Project management advanced

[General information 5](#_Toc3993959)

[Flows 6](#_Toc3993960)

[Event Flows 6](#_Toc3993961)

[Case 1: Het aanmaken van een event via frontend 6](#_Toc3993962)

[Case 2: Het aanmaken van een event via CRM 7](#_Toc3993963)

[Case 3: Het bewerken van een event 8](#_Toc3993964)

[Case 4: Het verwijderen van een event via frontend 9](#_Toc3993965)

[Case 5: Het verwijderen van een event via CRM 10](#_Toc3993966)

[Sessie Flows 11](#_Toc3993967)

[Case 6: Het aanmaken van een sessie via frontend 11](#_Toc3993968)

[Case 7: Het aanmaken van een sessie via CRM 12](#_Toc3993969)

[Case 8: Het bewerken van een sessie 13](#_Toc3993970)

[Case 9: Het verwijderen van een sessie via frontend 14](#_Toc3993971)

[Case 10: Het verwijderen van een sessie via CRM 15](#_Toc3993972)

[Bezoeker/werknemer Flows 16](#_Toc3993973)

[Case 11: Reservatie frontend online 16](#_Toc3993974)

[Case 12: Registratie bezoeker op event zelf 17](#_Toc3993975)

[Case 13: Reservatie bezoeker uitschrijven 18](#_Toc3993976)

[Case 14: Het bewerken van een bezoeker/werknemer 19](#_Toc3993977)

[Case 15: Bezoeker/werknemer wordt geblokkeerd (blacklist) 20](#_Toc3993978)

[Case 16: bezoekers/werknemers worden als groep aangemaakt 21](#_Toc3993979)

[Sessie inschrijven Flows 22](#_Toc3993980)

[Case 17: Bezoeker/werknemer schrijft zicht in voor een sessie 22](#_Toc3993981)

[Case 18: Bezoeker/werknemer schrijft zicht uit voor een sessie 23](#_Toc3993982)

[Case 19: Groep schrijft zich in voor sessie 24](#_Toc3993983)

[Case 20: Groep schrijft zich uit voor sessie 25](#_Toc3993984)

[Taken Flows 26](#_Toc3993985)

[Case 21: Taak toevoegen 26](#_Toc3993986)

[Case 22: Taak verwijderen/non-actief 27](#_Toc3993987)

[Case 23: Taak toewijzen aan een werknemer 28](#_Toc3993988)

[Case 24: Taak toewijzen aan een andere werknemer 29](#_Toc3993989)

[Betalings Flows 30](#_Toc3993990)

[Case 25: EHACKB badge opladen 30](#_Toc3993991)

[Case 26: Betalen met EHACKB badge 31](#_Toc3993992)

[Case 27: Bezoeker/werknemer vraagt om factuur 32](#_Toc3993993)

[Case 28: Bezoeker/werknemer vraagt creditnota aan 33](#_Toc3993994)

[Status Flows 34](#_Toc3993995)

[Case 29: Status update van het systeem (werken alle diensten nog naar behoren?) 34](#_Toc3993996)

[Messages 35](#_Toc3993997)

[Inleiding 35](#_Toc3993998)

[Consumerende services 36](#_Toc3993999)

[Consumerende diensten per message 36](#_Toc3994000)

[producerende services 38](#_Toc3994001)

[Producerende diensten per message 38](#_Toc3994002)

[Beschrijving messages 40](#_Toc3994003)

[Variabelen van message headers voor alle messages 40](#_Toc3994004)

[Organisatie message 41](#_Toc3994005)

[Variabelen van message body per message 41](#_Toc3994006)

[Benodigde variabelen per dienst 41](#_Toc3994007)

[Event message 42](#_Toc3994008)

[Variabelen van message body per message 42](#_Toc3994009)

[Benodigde variabelen per dienst 42](#_Toc3994010)

[Error message 43](#_Toc3994011)

[Variabelen van message body per message 43](#_Toc3994012)

[Benodigde variabelen per dienst 43](#_Toc3994013)

[Sessie message 44](#_Toc3994014)

[Variabelen van message body per message 44](#_Toc3994015)

[Benodigde variabelen per dienst 44](#_Toc3994016)

[Werknemer message 46](#_Toc3994017)

[Variabelen van message body per message 46](#_Toc3994018)

[Benodigde variabelen per dienst 46](#_Toc3994019)

[Bezoeker message 47](#_Toc3994020)

[Variabelen van message body per message 47](#_Toc3994021)

[Benodigde variabelen per dienst 47](#_Toc3994022)

[Groep message 48](#_Toc3994023)

[Variabelen van message body per message 48](#_Toc3994024)

[Benodigde variabelen per dienst 48](#_Toc3994025)

[Kalender message 49](#_Toc3994026)

[Variabelen van message body per message 49](#_Toc3994027)

[Benodigde variabelen per dienst 49](#_Toc3994028)

[Benodigdheden message 50](#_Toc3994029)

[Variabelen van message body per message 50](#_Toc3994030)

[Benodigde variabelen per dienst 50](#_Toc3994031)

[Oplaad message 51](#_Toc3994032)

[Variabelen van message body per message 51](#_Toc3994033)

[Benodigde variabelen per dienst 51](#_Toc3994034)

[Aankoop message 52](#_Toc3994035)

[Variabelen van message body per message 52](#_Toc3994036)

[Benodigde variabelen per dienst 52](#_Toc3994037)

[OnvoldoendeVermogen message 53](#_Toc3994038)

[Variabelen van message body per message 53](#_Toc3994039)

[Benodigde variabelen per dienst 53](#_Toc3994040)

[Badge message 54](#_Toc3994041)

[Variabelen van message body per message 54](#_Toc3994042)

[Benodigde variabelen per dienst 54](#_Toc3994043)

[Factuur message 55](#_Toc3994044)

[Variabelen van message body per message 55](#_Toc3994045)

[Benodigde variabelen per dienst 55](#_Toc3994046)

[Creditnota message 56](#_Toc3994047)

[Variabelen van message body per message 56](#_Toc3994048)

[Benodigde variabelen per dienst 56](#_Toc3994049)

[Ping message 57](#_Toc3994050)

[Variabelen van message body per message 57](#_Toc3994051)

[Benodigde variabelen per dienst 57](#_Toc3994052)

[Ok message 58](#_Toc3994053)

[Variabelen van message body per message 58](#_Toc3994054)

[Benodigde variabelen per dienst 58](#_Toc3994055)

[Taak message 59](#_Toc3994056)

[Variabelen van message body per message 59](#_Toc3994057)

[Benodigde variabelen per dienst 59](#_Toc3994058)

[Werk message 60](#_Toc3994059)

[Variabelen van message body per message 60](#_Toc3994060)

[Benodigde variabelen per dienst 60](#_Toc3994061)

[Reservatie message 61](#_Toc3994062)

[Variabelen van message body per message 61](#_Toc3994063)

[Benodigde variabelen per dienst 61](#_Toc3994064)

[GebruiksItem message 62](#_Toc3994065)

[Variabelen van message body per message 62](#_Toc3994066)

[Benodigde variabelen per dienst 62](#_Toc3994067)

[VerkoopsItem message 63](#_Toc3994068)

[Variabelen van message body per message 63](#_Toc3994069)

[Benodigde variabelen per dienst 63](#_Toc3994070)

[Registratie message 64](#_Toc3994071)

[Variabelen van message body per message 64](#_Toc3994072)

[Benodigde variabelen per dienst 64](#_Toc3994073)

[Locatie message 65](#_Toc3994074)

[Variabelen van message body per message 65](#_Toc3994075)

[Benodigde variabelen per dienst 65](#_Toc3994076)

Algemene informatie

* *Alle Messages Bevatten: UUID, Timestamp, VersieNummer, MessageType, GEGEVENS.*
* *Onder event verstaan we Ehackb.*
* *Onder dienst verstaan we facturatie, crm, frontend, planning, monitor, kassa*
* *Onder sessies verstaan we bv een lezing, workshop, gaming, …*
* *uitzonderingen*

Flows

# Event Flows

## Case 1: Het aanmaken van een event via frontend

**Uitvoerende dienst:** Frontend

1. Admin gaat naar dienst.
2. Admin geeft zijn gebruikergegevens op (email + wachtwoord).

Admin is zijn gebruikergegevens vergeten.

1. Admin maakt een event aan.
   1. Tijden zijn niet beschikbaar
2. Frontend genereert een UUID.
3. Frontend stuurt [eventMessage](#_Event_message) naar de queue.

**Rabbitmq - queue**

Planning haalt [eventMessage](#_Event_message) van de queue.

1. Planning kijkt of aangegeven tijden en locatie beschikbaar zijn.
   * Indien tijd en/of locatie niet beschikbaar is: [errorMessage](#_Error_message)
2. Voegt event toe op eventkalender.
3. Slaagt gegevens van event op.

Monitor haalt [eventMessage](#_Event_message)/[errormessage](#_Error_message) van de queue.

1. Monitor logt de creatie van het event of logt de errorMessage.

CRM haalt [eventMessage](#_Event_message) van de queue.

1. De dienst maakt een nieuw EVENT aan met gegevens *naam\* / locatie / start datum\* / eind datum\* / duur (Auto Filled) / budget / omschrijving / toegewezen aan / DELEGATES?*

Kassa haalt [eventMessage](#_Event_message) van de queue.

1. Kassa maakt een “kassa” aan met de naam van het event.

Facturatie haalt [eventMessage](#_Event_message) van de queue.

1. Facturatie maakt een product aan met de prijs voor het event.

## Case 2: Het aanmaken van een event via CRM

**Uitvoerende dienst:** CRM

1. Admin gaat naar dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin maakt een event aan.

Tijden zijn niet beschikbaar

1. Frontend genereert een UUID.
2. CRM stuurt [eventMessage](#_Event_message) naar de queue.

**Rabbitmq - queue**

Planning haalt [eventMessage](#_Event_message) van de queue.

1. Dienst kijkt of aangegeven tijden en locatie beschikbaar zijn.
   * Indien tijd en/of locatie niet beschikbaar is: [errorMessage](#_Error_message)
2. Dienst voegt event toe op eventkalender.
3. Dienst slaagt gegevens van event op.

Monitor haalt [eventMessage](#_Event_message)/ [errorMessage](#_Error_message) van de queue.

1. Monitor logt de creatie van het event of logt de errorMessage

Kassa haalt [eventMessage](#_Event_message) van de queue.

1. Kassa maakt een “kassa” aan met de naam van het event.

Facturatie haalt [eventMessage](#_Event_message) van de queue.

1. Facturatie maakt een product aan met de prijs voor het event.

## Case 3: Het bewerken van een event

*! Is hetzelfde als het aanmaken maar met hogere versieNummer !*

**Uitvoerende dienst:** CRM / Frontend

1. Admin gaat naar dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin past event aan, dat bestaat in het systeem (CRM/frontend).

Tijden zijn niet beschikbaar en/of locatie is niet beschikbaar

1. Dienst genereert een UUID Met versieNummer++.
2. Diest stuurt [eventMessage](#_Event_message) naar de queue.

**Rabbitmq - queue**

Planning haalt [eventMessage](#_Event_message) van de queue.

1. Planning ontvangt gegevens die moeten veranderen.
2. Planning controleert of de gewenste tijd en locatie beschikbaar zijn.
   * + Indien de tijd en locatie beschikbaar zijn: event met aangepaste gegevens wordt opgeslagen
     + Indien het event niet kan worden ingepland door overlap: [errorMessage](#_Error_message) naar de queue

Monitor haalt [eventMessage](#_Event_message)/[errorMessage](#_Error_message) van de queue.

1. Monitor logt het bewerken van het event of logt de errorMessage.

Kassa haalt [eventMessage](#_Event_message) van de queue.

1. Kassa verandert de gegevens van de sessie (prijs, plaats, ...)

Facturatie haalt [eventMessage](#_Event_message) van de queue

1. Past de nodige zaken voor het event aan (naam, prijs, …).

CRM ontvangt [eventMessage](#_Event_message)

1. In het CRM systeem wordt het ‘event’ gewijzigd, de veranderde gegevens zijn afhankelijk van de message: naam\* / locatie / start datum\* / eind datum\* / duur (Auto Filled) / budget / omschrijving / toegewezen aan / DELEGATES?

Frontend ontvangt [eventMessage](#_Event_message)

1. Frontend past de gegevens aan en slaagt ze op.

## Case 4: Het verwijderen van een event via frontend

*! Is hetzelfde als bewerken maar met hogere VersieNummer & actiefFlag op 0 !*

**Uitvoerende dienst:** Frontend

1. Admin gaat naar dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin verwijdert event.

Tijden zijn niet beschikbaar

1. Frontend genereert een UUID Met versieNummer++ & actiefFlag op 0.
2. Frontend stuurt [eventMessage](#_Event_message) naar de queue.

**Rabbitmq - queue**

Planning haalt [eventMessage](#_Event_message) van de queue.

1. Planning ontvangt gegevens die moeten veranderen en archiveert het event

Monitor haalt [eventMessage](#_Event_message) van de queue.

1. Monitor logt het verwijderen van het event, de message dus.

Kassa haalt [eventMessage](#_Event_message) van de queue.

1. Kassa verwijdert de kassa van dit event

Facturatie haalt [eventMessage](#_Event_message) van de queue

1. Verwijdert het product van dit event

CRM ontvangt [eventMessage](#_Event_message)

1. In het CRM systeem wordt het event op non-actief gezet.

## Case 5: Het verwijderen van een event via CRM

*! Is hetzelfde als bewerken maar met hogere versieNummer & actiefFlag op 0 !*

**Uitvoerende dienst:** CRM

1. Admin gaat naar de dienst (CRM).
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin verwijdert event.

Tijden zijn niet beschikbaar

1. CRM genereert een UUID Met versieNummer++ & actiefFlag op 0.
2. CRM stuurt [eventMessage](#_Event_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [eventMessage](#_Event_message) van de queue.

1. Planning ontvangt gegevens die moeten veranderen en archiveert het event

Monitor haalt [eventMessage](#_Event_message) van de queue.

1. Monitor logt het verwijderen van het event, de message dus.

Kassa haalt [eventMessage](#_Event_message) van de queue.

1. Kassa verwijdert de kassa van dit event

Facturatie haalt [eventMessage](#_Event_message) van de queue

1. Verwijdert het product van dit event

Frontend haalt [eventMessage](#_Event_message) van de queue.

1. Frontend zet het event op ‘non-actief’ en slaagt dit op.

# Sessie Flows

## Case 6: Het aanmaken van een sessie via frontend

**Uitvoerende dienst:** Frontend

1. Admin gaat naar de dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin maakt een sessie aan.

Tijden zijn niet beschikbaar

1. Frontend genereert een UUID.
2. Frontend stuurt [sessieMessage](#_Session_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [sessieMessage](#_Session_message)van de queue.

1. Service kijkt of aangegeven tijden/locatie beschikbaar zijn.
   * Beschikbaar:
     + De dienst voegt sessie toe op de sessiekalender en slaagt deze op.
   * Er is overlapping :
     + Er wordt een [errorMessage](#_Error_message) verstuurd naar de queue

Monitor haalt [sessieMessage](#_Session_message)/[errorMessage](#_Error_message) van de queue.

1. Monitor logt de creatie van de Session of de errorMessage.

CRM haalt [sessieMessage](#_Session_message) van de queue.

1. De dienst logt de creatie van de sessie.
2. De dienst maakt een nieuwe sessie aan met gegevens naam\* / locatie / start datum\* / eind datum\* / duur (Auto Filled) / budget / omschrijving / toegewezen aan / *DELEGATES?*

Kassa haalt [sessieMessage](#_Session_message) van de queue.

1. Kassa maakt een product aan met de naam van de sessie.

Facturatie haalt [sessieMessage](#_Session_message) van de queue

1. Maakt een product aan met een prijs voor de sessie.

## Case 7: Het aanmaken van een sessie via CRM

**Uitvoerende dienst:** CRM

1. Admin gaat naar de dienst
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin maakt een sessie aan.

Tijden zijn niet beschikbaar

1. CRM genereert een UUID.
2. CRM stuurt [sessieMessage](#_Session_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [sessieMessage](#_Session_message)van de queue.

1. Service kijkt of aangegeven tijden/locatie beschikbaar zijn.
   * Beschikbaar:
     + De dienst voegt sessie toe op sessiekalender en slaagt deze op.
   * Er is overlapping :
     + Er wordt een [errorMessage](#_Error_message) verstuurd naar de queue

Monitor haalt [sessieMessage](#_Session_message)/[errorMessage](#_Error_message) van de queue.

1. Monitor logt de creatie van de sessie of de errorMessage.

Kassa haalt [sessieMessage](#_Session_message) van de queue.

1. Kassa maakt een product aan met de naam van de sessie.

Facturatie haalt [sessieMessage](#_Session_message) van de queue

1. Maakt een product aan met een prijs voor de sessie.

Frontend haalt [sessieMessage](#_Session_message) van de queue.

1. Frontend slaagt informatie van de sessie op

## Case 8: Het bewerken van een sessie

*! Is hetzelfde als het aanmaken maar met hogere versieNummer !*

**Uitvoerende dienst:** CRM / Frontend

1. Admin gaat naar de dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin past event aan.

Tijden zijn niet beschikbaar

1. De dienst genereert een UUID Met versieNummer++.
2. De dienst stuurt [sessieMessage](#_Session_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [sessieMessage](#_Session_message) van de queue.

1. Planning kijkt of aangegeven tijden/locatie beschikbaar zijn.
   * Beschikbaar:
     + Planning voegt wijzigingen van session toe op de sessiekalender en slaagt deze op.
   * Er is overlapping :
     + Er wordt een [errorMessage](#_Error_message) verstuurd naar de queue

Monitor haalt [sessieMessage](#_Session_message)/[errorMessage](#_Error_message) van de queue.

1. Monitor logt het bewerken van de sessie of logt de errorMessage.

Kassa haalt [sessieMessage](#_Session_message) van de queue.

1. Kassa verandert de gegevens van de sessie (prijs, plaats, …)

Facturatie haalt [sessieMessage](#_Session_message) van de queue

1. Verandert een product (prijs,naam,…) voor de sessie.

CRM haalt [sessieMessage](#_Session_message) van de queue

1. In het CRM Systeem wordt de ‘sessie’ gewijzigd, de veranderde gegevens zijn afhankelijk van de Message: naam\* / locatie / start datum\* / eind datum\* / duur (Auto Filled) / budget / omschrijving / toegewezen aan / *DELEGATES?*

Frontend haalt [sessieMessage](#_Session_message) van de queue

1. Frontend past de gegevens aan en slaagt ze op.

## Case 9: Het verwijderen van een sessie via frontend

*! Is hetzelfde als bewerken maar met hogere versieNummer & actiefFlag op 0 !*

**Uitvoerende dienst:** Frontend

1. Admin gaat naar de dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin verwijdert event.

Tijden zijn niet beschikbaar

1. Frontend genereert een UUID Met versieNummer++ & actiefFlag op 0.
2. Frontend stuurt [sessieMessage](#_Session_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [sessieMessage](#_Session_message) van de queue.

1. Planning verwijdert de sessie uit de planning.

Monitor haalt [sessieMessage](#_Session_message) van de queue.

1. Monitor logt het verwijderen van de sessie.

Kassa haalt [sessieMessage](#_Session_message) van de queue.

1. Kassa verandert het product van de sessie (prijs, plaats, ...)

Facturatie haalt [sessieMessage](#_Session_message) van de queue

1. Facturatie verwijdert het product (prijs,naam, …) van de sessie.

CRM haalt [sessieMessage](#_Session_message) van de queue

1. In het CRM Systeem wordt de sessie gearchiveerd, de veranderde gegevens zijn afhankelijk van de Message: naam\* / locatie / start datum\* / eind datum\* / duur (Auto Filled) / budget / omschrijving / toegewezen aan / *DELEGATES?*

## Case 10: Het verwijderen van een sessie via CRM

*! Is hetzelfde als bewerken maar met hogere versieNummer & actiefFlag op 0 !*

**Uitvoerende dienst:** CRM

1. Admin gaat naar de dienst.
2. Admin geeft zijn gebruikergegevens op.

Admin is zijn gebruikergegevens vergeten.

1. Admin verwijdert event.

Tijden zijn niet beschikbaar

1. CRM genereert een UUID Met versieNummer++ & actiefFlag op 0.
2. CRM stuurt [sessieMessage](#_Session_message) naar de queue.

**Rabbitmq – queue**

Planning haalt [sessieMessage](#_Session_message) van de queue.

1. Planning verwijdert de sessie uit de planning.

Monitor haalt [sessieMessage](#_Session_message) van de queue.

1. Monitor logt het verwijderen van de sessie.

Kassa haalt [sessieMessage](#_Session_message) van de queue.

1. Kassa verandert het product van de sessie (prijs, plaats, ...)

Facturatie haalt [sessieMessage](#_Session_message) van de queue

1. Facturatie verwijdert het product (prijs,naam, …) van de sessie.

Frontend haalt [sessieMessage](#_Session_message) van de queue.

1. Frontend zet het event op ‘non-actief en slaagt dit op.

# Bezoeker/werknemer Flows

## Case 11: Reservatie frontend online

**Uitvoerende dienst:** Frontend

1. Bezoeker/werknemer geeft zijn gegevens op, verplicht naam en email

Bezoeker/werknemer is al geregistreerd? UPDATE MET VERSIENUMMER +1

1. Frontend genereert een UUID Met versieNummer++ & ActiefFlag op 1.
2. Frontend stuurt bezoekerMessage/werknemerMessage naar de queue.

**Rabbitmq – queue**

Kassa haalt bezoekerMessage/werknemerMessage van de queue.

1. Kassa slaat de bezoeker/werknemer op indien deze nog niet bestond

Monitor haalt bezoekerMessage/werknemerMessage van de queue.

1. Haalt de message van de queue en logt deze.

Facturatie haalt bezoekerMessage/werknemerMessage van de queue

1. De bezoeker/werknemer wordt binnen invoiceninja opgeslagen. Als de bezoeker/werknemer een spreker of sponsor is wordt hij opgeslagen als vendor ipv client.

CRM haalt bezoekerMessage/werknemerMessage van de queue.

1. Het CRM systeem ontvangt de message en de bezoeker/werknemer wordt als een ‘CONTACT’ EN ‘LEAD’ opgeslagen in het systeem.

*Indien de gebruiker teamleden opgeeft worden deze eveneens als ‘CONTACT’ EN ‘LEAD’ opgeslagen (Iemand die de Facebook Pagina Liked wordt een LEAD)*

1. Gegevens voor CONTACT (voornaam / achternaam\* / adres / Email / Lead Source)
2. Gegevens Voor LEAD (voornaam / achternaam\* / adres / email / Lead Source / status / Lead Source )
3. Bezoeker/werknemer wordt ingegeven als ‘Vooringeschreven’ (STATISTICS)

Planning haalt de bezoekerMessage/werknemerMessage van de queue.

1. Planning slaagt de message op om een persoonlijke planning per persoon te kunnen maken

## Case 12: Registratie bezoeker op event zelf

**Uitvoerende dienst:** Kassa

1. Bezoeker geeft zijn gegevens op.
2. Kassa genereert een UUID Met versieNummer++ & ActiefFlag op 1.
3. Kassa stuurt [bezoekerMessage](#_User_message) naar de queue.

**Rabbitmq – queue**

Monitor haalt [bezoekerMessage](#_User_message)van de queue.

1. Haalt [message](#_User_message) van de queue en logt deze.

Facturatie haalt [bezoekerMessage](#_User_message)van de queue

1. De bezoeker wordt binnen invoiceninja opgeslagen.

Frontend haalt [bezoekerMessage](#_User_message)van de queue.

1. Frontend slaagt de bezoeker op indien die nog niet in het systeem zit.

CRM haalt [bezoekerMessage](#_User_message) van de queue.

1. Het CRM systeem ontvangt de message en de gebruiker wordt als een ‘CONTACT’ EN ‘LEAD’ opgeslagen in het systeem.

*Indien de gebruiker teamleden opgeeft worden deze eveneens als ‘CONTACT’ EN ‘LEAD’ opgeslagen (Iemand die de Facebook Pagina Liked wordt een LEAD)*

1. Gegevens voor CONTACT (voornaam / achternaam\* / adres / Email / Lead Source)
2. Gegevens Voor LEAD (voornaam / achternaam\* / adres / email / Lead Source / status / Lead Source )
3. Bezoeker wordt ingegeven als ‘Vooringeschreven’ (STATISTICS)

Planning haalt de [bezoekerMessage](#_User_message) van de queue.

1. Planning slaagt de message op om een persoonlijke planning per persoon te kunnen maken

## Case 13: Reservatie bezoeker uitschrijven

**Uitvoerende dienst:** Frontend

1. Bezoeker schrijft zich uit voor EHACKB
2. Klikt de optie aan om *wel / niet* op de hoogte te blijven van volgende Events
3. Frontend stuurt een bezoekerMessage met verhoogd versionnummer naar queue

**Rabbitmq – queue**

*WEL OP DE HOOGTE:*

Kassa haalt bezoekerMessage van de queue.

1. Kassa neemt message van op.
2. Kassa archiveert de bezoeker van alle sessies.
3. Kassa zet de bezoeker op non-actief.

Planning haalt bezoekerMessage van de queue.

* 1. Planning stuurt geen notificaties naar bezoeker.

Monitor haalt bezoekerMessage van de queue.

1. Haalt message van de queue en logt deze.

Facturatie haalt bezoekerMessage van de queue

1. De bezoeker wordt binnen invoiceninja gearchiveerd.

Frontend haalt bezoekerMessage van de queue

1. FrontEnd ‘de-activates’ gebruiker.

CRM haalt bezoekerMessage van de queue.

1. Gebruiker blijft in het CRM systeem staan als (CONTACT &) LEAD

*NIET OP DE HOOGTE:*

1. Alle diensten zijn verplicht om deze bezoeker te verwijderen uit onze database.

## Case 14: Het bewerken van een bezoeker/werknemer

*! Is hetzelfde als aanmaken maar met hogere versieNummer !*

**Uitvoerende dienst:** Frontend/kassa

1. Bezoeker/werknemer gaat naar de dienst.
2. Bezoeker/werknemer geeft zijn gebruikergegevens op.
3. Bezoeker/werknemer past de nodige gegevens aan.
4. De dienst genereert een UUID Met VersieNummer++.
5. De dienst stuurt bezoekerMessage/werknemerMessage naar de queue.

**Rabbitmq – queue**

Planning haalt bezoekerMessage/werknemerMessage van de queue.

1. Planning vergelijkt het versienummer met het vorige om te zien of het hoger is, indien ja pas de veranderingen toe.

Monitor haalt bezoekerMessage/werknemerMessage van de queue.

1. Monitor logt het bewerken van de bezoeker/werknemer.

Kassa haalt bezoekerMessage/werknemerMessage van de queue.

1. Kassa verandert de gegevens van de bezoeker/werknemer.

Facturatie haalt bezoekerMessage/werknemerMessage van de queue

1. Na het controleren van het versienummer worden de veranderingen van de bezoeker/werknemer binnen het facturatiesysteem opgeslagen.

CRM haalt bezoekerMessage/werknemerMessage van de queue

1. CRM vergelijkt het versienummer met het vorige om te zien of het hoger is, indien ja pas de veranderingen toe.

Frontend haalt bezoekerMessage/werknemerMessage van de queue

1. Frontend vergelijkt het versienummer met het vorige om te zien of het hoger is, indien ja pas de veranderingen toe.

## Case 15: Bezoeker/werknemer wordt geblokkeerd (blacklist)

**Uitvoerende dienst:** Frontend

1. De admin meldt zich aan op de frontend
2. De admin zet de geblokkeerdFlag op 1 en verstuurd een bezoekerMessage/werknemerMessage met versieNummer++

**Rabbitmq – queue**

Planning haalt bezoekerMessage/werknemerMessage van de queue.

1. Planning plaatst de bezoeker/werknemer op non-actief.

Monitor haalt bezoekerMessage/werknemerMessage van de queue.

1. Monitor logt deze message.

CRM haalt bezoekerMessage/werknemerMessage van de queue.

1. CRM plaatst de bezoeker/werknemer op non-actief.

Kassa haalt bezoekerMessage/werknemerMessage van de queue.

1. Kassa plaatst de bezoeker/werknemer op non-actief.

Facturatie haalt bezoekerMessage/werknemerMessage van de queue.

1. Facturatie plaatst de bezoeker/werknemer op non-actief.

## Case 16: bezoekers/werknemers worden als groep aangemaakt

**Uitvoerende dienst:** Frontend

1. De bezoeker/werknemer meldt zich aan op de frontend
2. De werknemer/bezoeker die de groep wilt aanmaken heeft een ehb email-adres.
3. De bezoeker/werknemer vult in hoeveel leden er in de groep zitten
4. De bezoeker/werknemer geeft de voornaam, achternaam en email op van de andere leden
5. Indien de andere leden nog niet in een andere groep zijn ingeschreven wordt de groep aangemaakt.
6. Indien de leden nog niet in geregistreerd waren wordt er voor elk lid van de groep een bezoeker message verstuurd.
7. De frontend stuurt een [groepMessage](#_User_message) naar de queue

**Rabbitmq – queue**

Planning haalt [groepMessage](#_User_message) van de queue.

1. Planning gebruikt de message om de planning van de persoon te finetunen binnen de groep.

Monitor haalt [groepMessage](#_User_message) van de queue.

1. Monitor logt deze message.

CRM haalt de [groepMessage](#_User_message) van de queue.

1. CRM gebruikt de message om de groep op de hoogte te kunnen houden.

Kassa haalt de [groepMessage](#_User_message) van de queue.

1. Kassa gebruikt de om eventueel aan de kassa te weten wie er nog niet betaalt heeft voor de gaming sessie.

Facturatie haalt [groepMessage](#_User_message) van de queue.

1. Facturatie gebruikt de message om een correcte factuur aan te maken per individu.

# Sessie inschrijven Flows

## Case 17: Bezoeker/werknemer schrijft zicht in voor een sessie

**Uitvoerende dienst:** Frontend/kassa

1. De bezoeker/werknemer meldt zich aan op de dienst
2. De bezoeker/werknemer schrijft zich in voor één of meerdere sessies
3. De service valideert welke sessies nog beschikbaar zijn
4. Indien er een betaling gewenst ist kan de bezoeker/werknemer kiezen om die onmiddellijk te betalen (bij reservatie) of aan de kassa (bij registratie)
5. De dienst stuurt een [reservatieMessage](#_Reservation_message) naar de queue

**Rabbitmq – queue**

Planning haalt [reservatieMessage](#_Reservation_message)van de queue.

1. Planning valideert dat er nog plaatsen zijn, anders stuurt die een [errorMessage](#_Error_message).
2. Planning verlaagt het aantal plaatsen
3. Planning voegt de bezoeker/werknemer toe aan de sessie

Monitor haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Monitor logt deze message.

CRM haalt [reservatieMessage](#_Reservation_message)van de queue.

1. CRM plaatst de bezoeker/werknemer bij desbetreffende sessie.

Kassa haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Kassa koppelt de reservatie of registratie met de bezoeker/werknemer zodat ze kunnen zien bij registreren of de bezoeker/werknemer nog moet betalen, ja dan nee.

Facturatie haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Facturatie plaatst de reservatiekosten van de bezoeker/werknemer in de database.

## Case 18: Bezoeker/werknemer schrijft zicht uit voor een sessie

**Uitvoerende dienst:** Frontend/kassa

1. De bezoeker/werknemer meldt zich aan op de dienst
2. De bezoeker/werknemer schrijft zich uit voor één of meerdere sessies waarvoor hij ingeschreven is.
3. De dienst valideert welke sessies nog beschikbaar zijn
4. De dienst stuurt een [reservatieMessage](#_Reservation_message) naar de queue met de activeFlag 0

**Rabbitmq – queue**

Planning haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Planning verhoogt het aantal plaatsen
2. Planning verwijdert de bezoeker/werknemer van de sessie

Monitor haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Monitor logt deze message.

CRM haalt [reservatieMessage](#_Reservation_message) van de queue.

1. CRM verwijdert de bezoeker/werknemer van de desbetreffende sessie.

Kassa haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Kassa ontkoppelt de reservatie of registratie met de bezoeker/werknemer zodat ze kunnen zien bij registreren dat de bezoeker/werknemer niet meer moet betalen.

Facturatie haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Facturatie de reservatiekosten worden uit de database gehaald.
2. Indien al betaald krijgt de bezoeker/werknemer bij registratie zijn kosten van de sessie terug.

## Case 19: Groep schrijft zich in voor sessie

**Uitvoerende dienst:** Frontend/kassa

1. De admin van de groep meldt zich aan op de dienst
2. De admin van de groep schrijft zich in voor één of meerdere sessies
3. De dienst valideert welke sessies nog beschikbaar zijn
4. Indien er een betaling gewenst ist kan de bezoeker/werknemer kiezen om die onmiddellijk te betalen (bij reservatie) of aan de kassa (bij registratie)
5. De dienst stuurt een [reservatieMessage](#_Reservation_message) naar de queue

**Rabbitmq – queue**

Planning haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Planning valideert dat er nog plaatsen zijn, anders stuurt die een [errorMessage](#_Error_message).
2. Planning verlaagt het aantal plaatsen
3. Planning voegt de bezoekers/werknemers toe aan de sessie

Monitor haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Monitor logt deze message.

CRM haalt [r[reservatieMessage](#_Reservation_message)](#_Reservation_message)  van de queue.

1. CRM plaatst de bezoekers/werknemers bij desbetreffende sessie.

Kassa haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Kassa koppelt de reservatie of registratie met de bezoekers/werknemers zodat ze kunnen zien bij registreren of de bezoeker/werknemer nog moet betalen, ja dan nee.

Facturatie haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Facturatie plaatst de reservatiekosten van de bezoeker/werknemer in de database.

## Case 20: Groep schrijft zich uit voor sessie

**Uitvoerende dienst:** Frontend/kassa

1. De admin van de groep meldt zich aan op de dienst
2. De admin van de groep schrijft zich uit voor één of meerdere sessies waarvoor hij ingeschreven is.
3. De dienst valideert welke sessies nog beschikbaar zijn
4. De dienst stuurt een [reservatieMessage](#_Reservation_message) naar de queue met de actiefFlag 0

**Rabbitmq – queue**

Planning haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Planning verhoogt het aantal plaatsen
2. Planning verwijdert de groep van de sessie

Monitor haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Monitor logt deze message.

CRM haalt [reservatieMessage](#_Reservation_message) van de queue.

1. CRM verwijdert de groep van de desbetreffende sessie.

Kassa haalt [reservatieMessage](#_Reservation_message) an de queue.

1. Kassa ontkoppelt de reservatie of registratie met de bezoekers/werknemers zodat ze kunnen zien bij registreren dat de bezoekers/werknemers niet meer moet betalen.

Facturatie haalt [reservatieMessage](#_Reservation_message) van de queue.

1. Facturatie de reservatiekosten worden uit de database gehaald.
2. Indien al betaald krijgen de bezoekers/werknemers bij registratie de kosten van de sessie terug.

# Taken Flows

## Case 21: Taak toevoegen

**Uitvoerende diensr:** Frontend

1. Admin gaat naar de dienst
2. Admin geeft zijn gebruikersgegevens op
3. Admin geeft de gegevens van de taak op
4. De gegevens van de taak worden via een [taakMessage](#_Task_message) verstuurd naar de queue

**Rabbitmq – queue**

Planning haalt [taakMessage](#_Task_message) van de queue.

1. Planning zet de data van de taak binnen een event/sessie.

Monitor haalt [taakMessage](#_Task_message) van de queue.

1. haalt [taakMessage](#_Task_message) van de queue en logt deze.

CRM //.

Kassa //.

Facturatie //.

## Case 22: Taak verwijderen/non-actief

**Uitvoerende dienst:** Frontend

1. Admin gaat naar de dienst
2. Admin geeft zijn gebruikersgegevens op
3. Admin verwijdert de taak
4. De taak wordt op non-actief gezet
5. Frontend stuurt [taakMessage](#_Task_message) naar de queue

**Rabbitmq – queue**

Planning haalt [taakMessage](#_Task_message) van de queue.

1. Planning zet de data van de taak binnen een event/sessie op non-actief.

Monitor haalt [taakMessage](#_Task_message) van de queue.

1. Haalt [taakMessage](#_Task_message) van de queue en logt deze.

CRM //.

Kassa //.

Facturatie //.

## Case 23: Taak toewijzen aan een werknemer

**Uitvoerende dienst**: Frontend

1. Admin gaat naar de dienst
2. Admin geeft zijn gebruikersgegevens op
3. Admin koppelt een taak aan een werknemer
4. De gegevens worden via een [werkMessage](#_Work_message) verstuurd

**Rabbitmq – queue**

Planning haalt [werkMessage](#_Work_message) van de queue.

1. Planning voegt werknemer toe aan taak binnen een event/sessie.

Monitor haalt [werkMessage](#_Work_message) van de queue.

1. Haalt message van de queue en logt deze.

Frontend haalt [werkMessage](#_Work_message) van de queue.

1. Frontend ‘update’ werknemer en zijn taken. (non-actief)

CRM //.

Kassa //.

Facturatie //.

## Case 24: Taak toewijzen aan een andere werknemer

**Uitvoerende dienst:** Frontend

1. Admin gaat naar de dienst
2. Ontkoppelt de taken van de werknemer
3. Taken toewijzen aan andere werknemer(s)
4. Er wordt een message verstuurd van de taken die aan een bepaalde werknemer zijn toegevoegd
5. De gegevens worden via een [werkMessage](#_Work_message) verstuurd naar de queue

**Rabbitmq – queue**

Planning haalt [werkMessage](#_Work_message) van de queue.

1. Planning voegt werknemer toe aan taak binnen een event/sessie.

Monitor haalt [werkMessage](#_Work_message) van de queue.

1. Haalt [message](#_Work_message) van de queue en logt deze.

Frontend haalt [werkMessage](#_Work_message) van de queue.

1. FrontEnd ‘update’ werknemer en zijn taken. (non-Actief)

CRM //.

Kassa //.

Facturatie //.

# Betalings Flows

## Case 25: EHACKB badge opladen

**Uitvoerende dienst:** Kassa

1. Bezoeker meldt zich bij de Kassa

Geen badge = Nieuwe badge aanmaken

1. Bezoeker beslist om een bepaald bedrag op zijn badge te storten

Niet Betalen: Geen message sturen

1. Kassa genereert oplaadMessage en stuurt die naar de queque.

**Rabbitmq – queue**

Monitor haalt oplaadMessage van de queue.

1. Monitor haalt message van de queue en logt deze
2. Monitor slaat alarm indien er verdachte gebeurtenissen plaatsvinden. -> errorMessage

Facturatie haalt oplaadMessage van de queue

1. Facturatie voegt deze toe aan de databank, om zo de factuur naderhand te kunnen opstellen.

Frontend haalt oplaadMessage van de queue.

1. Frontend haalt message op om zo te kunnen tonen hoeveel de bezoeker heeft uitgegeven (Dit kan enkel getoond worden wanneer de bezoeker zich aanmeldt op de site)

CRM //.

Planning //.

Moet werknemer ook een zijn badge kunnen opladen?

## Case 26: Betalen met EHACKB badge

**Uitvoerende dienst:** Kassa

1. Bezoeker/werknemer betaalt met zijn badge aan de kassa
2. Per product wordt een [aankoopMessage](#_Payment_message) gemaakt
3. Kassa genereert [aankoopMessage](#_Payment_message) en stuurt die naar de queque.

**Rabbitmq – queue**

Monitor haalt [aankoopMessage](#_Payment_message) van de queue.

1. haalt [aankoopMessage](#_Payment_message) /[onvoldoendeVermogenMessage](#_InsufficientFunds_message) van de queue en logt deze
2. slaat alarm indien er verdachte gebeurtenissen plaatsvinden

Facturatie haalt [aankoopMessage](#_Payment_message) van de queue

1. Voegt deze toe aan de databank.
2. Voegt per aankoop een record toe aan de databank van de bezoeker/werknemer.

Frontend haalt [aankoopMessage](#_Payment_message) van de queue.

1. Haalt message op om zo te kunnen tonen hoeveel de bezoeker/werknemer heeft uitgegeven aan welk item (dit kan enkel getoond worden wanneer de bezoeker/werknemer zich aanmeldt op de site)

Kassa haalt [onvoldoendeVermogenMessage](#_InsufficientFunds_message) van de queue

1. Kassa dubbelcheckt of bv grote aankopen wel correct zijn.

CRM //.

Planning //.

## Case 27: Bezoeker/werknemer vraagt om factuur

**Uitvoerende dienst:** Frontend

1. Bezoeker/werknemer meldt zich aan.
2. Bezoeker/werknemer vraagt om factuur te versturen naar email
3. Frontend genereert een [factuurMessage](#_InvoiceRequest_message) en stuurt die naar de queue

**Rabbitmq – queue**

Monitor haalt [factuurMessage](#_InvoiceRequest_message) van de queue.

1. haalt message van de queue en logt deze.

Facturatie haalt [factuurMessage](#_InvoiceRequest_message) van de queue

1. Facturatie stelt op basis van de gegevens in de databank een factuur op met alle aankopen.
2. Facturatie verstuurt de factuur naar zijn email.

Frontend haalt [factuurMessage](#_InvoiceRequest_message) van de queue.

1. Haalt message op om factuur te kunnen tonen op de site.

Wordt de factuur da nook getoond via de site? Lijkt me niet logisch?

CRM haalt de [factuurMessage](#_InvoiceRequest_message) van de queue.

1. CRM slaagt de factuur op.

Planning //.

## Case 28: Bezoeker/werknemer vraagt creditnota aan

**Uitvoerende dienst:** Frontend

1. Bezoeker/werknemer meldt zich aan.
2. Bezoeker/werknemer vraagt om factuur te versturen naar email met aanpassingen
3. Frontend genereert een [creditnotaMessage](#_Creditnota_message) en stuurt die naar de queue

**Rabbitmq – queue**

Monitor haalt [creditnotaMessage](#_Creditnota_message) van de queue.

1. haalt [creditnotaMessage](#_Creditnota_message) van de queue en logt deze

Facturatie haalt [creditnotaMessage](#_Creditnota_message) van de queue

1. Facturatie stelt op basis van de gegevens in de databank een creditnota op met alle wijzigingen.
2. Facturatie verstuurt de creditnota naar zijn email.

Frontend haalt [creditnotaMessage](#_Creditnota_message) van de queue.

1. Haalt message op om factuur te kunnen tonen op de site.

Wordt de factuur da nook getoond via de site? Lijkt me niet logisch?

CRM haalt de [creditnotaMessage](#_Creditnota_message) van de queue.

1. CRM slaagt de factuur op

Planning //.

# Status Flows

## Case 29: Status update van het systeem (werken alle diensten nog naar behoren?)

**Uitvoerende dienst:** Frontend

1. Monitor stuurt een [pingMessage](#_Alive_message) naar elke dienst om de 5 min. Reageren die niet dan stuurt monitor een ping naar de technici.

**Rabbitmq – queue**

Planning haalt de [pingMessage](#_Alive_message)van de queue.

1. Planning stuurt een [okMessage](#_isAlive_message)

Frontend haalt de [pingMessage](#_Alive_message) van de queue.

1. Frontend stuurt een [okMessage](#_isAlive_message)

Monitor haalt de [pingMessage](#_Alive_message)van de queue.

1. Monitor stuurt een [okMessage](#_isAlive_message)

CRM haalt de [pingMessage](#_Alive_message) van de queue.

1. CRM stuurt een [okMessage](#_isAlive_message)

Kassa haalt de [pingMessage](#_Alive_message) van de queue.

1. Kassa stuurt een [okMessage](#_isAlive_message)

Facturatie haalt de [pingMessage](#_Alive_message) van de queue.

1. Facturatie stuurt een [okMessage](#_isAlive_message)

Messages

# Inleiding

De diensten communiceren enkel m.b.v. berichten, voortaan messages genoemd. Messages bevatten data in de vorm van een xml bestand dat gevalideerd dient te worden a.d.h.v. een xsd bestand. Je kan deze bestanden terugvinden op github.

De xml data is opgesplitst in 2 delen, een header en de body. Alle messages bevatten dezelfde data in de header. Dat is:

* Het type bericht.
* Het versienummer.
* De tijd van verzending.
* De naam van de dienst die het bericht verzendt.

Het type bericht is belangrijk om te bepalen of een dienst het bericht al dan niet moet verwerken, en hoe de service het bericht moet verwerken.

Het versienummer wordt gebruikt voor veiligheidsredenen, en om het type operatie op de meegegeven entiteit te bepalen.

Het tijdstip van verzending en de naam van de service worden voornamelijk voor te loggen meegegeven.

# Consumerende diensten

Onderstaande tabel geeft een overzicht van de verschillende types messages, en welke diensten deze berichten consumeren.

## Consumerende diensten per message

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | HAALT DE DIENST DE MESSAGE VAN DE QUEUE? | | | | | | | |
|  |  | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM |
|  | [Organisatie message](#_Organisatie_message) |  | ? |  |  |  | ? |
|  | [Event message](#_Inhoud) |  |  |  |  |  |  |
|  | [Error message](#_Inhoud_1) |  |  |  |  |  |  |
|  | [Sessie message](#_Inhoud_2) |  |  |  |  |  |  |
|  | [Werknemer message](#_Werknemer_message) |  |  |  |  |  |  |
|  | [Bezoeker message](#_Bezoeker_message) |  |  |  |  |  |  |
|  | [Groep message](#_Groep_message) |  |  |  |  |  |  |
|  | [Kalender message](#_Kalender_message) |  |  |  |  |  |  |
|  | [Benodigdheden message](#_Benodigdheden_message) |  |  |  |  |  |  |
|  | [Oplaad message](#_Oplaad_message) |  |  |  |  |  |  |
|  | [Aankoop message](#_Aankoop_message) |  |  |  |  |  |  |
|  | [OnvoldoendeVermogen message](#_OnvoldoendeVermogen_message) |  |  |  |  |  |  |
|  | [Badge message](#_Kaart_message) |  |  |  |  |  |  |
|  | [Factuur message](#_Inhoud_8) |  |  |  |  |  |  |
|  | [Creditnota message](#_Inhoud_9) |  |  |  |  |  |  |
|  | [Ping message](#_Ping_message) |  |  |  |  |  |  |
|  | [isAlive message](#_Ok_message) |  |  |  |  |  |  |
|  | [Taak message](#_Inhoud_12) |  |  |  |  |  |  |
|  | [shift message](#_Inhoud_13) |  |  |  |  |  |  |
|  | [Reservatie message](#_Inhoud_14) |  |  |  |  |  |  |
|  | [Gebruiksitem message](#_GebruiksItem_message) |  |  |  |  |  |  |
|  | [Verkoopsitem message](#_VerkoopsItem_message) |  |  |  |  |  | ? |
|  | [Registratie message](#_Registratie_message) |  |  |  |  |  |  |
|  | locatiemessage |  |  |  |  |  |  |

Vraag CRM:

* Wat is locatiemessage?
* Het zou goed zijn als we enkel totaal van producten en bedragen zouden kunnen bijhouden, ipv elke message  
  bv: Totaal verkochte cola’s, totaal verkochte cakes, .. => Dus mss een stockmessage maken die we dan van de kassa krijgen?

# producerende diensten

Onderstaande tabel geeft een overzicht van de verschillende types messages, en welke diensten deze messages produceren.

## Producerende diensten per message

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PLAATST DE SERVICE DE MESSAGE OP DE QUEUE? | | | | | | | |
|  |  | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM |
|  | [Organisatie message](#_Organisatie_message) |  | ? |  |  |  | ? |
|  | [Event message](#_Inhoud) |  | ? |  |  |  |  |
|  | [Error message](#_Inhoud_1) |  |  |  |  |  |  |
|  | [Sessie message](#_Inhoud_2) |  |  |  |  |  |  |
|  | [Werknemer message](#_Werknemer_message) |  | ? |  |  |  |  |
|  | [Bezoeker message](#_Bezoeker_message) |  |  |  |  |  |  |
|  | [Groep message](#_Groep_message) |  |  |  |  |  |  |
|  | [Kalender message](#_Kalender_message) |  |  |  |  |  |  |
|  | [Benodigdheden message](#_Benodigdheden_message) |  |  |  |  |  |  |
|  | [Oplaad message](#_Oplaad_message) |  |  |  |  |  |  |
|  | [Aankoop message](#_Aankoop_message) |  |  |  |  |  |  |
|  | [OnvoldoendeVermogen message](#_OnvoldoendeVermogen_message) |  |  |  |  |  |  |
|  | [Badge message](#_Kaart_message) |  |  |  |  |  |  |
|  | [Factuur message](#_Inhoud_8) |  |  |  |  |  |  |
|  | [Creditnota message](#_Inhoud_9) |  |  |  |  |  |  |
|  | [Ping message](#_Ping_message) |  |  |  |  |  |  |
|  | [isAlive message](#_Ok_message) |  |  |  |  |  |  |
|  | [Taak message](#_Inhoud_12) |  |  |  |  |  |  |
|  | [Shift message](#_Inhoud_13) |  |  |  |  |  |  |
|  | [Reservatie message](#_Inhoud_14) |  |  |  |  |  |  |
|  | [Gebruiksitem message](#_GebruiksItem_message) |  |  |  |  |  |  |
|  | [Verkoopsitem message](#_VerkoopsItem_message) |  |  |  |  |  |  |
|  | [Registratie message](#_Registratie_message) |  |  |  |  |  |  |
|  | locatiemessage |  |  |  |  |  |  |

# Beschrijving messages

In dit deel geven we een gedetailleerde beschrijving van alle messages. Zoals eerder vermeld bevatten alle messages dezelfde typen data in de header.

### Variabelen van message headers voor alle messages

|  |  |  |
| --- | --- | --- |
|  | Type | Opmerking |
| type | String | Zie titel message |
| versienummer | Integer |  |
| verzender | Enum String | Naam dienst zonder hoofdletter |
| timestamp | DateTime | Formaat: 2019-03-18T20:30:19+01:00 |

## Organisatie [message](#_Waar_nodig?)

In deze message komt te alle informatie over de organisatie te staan. Dit is belangrijk bij het opstellen van een factuur.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| organisatieUUID | UUID | Uniek ID over het hele system |
| eventUUID | UUID |  |
| verantwoordelijke | String |  |
| telefoonnummer | int |  |
| email | String |  |
| adres | String |  |
| btwnummer | string |  |
| bankrekeningnummer | string |  |
| website | String | 1 actief, 0 geannuleerd |
| isActief | Boolean | Geeft aan of user op event aanwezig is |

### 

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| organisatieUUID |  |  |  |  |  |  |
| eventUUID |  |  |  |  |  |  |
| verantwoordelijke |  |  |  |  |  |  |
| telefoonnummer |  |  |  |  |  |  |
| email |  |  |  |  |  |  |
| adres |  |  |  |  |  |  |
| btwnummer |  |  |  |  |  |  |
| bankrekeningnummer |  |  |  |  | ? |  |
| website |  |  |  |  |  |  |
| isActief |  |  |  |  |  |  |

Moet bankrekeningnummer niet aanpasbaar zijn door admin of andere medewerker?

## Event [message](#_Waar_nodig?)

De event message zal alle nodige informatie over het event (Ehackb) meegeven.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| eventUUID | UUID | Uniek ID over het hele system |
| naam | String | Naam van het event: EhackB |
| organisatieUUID | UUID |  |
| omschrijving | String | Korte omschrijving van het event |
| startDatum | Datum | Formaat: DD/MM/YY HH:MM |
| eindDatum | Datum | Formaat: DD/MM/YY HH:MM |
| adres | String | Plaats van het event |
| prijs | Double | Hoeveel kost de opzet van het event |
| beschikbarePlaatsen | Int | Hoeveel man kan er komen |
| isActief | Boolean | Staat het event op actief of geannuleerd |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| eventUUID |  |  |  |  |  |  |
| naam |  |  |  |  |  |  |
| organisatieUUID |  |  |  |  |  |  |
| omschrijving |  |  |  |  |  |  |
| startDatum |  |  |  |  |  |  |
| eindDatum |  |  |  |  |  |  |
| adres |  |  |  |  |  |  |
| prijs |  |  |  |  |  |  |
| beschikbarePlaatsen |  |  |  |  |  |  |
| isActief |  |  |  |  |  |  |

## Error [message](#_Waar_nodig?)

De **errorMessage** wordt verzonden wanneer er een onverwachte fout optreedt waarbij een administrator mogelijks moet ingrijpen.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| errorUUID | UUID | Uniek ID over het hele system |
| errorBericht | String | Dient ter verduidelijking van de error. Bv. Bezoeker/werknemer bestaat al, nullpointerexception, timeslot is niet beschikbaar… |
| type | Char | Geeft de ernst van de error weer. Critical-warning-info |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | * Kassa | Planning | Monitoring | Frontend | CRM | |
| errorUUID |  |  |  |  |  |  |
| errorBericht |  |  |  |  |  |  |
| type |  |  |  |  |  |  |

## Sessie [message](#_Waar_nodig?)

De sessie message dient om mee te geven welke sessie er wordt aangemaakt, gaming, spreker, create corner, …

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| sessieUUID | UUID | Uniek ID over het hele system |
| naam | String |  |
| omschrijving | String |  |
| organisatieUUID | UUID | UUID van de organisatie |
| gastspreker | String | Voor en achternaam spreker |
| startDatum | Datum | Formaat: DD/MM/YY HH:MM |
| eindDatum | Datum | Formaat: DD/MM/YY HH:MM |
| plaats | String | Plaats van sessie |
| prijs | Double | Hoeveel kost de opzet van deze sessie |
| kostPrijsSessie | Double | Hoeveel kost het inschrijven voor sessie |
| beschikbarePlaatsen | Int | Hoeveel man kan er komen |
| benodigdheden |  | Wat zijn de nodige resource voor deze sessie: hardware,… |
| isReserverend | Boolean | Geeft aan of je moet reserveren voor deze sessie of niet |
| isActief | Boolean | Is deze sessie actief of gecanceled |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM |
| sessionUUID |  |  |  |  |  |  |
| naam |  |  |  |  |  |  |
| omschrijving |  |  |  |  |  |  |
| organisatieUUID |  |  |  |  |  |  |
| gastspreker |  |  |  |  |  |  |
| startDatum |  |  |  |  |  |  |
| eindDatum |  |  |  |  |  |  |
| plaats |  |  |  |  |  |  |
| prijs |  |  |  |  |  |  |
| kostPrijsSessie |  |  |  |  |  |  |
| beschikbarePlaatsen |  |  |  |  |  |  |
| benodigdheden |  |  |  |  |  |  |
| isReserverend |  |  |  |  |  |  |
| isActief |  |  |  |  |  |  |

## Werknemer [message](#_Waar_nodig?)

In de werknemer message wordt alle informatie over een werknemer bewaard. Werknemer en bezoeker zijn opgesplitst omdat er zo meer ruimte is voor uitbreidbaarheid, bv wannneer er een extra veld bijkomt omdat de werknemers een t-shirt krijgen om in te werken.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| werknemerUUID | UUID | Uniek ID over het hele system |
| badgeUUID | UUID |  |
| achteraam | String |  |
| voornaam | String |  |
| email | String |  |
| adres | String |  |
| btwummer | string |  |
| bankrekeningnummer | string |  |
| typeWerknemer | String | Geeft het type van werknemer aan: admin, sponsor, spreker, gewone werknemer |
| betaalStatus | Double |  |
| isAanwezig | Boolean | Geeft aan of werknemer op event aanwezig is |
| isGeblokkeerd | Boolean | Geeft aan of deze werknemer geblokkeerd is |

### Benodigde variabelen per dienst

Indien je dienst niets nodig heeft, verwijder dan je kolom!!!!

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| werknemerUUID |  |  |  |  |  |  |
| badgeUUID |  |  |  |  |  |  |
| achternaam |  |  |  |  |  |  |
| voornaam |  |  |  |  |  |  |
| email |  |  |  |  |  |  |
| adres |  |  |  |  |  |  |
| typeWerknemer |  |  |  |  |  |  |
| betaalStatus |  |  |  |  |  |  |
| isAanwezig |  |  |  |  |  |  |
| isGeblokkeerd |  |  |  |  |  |  |

## Bezoeker [message](#_Waar_nodig?)

Een **bezoekerMesage** bevat informatie over een bezoeker. De bezoeker wordt aangemaakt als er een reservatie wordt gemaakt in een van de systemen die dit toelaten.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| bezoekerUUID | UUID | Uniek ID over het hele system |
| badgeUUID | UUID |  |
| achteraam | String |  |
| voornaam | String |  |
| email | String |  |
| adres | String |  |
| btwnummer | string |  |
| bankrekeningnummer | string |  |
| betaalStatus | Double |  |
| isAanwezig | Boolean | Geeft aan of bezoeker op event aanwezig is |
| isGeblokkeerd | Boolean | Geeft aan of deze bezoeker geblokkeerd is |
| isActief | Boolean | Geeft aan of de reservatie geannuleerd is |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| bezoekerUUID |  |  |  |  |  |  |
| badgetUUID |  |  |  |  |  |  |
| achternaam |  |  |  |  |  |  |
| voornaam |  |  |  |  |  |  |
| email |  |  |  |  |  |  |
| adres |  |  |  |  |  |  |
| btwnummer |  |  |  |  |  |  |
| bankrekeningnummer |  |  |  |  |  |  |
| BetaalStatus |  |  |  |  |  |  |
| isAanwezig |  |  |  |  |  |  |
| isGeblokkeerd |  |  |  |  |  |  |
| isActief |  |  |  |  |  |  |

## Groep [message](#_Waar_nodig?)

De groep message wordt gebruikt om zich in te kunnen schrijven voor de gaming sessie. Hierbij is er minstens één person uit de groep die een student is op ehb, enkel de persoon met een email van EhB kan een groep aanmaken. Wanneer een groep message verstuurd wordt kan het dus ook zijn dat er meerdere bezoeker messages verstuurd worden indien deze personen nog niet gekend waren. Om deel te nemen aan de gaming wordt er per lid van de groep ook een nog vast te leggen bedrag aangerekend om mee te mogen doen.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| groepUUID | UUID | Uniek ID over het hele system |
| groepsnaam | String |  |
| adminEmail | String | Moet iemand binnen ehb zijn |
| betaalStatus | Double |  |
| isAanwezig | Boolean | Geeft aan of de groep op event aanwezig is |
| isGeblokkeerd | Boolean | Geeft aan of de groep geblokkeerd is |
| isActief | Boolean | Geeft aan of de reservatie geannuleerd is |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Planning | Monitoring | Frontend | CRM | |
| groepUUID |  |  |  |  |  |
| groepsnaam |  |  |  |  |  |
| adminEmail |  |  |  |  |  |
| betaalStatus |  |  |  |  |  |
| isAanwezig |  |  |  |  |  |
| isGeblokkeerd |  |  |  |  |  |
| isActief |  |  |  |  |  |

## Kalender [message](#_Waar_nodig?)

De kalender message wordt gebruikt om onder andere een planning per bezoeker te kunnen genereren.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| kalenderUUID | UUID | Uniek ID over het hele system |
| link | String | Link naar de kalender drive van Google |
| type | String |  |
| isActief | Boolean |  |

### Benodigde variabelen per dienst

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Planning | Monitoring | Frontend |
| kalenderUUID |  |  |  |
| link |  |  |  |
| type |  |  |  |
| isActief |  |  |  |

## Benodigdheden [message](#_Waar_nodig?)

In deze message komen alle spullen die nodig zijn om een bepaalde sessie te laten doorgaan, bv stekkerdozen, tafels, stoelen, schermen,… Dit is handig voor monitoring om later statistieken van te nemen alsook de planning. Wanneer er dingen zijn die niet vanuit EhB beschikbaar zijn kan hiervoor ook een kost worden bijgehouden die wederom handig kan zijn voor monitoring om volgende jaren een accurate inschatting te maken van de algemene kosten van het event.

Elke benodigheden message bestaat hoogstwaarschijnlijk uit meerdere gebruiksItem messages.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| benodigdhedenUUID | UUID | Uniek ID over het hele systeem |
| sessieUUID | UUID | Uniek ID over het hele systeem, geeft aan voor welk event deze benodigdheden nodig zijn |
| naam | String | Bv alle benodigdheden voor de gamingsessie |
| kostprijsBenodigdheden | Double | De totale kostprijs voor alle items die nodig zijn voor een bepaalde sessie |
| gebruiksItemUUID | UUID | Bv. Stekkerdoos. Elk item binnen het benodigheden ‘pakket’ heeft dan ook een actiefFlag. Door in xml gebruik te maken van nesting |

### Benodigde variabelen per dienst

|  |  |  |
| --- | --- | --- |
| Naam | Planning | Monitoring |
| benodigdhedenUUID |  |  |
| sessieUUID |  |  |
| naam |  |  |
| kostprijsBenodigdheden |  |  |
| gebruiksItemUUID |  |  |

### Oplaad [message](#_Waar_nodig?)

De oplaad message is belangrijk om geld te kunnen zetten op je badge. De badge wordt gebruikt om op het event van de ene locatie naar de andere te kunnen gaan maar ook om bijvoorbeeld aan de bar een drankje mee te betalen. Het is dus de bedoeling dat de bezoeker/werknemer bij aanvang van het event (of gedurende) een bepaald bedrag zet op de badge. Met deze badge kan men tijdens het event dan makkelijk eten en drinken kopen zonder met geld rond te lopen. De badge opladen kan enkel aan de kassa.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| oplaadUUID | UUID | Uniek ID over het hele system |
| badgeUUID | UUID |  |
| bedrag | Double | Hoeveel geld je op je kaart van het event zet |
| betaalmethode | String | Hoe betaald? Met creditkaart, contant, visa, paypal,… |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Monitoring | CRM | |
| oplaadUUID |  |  |  |  |
| badgeUUID |  |  |  |  |
| bedrag |  |  |  |  |
| betaalmethode |  |  |  |  |

## Aankoop [message](#_Waar_nodig?)

Een aankoopMessage bevat informatie over de aankoop van één type verkoopsItem tijdens het event (bijvoorbeeld 3 cola’s). Bij een aankoop van verschillende types zal er per type een aparte aankoopMessage worden verstuurd.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| aankoopUUID | UUID | Uniek ID over het hele system |
| badgeUUID | UUID |  |
| verkoopsItemUUID | UUID | UUID van het item dat je verkoopt |
| oplaadUUID | UUID | UUID van de oplaad message. |
| hoeveelheid | int | Hoeveel van een bepaald verkoopsitem koop je (bv 2 cola’s) |
| hotaalprijs | Double | Hoeveel kost de aankoop van de verkoopsitems samengeteld |
| prijsPerStuk | Double |  |

### Benodigde variabelen per dienst

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Monitoring |
| aankoopUUID |  |  |  |
| verkoopsItemUUID |  |  |  |
| oplaadUUID |  |  |  |
| badgeUUID |  |  |  |
| hoeveelheid |  |  |  |
| totaalprijs |  |  |  |
| prijsPerStuk |  |  |  |

CRM WILT TOTAAL OP HET EINDE VAN DE DAG

## OnvoldoendeVermogen [message](#_Waar_nodig?)

Wanneer je niet meer voldoende geld hebt staan op je badge wordt er een onvoldoendeVermogen message verstuurd.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| onvoldoendeVermogenUUID | UUID | Uniek ID over het hele system |
| badgeUUID | UUID | UUID van de badge |
| oplaadUUID | UUID | UUID van de message met bijhorend |
| bedragAankoop | Int | Hoeveel man kan er komen |

### Benodigde variabelen per dienst

|  |  |  |
| --- | --- | --- |
| Naam | Kassa | Monitoring |
| onvoldoendeVermogenUUID |  |  |
| oplaadUUID |  |  |
| badgeUUID |  |  |
| bedragAankoop |  |  |

## Badge [message](#_Waar_nodig?)

Een **badgeMessage** bevat informatie over een badge. De badge wordt aangemaakt bij het registreren van de persoon aan de kassa. Het saldo van een bage wordt bij elke storting en aankoop geüpdatet.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| badgeUUID | UUID | Uniek ID over het hele system |
| werknemerUUID/bezoekerUUID | UUID | UUID van de bezoeker of werknemer |
| badgeNummer | Int |  |
| saldo | Double |  |

### Benodigde variabelen per dienst

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Monitoring | CRM |
| badgeUUID |  |  |  |  |
| werknemerUUID/bezoekerUUID |  |  |  |  |
| badgeNummer |  |  |  |  |
| saldo |  |  |  |  |

## Factuur [message](#_Waar_nodig?)

Wanneer de bezoeker een factuur aanvraagt online zal deze message verstuurd worden met alle nodige gevens, welke items aangekocht, informatie van de organisatie, totaal bedrag, ….

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| factuurUUID | UUID | Uniek ID over het hele system |
| werknemerUUID/bezoekerUUID | UUID | UUID van de werknemer of bezoeker |
| oplaadUUID | UUID |  |
| aankoopUUID | UUID |  |
| badgeUUID | UUID |  |
| organisatieUUID | UUID |  |
| sessieUUID | UUID |  |
| totaalprijs | Double | Totaal aan uitgaven op het event |

### Benodigde variabelen per dienst

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Naam | Facturatie | Monitoring | CRM | |
| factuurUUID |  |  |  |
| werknemerUUID/bezoekerUUID |  |  |  |
| oplaadUUID |  |  |  |
| aankoopUUID |  |  |  |
| badgeUUID |  |  |  |
| organisatieUUID |  |  |  |
| sessieUUID |  |  |  |
| totaalprijs |  |  |  |

## Creditnota [message](#_Waar_nodig?)

Wanneer er iets moet aangepast worden in de factuur zal er een creditnota message verstuurd worden.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| creditnotaUUID | UUID |  |
| factuurUUID | UUID | Uniek ID over het hele system |
| werknemerUUID/bezoekerUUID | UUID | UUID van de werknemer of bezoeker |
| oplaadUUID | UUID |  |
| aankoopUUID | UUID |  |
| badgeUUID | UUID |  |
| organisatieUUID | UUID |  |
| sessieUUID | UUID |  |
| totaalprijs | Double | Totaal aan uitgaven op het event |

Moeten hier nog alle velden van de factuur instaan?

### Benodigde variabelen per dienst

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Monitoring | CRM | |
| creditnotaUUID |  |  |  |  |
| factuurUUID |  |  |  |  |
| werknemerUUID/bezoekerUUID |  |  |  |  |
| oplaadUUID |  |  |  |  |
| aankoopUUID |  |  |  |  |
| badgeUUID |  |  |  |  |
| organisatieUUID |  |  |  |  |
| sessieUUID |  |  |  |  |
| totaalprijs |  |  |  |  |

## Ping [message](#_Waar_nodig?)

Met de ping message gaat men nagaan of alle diensten (kassa, facturatie, monitoring, …) nog actief zijn. Indien men een negatief antwoord terug krijgt zal men de technische dienst inlichten (een ping naar hen sturen )

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| pingUUID | UUID | Uniek ID over het hele system |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| pingUUID |  |  |  |  |  |  |

## isAlive [message](#_Waar_nodig?)

De isAlive message wordt verstuurd vanuit de dienst zelf die eerst een ping message heeft ontvangen. Het is een bevestiging dat alles in orde is.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| isAliveUUID | UUID | Uniek ID over het hele system |
| isActief | Boolean |  |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| isAliveUUID |  |  |  |  |  |  |
| isActief |  |  |  |  |  |  |

Is overbodig , monitoring heeft ingebouwde functie om te controleren of systemen nog operationeel zijn.

## Taak [message](#_Waar_nodig?)

De taak message heft als nut het beschrijven van alle taken die er zijn op het event. Doormiddel van de verschillende taken kan men deze nadien met een werk message toewijzen aan de werknemers.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| taakUUID | UUID | Uniek ID over het hele system |
| naam | String | Naam van de taak |
| omschrijving | String | Omschrijving van de taak |
| startDatum | Datum | Formaat: DD/MM/YY HH:MM |
| eindDatum | Datum | Formaat: DD/MM/YY HH:MM |
| aantalPersonen | Int | Aantal personen die vereist is om de taak te kunnen voltooien |
| isActief | Booeanl | Bepaalt of de taak actief is of niet. |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Naam | Planning | Monitoring | Frontend | CRM | |
| taakUUID | ✔ | ✔ | ✔ | ✔ |
| naam | ✔ | ✔ | ✔ | ✔ |
| omschrijving | ✔ | ✔ | ✔ | ✔ |
| startDatum | ✔ | ✔ | ✔ | ✔ |
| eindDatum | ✔ | ✔ | ✔ | ✔ |
| aantalPersonen | ✔ | ✔ | ✔ | ✔ |
| isActief | ✔ | ✔ | ✔ | ✔ |

## shift [message](#_Waar_nodig?)

Met deze message wordt een bepaalde taak toegewezen aan één of meerdere werknemers.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| shiftUUID | UUID | Uniek ID over het hele system |
| werknemerUUID | UUID | Uniek ID van user over het hele systeem |
| taakUUID | UUID | Uniek ID van taak over het hele systeem. |

Hoe bepaal je of er een persoon bezig is met de taak. Wat gebeurt er als de persoon afwezig/blacklisted is?

### Benodigde variabelen per dienst

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Naam | Planning | Monitoring | Frontend | CRM | |
| shiftUUID | ✔ | ✔ | ✔ | ✔ |
| werknemerUUID | ✔ | ✔ | ✔ | ✔ |
| taakUUID | ✔ | ✔ | ✔ | ✔ |

## Reservatie [message](#_Waar_nodig?)

Reservatie gebeurt online en registratie pas aan de kassa. Doormiddel van registratie weet men wie er effectief komt opdagen van diegene die zich online hebben gereserveerd.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| reservatieUUID | UUID | Uniek ID over het hele system |
| werknemerUUID/bezoekerUUID | UUID | Uniek ID van user over het hele systeem |
| sessieUUID | UUID | Uniek ID van iedere sessie over het hele systeem |
| betaalStatus | Int | 1 = betaald, 0 niet betaald |
| isActief | Int | 1 = actief, 0 niet actief |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Planning | Monitoring | Frontend | CRM | |
| reservatieUUID |  | ✔ |  | ✔ | ✔ |
| werknemerUUID/bezoekerUUID |  | ✔ |  | ✔ | ✔ |
| sessieID |  | ✔ |  | ✔ | ✔ |
| betaalStatus |  | ✔ |  | ✔ | ✔ |
| isActief |  | ✔ |  | ✔ | ✔ |

## GebruiksItem [message](#_Waar_nodig?)

Met de gebruiksItem message leggen we vast welke materialen er allemaal zijn op het event/sessie. Gaande van banken, stoelen, tafels, schermen, stekkerdozen , … Hier wordt per item een uniek uuid vastgelegd en hoeveel er beschikbaar zijn om die erna bij de benodigheden message te kunnen aanduiden welk er wordt gebruikt per sessie. De isAanwezig flag duidt aan of EhB zelf het materiaal kan voorzien of niet.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| gebruiksItemUUID | UUID | Uniek ID over het hele system |
| naam | string |  |
| hoeveelheidBeschikbaar | Int | Hoevel van het items zijn er ter beschikking |
| isAanwezig | boolean | 1 = de organisatie heeft het zelf, 0 = item moet gehuurd worden |

### Benodigde variabelen per dienst

|  |  |  |
| --- | --- | --- |
| Naam | Planning | Monitoring |
| gebruiksItemUUID | ✔ |  |
| naam | ✔ |  |
| hoeveelheidBeschikbaar | ✔ |  |
| isAanwezig | ✔ |  |

## VerkoopsItem [message](#_Waar_nodig?)

De verkoopsItem message wordt gebruikt om te weten welke items er allemaal verkocht worden. Per item dat verkocht kan worden op het event wordt er dus een message aangemaakt, bv een message voor een cola, een water, een broodje ,…

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| verkoopsItemUUID | UUID | Uniek ID over het hele system |
| naam | String | Naam van het product. Bv. cola |
| prijsPerStuk | Double |  |
| aankoopprijs | Double |  |
| leverancier | String | Naam van de leverancier |
| isActief | Int | 1 = actief, 0 niet actief |

Ik vermoed niet dat we effectief ook een leverancier message moeten hebben met bijhorende flow om drank enzo te bestellen? Vandaar enkel de naam van de leverancier

### Benodigde variabelen per dienst

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Monitoring |
| verkoopsItemUUID |  |  |  |
| naam |  |  |  |
| prijsPerStuk |  |  |  |
| aankoopprijs |  |  |  |
| leverancier |  |  |  |
| isActief |  |  |  |

CRM ZOU GRAAG WETEN OP HET EINDE VAN DE DAG WAT HET TOTAAL IS

## Registratie [message](#_Waar_nodig?)

De registratiemessage kan enkel vanuit de kassa verstuurd worden. Het is pas wanneer je je fysiek aanmeldt op het event dat de kassa een registratie message kan versturen.

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| registratieUUID | UUID | Uniek ID over het hele system |
| werknemerUUID/bezoekerUUID | UUID | Uniek ID van user over het hele systeem |
| betaalStatus | Int | 1 = betaald, 0 niet betaald |
| isAanwezig | Int | 1 = actief, 0 niet actief |

### Benodigde variabelen per dienst

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Naam | Facturatie | Kassa | Planning | Monitoring | Frontend | CRM | |
| registratieUUID |  |  | ✔ |  | ✔ | ✔ |
| werknemerUUID/bezoekerUUID |  |  | ✔ |  | ✔ | ✔ |
| betaalStatus |  |  | ✔ |  | ✔ | ✔ |
| isAanwezig |  |  | ✔ |  | ✔ | ✔ |

## Locatie [message](#_Waar_nodig?)

### Variabelen van message body per message

|  |  |  |
| --- | --- | --- |
| Naam | Type | Opmerking |
| locatieUUID | UUID | Uniek ID over het hele system |
| locatie | String | Plaats waar het gebeurt |

### Benodigde variabelen per dienst

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Planning | Monitoring | Frontend |
| locatieUUID | ✔ |  | ✔ |
| locatie | ✔ |  | ✔ |