

**Disclaimer** | De lay-out van de PDF-versie voor dit document kan verschillen met de markdown versie, voor een accurate weergave zie markdown bestand in bitbucket.

## Overdrachtsdocument

- Gebruik van Docker
  - Docker Compose
  - Gebruik van Docker tijdens ontwikkeling
  - Gebruik van Docker na uitrol systeem
- Aanbevelingen

## Gebruik van Docker

### Docker Compose

Docker Compose is een applicatie om gemakkelijk meerdere containers mee op te starten. Dit bestand is te vinden onder `docker-compose.yaml`.

Om gebruik te maken van Docker Compose wordt het volgende commando gedraaid in de root directory (`/football-league-manager`) van het project:

```
docker-compose -f docker-compose.yaml -f docker-compose.dev.yaml up --build --force-recreate -d
```

### Gebruik van Docker tijdens ontwikkeling

Tijdens de ontwikkeling van het project wordt gebruik gemaakt van Docker; dit heeft als reden dat we door middel van Docker de omgeving van de applicatie kunnen wegabstraheren. Op deze manier hoeven wij ons zelf niet druk te maken over het installeren en onderhouden van installaties van bijvoorbeeld SQL Server en/of Linux.

Wij gebruiken Docker zo dat elke keer zodra de containers worden opgestart, alle data in de database wordt gereset. Dit houdt in dat alle mockdata scripts opnieuw worden gedraaid en de bestaande data in de docker omgeving wordt verwijderd. Zo beginnen we elke keer met een schone database. Het voordeel hiervan is dat op elk moment de database hetzelfde is zoals gedefinieerd in de SQL bestanden.

Zo is er altijd databaseinhoud om mee te testen en is de database consistent tussen de verschillende ontwikkelaars. Als iemand een schema, view of een procedure aanpast wordt deze wijziging meegecommit met de code. Zo lopen versies nooit uit elkaar.

### Gebruik van Docker na uitrol systeem

Om gebruik te maken van Docker voor het uitrollen van de applicatie is de volgende aanpassing gemaakt:

Binnen `entrypoint.sh` staat de volgende functie:

```
function run_sql_files {
    if [[ "$ENV" == "DEV" ]]; then
        find ./ -type f \( -iname "*.sql" -not -iname "CREATE_DATABASE.sql" \)
    else
        find ./ -type f \( -iname "*.sql" -not -iname "CREATE_DATABASE.sql" -not -iname "*-INSERT_*.sql" \)
    fi
    | sort -n -t / -k 3.1 \
    | xargs -I {} sh -c "echo === Executing '{}' === && /opt/mssql-tools/bin/sqlcmd -S 'localhost' -i {}"
    echo "=== DONE RUNNING FILES ==="
}
```

Deze methode insert alleen mock data als `docker-compose -f docker-compose.yaml -f docker-compose.prod.yaml up --build --force-recreate -d` wordt aangeroepen en creëert alleen een database zonder bestaande data.

data als `docker-compose -f docker-compose.yaml -f docker-compose.prod.yaml up --build --force-recreate -d` wordt aangeroepen.

Daarnaast is de data in de ontwikkelomgeving database niet persistent. Elke keer als de docker ontwikkelomgeving opnieuw wordt opgestart wordt de data in de database gereset. Voor het behouden van data kan gebruik gemaakt worden van volumes, deze kunnen met het volgende commando worden toegepast:

Voor productie (met data persistentie):

```
docker-compose -f docker-compose.yaml -f docker-compose.prod.yaml up --build --force-recreate -d
```

Voor ontwikkelaars (geen data persistentie):

```
docker-compose -f docker-compose.yaml -f docker-compose.dev.yaml up --build --force-recreate -d
```

## Aanbevelingen

- Om ervoor te zorgen dat periodiek data uit de Microsoft SQL Server database wordt omgezet naar JSON en vervolgens wordt overgezet naar MongoDB kan gebruik gemaakt worden van cronjobs.
- JSON wordt op dit moment niet netjes geformat, het sorteren van de JSON na het exporten van data uit de SQL Server kan handig zijn om de JSON als object te gebruiken. (Bijvoorbeeld object CLUB met de key HOME\_CLUB\_NAME die vervolgens de key PERSON\_ID heeft).

Voorbeeld van ongeformateerde JSON:

```
{
  "club_name": "naam",
  "referee_person_id": 12
}
```

Voorbeeld van geformateerde JSON:

```
{
  "club": {
    "name": "naam",
    // ...
  },
  "referee": {
    "id": 12,
    "naam": "bla",
    // ...
  }
}
```

Formateren kan door middel van de SQL statement: FOR JSON PATH (hiervoor moet elke kolom in de SELECT een alias krijgen).