



Manual

Voorraadbeheer

Methode

Doel

Het ordenen en bijhouden van de noodzakelijke materialen en reagentia voor het goed functioneren van het laboratorium en zorgen voor continuïteit van zorg.

Principe

Een goed voorraadbeheer systeem zou veel werk moeten besparen en fouten moeten voorkomen. Er moet gestreefd worden naar een zo veel mogelijk geautomatiseerd systeem zodat er zo weinig mogelijk menselijke fouten gemaakt kunnen worden.

Overzicht

Het voorraadbeheer wordt over verschillende tabbladen bijgehouden:

- voorraadbeheer
- minimum voorraad
- te bestellen
- besteld
- opgebruikte reagentia
- vervallen reagentia
- statistieken
- configuration
- jaarrapporten

Voorraadbeheer

De gegevens van de producten die op dit moment in gebruik zijn, worden opgeslagen in het tabblad '**voorraadbeheer**':

Naam kolom	Beschrijving
identificatie van het reagens	Naam van het reagens. Een dropdown menu gebaseerd op de items in het tabblad minimum



Manual

	voorraad. Een item dat niet in de lijst staat kan manueel ingegeven worden.
fabrikant	Naam van de firma. Een dropdown menu gebaseerd op de items op het tabblad configuratie. Een item dat niet in de lijst staat kan manueel ingegeven worden.
lotnummer	Lotnummer van het reagens.
hoeveelheid besteld	De totale hoeveelheid besteld van het reagens.
hoeveelheid in gebruik	Het aantal items in gebruik, maar nog niet opgebruikt, kan hier genoteerd worden.
hoeveelheid opgebruikt	De totale hoeveelheid van de reagens die al opgebruikt is.
ontvangstdatum	De datum waarop de bestelling ontvangen is.
houdbaarheidsdatum	De houdbaarheidsdatum die op het reagens staat. Indien er geen opstaat mag dit veld leeg blijven.
alarm	Automatische cel die het aantal dagen tot verval berekent, staat op -44758 indien er geen vervaldatum opgegeven is.
ingebruikname	Automatisch ingevulde cel die automatisch ingevuld wordt wanneer er voor het eerst een getal ingevuld wordt in de "hoeveelheid opgebruikt" kolom. Vult de datum van de huidige dag in.
opgebruikt	Automatisch ingevulde cel die automatisch ingevuld wordt wanneer het getal in "hoeveelheid besteld" gelijk is aan het getal in "hoeveelheid opgebruikt". Vult de datum van de huidige dag in.
buiten gebruik	Automatisch ingevulde cel die automatisch ingevuld wordt wanneer het item vervallen is. Vult de datum van de huidige dag in.
verificatie	Een cel waar ingevuld wordt of lotnummer al geverifieerd is, indien het geverifieerd is, vul de datum in.

'Ingebruikname' en **'opgebruikt'** worden automatisch ingevuld met de datum wanneer de voorwaarden vervuld zijn. **'Ingebruikname'** wordt ingevuld vanaf de **'hoeveelheid opgebruikt'** meer dan 0 is. **'Opgebruikt'** wordt automatisch ingevuld wanneer de **'hoeveelheid**



Manual

besteld' gelijk is aan de **'hoeveelheid opgebruikt'**. De functie die hiervoor gebruikt wordt noemt onEdit(e).

Iedere nacht controleert het programma of de tabel **'Opgebruikt'** ingevuld is. Als dat het geval is, verplaats het programma deze lijn naar het tabblad **'opgebruikte reagentia'**. Daarnaast wordt ook nog de datum van de actie genoteerd. De functie die hiervoor gebruikt wordt noemt usedUp().

Minimum voorraad

Op het tabblad **'minimum voorraad'** staat een lijst met alle items en hun minimum toegelaten hoeveelheid:

Naam kolom	Beschrijving
identificatie van het reagens	Naam van het reagens. Een dropdown menu gebaseerd op de items in het tabblad minimum voorraad. Een item dat niet in de lijst staat kan manueel ingegeven worden.
fabrikant	Naam van de firma. Een dropdown menu gebaseerd op de items in het tabblad configuratie. Een item dat niet in de lijst staat kan manueel ingegeven worden.
actieve voorraad	De huidige voorraad van het item op dit moment. De actieve voorraad wordt iedere nacht berekend.
minimum voorraad	De minimum voorraad voor het item. Als de actieve voorraad gelijk is aan de minimum voorraad, wordt het item automatisch in het tabblad 'te bestellen' geplaatst.

Iedere nacht wordt de lijst met items uit het tabblad **'minimum voorraad'** afgelopen. Voor elk item in de lijst berekent dit de totale hoeveelheid van het item. Daarna wordt de hoeveelheid vergeleken met de lijst met minimale hoeveelheden van items. Indien het resultaat lager is, wordt het item in het tabblad **'te bestellen'** geplaatst. De datum en de hoeveelheid wordt erbij geplaatst. Als het item al in **'te bestellen'** of **'besteld'** staat, wordt het item niet overgenomen, maar wordt enkel de hoeveelheid upgedate. De functie die hiervoor gebruikt wordt, noemt minimumSupply().

Een nieuw soort item moet ook ingevuld worden in de itemlijst in **'minimum voorraad'**. De firma en de minimale hoeveelheid moeten ook ingevuld worden. Indien dit niet gebeurt wordt dit item niet meegenomen in de bestelcyclus en zal dit item ook niet vermeld worden in het jaarrapport.

Manual

Te bestellen

De gegevens van de producten die op dit moment in afwachting zijn om besteld te worden, worden opgeslagen in het tabblad **'te bestellen'**:

Naam kolom	Beschrijving
datum minimum overschreden	De eerste keer dat het item onder de minimum hoeveelheid komt, wordt hier automatisch de datum ingevuld.
identificatie van het reagens	Naam van het reagens. Een dropdown menu gebaseerd op de items in het tabblad configuratie. Een item dat niet in de lijst staat kan manueel ingegeven worden.
fabrikant	Naam van de firma. Een dropdown menu gebaseerd op de items in het tabblad minimum voorraad. Een item dat niet in de lijst staat kan manueel ingegeven worden.
actieve voorraad	De huidige voorraad van het item op dit moment. De actieve voorraad wordt iedere nacht berekend.
besteld?	Kan aangevinkt worden. Druk na het aanvinken op de knop 'bestel', in het menu 'specialised functions'. Deze functie verplaatst het item en alle bijbehorende data naar het tabblad 'besteld'. Het vinkje wordt automatisch weer verwijderd.

Duid alle items in **'te bestellen'** aan die je besteld hebt. Dit doe je door de checkboxes aan te duiden in de kolom **'besteld?'**. Druk dan in het menu op 'Specialised functions' en dan op **'besteld'**. De functie controleert of er checkboxes in het tabblad **'te bestellen'** aangevinkt zijn. Als er items aangevinkt zijn, verwijdert dit het item uit het tabblad **'te bestellen'** en verplaatst het naar **'besteld'**. Plaats de datum van de actie in **'besteld'** en verwijder het vinkje bij **'te bestellen'**. De functie die hiervoor gebruikt wordt noemt orderItems().

Indien een item besteld wordt dat niet in de lijst van **'te bestellen'** staat, om bijvoorbeeld boven de portokosten te komen, kan een item manueel in de lijst met **'besteld'** toegevoegd worden. Selecteer het item in een lege cel in het tabblad **'besteld'** en duid ook de firma aan. De actieve voorraad mag leeg gelaten worden, deze zal overnacht berekend worden. In de cel **'datum minimum voorraad overschreden'** zal automatisch de tekst " Besteld voor minimale



Manual

hoeveelheid overschreden was.” verschijnen. Daarnaast zal ook de datum van de bestelling automatisch ingevuld worden. De functie die hiervoor gebruikt wordt noemt onEdit(e).

Besteld

De gegevens van de producten die op dit moment in bestelling zijn worden opgeslagen. Dit gebeurt op het tabblad **'besteld'**:

Naam kolom	Beschrijving
datum minimum overschreden	De eerste keer dat het item onder de minimum hoeveelheid komt, wordt hier automatisch de datum ingevuld. Indien het item manueel in de lijst met bestelde items toegevoegd wordt, wordt hier vanzelf de zin 'Besteld voor minimale hoeveelheid overschreden was.' ingevuld.
identificatie van het reagens	Naam van het reagens. Een dropdown menu gebaseerd op de items in het tabblad configuratie. Een item dat niet in de lijst staat kan manueel ingegeven worden.
fabrikant	Naam van de firma. Een dropdown menu gebaseerd op de items in het tabblad minimum voorraad. Een item dat niet in de lijst staat kan manueel ingegeven worden.
actieve voorraad	De huidige voorraad van het item op dit moment. De actieve voorraad wordt iedere nacht berekend.
datum van bestelling	Na het bestellen wordt hier automatisch de datum van de bestelling ingevuld
besteld?	Kan aangevinkt worden. Druk na het aanvinken op de knop 'ontvangen', in het menu 'specialised functions'. Deze functie verplaatst het item en alle bijbehorende data naar het tabblad 'statistieken'. Het vinkje wordt automatisch weer verwijderd.

Duid alle items in **'besteld'** aan die geleverd zijn. Dit doe je door de checkboxes aan te duiden in de kolom **'toegekomen?'**. Druk dan in het menu op 'Specialised functies' en dan op **'toegekomen'**. De functie controleert of er checkboxes in het tabblad **'besteld'** aangevinkt zijn. Als er items aangevinkt zijn, verwijdert dit het item uit het tabblad **'besteld'** en verplaatst het naar



Manual

'**statistieken**'. Plaats de datum van de actie in '**statistieken**' en verwijder het vinkje bij '**besteld**'. De functie die hiervoor gebruikt wordt noemt itemArrived().

Indien een item besteld wordt dat niet in de lijst van '**te bestellen**' staat, om bijvoorbeeld boven de portokosten te komen, kan een item manueel in de lijst met '**besteld**' toegevoegd worden. Selecteer het item in een lege cel in het tabblad '**besteld**' en duid ook de firma aan. De actieve voorraad mag leeg gelaten worden, deze zal overnacht berekend worden. In de cel '**datum minimum voorraad overschreden**' zal automatisch de tekst " Besteld voor minimale hoeveelheid overschreden was." verschijnen. Daarnaast zal ook de datum van de bestelling automatisch ingevuld worden. De functie die hiervoor gebruikt wordt noemt onEdit(e).

Statistieken

De gegevens van de producten die ooit besteld geweest zijn, worden opgeslagen. Dit gebeurt in het tabblad '**statistieken**':

Naam kolom	Beschrijving
datum minimum overschreden	De eerste keer dat het item onder de minimum hoeveelheid komt, wordt hier automatisch de datum ingevuld.
identificatie van het reagens	Naam van het reagens.
fabrikant	Naam van de firma. Een dropdown menu gebaseerd op de items in het tabblad configuratie. Een item dat niet in de lijst staat kan manueel ingegeven worden.
actieve voorraad	De huidige voorraad van het item op dit moment. De actieve voorraad wordt iedere nacht berekend.
datum van bestelling	Na het bestellen wordt hier automatisch de datum van de bestelling ingevuld.
datum toegekomen	Hier wordt automatisch de datum van aankomst ingevuld.

In dit tabblad wordt de data van de volledige bestelling opgeslagen.

Configuratie

In het tabblad '**Configuratie**' kunnen instellingen van het voorraadbeheer gewijzigd worden.

item selection:



Manual

Bij 'item selection' kan een item geselecteerd worden voor het genereren van een rapport voor dit item. Dit maakt een lijst met alle verschillende bestellingen waar er nog een reagens van aanwezig is in het labo. Daarnaast worden ook alle lotnummers en vervaldatum meegegeven. De volledige hoeveelheid aan reagens wordt ook berekend. Het item rapport wordt automatisch gegenereerd en geplaatst in de map '**Item rapports automatically generated**'. Het bestand staat in een Google Documents formaat en krijgt de naam van het item toegewezen, plus de datum en tijd van het generen. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is *totalSupply()*.

E-mailadressen waar een melding van vervallen producten naar toe gezonden worden:

E-mailadressen die hier ingevuld worden, krijgen een mail als de vervaldatum overschreden wordt. Vul het e-mailadres zoveel mogelijk links in op de tabel. Als er witruimte tussen gelaten wordt, zal dit e-mailadres niet opgepikt worden door het systeem. De functie plaatst de datum van verval in de kolom van '**vervallen product**' en verplaatst de gegevens in een aparte tabel '**Vervallen producten**'. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is *expiredProduct()*.

E-mailadressen waar een waarschuwing voor bijna vervallen producten naar toe gezonden wordt (14 dagen voor het verval van het product):

E-mailadressen die hier ingevuld worden, krijgen een mail als het product over 14 dagen vervalst. Vul het e-mailadres zoveel mogelijk links in op de tabel. Als er witruimte tussen gelaten wordt, zal dit e-mailadres niet opgepikt worden door het systeem. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is *almostExpiredProducts()*.

Gemiddelde besteltijd firma's:

Een lijst met firma's kan teruggevonden worden in de tabel. Als een nieuwe firma gebruikt wordt, kan deze gewoon onder de laatste firma toegevoegd worden. De firma is dan ook beschikbaar in het firmaselectievak in de rest van de software. Iedere nacht wordt automatisch alle data herberekend en wordt de gemiddelde besteltijd en het aantal bestellingen weergegeven in de tabel. Daarnaast wordt de gemiddelde besteltijd ook weergegeven in de grafiek. De gemiddelde besteltijd wordt berekend aan de hand van de datum van bestelling tegenover de datum van aankomst. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is *averageOrderTime()*.

Jaarrapporten

In het tabblad '**jaarrapporten**' staat een overzicht van het gebruik van items over 1 jaar. De data wordt hier permanent opgeslagen.



Manual

Specialized functions

Verschillende functies kunnen geselecteerd worden door de gebruiker. Een nieuw menu is hiervoor aangemaakt met de naam '**Specialized functions**'. Dit is terug te vinden in het overzicht boven het voorraadbeheer naast het menu '**help**'. Wanneer het menu geselecteerd wordt, krijg je een overzicht van de verschillende functies. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is onOpen().

Maak rapport voor 1 item

[Zie tabblad configuratie.](#)

Maak een jaarrapport

Druk in het menu op 'Specialised functies' en dan op '**Maak een jaarrapport**'. Vul het jaar in waarvan je een jaarrapport wilt generen. Het jaar moet in het yyyy format genoteerd worden (bijvoorbeeld 2023). Wanneer het jaar niet in het correcte format ingevuld wordt, krijg je een errormelding. Wanneer er wel een correct jaar ingegeven wordt, wordt voor elk item de totale hoeveelheid van dat jaar berekend.

De gegevens worden geplaatst in het tabblad '**jaarrapporten**'. De lijst met items in '**minimum voorraad**' wordt gebruikt, nieuwe items worden onderaan de lijst van '**jaarrapporten**' toegevoegd.

De berekening houdt rekening met wanneer items opgebruikt zijn en welke items al in gebruik zijn. De gegeven waarde is een ruwe berekening die niet volledig zal kloppen voor elk item. Daarna worden de items gesorteerd in volgorde van meest gebruikte items en maakt de functie een grafiek van de 6 meest gebruikte items. De naam van de functie die hiervoor hoofdzakelijk gebruikt wordt is makeYearRapport().

Bestel

[Zie tabblad te bestellen.](#)

Toegekomen

[Zie tabblad besteld.](#)

Time triggers instellen

Ga naar **Extensions => App Scripts => Triggers => Add Triggers**.



Manual

Add Trigger for Inventory Management System

Choose which function to run

expiredProduct

Choose which deployment should run

Head

Select event source

Time-driven

Select type of time based trigger

Day timer

Select time of day

Midnight to 1am

(GMT+02:00)

Failure notification settings

+

Notify me daily

Cancel

Save

Selecteer de gewenste functie, selecteer “**Time-Driven**” bij **select event source**. Kies dan een gewenste tijd en druk op **save**.

Named ranges

De code gebruikt de named ranges in het Google Sheets document. Deze mogen niet verwijderd of gewijzigd worden. Indien dit wel gebeurt wordt de software best teruggezet via versiebeheer (zie archivering). Indien nodig kan een named range ingesteld worden. Dit kan gedaan worden via **Gegevens => Benoemde bereiken => nieuw bereik toevoegen**.

Troubleshooting

De volgorde van de tabbladen mag niet veranderd worden! Dit zal leiden tot het falen van de meeste scripts!

Verantwoordelijke: Yoram Vandenhouwe

DocID: 2024 az zeno manual voorraadbeheer

Versie: 1

Aanmaakdatum: 30-04-2024

Revisiedatum: 30-04-2026

Printdatum: enkel DMS online versie is van kracht

Pagina 9 / 46



Manual

De volgorde van de kolommen mag niet veranderd worden! Dit zal leiden tot het falen van de meeste scripts!

De code gebruikt de named ranges in het Google Sheets document. Deze mogen niet verwijderd of gewijzigd worden!

Error	Mogelijke Oorzaak	Advies
Datums worden in verkeerde cellen ingevuld	De kolommen zijn verplaatst.	Vergelijk de kolommen van de software met de kolommen in deze manual. Indien die niet overeenkomen wordt de software best teruggedraaid naar een vorige versie. Zie archief.
	De tabbladen zijn verplaatst.	Vergelijk de tabbladen van de software met de tabbladen in deze manual. Indien die niet overeenkomen wordt de software best teruggedraaid naar een vorige versie. Zie archief.
Een item komt niet voor in het jaarverslag	Het item staat niet in de lijst met minimum voorraad	Voeg het item toe en voer de 'jaarrapport' functie opnieuw uit.
Een item wordt niet automatisch in het tabblad 'te bestellen' geplaatst.	Het item staat niet in de lijst met minimum voorraad of er is geen minimum voorraad ingevuld	Voeg het item toe onderaan de lijst en vul een minimale hoeveelheid in.



Manual

Onderhoud

Het systeem is gebaseerd op de dienstverlening van Google. Als er een update van de werking van Google sheets gebeurt, kan dit een effect hebben op de werking van het voorraadbeheer. De werking van het voorraadbeheer moet dus gecontroleerd worden bij een relevante verandering aan de software.






Het systeem is daarnaast nog steeds deels afhankelijk van het correct invoeren en handelen van de gebruikers. Het is daarom aangeraden om jaarlijks de huidige voorraad in het systeem te vergelijken met de actuele voorraad in het laboratorium. De voorraad kan dan manueel aangepast worden.

Archivering

Alle veranderingen en de verantwoordelijke voor deze aanpassingen, worden opgeslagen in de versiegeschiedenis. Voor het bekijken van de versiegeschiedenis ga naar **Bestand => versiegeschiedenis => versiegeschiedenis bekijken**. Aan de rechterkant van het scherm kan je dan een overzicht zien van de verschillende versies. Deze zijn selecteerbaar. Wanneer een versie geselecteerd is, worden alle wijzigingen van die versie tegenover de vorige versie aangeduid in het groen.

Een vorige versie kan hersteld worden wanneer er fouten gemaakt zijn. Selecteer de gewenste versie en druk op de knop '**deze versie herstellen**'. Alle data en wijzigingen aan die data zullen teruggezet worden naar deze versie.

Gerelateerde documenten

-  2024 az zeno voorraadbeheersysteem
-  Bestellijst Reagentia en verbruiksartikelen
-  2019 az zeno werkinstructie bestelling verbruiksgoederen
-  2024 az zeno validatie voorraadbeheer
-  AZ ZENO Logboeken Pathologie 2014 samengevoegd



Manual

Wijzigingen

Literatuur

Praktijkrichtlijn laboratorium voor pathologische anatomie

Scripts

Hieronder is een volledige kopie van de JavaScript code, na de validatie, terug te vinden. Dit werd toegevoegd op vraag van de werkplaats.

```
/* eslint-disable require-jsdoc */
/* eslint-disable no-unused-vars */
/* eslint-disable max-len */
// Project: Inventorymanagement system
// Function: a semi-automatic Inventorymanagement system.
// This application is developed for the pathology labo of AZ Zeno.
// Name: Yoram Vandenhoutte
// Start of project: 13/02/2024
// Implementation: 14/05/2024
// Version: 1

/**
 *Declaration
 */

// Get SpreadsheetUrl.
const sheetUrl = SpreadsheetApp.getActive().getUrl();
// Get all the sheets in order.
const sheets = SpreadsheetApp.getActive().getSheets();
const voorraadbeheer = sheets[0];
```



Manual

```
const minimumVoorraad = sheets[1];
const teBestellen = sheets[2];
const besteld = sheets[3];
const opgebruikteReagentia = sheets[4];
const vervallenReagentia = sheets[5];
const statistieken = sheets[6];
const configuration = sheets[7];
const jaarrapporten = sheets[8];

// Get data from named ranges. Tab by tab.
const
voorraadbeheerId=SpreadsheetApp.getActive().getRangeByName('voorraadbeheerId');
const voorraadbeheerIdColumn=voorraadbeheerId.getColumn();
const
voorraadbeheerHoeveelheidBesteld=SpreadsheetApp.getActive().getRangeByName('voorraadbeheerHoeveelheidBesteld');
const
voorraadbeheerHoeveelheidBesteldColumn=voorraadbeheerHoeveelheidBesteld.getColumn();
const
voorraadbeheerLotnummer=SpreadsheetApp.getActive().getRangeByName('voorraadbeheerLotnummer');
const
voorraadbeheerLotnummerColumn=voorraadbeheerLotnummer.getColumn();
const
voorraadbeheerHoeveelheidOpgebruikt=SpreadsheetApp.getActive().getRangeByName('voorraadbeheerHoeveelheidOpgebruikt');
const
voorraadbeheerHoeveelheidOpgebruiktColumn=voorraadbeheerHoeveelheidOpgebruikt.getColumn();
const
voorraadbeheerHoudbaarheidsDatum=SpreadsheetApp.getActive().getRangeByName('voorraadbeheerHoudbaarheidsDatum');
const
```



Manual

```
voorraadbeheerHoudbaarheidsDatumColumn=voorraadbeheerHoudbaarheidsDatum
.getColumn();
const
voorraadbeheerAlarm=SpreadsheetApp.getActive().getRangeByName('voorraad
beheerAlarm');
const voorraadbeheerAlarmColumn=voorraadbeheerAlarm.getColumn();
const
voorraadbeheerIngebruikname=SpreadsheetApp.getActive().getRangeByName('
voorraadbeheerIngebruikname');
const
voorraadbeheerIngebruiknameColumn=voorraadbeheerIngebruikname.getColumn
();
const
voorraadbeheerOpgebruikt=SpreadsheetApp.getActive().getRangeByName('voo
rraadbeheerOpgebruikt');
const
voorraadbeheerOpgebruiktColumn=voorraadbeheerOpgebruikt.getColumn();
const
voorraadbeheerBuitenGebruik=SpreadsheetApp.getActive().getRangeByName('
voorraadbeheerBuitenGebruik');
const
voorraadbeheerBuitenGebruikColumn=voorraadbeheerBuitenGebruik.getColumn
();

const
minimumVoorraadId=SpreadsheetApp.getActive().getRangeByName('minimumVoo
rraadId');
const minimumVoorraadIdColumn=minimumVoorraadId.getColumn();
const
minimumVoorraadFirm=SpreadsheetApp.getActive().getRangeByName('minimumV
oorraadFirm');
const minimumVoorraadFirmColumn=minimumVoorraadFirm.getColumn();
const
minimumVoorraadActieveVoorraad=SpreadsheetApp.getActive().getRangeByNam
e('minimumVoorraadActieveVoorraad');
```

Verantwoordelijke: Yoram Vandenhouwe

DocID: 2024 az zeno manual voorraadbeheer

Versie: 1

Aanmaakdatum: 30-04-2024

Revisiedatum: 30-04-2026

Printdatum: enkel DMS online versie is van kracht

Pagina 14 / 46



Manual

```
const
minimumVoorraadActieveVoorraadColumn=minimumVoorraadActieveVoorraad.get
Column();
const
minimumVoorraadMinimumVoorraad=SpreadsheetApp.getActive().getRangeByNam
e('minimumVoorraadMinimumVoorraad');
const
minimumVoorraadMinimumVoorraadColumn=minimumVoorraadMinimumVoorraad.get
Column();

const
teBestellenDatumMinimum=SpreadsheetApp.getActive().getRangeByName('teBe
stellenDatumMinimum');
const
teBestellenDatumMinimumColumn=teBestellenDatumMinimum.getColumn();
const
teBestellenId=SpreadsheetApp.getActive().getRangeByName('teBestellenId'
);
const teBestellenIdColumn=teBestellenId.getColumn();
const
teBestellenFirma=SpreadsheetApp.getActive().getRangeByName('teBestellen
Firma');
const teBestellenFirmaColumn=teBestellenFirma.getColumn();
const
teBestellenActieveVoorraad=SpreadsheetApp.getActive().getRangeByName('t
eBestellenActieveVoorraad');
const
teBestellenActieveVoorraadColumn=teBestellenActieveVoorraad.getColumn()
;
const
teBestellenBesteld=SpreadsheetApp.getActive().getRangeByName('teBestell
enBesteld');
const teBestellenBesteldColumn=teBestellenBesteld.getColumn();

const
```



Manual

```
besteldDatumMinimum=SpreadsheetApp.getActive().getRangeByName('besteldDatumMinimum');
const besteldDatumMinimumColumn=besteldDatumMinimum.getColumn();
const besteldId=SpreadsheetApp.getActive().getRangeByName('besteldId');
const besteldIdColumn=besteldId.getColumn();
const
besteldFirma=SpreadsheetApp.getActive().getRangeByName('besteldFirma');
const besteldFirmaColumn=besteldFirma.getColumn();
const
besteldActieveVoorraad=SpreadsheetApp.getActive().getRangeByName('besteldActieveVoorraad');
const besteldActieveVoorraadColumn=besteldActieveVoorraad.getColumn();
const
besteldDatumBestelling=SpreadsheetApp.getActive().getRangeByName('besteldDatumBestelling');
const besteldDatumBestellingColumn=besteldDatumBestelling.getColumn();
const
besteldToegekomen=SpreadsheetApp.getActive().getRangeByName('besteldToegekomen');
const besteldToegekomenColumn=besteldToegekomen.getColumn();

const
opgebruikteReagentiaId=SpreadsheetApp.getActive().getRangeByName('opgebruikteReagentiaId');
const opgebruikteReagentiaIdColumn=opgebruikteReagentiaId.getColumn();
const
opgebruikteReagentiaHoeveelheidOpgebruikt=SpreadsheetApp.getActive().getRangeByName('opgebruikteReagentiaHoeveelheidOpgebruikt');
const
opgebruikteReagentiaHoeveelheidOpgebruiktColumn=opgebruikteReagentiaHoeveelheidOpgebruikt.getColumn();
const
opgebruikteReagentiaOpgebruikt=SpreadsheetApp.getActive().getRangeByName('opgebruikteReagentiaOpgebruikt');
const
```

Verantwoordelijke: Yoram Vandenhouwe

DocID: 2024 az zeno manual voorraadbeheer

Versie: 1

Aanmaakdatum: 30-04-2024

Revisiedatum: 30-04-2026

Printdatum: enkel DMS online versie is van kracht

Pagina 16 / 46



Manual

```
opgebruikteReagentiaOpgebruiktColumn=opgebruikteReagentiaOpgebruikt.getColumn();

const
statistiekenDatumOverschreden=SpreadsheetApp.getActive().getRangeByName(
'statistiekenDatumOverschreden');
const
statistiekenDatumOverschredenColumn=statistiekenDatumOverschreden.getColumn();
const
statistiekenId=SpreadsheetApp.getActive().getRangeByName('statistiekenId');
const statistiekenIdColumn=statistiekenId.getColumn();
const
statistiekenFirma=SpreadsheetApp.getActive().getRangeByName('statistiekenFirma');
const statistiekenFirmaColumn=statistiekenFirma.getColumn();
const
statistiekenDatumBestellingen=SpreadsheetApp.getActive().getRangeByName(
'statistiekenDatumBestellingen');
const
statistiekenDatumBestellingenColumn=statistiekenDatumBestellingen.getColumn();
const
statistiekenDatumToegekomen=SpreadsheetApp.getActive().getRangeByName(
'statistiekenDatumToegekomen');
const
statistiekenDatumToegekomenColumn=statistiekenDatumToegekomen.getColumn();

const
jaarrapportenId=SpreadsheetApp.getActive().getRangeByName('jaarrapportenId');
const jaarrapportenIdColumn=jaarrapportenId.getColumn();
// Create array to store all the links.
```



Manual

```
const links = [];
// For each sheet in the spreadsheet add an array element to our array
with the string of the URL for that sheet.
sheets.map((sheet)=>links.push(sheetUrl+'#gid='+sheet.getSheetId()));
const voorraadbeheerlink = links[0]; // Link to the voorraadbeheer tab.
const vervallenReagentialink = links[5]; // Link to the vervallen
reagentia tab.

const maxrange = voorraadbeheer.getRange('A2:O1100'); // Total range to
sort items in the voorraadbeheer tab.

/**
 * These functions trigger when the spreadsheets is opened.
 */
// Add a custom menu to the active spreadsheet, including a separator
and a submenu. This is added when opening the spreadsheet.
function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('Specialised Functions') // Create a new option in
the menu.
    .addItem('Maak rapport voor 1 item', 'totalSupply') // Add an
item with the name of the function and then the link to the funtion.
    .addSeparator() // Adds a line between functions.
    .addItem('Maak een jaarrapport', 'makeYearRapport')
    .addSeparator()
    .addItem('Bestel', 'orderItems')
    .addSeparator()
    .addItem('Toegekomen', 'itemArrived')
    .addToUi();
}

/**
 * Creates a series of functionalities where the conditions are checked
each time a change is made to the spreadsheet.
 */
```



Manual

```
function onEdit(e) {  
    // Declaration  
    const activerange = e.range; // Selects the active range.  
    const activerow = activerange.getRow(); // Selects the active range.  
    const activeColumn = activerange.getColumn(); // Selects the active  
range.  
    // This checks to see if the active sheet is the besteld tab. If  
there is an item in the list but no date of the below minimal supply  
add date in besteld column and message in below minimal supply column.  
    if (SpreadsheetApp.getActiveSheet().getName() ===besteld.getName() &&  
activeColumn === 2) {  
        const addItem = besteld.getRange(activerow,  
besteldIdColumn).getValue(); // Retrieves the name of the item in the  
active row.  
        const datePresent =besteld.getRange(activerow,  
besteldDatumMinimumColumn).getValue(); // Retrieves the date of the  
item in the active row.  
        if (addItem !== '' && datePresent ==='') { // If an item is filled  
in the row but there is no date present:  
            _addDate(besteld, activerow, 5); // Add date in active row.  
            besteld.getRange(activerow,  
besteldDatumMinimumColumn).setValue('Besteld voor minimale hoeveelheid  
overschreden was.');// Set message in active row where the date of  
below minimal supply should be.  
        }  
    }  
  
    // Check on edit if the number of items in that row is equal to the  
max number of items in that row. If this matches then a the current date  
is placed in column 11 ( "opgebruikt").  
    if (SpreadsheetApp.getActiveSheet().getName()  
===voorraadbeheer.getName() && activeColumn ==  
voorraadbeheerHoeveelheidOpgebruiktColumn) { // Check to see if the  
active cell is in the voorraadbeheer tab.
```



Manual

```
const maxitems = voorraadbeheer.getRange(activerow,
voorraadbeheerHoeveelheidBesteldColumn).getValue(); // Retrieves the
total number of orderd items for the current line.

const currentstock = voorraadbeheer.getRange(activerow,
voorraadbeheerHoeveelheidOpgebruiktColumn).getValue(); // Retrieves the
current number of orderd items for the current line.

const dateUsedUp =voorraadbeheer.getRange(activerow,
voorraadbeheerOpgebruiktColumn).getValue(); // Retrieves the date for
when this item was used up, for the current line.

if (maxitems === currentstock && maxitems !== 0 && dateUsedUp ===
'') { // If the maximum number of items is equal to the current number
of used up items and the date is not already filled in:
    _addDate(voorraadbeheer, activerow,
voorraadbeheerOpgebruiktColumn); // Add the current date.
}
}

// Checks on the active line when a product is used for the first
time, when it is: add date in column 10 "ingebruikname".
if (SpreadsheetApp.getActiveSheet().getName() ===
voorraadbeheer.getName() && activeColumn ===
voorraadbeheerHoeveelheidOpgebruiktColumn) {
    const huidigAantal = voorraadbeheer.getRange(activerow,
voorraadbeheerHoeveelheidOpgebruiktColumn).getValue(); // Retrieves the
current number of orderd items for the current line.

    const dateFirstUsed =voorraadbeheer.getRange(activerow,
voorraadbeheerIngebruiknameColumn).getValue(); // Retrieves the date
for when this item was first used, for the current line.

    if (huidigAantal !== 0 && dateFirstUsed === '' ) { // If the number
of items is larger then 0 and the date for the first in use is not
filled in:
        _addDate(voorraadbeheer, activerow,
voorraadbeheerIngebruiknameColumn); // Add the current date.
    }
}
```

Verantwoordelijke: Yoram Vandenhouwe

DocID: 2024 az zeno manual voorraadbeheer

Versie: 1

Aanmaakdatum: 30-04-2024

Revisiedatum: 30-04-2026

Printdatum: enkel DMS online versie is van kracht

Pagina 20 / 46



Manual

```
}

/**
 * These functions use a time trigger to activate
 */
/* Activates using a trigger in the Google App Script application. If
the product is expired, move all data of this product to a separate
sheet to store the data.
Then send an email to all mail addresses in the config sheet. */
function expiredProduct() {
    let row = 2; // Start of the table.
    // Declaration
    const expired = 0; // Expiration date.
    let currentcell = voorraadbeheer.getRange(row,
    voorraadbeheerIdColumn).getValue(); // Retrieve the name of the item
    for the current row.

    // While the row is not empty, check each row to see if the product
    is expired:
    while ( currentcell !== '') {
        const expirationdate = voorraadbeheer.getRange(row,
    voorraadbeheerAlarmColumn).getValue();
        // If expired there are 0 days let till expiration and the column
    for expired date is still empty:
        if (expirationdate === expired) {
            _addDate(voorraadbeheer, row, voorraadbeheerBuitenGebruikColumn);
        }
        // Add the date in column 12.

        // Cuts the row and places the data in a new line in sheet
    'vervallen reagentia'.
        const expiredProduct = voorraadbeheer.getRange(row, 1, 1, 15);
        const destRange =
    vervallenReagentia.getRange(vervallenReagentia.getLastRow()+1, 1);
        expiredProduct.copyTo(destRange, {contentsOnly: false});
        expiredProduct.clear();
    }
```



Manual

```
// Get list of emails in the config tab.
const emailList= _getRowOfData(configuration, 8);
// Send an email if a product is expired.
MailApp.sendEmail({to: emailList,
  subject: 'automatic mail-Expired product',
  htmlBody: 'The product, ' + currentcell + ', has expired and was
placed in the tab vervallen reagentia on the last row. For more
information use the link:' + vervallenReagentialink,
  });
  voorraadbeheer.getRange(row,
voorraadbeheerAlarmColumn).setValue('=DAYS360(configuration!$B$2,H'+row+
')');
}
  row = row +1; // While the row is not empty, check each row to see
if the product is expired.
  currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
}
  maxrange.sort(7); // Sort complete range based on column.
}

// This function uses a trigger to find all products that are almost
expired and send an email to specific users.
function almoustExpiredProducts() {
  let row = 2; // Start of the table.
  const almoustexpired = 14; // Number of days before the product
expires.
  let currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
  Logger.log(currentcell);
  // As long as the current cell is not empty the function goes over
the table and will compare each time the expiration date with the
number of days till it expires.

  while ( currentcell !== '' ) {
```



Manual

```
const expiredate = voorraadbeheer.getRange(row,
voorraadbeheerAlarmColumn).getValue();

const alreadyusedUp =voorraadbeheer.getRange(row,
voorraadbeheerOpgebruiktColumn).getValue();

if (expiredate === almoustexpired && alreadyusedUp === '' ) { // If
the product is 14 days befor expiration.

    Logger.log(currentcell);
    // Get list of emails in the config tab.
    const emailList= _getRowOfData(configuration, 9);
    // Sends an email to the users.
    MailApp.sendEmail({to: emailList,
        subject: 'automatische mail- Bijna Vervallen product',
        htmlBody: 'Het product '+currentcell+' op rij '+row+' zal over
14 dagen vervallen: '+ voorraadbeheerlink,
    });
}

row = row +1; // Go the the next empty row.
currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
}
}

// Calculates the averige time it takes between the order date and the
date of the arrival. This is calculated based on the firm and uses the
data of the tab statestieken.
// Builds a graph with all the firms and the average order length.
function averageOrderTime() {
    const miliSecondsDay = 1000 * 60 * 60 * 24; // Number of milliseconds
in 1 day.
    let firmRow = 15; // First row of the table with firms.
    let firm = configuration.getRange(firmRow, 1).getValue(); // First
firm of the table.
    while (firm !== '') { // While there are firms in the table.
        let totalDaysTillArive =0; // Initialize parameter; total number of
days for 1 firm.
```



Manual

```
let numberOfOrders = 0; // Initialize parameter; total number of
orders for 1 firm.

let averageOrderTime= 0; // Initialize parameter; average number of
days for 1 firm.

let statistiekenRow = 2; // Initialize parameter; first row of the
tab statestieken.

let statFirm = statistieken.getRange(statistiekenRow,
statistiekenFirmaColumn).getValue(); //
while (statFirm !== '') {
    if (statFirm === firm) { // If the firm in statistieken has the
same name as the firm in the list in configuration.
        const startDate =statistieken.getRange(statistiekenRow,
statistiekenDatumBestellingenColumn).getValue(); // Get the start date
of the order.

        const endDate = statistieken.getRange(statistiekenRow,
statistiekenDatumToegekomenColumn).getValue(); // Get the date of
arrival of the order.

        const first =startDate.getTime(); // Get time in milliseconds
sinds Epoc.

        const last =endDate.getTime(); // Get time in milliseconds sinds
Epoc.

        const mili = last - first; // Calculate the defenrence in
milliseconds.

        const daysTillArive= mili/ miliSecondsDay; // Divide the
difference in milliseconds by the number of milliseconds in a day.

        totalDaysTillArive= totalDaysTillArive +daysTillArive; // Add
to the total number of days of orders.

        numberOfOrders= numberOfOrders +1; // Total number of orders.
    }

    statistiekenRow = statistiekenRow +1; // Select the next value in
the tab of statistieken.

    statFirm = statistieken.getRange(statistiekenRow,
statistiekenFirmaColumn).getValue();
}
```




Manual

```
averageOrderTime =totalDaysTillArive/numberOfOrders; // Calculate
the averige lenght of orders the selected firm.
configuration.getRange(firmRow, 2).setValue(averageOrderTime); //
Put the averige lenght of orders in the table.
configuration.getRange(firmRow, 3).setValue(numberOfOrders); // Put
the total number of orders in the table.

firmRow = firmRow +1; // Select the next value in the tab of
configuration.
firm = configuration.getRange( firmRow, 1).getValue();
}

const allCharts =configuration.getCharts(); // Select all charts in
the tab configuration and removes them.
for (const i in allCharts) {
  const chart = allCharts[i];
  configuration.removeChart(chart);
}
firmRow=firmRow-1; // End of the firm table.
const chart = configuration.newChart() // Make a new chart.
  .setChartType(Charts.ChartType.BAR) // Make a barplot chart.
  .setOption('title', 'Gemiddelde besteltijd bij firma\'s' ) //
Give a title.
  .setOption('titleTextStyle.alignment', 'center') // Center the
title.
  .setOption('hAxis.title', 'Firma\'s') // Give a name to the
horizontal axis.
  .setOption('vAxis.title', 'Gemiddeld besteltijd') // Give a name
to the vertical axis.
  .addRange(configuration.getRange('A15:B'+firmRow)) // Select the
range.
  .setOption('colors', ['#1FD0E9']) // Give a color scheme.
  .setOption('height', 386)
```



Manual

```
.setOption('width', 625)
.setPosition(12, 4, 1, 52) // Set the position on the sheet.
.build();

configuration.insertChart(chart); // Adds the new chart.
}

// This function loops over a list in the tab minimum voorraad, for
each item in the list it loops over the list in the tab voorraadbeheer
and calculated the total supplies present for the item in minimum
voorraad.
// The calculation is done by subtracting the used items from the total
items. Next the item is compared to a list of minimum item kwantities,
if the item is lower then the minimum than the te bestellen page and
the besteld page are checked to see if the item is present in the first
30 lines. If the item is present then update the current supplies, if
not present search the first empty line and add the itemname, the
active supply and the current date.
function minimumSupply() {
    let rowMin= 2; // Begin row of the tab minimum voorraad.

    let itemId = minimumVoorraad.getRange(rowMin,
minimumVoorraadIdColumn).getValue(); // The name of the item on
specified row on the minimum voorraad tab.

    // As long as the current cell is not empty the function goes over
the table in minimum voorraad.
    while ( itemId !== '' ) {
        let row = 2; // Begin row of the tab voorraad.
        // Loop over all items in the tab voorraadbeheer.
        let selectedText = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue(); // The name of the item on
specified row on the voorraadbeheer tab.

        let totalitems = 0; // Set the total counted items to 0.
```



Manual

```
while (selectedText !== '') { // Loop over all items in the tab
  voorraadbeheer untill there is an empty row.

  if (itemId === selectedText) { // If the names of the items in
    minimum voorraad and in the voorraadbeheer tab match.

    // Calculate the current number of items by taking the maximum
    number of items and subtracting the used items.

    const maxitem = voorraadbeheer.getRange(row,
    voorraadbeheerHoeveelheidBesteldColumn).getValue();

    const useditem = voorraadbeheer.getRange(row,
    voorraadbeheerHoeveelheidOpgebruiktColumn).getValue();

    currentitem = maxitem- useditem;

    // Added the number of used items in this instance to the
    number of total items.

    totalitems = totalitems + currentitem;
  }

  row = row +1; // Go to the next not empty row in the
  voorraadbeheer tab.

  selectedText = voorraadbeheer.getRange(row,
  voorraadbeheerIdColumn).getValue(); // Get the next name in the table.
}

// Sets the calculated number of supplies in the tabel next to the
correct item.

minimumVoorraad.getRange(rowMin,
minimumVoorraadActieveVoorraadColumn).setValue(totalitems);

const minimumSupply = minimumVoorraad.getRange(rowMin,
minimumVoorraadMinimumVoorraadColumn).getValue(); // Get the minimum
supply for a product from the tab minimum voorraad.

if (totalitems < minimumSupply || totalitems === 0 ) { // If the
current value is lower then the minimum supply or equal to zero.

  const resultTeBestellen = _itemPresentInList(teBestellen, itemId,
```



Manual

```
2); // Look if the item is already present in the tab te bestellen and
notate the row if so.

    const inListTeBestellen=resultTeBestellen.alreadyInList; // If
values was in list then this is true.
    const itemRowTeBestellen=resultTeBestellen.rowWithItem;

    const resultBesteld = _itemPresentInList(besteld, itemId, 2); //
Look if the item is already present in the tab besteld and notate the
row if so.
    const inListBesteld=resultBesteld.alreadyInList; // If values was
in list then this is true.
    const itemRowBesteld=resultBesteld.rowWithItem;

    if (inListTeBestellen=== true) { // If the item was in the list
te bestellen then update the number of items to the current supply.
        teBestellen.getRange(itemRowTeBestellen,
teBestellenActieveVoorraadColumn).setValue(totalitems);
    }
    if (inListBesteld=== true) { // If the item was in the list
besteld then update the number of items to the current supply.
        besteld.getRange(itemRowBesteld,
besteldActieveVoorraadColumn).setValue(totalitems);
    }
    if (inListTeBestellen === false && inListBesteld=== false ) { //
If it is not present in both lists above then add it to the list of te
bestellen items.
        let rowTeBes=2;
        let itemIdTeBest = teBestellen.getRange(rowTeBes,
teBestellenIdColumn).getValue();

        while (itemIdTeBest!=='') { // Find the next empty line of te
bestellen.
            rowTeBes = rowTeBes +1;
            itemIdTeBest = teBestellen.getRange(rowTeBes,
teBestellenIdColumn).getValue();
```



Manual

```
}
    _addDate(teBestellen, rowTeBes, 1); // Add the date.
    teBestellen.getRange(rowTeBes,
teBestellenIdColumn).setValue(itemId); // Add the name of the item.
    const itemFirm = minimumVoorraad.getRange(rowMin,
minimumVoorraadFirmColumn); // The name of the firm on specified row on
the minimum voorraad tab.
    const firmRange = teBestellen.getRange(rowTeBes,
teBestellenFirmaColumn); // Add firm to the tab te bestellen.
    itemFirm.copyTo(firmRange,
SpreadsheetApp.CopyPasteType.PASTE_VALUES, false);

    teBestellen.getRange(rowTeBes,
teBestellenActieveVoorraadColumn).setValue(totalitems); // Add the
current number of items.
}
}

rowMin = rowMin +1; // Go to the next not empty row

itemId = minimumVoorraad.getRange(rowMin,
minimumVoorraadIdColumn).getValue(); // Get the name of the item of the
row.
}
}

// This function uses a trigger to find all products used up and moves
the data to a seperate sheet.
function usedUp() {
    try {
        let row = 2; // Start of the table.
        let currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
```

Verantwoordelijke: Yoram Vandenhouwe

DocID: 2024 az zeno manual voorraadbeheer

Versie: 1

Aanmaakdatum: 30-04-2024

Revisiedatum: 30-04-2026

Printdatum: enkel DMS online versie is van kracht

Pagina 29 / 46



Manual

```
// As long as the current cell is not empty the function loops over
the table and will compare each time the expirationdate with the number
of days till it expires.

while ( currentcell !== '' ) {
    const alreadyusedUp =voorraadbeheer.getRange(row,
    voorraadbeheerOpgebruiktColumn).getValue();

    if ( alreadyusedUp !== '' ) { // If the product is not already
used up:
        const currentrow = voorraadbeheer.getRange(row, 1, 1, 15);
        const destRange =
opgebruikteReagentia.getRange(opgebruikteReagentia.getLastRow()+1, 1);
        currentrow.copyTo(destRange, {contentsOnly: false});
        currentrow.clear();
        voorraadbeheer.getRange(row,
    voorraadbeheerAlarmColumn).setValue('=DAYS360 (configuratie!$B$2,H'+row+
    ') ');
    }
    row = row +1; // Go to the next not empty row
    currentcell = voorraadbeheer.getRange(row,
    voorraadbeheerIdColumn).getValue();
}

maxrange.sort(7); // Sort the whole range on column 7
} catch (error) {
    SpreadsheetApp.getUi().alert(error);
}
}

/**
 * These functions use manual trigger created in the funtion onOpen().
 */

// Calculates the total supply of a selected item.
```



Manual

```
function totalSupply() {
  try {
    let row = 2; // Start of the table.
    // start of itemarray
    const itemarr = [['Item name', 'Lotnumber', 'Expiration date ',
'Number of items']];
    // Get the value of the cell selected in the spreadsheet.
    const selectedText = configuration.getRange(4, 2).getValue();
    // Selects the first tab.
    voorraadbeheer.activate();
    // Get the name of the first item in the list.
    let currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
    if (selectedText === '') { // Gives an error message if the
selected cell is empty.
      SpreadsheetApp.getUi().alert('Select the item you want to check
the inventory for out of the list of in the minimum voorraad tab. Then
run the function again.');
```



Manual

```
const useditem = voorraadbeheer.getRange(row,
voorraadbeheerHoeveelheidOpgebruiktColumn).getValue();
currentitem = maxitem- useditem;
// Adding the number of used items in this instance to the
number of total items.
totalitems = totalitems + currentitem;
const arr = []; // Generates a new array and adds the name of
the item
arr.push(selectedText);
// Takes the expiration date out of the selected line.
Transforms it to string then slices the string to keep relevant
information.
const expirationdateObj = voorraadbeheer.getRange(row,
voorraadbeheerHoudbaarheidsDatumColumn).getValue();
const expirationdateString= expirationdateObj.toString();
const expirationdate = expirationdateString.slice(0, 16);
const lotNumber = voorraadbeheer.getRange(row,
voorraadbeheerLotnummerColumn).getValue();
// Adds items to array.
arr.push(lotNumber);
arr.push(expirationdate);
arr.push(currentitem);
// Adds the new array to the existing table.
itemarr.push(arr);
}
row = row +1; // Go to the next line.
currentcell = voorraadbeheer.getRange(row,
voorraadbeheerIdColumn).getValue();
}
// This function takes a name of an item, a number of items in
total and an array to construct a table.
_createDoc(selectedText, totalitems, itemarr);
} catch (error) {
Logger.log(error);
SpreadsheetApp.getUi().alert(error);
```




Manual

```
}  
}  
  
// Checks to see if items are checked of in the tab te bestellen. If  
checked, places removes the item from the te bestellen tab and places  
the item in the tab besteld. It also adds the date in the new tab en  
resets the checkmark in the old tab.  
function orderItems() {  
  try {  
    for (let i = 2; i < 60; i++) { // Loop over the first 60 items in  
the given tab, in the given column.  
      const orderd = teBestellen.getRange(i,  
teBestellenBesteldColumn).getValue(); // Get the value of the checkbox  
of the item in the list.  
      if (orderd === true) { // Check if the item's checkbox was  
checked, if checked proceed with following code, if not go to next  
line.  
        const currentrow = teBestellen.getRange(i, 1, 1, 4); // Get all  
data of the the current line.  
  
        let rowBes=2; // The first row that will be used.  
        let itemId = besteld.getRange(rowBes,  
besteldIdColumn).getValue(); // The value of the name of the row.  
        while (itemId!='') { // Find the first empty line in the tab  
besteld.  
          rowBes = rowBes +1;  
          itemId = besteld.getRange(rowBes,  
besteldIdColumn).getValue();  
        }  
        const destRange = besteld.getRange(rowBes, 1, 1, 4); // Select  
the range where an empty cell was found.  
        currentrow.copyTo(destRange,  
SpreadsheetApp.CopyPasteType.PASTE_VALUES, false); // Copy all data to  
the new line in the besteld tab and clear the old line.  
        currentrow.clear();  
      }  
    }  
  }  
}
```



Manual

```
        _addDate(besteld, rowBes, 5); // Add the date of this action.
        teBestellen.getRange(i,
teBestellenBesteldColumn).setValue(false); // Reset the checkbox.
    }
}
} catch (error) {
    SpreadsheetApp.getUi().alert(error);
}
}

// Checks to see if items are checked of in the tab besteld. If
checked, places removes the item from the besteld tab and places the
item in the tab statistieken. It also adds the date in the new tab en
resets the checkmark in the old tab.
function itemArrived() {
    try {
        for (let i = 2; i < 60; i++) { // Loop over the first 60 items in
the given tab, in the given column.
            const orderd = besteld.getRange(i,
besteldToegekomenColumn).getValue(); // Get the value of the checkbox
of the item in the list.
            if (orderd === true) { // Check if the item's checkbox was
checked, if checked proceed with following code, if not go to next
line.
                const currentrow = besteld.getRange(i, 1, 1, 5); // Get all
data of the the current line.
                const destRange =
statistieken.getRange(statistieken.getLastRow()+1, 1, 1, 5); // Select
the range where an empty cell was found.
                const rowStat= statistieken.getLastRow()+1; // Get the value of
this row.
                currentrow.copyTo(destRange,
SpreadsheetApp.CopyPasteType.PASTE_VALUES, false); // Copy all data to
the new line in the besteld tab and clear the old line.
            }
        }
    }
}
```



Manual

```
currentrow.clear();

    _addDate(statistieken, rowStat,
statistiekenDatumToegekomenColumn); // Add the date of this action.
    besteld.getRange(i, besteldToegekomenColumn).setValue(false);
// Reset the checkbox.
    }
}
} catch (error) {
    SpreadsheetApp.getUi().alert(error);
}
}

// Make a function to calculate the total number of times an item was
used in the selected year. The calculation is based on the data from
voorraadbeheer tab
// and the tab opgebruikte reagentia. Builds a graph with the 6 most
used items.
function makeYearRapport() {
    try {
        const ui = SpreadsheetApp.getUi();
        const response = ui.prompt('Jaar van het rapport', 'Vul een datum
in tussen 2000 en 2100.', ui.ButtonSet.YES_NO); // Give a prompt to
        // Process the user's response.
        let givenYearDateString=response.getResponseText();
        let givenYearDate = Number(givenYearDateString);
        if (response.getSelectedButton() == ui.Button.YES) { // If they
click yes execute the code below.
            if (givenYearDate >= 2100 || givenYearDate <= 2000) { // Check if
the date is in the yyyy format and is a possible date.
                SpreadsheetApp.getUi().alert('The year needs to be between 2000
and 2100.');// Give the user an errormessage to give a correct date.
            }
            return;
        }
    }
}
```



Manual

```
}  
} else if (response.getSelectedButton() == ui.Button.NO) { // Stop  
the code is "no" is selected. Stop the funtion.  
    return;  
} else { // If clicked away. Stop the funtion.  
    return;  
}  
  
// Loop over the year rapport lists.  
let yearRapportColumnCheck=2;  
let yearRapportColumn= 2;  
let yearRapportColumnName=jaarrapporten.getRange(1,  
yearRapportColumnCheck).getValue();  
let yearIsAlreadyPresent= false;  
while (yearRapportColumnName !== '') {  
    if (yearRapportColumnName===givenYearDate) { // Check if the  
given year is already present. If so overwrite this data with the new  
data below.  
        yearIsAlreadyPresent= true;  
        yearRapportColumn =yearRapportColumnCheck;  
    }  
    yearRapportColumnCheck=yearRapportColumnCheck +1;  
    yearRapportColumnName=jaarrapporten.getRange(1,  
yearRapportColumnCheck).getValue();  
}  
if (yearIsAlreadyPresent=== false) {  
    yearRapportColumn= jaarrapporten.getLastColumn()+1; // Gets the  
last empty column.  
}  
jaarrapporten.getRange(1,  
yearRapportColumn).setValue(givenYearDate);  
let rowMin= 2; // Begin row of the tab minimum voorraad.  
let itemIdMin = minimumVoorraad.getRange(rowMin,  
minimumVoorraadIdColumn).getValue(); // The name of the item on  
specified row on the minimum voorraad tab.  
// As long as the current cell is not empty the function goes over
```



Manual

```
the table in minimum voorraad.
    while ( itemIdMin !== '' ) { // Loop over all the items in the
minimum voorraad tab.
        //
        // Loops over all items in the tab opgebruikteReagentia and
counts the items with the correct name which are used up.
        let rowOpgbrRea = 2; // Begin row of the tab
opgebruikteReagentia.
        let selectedTextopgbrRea =
opgebruikteReagentia.getRange(rowOpgbrRea,
opgebruikteReagentiaIdColumn).getValue(); // The name of the item on
specified row on the opgebruikteReagentia tab.
        let totalitems = 0; // Initialize parameter; set the total
counted items to 0.
        while (selectedTextopgbrRea !== '' ) { // Loop over all items in
the tab opgebruikteReagentia untill there is an empty row.
            if ( itemIdMin === selectedTextopgbrRea ) { // If the names of
the items in minimum voorraad and in the voorraadbeheer tab match.
                const expirationYear =
opgebruikteReagentia.getRange(rowOpgbrRea,
opgebruikteReagentiaOpgebruiktColumn).getValue().getFullYear(); // Gets
the year in yyyy format.
                if ( expirationYear === givenYearDate ) { // Checks to see if
the year is the same as the given year.
                    const useditem = opgebruikteReagentia.getRange(rowOpgbrRea,
opgebruikteReagentiaHoeveelheidOpgebruiktColumn).getValue(); // Gets
the used items and adds them to the total.
                    // Adding the number of used items in this instance to the
number of total items.
                    totalitems = totalitems + useditem;
                }
            }
            rowOpgbrRea = rowOpgbrRea + 1; // Go to the next not empty row
in the opgebruikteReagentia tab.
```



Manual

```
selectedTextopgbrRea =
opgebruikteReagentia.getRange(rowOpgbrRea,
opgebruikteReagentiaIdColumn).getValue(); // Get the next name in the
table.
}
// Loops over all items in the tab voorraadbeheer and counts the
items with the correct name which have been started that year
let rowVoorraad = 2; // Begin row of the tab voorraadbeheer.
let selectedTextVoor = voorraadbeheer.getRange(rowVoorraad,
voorraadbeheerIdColumn).getValue(); // The name of the item on
specified row on the voorraadbeheer tab.
while (selectedTextVoor !== '') { // Loop over all items in the
tab voorraadbeheer untill there is an empty row.
if (itemIdMin === selectedTextVoor) { // If the names of the
items in minimum voorraad and in the voorraadbeheer tab match.
const initiationYear = voorraadbeheer.getRange(rowVoorraad,
voorraadbeheerIngebruiknameColumn).getValue().getFullYear(); // Gets
the year in YYYY format.
if (initiationYear === givenYearDate ) { // Checks to see if
the year is the same as the given year.
const useditem = voorraadbeheer.getRange(rowVoorraad,
voorraadbeheerHoeveelheidOpgebruiktColumn).getValue(); // Gets the used
items and adds them to the total.
// Adding the number of used items in this instance to the
number of total items.
totalitems = totalitems + useditem;
}
}
rowVoorraad = rowVoorraad +1; // Go to the next not empty row
in the opgebruikteReagentia tab.

selectedTextVoor = voorraadbeheer.getRange(rowVoorraad,
voorraadbeheerIdColumn).getValue(); // Get the next name in the table.
}
```



Manual

```
let rowYear= 2; // Begin row of the tab jaar rapport tab.
let itemIdYear = jaarrapporten.getRange(rowYear,
jaarrapportenIdColumn).getValue(); // The name of the item on specified
row on the jaar rapport tab.

let alreadyInList = false; // Sets the boolean to check if an item
is already present in the list.
while (itemIdYear !== '') { // Loop over the list in jaar rapport
tab.
    if (itemIdMin=== itemIdYear) { // Check if the item is already
in the list.
        alreadyInList = true; // Initialize parameter; if the item is
already in the list, set this to true.
        jaarrapporten.getRange(rowYear, yearRapportColumn
).setValue(totalItems);
    }
    rowYear = rowYear +1; // Go to the next not empty row

    itemIdYear = jaarrapporten.getRange(rowYear,
jaarrapportenIdColumn).getValue(); // Get the name of the item of the
row.
}
// If the item is not on the jaar list, add it at the bottom.
if (alreadyInList === false) {
    const destRange =jaarrapporten.getLastRow()+1;
    jaarrapporten.getRange(destRange,
jaarrapportenIdColumn).setValue(itemIdMin);

    jaarrapporten.getRange(destRange, yearRapportColumn
).setValue(totalItems);
}

//
rowMin = rowMin +1; // Go to the next not empty row
```



Manual

```
        itemIdMin = minimumVoorraad.getRange(rowMin,
minimumVoorraadIdColumn).getValue(); // Get the name of the item of the
row.
    }

    const maxSortRange =
jaarrapporten.getRange(jaarrapporten.getLastRow(),
jaarrapporten.getLastColumn()).getA1Notation(); // Get A1 range
notation for the last row and column in the table.

    const sortRange = jaarrapporten.getRange('A2:'+maxSortRange); //
Select the sorting range.

    // Sorts descending by the new column.
    sortRange.sort({column: yearRapportColumn, ascending: false});
    _jaarrapportenChart(yearRapportColumn, givenYearDate);
} catch (error) {
    SpreadsheetApp.getUi().alert(error);
}
}

/**
 * These functions are only used by other funtions
 */
// Function to add the current date in a given cell.
function _addDate(sheet, row, column) {
    let time = new Date(); // Create a new date variable.
    const timeZone = SpreadsheetApp.getActive().getSpreadsheetTimeZone();
    time = Utilities.formatDate(time, timeZone, 'dd/MM/yy'); // Get local
time in dd/MM/yy format.
    sheet.getRange(row, column).setValue(time); // Add date to given
cell.
}

// This function takes the name of the tab and the name of an item and
the column number. It will then check if it can find the name in the
specified column.
```




Manual

```
// Returns an array: alreadyInList is true when the item is present,
false if not. rowWithItem: is 0 when not found, otherwise it will be
the row number where the name was found.
function _itemPresentInList(sheet, itemId, columnNumber) {
  let alreadyInList = false; // Set the default as false
  let rowWithItem = 0; //
  for (let i = 2; i < 30; i++) { // Loop over the first 30 items in the
given tab, in the given column.
    const itemInList = sheet.getRange(i, columnNumber).getValue(); //
Get the name of the item in the list.
    if (itemInList === itemId) { // Compare the name of the item in the
list to the given name in the function.
      alreadyInList = true; // If the same name is found set variables
to true and note the number of the line.
      rowWithItem = i;
    }
  }
  return { // Return array with boolean to see if item is present and
integer of the row where it was found ( 0 if not found)
    alreadyInList: alreadyInList,
    rowWithItem: rowWithItem,
  };
}

// Searches a file in the drive, checks to see if the given foldername
exist in the drive, if not, the script creates it, then moves the file
in the folder.
function _moveFile(nameOfDestination, idOfFile) {
  let folderpresent = false; // Initiates parameter to check if folder
is available
  let folderid = ''; // Initiates the ID string for the folder

  const folders = DriveApp.getFoldersByName('Item rapports automatically
generated'); // Get folder by name.
```



Manual

```
// Loops over all folders and compares the name to the name given.
while (folders.hasNext()) {
    const folder = folders.next();
    folderpresent = true;
    folderid = folder.getId();
}

// If the folder does not exist, create the folder and get the ID
if (folderpresent === false) {
    folderid = DriveApp.createFolder(nameOfDestination).getId();
}

// Get the folder and move file to folder.
const correctfolder = DriveApp.getFolderById(folderid);
DriveApp.getFileById(idOfFile).moveTo(correctfolder);
const fileUrl=DriveApp.getFileById(idOfFile).getUrl();

SpreadsheetApp.getUi().alert('The file is available the
'+nameOfDestination+' folder. ', 'This is the link to the file: \n
'+fileUrl, SpreadsheetApp.getUi().ButtonSet.OK);
}

// This function gets all the data from 1 row from a selected sheet en
concatinates its in a comma sepperated string.
function _getRowOfData(sheet, rowNumber) {
    let columnNumber = 2; // Starting with column 2
    // Uses the coördinates to get the chosen value
    let currentcell = sheet.getRange(rowNumber, columnNumber).getValue();
    // Loops over the row untill a cell is empty en concatinales each new
    item with the previous items.
    let emailList= '';
    while ( currentcell !== '') {
        const emailItem = sheet.getRange(rowNumber,
columnNumber).getValue();
```



Manual

```
emailList= emailList +',' + emailItem;

columnNumber = columnNumber +1; // go to the next not empty row
currentcell = sheet.getRange(rowNumber, columnNumber).getValue();
}
emailList=emailList.slice(1); // Removes the comma at the start of
the string.

return emailList; // Return the string.
}

// This function take a name of an item, a number of items in total and
an array to construct a table.
function _createDoc(itemname, totalitems, itemarr) {
    // Style of the title
    const titleStyle = {};
    titleStyle[DocumentApp.Attribute.HORIZONTAL_ALIGNMENT] =
        DocumentApp.HorizontalAlignment.CENTER;
    titleStyle[DocumentApp.Attribute.FONT_FAMILY] = 'Calibri';
    titleStyle[DocumentApp.Attribute.FONT_SIZE] = 20;
    titleStyle[DocumentApp.Attribute.BOLD] = true;
    // Style of the basic text
    const textStyle = {};
    textStyle[DocumentApp.Attribute.HORIZONTAL_ALIGNMENT] =
        DocumentApp.HorizontalAlignment.LEFT;
    textStyle[DocumentApp.Attribute.FONT_FAMILY] = 'Calibri';
    textStyle[DocumentApp.Attribute.FONT_SIZE] = 12;
    textStyle[DocumentApp.Attribute.BOLD] = false;
    // Style of the solution text
    const solutionStyle = {};
    solutionStyle[DocumentApp.Attribute.HORIZONTAL_ALIGNMENT] =
        DocumentApp.HorizontalAlignment.CENTER;
    solutionStyle[DocumentApp.Attribute.FONT_FAMILY] = 'Calibri';
    solutionStyle[DocumentApp.Attribute.FONT_SIZE] = 14;
    solutionStyle[DocumentApp.Attribute.BOLD] = true;
```



Manual

```
solutionStyle[DocumentApp.Attribute.UNDERLINE] = true;
// Style of the table
const tableStyle = {};
tableStyle[DocumentApp.Attribute.HORIZONTAL_ALIGNMENT] =
    DocumentApp.HorizontalAlignment.CENTER;
tableStyle[DocumentApp.Attribute.FONT_FAMILY] = 'Calibri';
tableStyle[DocumentApp.Attribute.FONT_SIZE] = 12;
tableStyle[DocumentApp.Attribute.BOLD] = false;
solutionStyle[DocumentApp.Attribute.UNDERLINE] = false;

// Generates the time from google
let time = new Date();
const timeZone = SpreadsheetApp.getActive().getSpreadsheetTimeZone();
time = Utilities.formatDate(time, timeZone, 'dd/MM/yy HH:mm');

// Generate the name of the document and get the ID
const newdoc= DocumentApp.create('Item rapport : '+itemname+' '+
time);
const docid= newdoc.getId();
// Open a document by ID.
const doc = DocumentApp.openById(docid);
const body = doc.getBody();

// Append a paragraph to the document body section directly.
const title =body.appendParagraph(itemname);
// Apply the custom style.
title.setAttributes(titleStyle);
// Creates a horizontal line.
body.appendHorizontalRule();
// Generates and adds the main body of text
const standardtext= body.appendParagraph('This document was
automatically generated and uses the data of 2024-Voorraadbeheer to
calculate the quantity of the requested product. This rapport was
generated on '+time+'.');
// Apply the custom style.
```



Manual

```
standardtext.setAttributes(textStyle);
// Creates a paragraph break.
body.appendParagraph('');
// Generates and adds the solution line.
const solution =body.appendParagraph('The total number of times item
'+itemname+' is available is '+ totalitems+'.');
// Apply the custom style.
solution.setAttributes(solutionStyle);

// Build a table from the array.
const table =body.appendTable(itemarr);
// Apply the custom style.
table.setAttributes(tableStyle);
// Searches a file in the drive, checks to see if the given
foldername exist in the drive, if not, the script creates it, then
moves the file in the folder.
_moveFile('Item rapports automatically generated', docid);
}

function _jaarrapportenChart(yearRapportColumn, givenYearDate) {
    // Create a range to build a chart.
    let firstCell = jaarrapporten.getRange(2, yearRapportColumn); //
Select the first row.
    let lastCell= jaarrapporten.getRange(8, yearRapportColumn); // Select
the last row.
    firstCell= firstCell.getA1Notation(); // Change first row range to A1
notation.
    lastCell= lastCell.getA1Notation(); // Change last row range to A1
notation.

    const chart = jaarrapporten.newChart() // Make a new chart.
        .setChartType(Charts.ChartType.BAR) // Make a barplot chart.
        .setOption('title', 'Top 6 most frequently used items of
'+givenYearDate) // Give a title.
        .setOption('titleTextStyle.alignment', 'center') // Center the
```



Manual

```
title.  
    .setOption('hAxis.title', 'Number of items') // Give a name to  
the horizontal axis.  
    .setOption('vAxis.title', 'Itemnames') // Give a name to the  
vertical axis.  
    .addRange(jaarrapporten.getRange('A2:A8')) // Select the item  
name range.  
    .addRange(jaarrapporten.getRange(firstCell+':'+lastCell)) //  
Select the range of the values.  
    .setOption('colors', ['#1FD0E9']) // Give a color scheme.  
    .setPosition(2, 11, 1, 0) // Set the position on the sheet.  
    .build();  
jaarrapporten.insertChart(chart); // Adds the new chart.  
}
```