**INT0001.Ladok2.Events**

**Beskrivning**

Denna integrations syfte är att synkronisera händelser från Ladok2 ”uppdok” till flertal system i Uppsala universitets, främst Uppsala universitets ActiveDirectory.

**Flödesbeskrivning**

* Biztalk pollar händelser från SQL via receive porten INT0001.Ladok2.Events.  
  SQL proceduren INT0001.Ladok2.Events.getNext anropas som ser till att händelser som har beroenden hanteras efter att händelserna de är beroende av har hanterats. BizTalk SQL pollning är satt på ganska hög nivå då det kan gå kanske långa perioder då det inte händer något.   
  Endast 100 st. meddelanden hämtas per gång men pollning efter nya meddelanden sker kontinuerligt så länge det finns meddelanden kvar att hämta. Batchning på detta sätt utförs för att inte orsaka minnesproblem
* PromoteAck pipeline promotar AckRequired för att se till att en ACK skapas av BizTalk när meddelandet nått sin mottagare.
* Meddelanden med status KurstillfalleTillStatusEvent som mappas om till meddelandet KurstillfalleTillStatusEnvelope. Via MySQLClientHelper hämtas kursbeskrivningar, semesters samt institut kod och dess beskrivningar.
* Meddelandet KurstillfalleTillStatusEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_ KurstillfalleTillStatusEvent.
* Meddelanden med status KurspaketeringUppdateradEvent som mappas om till meddelandet KurspaketeringUppdateradEnvelope. Via MySQLClientHelper hämtas programbeskrivningar, semesters samt institut kod och dess beskrivningar.
* Meddelandet KurspaketeringUppdateradEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_ KurspaketeringUppdateradEvent.
* Meddelanden med status ForvantatDeltagandeSkapadEvent mappas om till meddelandet ForvantatDeltagandeSkapadEnvelope. . Via MySQLClientHelper hämtas studentens fullständiga namn.
* Meddelandet ForvantatDeltagandeSkapadEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_ForvantatDeltagandeSkapadEvent.
* Meddelanden med status RegistreringEvent mappas om till meddelandet RegistreringEnvelope.
* Meddelandet RegistreringEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_RegistreringEvent.
* Meddelanden med status OmregistreringEvent mappas om till meddelandet OmregistreringEnvelope.
* Meddelandet OmregistreringEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_OmregistreringEvent.
* Meddelanden med status AterbudEvent mappas om till meddelandet AterbudEnvelope.
* Meddelandet AterbudEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_AterbudEvent.
* Meddelanden med status AterkalladOmregistreringEvent mappas om till meddelandet AterkalladOmregistreringEnvelope.
* Meddelandet AterkalladOmregistreringEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_AterkalladOmregistreringEvent.
* Meddelanden med status AterkalladRegistreringEvent mappas om till meddelandet AterkalladRegistreringEnvelope.
* Meddelandet AterkalladRegistreringEnvelope skickas till MSMQ kön SD-DISTRIBUTION via send porten INT0001.SD.Send\_AterkalladRegistreringEvent.
* Alla meddelanden som skickas till MSMQ kön SD-DISTRIBUTION skapar en ACK (se #2). Pipelinen ProcessAck skapar ett ContextCollection meddelande från ACK meddelandets context, detta då ett Ack meddelande inte innehåller en body pilelinen mappar även om det nya ACK meddelandet till ett SQL anrop.
* ContextCollection meddelandet som skapas av pipelinen ProcessAck mappas om till ett SQL anrop av typen INT0001.Ladok2.Events.updateId.
* SQL anropet skickas till SQL som kör proceduren INT0001.Ladok2.Events.updateId via sendporten INT0001.Ladok2.Send\_Update.

**Komponenter**

**Pipeline komponenter**

* BizTalk.PipelineComponents.PromoteTypedContext
* BizTalk.PipelineComponents.ContextToXml
* BizTalkComponents.PipelineComponents.XSLTransform

**Utilities**

* Shared.Utilities.MySQLClientHelper