poistaVanhentuneetTuotteet()`. Tämä metodi katsoo, mitkä tuotteet ovat vanhentuneita ja kutsuu sen jälkeen `RoskakoriEJB:n lisääTuotteetRoskakoriin()`-metodia sekä `KirjanpitoEJB:n kirjaaVanhentuneetTuotteet()`-metodia. Olemme määritelleet, että `poistaVanhentuneetTuotteet()`-metodin suoritus on aina yksi transaktio, mutta kuuluuko tähän transaktioon silloin myös `lisääTuotteetRoskakoriin()`- ja `kirjaaVanhentuneetTuotteet()`-metodien suoritus? Entä jos nämä metodit kutsuvat sisäisessä toteutuksessaan edelleen joitain toisia komponentteja?

Alustan hallitsemille transaktioille (Container-Managed Transactions) on mahdollista määritellä erityyppisiä määreitä joilla kerrotaan, minkä tasoisesta transaktiosta on kyse. Voimme esimerkiksi määritellä, että tietyn metodin on aina kuuluttava johonkin jo olemassa olevaan transaktioon tai että metodin suoritus aloittaa aina uuden transaktion. J2EE tarjoaa seuraavat transaktiomääreet:

Määre	Selitys
Required	Metodi on suoritettava aina osana jotakin transaktiota. Jos metodia kutsuvassa ohjelmassa on jo käynnissä transaktio, tulee metodin toiminnallisuus osaksi tätä transaktiota. Jos transaktiota ei ole vielä käynnissä, aloitetaan kokonaan uusi transaktio. Required on selvästi yleisin transaktion määre, sitä voi käyttää oletusarvoisesti transaktion tyyppinä jos ole erityistä syytä käyttää jotakin toista tyyppiä.
RequiresNew	Metodin suorittaminen käynnistää aina uuden itsenäisen transaktion. Mikäli metodia kutsuvassa sovelluksessa on jo transaktio käynnissä, odottaa tämä transaktio niin kauan, että uusi transaktio saadaan suoritettua loppuun. Kutsuvan sovelluksen transaktio ja uusi aloitettava transaktio eivät ole toisistaan riippuvaisia, vaan niiden commit- ja rollback-operaatiot suoritetaan toisistaan erillisinä.
Mandatory	Metodi vaatii toimiakseen aina jo valmiin, olemassa olevan transaktion. Metodia täytyy kutsua jonkin toisen transaktion sisältä, muussa tapauksessa metodin kutsuminen aiheuttaa <code>TransactionRequiredException</code> -poikkeuksen.