



TECH SCREENING .NET

CASUS

TECH SCREENING .NET

1	INTRODUCTIE	3
2	BACKEND	4
3	FRONTEND	5
4	QUALITY ASSURANCE	5
5	EXTRA REQUIREMENTS	5
6	FOCUS	7
7	RULES	7

INTRODUCTIE

Het leven van een IT Rockstar gaat natuurlijk niet over rozen. Zowel zakelijk als privé hebben wij dagelijks te maken met de meest uiteenlopende irritaties. Als Rockstar in hart en nieren doe je er natuurlijk alles aan om die ongemakken weg te automatiseren en als de wereld daar ook beter van wordt is dat uiteraard mooi meegenomen.

Daarom bieden we jou graag een technische challenge aan! De uitdaging is als volgt:

Zoals onze naam doet vermoeden hebben we een bijzondere link met muziek, en dan vooral keiharde Rockmuziek. Dat genre is enorm uiteenlopend en heeft een prachtige historie. Vanwege al die variatie zou het handig zijn om met een paar clicks de juiste band voor het moment te vinden! Van onze IT Rockstars wordt verwacht dat ze zo'n app snel en kundig in elkaar zetten, dan maak je altijd een top indruk bij je collega's en vrienden! Aan jou dus om de volgende specs te vertalen naar de nieuwe appTM.

We verwachten dat de app te openen is met een IDE en dat de backend code geschreven is in C#. De versies van zowel de IDE als de taal zijn open voor eigen invulling, zo ook de exacte invulling van de case.

Good luck and may the force be with you!

BACKEND

De backend moet JSON bestanden met bands en nummers uit kunnen lezen en opslaan in een database. Op de opgeslagen data moeten CRUD (Create/Read/Update/Delete) operaties uitgevoerd kunnen worden.

Onder CRUD functionaliteit verstaan we:

- › (C): Het kunnen toevoegen van nieuwe band(s) of nummers met alle details die ook in de JSON staan waarbij de naam natuurlijk niet dubbel voor mag komen;
- › (R): Het kunnen zoeken naar een band op naam of nummers op genre (voor wanneer je echt zin hebt in Finnish Death Metal);
- › (U): Het kunnen updaten van alle informatie gerelateerd aan bands en nummers op basis van de bijbehorende id's;
- › (D): Het kunnen verwijderen van een band of nummer op basis van de id.

Maak hierbij gebruik van het volgende:

- › een ASP.NET Core Web API project
- › een ORM (bijvoorbeeld Entity Framework of NHibernate)
- › een database naar keuze

Om de JSON files in de database te verwerken volstaat voor nu een enkel endpoint, bijvoorbeeld `"/api/import"`. Wanneer er een PUT naar dat endpoint verstuurt wordt, moet er in grote lijnen het volgende gebeuren:

- › De volgende bestanden worden geautomatiseerd gedownload en gedeserialiseerd:
<https://raw.githubusercontent.com/Team-Rockstars-IT/MusicLibrary/v1.0/artists.json> en
<https://raw.githubusercontent.com/Team-Rockstars-IT/MusicLibrary/v1.0/songs.json>
- › We zijn voor de songs enkel geïnteresseerd in nummers die uitgebracht zijn voor 2016 en waarbij het genre "Metal" bevat
- › Nummers of bands die in de JSON files wél voorkomen en in de database niet, worden toegevoegd en opgeslagen
- › Probeer zo efficiënt mogelijk (lees: met zo min mogelijk database operaties) de nummers en artiesten naar de database te sturen

FRONTEND

Bij deze opdracht wordt er geen frontend implementatie verwacht en moet het mogelijk zijn de web API te testen met bijvoorbeeld Postman of cURL.

QUALITY ASSURANCE

We zien nu al aankomen dat dit een super populaire app wordt. En bij een veelgebruikte app horen goede unit tests en goed leesbare code. Deze applicatie gaat immers nog lang onderhouden moeten worden door mede-developers. We verwachten dan ook unit tests voor de backend en duidelijke naamgeving en documentatie waar nodig. Op z'n minst verwachten we:

- › Een endpoint die Swagger documentatie van jouw API opgeleverd (bijvoorbeeld middels Swashbuckle)
- › >90% unit-test dekking van minimaal 1 class

EXTRA REQUIREMENTS

Om jou als potential IT Rockstar extra te challengen, willen we jou vragen om uit onderstaande lijstje nog een aantal extra requirements te selecteren die je toevoegt in jouw applicatie. De keuze voor de requirements is vrij, het aantal niet.

Zonder te veel in hokjes te denken...hieronder een indicatie voor de aantallen. Meer mag altijd!

Junior:	minimaal 1 extra requirement
Medior:	minimaal 3 extra requirements
Senior:	minimaal 4 extra requirements

Waar moet je dan uit kiezen? Zie hieronder:

- › Zorg ervoor dat bijvoorbeeld een artiest zonder naam of een song zonder titel een melding teruggeeft.
- › Voorzie iedere Controller, Service en Repository van minstens 2 geslaagde unit-tests.
- › De Read operatie die zoekt op een genre moet supersnel zijn. Zorg daarom dat er een cache zit tussen de API en de database.

FOCUS

De focus van deze hele exercitie ligt op de functionaliteit. Daarnaast ontvangen we graag ook vooraf toelichting over bepaalde shortcuts die genomen zijn en een aanpak/ architectuur keuze/ strategie die je gebruikt hebt.

Verder wordt alle code beoordeeld op de volgende aspecten:

- › Security
- › Performance
- › Code style

RULES

Ondanks dat wij niet van regels en verplichtingen houden, zijn er toch een aantal zaken waar je rekening mee moet houden:

- › Je mag maximaal 4 uur van jouw kostbare tijd aan de casus spenderen. Op deze wijze krijgen wij het meest realistische beeld van jouw skills, vaardigheden en snelheid.
- › Het is overigens geen schande als je het binnen deze tijd niet 'af' krijgt!
- › Kwaliteit is in deze beter dan kwantiteit, dus zorg dat hetgeen dat af is ook goed is.
- › De applicatie moet met simpele commando's te runnen zijn en voorzien zijn van duidelijke instructies hoe dit te doen
- › Je levert de solution minimaal 48 uur van te voren aan bij iemand van de Team Rockstars IT Techboard. Doe dat alsjeblieft per mail, in een zipje, eventueel gecombineerd met de git map. De gegevens van de techboarder worden verstrekt door de agent die jouw kwalificatietraject begeleidt.
- › Na het gesprek volgt er een definitieve GO / NO GO vanuit de Rockstar! Bij een GO ontvang je binnen 24 uur een contractvoorstel vanuit jouw Agent!

- › We verwachten dat de appTM veel vaker up is dan github. Zorg er daarom voor dat de API ook om kan gaan met sporadische onbereikbaarheid van github of github die 5xx fouten teruggeeft.
- › We willen alleen de "select few" toegang geven tot de API, want net zoals Rockstars is de app ook zeer exclusief. Zorg ervoor dat we de geaccepteerde API keys in kunnen stellen en dat die daarna bij API aanvragen gevalideerd worden.
- › Introduceer het CQS pattern in de applicatie.
- › Los van dat de app zeer exclusief is, willen we eigenlijk ook dat alleen Rockstars erbij kunnen. Maak daarom een simpel login endpoint die enkel teamrockstars.nl mail adressen accepteert (samen met het super geheime code woord: HappyDevelopers) en een token teruggeeft. Enkel met dat token zijn dan vervolgens de endpoints aan te roepen.
- › Voor de Docker die-hards/experts, gebruik Docker Compose om de applicatie en een losse container voor jouw database in één keer op te starten



WWW.TEAMROCKSTARS.NL