# Sprint 0.35

Start sprint 6 oktober 2020

In de vorige weken is een Spring Boot opzet gemaakt waarin twee Spring Boot microservices: Koersenmodule en het Beleggingspakket:

Folder op MAC van BKWI:

/Users/abakker/Pakket/ideaal-beleggen

* stockprices is hiervan een subfolder, GIT repo bewaart ook de koersinformatie
* Webapplicatie is hiervan een subfolder, bevat de subfolders voor de microservices

Koersenmodule

Beleggingspakket

Beleggingspakket wordt buiten url om opgestart.

Koersenmodule heeft momenteel een REST endpoint:

localhost:8080/koersen/verversen  
localhost:8080/koersen/verversen?eindJaar=2020&eindMaand=08

Bij het verversen wordt de laatste maand uit de koersbestanden overschreven.

Bij IEX blijkt het zo te zijn dat de dagkoersen op een aparte pagina staan. De koersen van de dag zelf worden pas in de avond of op de volgende dag beschikbaar.

## Fouten:

|  |
| --- |
| B001 Opgelost  Er is een storende fout in de Portefeuillemodule  Die zorgt ervoor dat je “voorkennis” hebt bij het droog beleggen.  Bij aankoop bestens wordt op de openingskoers van de dag gekocht die al zichtbaar is. Bovendien wordt direct na aankoop de slotkoers van de ***vorige*** dag getoond in portefeuilleoverzicht bij de waarde van de positie in plaats van de slotkoers van de laatste dag.  Bij klikken van 1 beursdag moet gekocht worden tegen de koersen van de *volgende* dag. De knop moet zijn uitgegrijsd als de laatst in de grafiek getoonde candle de candle is van de laatste beursdag in de reeks.  Waarschijnlijk niet moeilijk te verhelpen. Geschat op halve dag  Za 10 okt opgelost. Bleek dat ten onrechte twee beursdagen naar rechts werd geschoven bij klikken op beursdag naar rechts bij “retro” beleggen. |

## Wensen:

|  |
| --- |
| W001 Portefeuille op kunnen slaan. Droog beleggen met actuele situatie.  Betekent dat “onthouden” wordt op welke datum we waren gebleven met de portefeuille. Zodra via de portefeuille het grafiekenscherm wordt geopend wordt de grafiek geopend tot op de laatste handelsdag waarop je de grafiek zag.  Schatting: ongeveer 1 dag |
| W002 Tijdelijke dagkoers voor huidige handelsdag ophalen en tonen  Vaak is er behoefte aan om in de grafieken en bij het maken van beleggingsbeslissingen deze laatste dag in de grafieken op te nemen. Wens is dan ook om de dagkoersen erbij te hebben in de grafiek.   * Eventueel in de grafiek een indicatie opnemen bij de laatste candle (zodat duidelijk is dat de koersen bij de laatste candle nog niet definitief zijn * Al dan niet de tijdelijke dagkoers in de koersbestanden op te nemen   Voor het alternatief waarbij de tijdelijke dagkoers niet in het koersbestand wordt opgenomen: Het is mogelijk om vanuit de koersgrafiek een call naar de koersmodule op te nemen zodat voor dat fonds de dagkoers wordt opgehaald en in de grafiek wordt getoond.  Geschat op 1 dag werk |
| W003 Signalen van indicatoren genereren en tonen  De indicatoren geven signalen: bijvoorbeeld het opwaarts doorkruisen van de korte en de langere periode lijn van de MAC.   * Deze signalen bij de indicatoren opnemen en de koop/verkoop momenten opvraagbaar bij de indicator * De signalen zichtbaar maken in de grafiek   Geschat op 1 dag per indicator |
| W004 Betekenis van signalen onderzoeken (statistisch) en zichtbaar maken  Bepaalde vragen rondom de betrouwbaarheid van signalen van indicatoren inzichtelijk maken via een aantal queries op historische data: bijvoorbeeld, hoe groot de kans is bij een macd koopsignaal dat de koers inderdaad gaat stijgen. Gemiddelde stijging en daling binnen een week na dat signaal.  In principe kun je hier een tijdelijke functie voor maken die in het grafiekenscherm dit inzichtelijk maakt bij de verschillende indicatoren die al in beeld zijn via een stukje tekst.  Geschat op 1 dag per indicator |

Alle vier de wijzigingen zijn ongeveer even begeerlijk. Gezien de opbouw van de manier waarop we droog willen beleggen is het waarschijnlijk het verstandigst om bij W001 te beginnen.

## W001 Portefeuille op kunnen slaan. Droog beleggen met actuele situatie.

**Vraag: Wanneer persisteren? Wanneer ophalen?**

Wil nu bij het opslaan in het Portefeuilleobject dit ook opslaan op schijf. Er zijn verschillende mogelijkheden:

* Iedere keer dat er een mutatie optreedt in de portefeuille deze doorgeven aan een speciaal daarvoor bedoelde service
* Aan het einde van de sessie met de portefeuille vragen of deze op schijf moet worden opgeslagen.

Op dit moment is het portefeuille object zowel de (tijdelijke) persistentie als de intelligentie om met orders en dergelijke om te gaan.

**Vraag: hoe op latere datum verder gaan met de portefeuille**

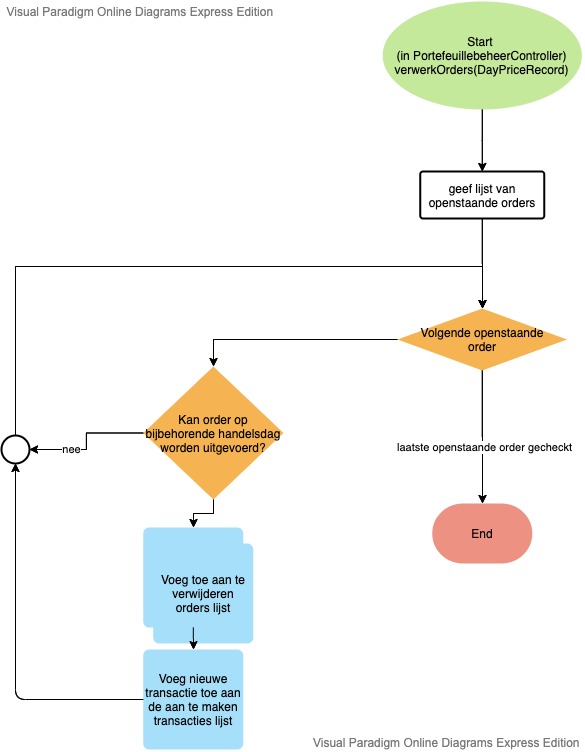
Daarnaast is het van belang dat met een portefeuille “verder” kan worden gegaan. Op dit moment staat er een veldje op het hoofdscherm van waaruit je “terug” kunt gaan in de tijd. Eigenlijk hoort dit gegeven bij de portefeuille (laatste handelsdag in de grafiek komt overeen met de huidige datum minus een hoeveelheid dagen terug in de tijd).

Als je op een latere dag in de tijd het pakket weer opnieuw opstart moet het aantal handelsdagen in het verleden worden gebruikt bij die portefeuille om ervoor te zorgen dat grafieken slechts tot de handelsdag worden getoond waar je het laatst was gebleven.

Processing van orders

In de portefeuillebeheer-controller bevindt zich een methode verwerkOrders(dpr).

Het day-price-record bevat de gegevens van de eindedagkoers op de handelsdag (datum plus open/hoog/laag/slot). Er worden twee lijsten opgebouwd, te verwijderen orders en toe te voegen transacties. Dit aan de hand van de openstaande orders uit het **orders** object.



Vervolgens worden de orders via de de portefeuillebeheercontroller (het pfController object) verwerkt (processMatchedOrders() methode). De orders en de transacties in respectievelijk het orders object en het transacties object worden bijgewerkt. Tevens wordt het rekeningtegoed in de portefeuillebeheercontroller bijgewerkt. Vervolgens wordt het schermobject bijgewerkt met de nieuwe gegevens (addTransactionsToScreen(), addOrdersToScreen() en addPositionsToScreen() van de pfController) en worden de orders getoond via de methode toonOrders().

|  |
| --- |
| result = pfController.verwerkOrders(dpr); pfController.processMatchedOrders(); pfController.addOrdersToScreen(); pfController.addTransactionsToScreen(); pfController.addPositionsToScreen(dpr.getYear(),  dpr.getMonth(), dpr.getDay()); pfController.toonOrders(); |

Reden om de orders pas achteraf te verwerken in het orders object is dat de synchronisatie anders niet goed verloopt. In bijvoorbeeld addOrdersToScreen wordt het nieuwe rekeningtegoed in het scherm getoond in de textbox txtRekeningTegoed.

We hebben dus met de volgende informatie in de volgende objecten te maken:

**pfController**

rekeningTegoed

**transactions** – bevat een lijst van transacties

**orders** – bevat een lijst vannog openstaandeorders

**portefeuille->posities –** bevat een lijst van opgebouwde posities

Posities en rekeningtegoed zijn het gevolg van de transacties die hebben plaatsgevonden.

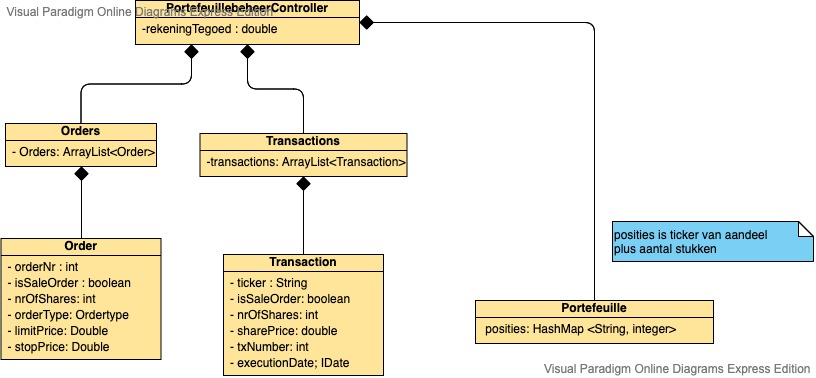
**GrafiekenschermController->candleStickClassObject**

Het candlestickClassObject van het grafiekenscherm bevat de naam van het gekozen aandeel, het eindedagprijzenarray van het aandeel over de totaal bekende periode, het aantal dagen van de koersreeks die moet worden getoond, de startindex en de eindindex van de op dat moment getoonde candles en het laatste DayPriceRecord die bij deze eindindex hoort.

|  |
| --- |
| public CandlestickClass(ArrayList<DayPriceRecord> dayPriceArray,  String gekozenAandeel,  int aAantalBeursdagenKoersreeks,  int aAantalDagenRetro) throws Exception {   myDayPriceArray = dayPriceArray;  init(gekozenAandeel,  aAantalBeursdagenKoersreeks,  aAantalDagenRetro); } |

In feite hoort de laatste handelsdag van de portefeuille bij de portefeuille. Op dit moment wordt vanuit de grafiek via checkUitvoerenOrders() het candlestickobject uitgevraagd voor de eindindex en via het priceArray en het main object doorgegeeven via main.verwerkOrders(daypricerecord).

Het portefeuille object zelf bevat op dit moment alleen de posities.



Portefeuillebeheer-objecten

* Orders en Transactions moeten worden verplaatst naar de portefeuille.
* Laatste handelsdag van de portefeuille moet opgenomen worden in de portefeuille.

Deze speelt een rol bij het tonen van grafieken die handelsdagen tonen die “nieuwer” zijn.

* rekeningTegoed moet verplaatst worden naar de portefeuille.
* Er moeten Opslaan() en Ophalen() methoden in het portefeuille object worden opgenomen.

Grafiekenscherm en gebruikersinteractie

* Er moet een methode komen om een grafiekenscherm te openen gegeven het gebruik in combinatie met een portefeuille. (Voorlopig hebben we slechts 1 portefeuille). Openen van het grafiekenscherm gebeurt op dit moment vanuit het hoofdscherm.
* De grafiek moet bij het starten in combinatie met een portefeuille eindigen op de laatste handelsdag die in de portefeuille staat geregistreerd.
* Belangrijk is daarbij dat het niet mogelijk is om “nieuwere” beursdagen op te vragen vanuit het grafiekenscherm zonder dat er orders worden gecheckt en verwerkt voor deze handelsdagen. Op dit moment zijn er nog twee verschillende knoppen in het grafiekenscherm (“beursdagen naar rechts” en “retro beleggen – 1 beursdag”).
* Vanuit het portefeuillebeheer scherm moet het mogelijk zijn om de portefeuille op te slaan en op te halen.