## De data-service

**De service**

Om onze databank aan te spreken gaan we een rest service voorzien waar we enkele simplistische methodes toekennen die het mogelijk maken om: de ids van de gewijzigde data periodiek op te halen aan de hand van een datum die fungeert als ondergrens en een endpoint waar we één element kunnen ophalen aan de hand van een id. Dit maakt het mogelijk om via een schedular systematisch de gewijzigde producten, winkels en prijzen uit onze databank op te halen en naar AEM te versturen. Of dat dit effectief de beste plan aanpak is voor onze drie modellen kunnen we later nog herzien en, indien dit niet het geval is, onze service aanpassen.

We gebruiken hiervoor een Java Spring-boot service omdat deze makkelijk in opzet zijn maar gezien het een rest service is kan hier perfect voor een andere programmeertaal gekozen worden. Onze configuratie zullen we doen via een application.yml bestand, deze wordt automatisch door de laatste versie van Spring-boot ondersteund.

Om te connecteren met onze Cassandra cluster maken we gebruik van de Datastax java-driver. Datastax is een bedrijf dat een commerciële versie van Apache Cassandra aanbiedt alsook support voorziet. Hun Java-driver is open-source en dus gratis te gebruiken, in mijn ervaring biedt deze ook een betere ondersteuning voor het gebruik van UDTs tegenover de JPA-implementatie voor Cassandra. Om deze driver te gebruiken moeten we onze klassen die overeenstemmen met een tabel in onze databank annoteren met ‘@Table’. De velden in deze klassen krijgen nog de annotatie ‘@Column’ en ‘@PartitionKey’ als het gaat om een primary key kolom. De types van onze databank voorzien we ook klassen voor die we annoteren met ‘@Field’. De velden hier annoteren we met ‘@Field’. Eenmaal we onze tabellen hebben vertaald naar overeenstemmende klassen volstaat het om een sessie met onze cluster aan te maken, bij het aanmaken hiervan volstaat het om het IP-adres van één node te voorzien, de driver zal dan via discovery de locatie van de overige nodes vinden en zo fouttolerantie te waarborgen. Het is toch aan te raden om meerdere IP-adressen mee te geven zodat wanneer er een node niet beschikbaar is tijdens het aangaan van de sessie, deze kan terugvallen op een andere node.

Verder voorzien we drie rest controllers: één voor elk hoofdmodel dat we hebben voorzien. Voor onze AEM-applicatie te kunnen voeden moeten we minstens twee endpoints per controller moeten voorzien, één voor het ophalen van de gewijzigde IDs per model en één voor het effectief ophalen van de data.

**Docker**

Eenmaal onze services operationeel zijn is de volgende stap deze te deployen naar onze AWS-omgeving