Scowin API's MICROSERVICES

By

V Nikhil Kumar & Sapti Sunil

CONTENTS

Introduction

We have developed an application for Vaccination tracker for Schools, we have created it for selected User Stories for one of the primary actors - School Coordinator. We have implemented three microservices, Student Module, Vaccination Module and Vaccination Reports Module. In the student module we are able to add/edit students, get the student details. In the vaccination module we can add new vaccination details of the students and even edit them. In the Vaccination Report module we can get the details of students along with details of their vaccination.

Technical Implementation

To develop APIs we have used Python Django for Student Module and Vaccination Module, For Vaccination Report we have used Flask. Used Postgres DB for Database storage for Student and Vaccination Module. RabbitMQ is used to communicate between the microservices. Docker is used to containerize the application and to expose the APIs to end users using NGINX as a gateway. As there are many docker containers used docker-compose to integrate all the docker containers. Used Postman to test the APIs.

Microservices for end-users

Student Microservices :

This microservice provides users with the data respective to student details and ability to create/edit/delete students.

GET - http://localhost/students/students - To get details of the students

POST - http://localhost/students/students - To create students

PUT - http://localhost/students/students/{id} - To edit student details

Vaccine Microservices :

This microservice provides users with the data respective to vaccination details and ability to create/edit/delete vaccination for users

GET -   - To get the vaccination details

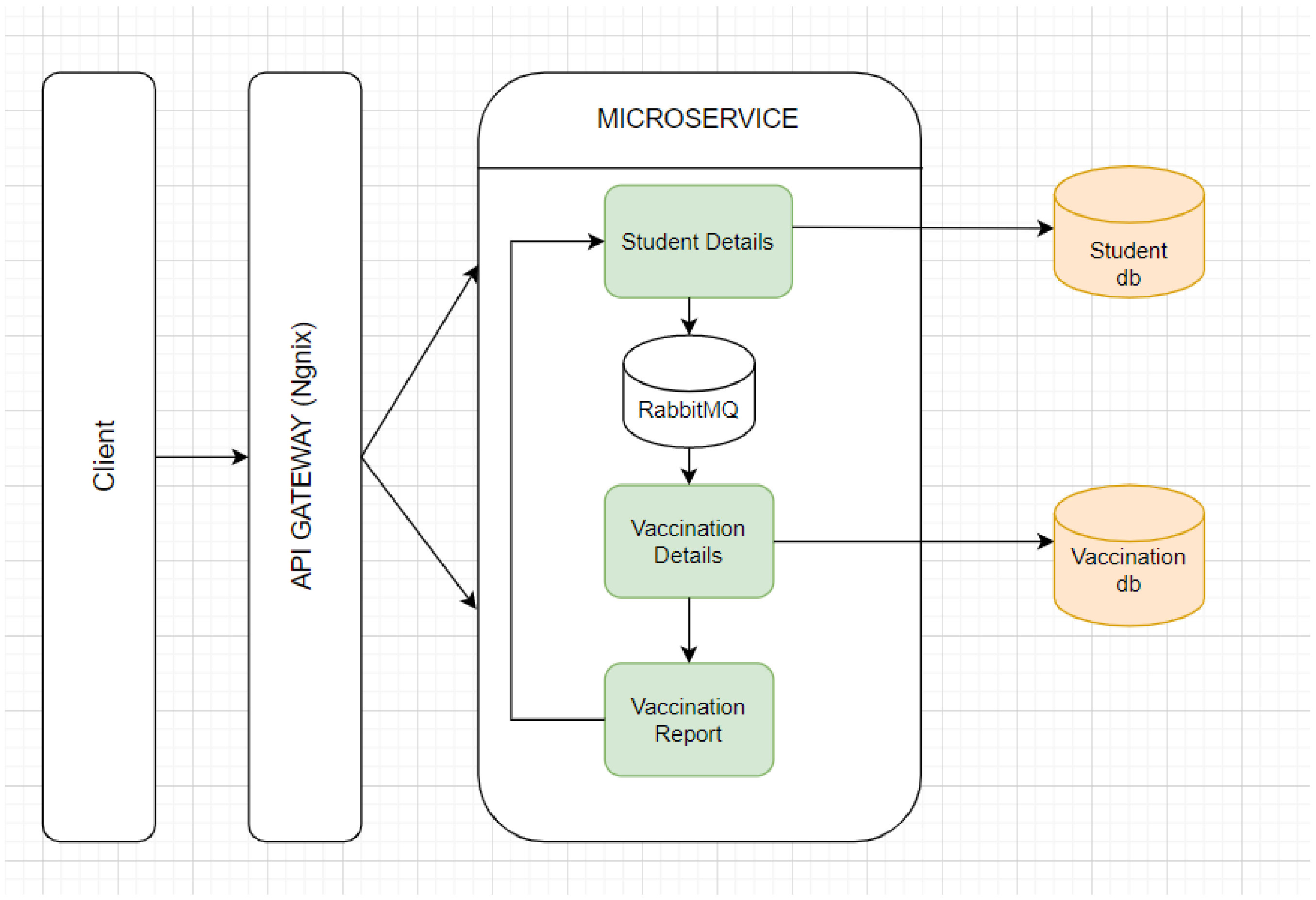
POST - http://localhost/vaccination/student-vaccination - To vaccinate students

Vaccination Report Microservices :

This microservice provides users with the data respective to student details along with their vaccination details

GET - http://localhost/reports/ - To get the vaccination details along with students details

Microservices Architecture



The client requests are handled by API Gateway(Ngnix). Ngnix sends the request to the corresponding microservice based on the configuration. We have different microservices namely Student Details, Vaccination Details and Vaccination Report. Here the communication between the student detail container and vaccination detail container is done using RabbitMQ. RabbitMQ is a messaging broker an intermediary for messaging. Vaccination Report interacts with the vaccination details and student details microservices to get the data required by the users.

Design Patterns

Decomposition Pattern :

Decompose by sub domain
To decompose the application into different microservices we have user Decompose by sub domain pattern, as in the school system sub-domains might be required at multiple places and the same Microservices can be used there. eg: Students Microservice can be used in Library Application, Accounting Application.

Integration Pattern :

API Gateway Pattern
API Gateway is the single point of entry for any microservice call. It can work as a proxy service to route a request to the concerned microservice. So that the end users don't have to worry about the different microservices we used this pattern.

API Composition Pattern
When breaking the business functionality into several microservices, it becomes necessary to think about how to merge the data returned by each service. For this case we have used an API composition pattern to integrate data from different microservices to show it to end users.

Database Pattern :

Database per service Pattern
To make the services loosely coupled, to make them independent we use this pattern.

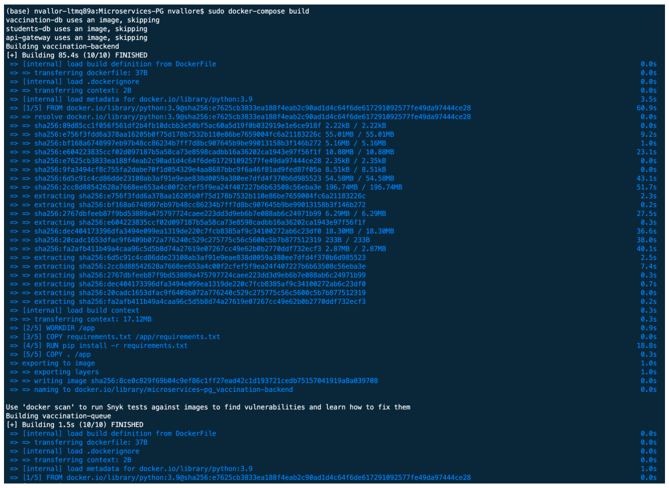
Saga Pattern
Using the Choreography Saga pattern we can asynchronously communicate between different microservices to take an action based on an event.

Communication with services

To communicate between student and vaccination microservices we have used RabbitMQ asynchronous messaging queue and used subscribers and publishers to do respective actions based on the messages. Used Django ORM to communicate with postgresDB. In Vaccination Reports to communicate with different APIs in the other Docker Container within the same network, we have used docker-name to communicate.

Github Link & Execution Instructions

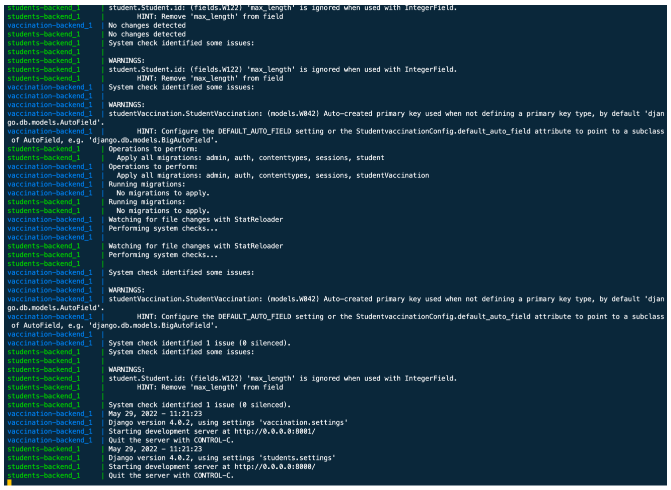
- GIT Repository https://github.com/nvallore/scowin-microservices
- Install Docker and start it
- Clone the GitHub repository
- Open the terminal and navigate to the repository folder
- Run command
  $ docker-compose build



```
(base) nvallor-ltmq89a:Microservices-PG nvallore$ sudo docker-compose build
vaccination-db uses an image, skipping
students-db uses an image, skipping
api-gateway uses an image, skipping
Building vaccination-backend
[+] Building 85.4s (10/10) FINISHED
 => [internal] load build definition from DockerFile                                                                                   0.0s
 => => transferring dockerfile: 37B                                                                                                    0.0s
 => [internal] load .dockerignore                                                                                                      0.0s
 => => transferring context: 2B                                                                                                        0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                                          3.5s
 => [1/5] FROM docker.io/library/python:3.9@sha256:e7625cb3833ea188f4eab2c90ad1d4c64f6de617291092577fe49da97444ce28                   60.9s
 => => resolve docker.io/library/python:3.9@sha256:e7625cb3833ea188f4eab2c90ad1d4c64f6de617291092577fe49da97444ce28                    0.0s
 => => sha256:89d85cc1f056f561df2b4fb10dcbb3e50bf5ac60a5d19f0b032919e1e6ce918f 2.22kB / 2.22kB                                         0.0s
 => => sha256:e756f3fdd6a378aa16205b0f75d178b7532b110e86be7659004fc6a21183226c 55.01MB / 55.01MB                                       9.2s
 => => sha256:bf168a6748997eb97b48cc86234b7ff7d8bc907645b9be99013158b3f146b272 5.16MB / 5.16MB                                         1.0s
 => => sha256:e604223835ccf02d097187b5a58ca73e8598cadbb16a36202ca1943e97f56f1f 10.88MB / 10.88MB                                      23.1s
 => => sha256:e7625cb3833ea188f4eab2c90ad1d4c64f6de617291092577fe49da97444ce28 2.35kB / 2.35kB                                         0.0s
 => => sha256:9fa3494cf8c755fa2dabe70f1d054329e4aa8687bbc9f6a46f81ad9fed87f05a 8.51kB / 8.51kB                                         0.0s
 => => sha256:6d5c91c4cd86dde23108ab3af91e9eae838d0059a380ee7dfd4f370b6d985523 54.58MB / 54.58MB                                      43.1s
 => => sha256:2cc8d88542628a7668ee653a4c00f2cfef5f9ea24f407227b6b63508c56eba3e 196.74MB / 196.74MB                                    51.7s
 => => extracting sha256:e756f3fdd6a378aa16205b0f75d178b7532b110e86be7659004fc6a21183226c                                              2.3s
 => => extracting sha256:bf168a6748997eb97b48cc86234b7ff7d8bc907645b9be99013158b3f146b272                                              0.2s
 => => sha256:2767dbfeeb87f9bd53889a475797724caee223dd3d9eb6b7e088ab6c24971b99 6.29MB / 6.29MB                                        27.5s
 => => extracting sha256:e604223835ccf02d097187b5a58ca73e8598cadbb16a36202ca1943e97f56f1f                                              0.3s
 => => sha256:dec404173396dfa3494e099ea1319de220c7fcb8385af9c34100272ab6c23df0 18.30MB / 18.30MB                                      36.6s
 => => sha256:20cadc1653dfac9f6409b072a776240c529c275775c56c5600c5b7b877512319 233B / 233B                                            38.0s
 => => sha256:fa2afb411b49a4caa96c5d5b8d74a27619e07267cc49e62b0b2770ddf732ecf3 2.87MB / 2.87MB                                        40.1s
 => => extracting sha256:6d5c91c4cd86dde23108ab3af91e9eae838d0059a380ee7dfd4f370b6d985523                                              2.5s
 => => extracting sha256:2cc8d88542628a7668ee653a4c00f2cfef5f9ea24f407227b6b63508c56eba3e                                              7.4s
 => => extracting sha256:2767dbfeeb87f9bd53889a475797724caee223dd3d9eb6b7e088ab6c24971b99                                              0.3s
 => => extracting sha256:dec404173396dfa3494e099ea1319de220c7fcb8385af9c34100272ab6c23df0                                              0.7s
 => => extracting sha256:20cadc1653dfac9f6409b072a776240c529c275775c56c5600c5b7b877512319                                              0.0s
 => => extracting sha256:fa2afb411b49a4caa96c5d5b8d74a27619e07267cc49e62b0b2770ddf732ecf3                                              0.2s
 => [internal] load build context                                                                                                      0.3s
 => => transferring context: 17.12MB                                                                                                   0.3s
 => [2/5] WORKDIR /app                                                                                                                 0.9s
 => [3/5] COPY requirements.txt /app/requirements.txt                                                                                  0.0s
 => [4/5] RUN pip install -r requirements.txt                                                                                         18.8s
 => [5/5] COPY . /app                                                                                                                  0.3s
 => exporting to image                                                                                                                 1.0s
 => => exporting layers                                                                                                                1.0s
 => => writing image sha256:8ce0c829f69b04c9ef86c1ff27ead42c1d193721cedb75157041919a8a039708                                           0.0s
 => => naming to docker.io/library/microservices-pg_vaccination-backend                                                                0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
Building vaccination-queue
[+] Building 1.5s (10/10) FINISHED
 => [internal] load build definition from DockerFile                                                                                   0.0s
 => => transferring dockerfile: 37B                                                                                                    0.0s
 => [internal] load .dockerignore                                                                                                      0.0s
 => => transferring context: 2B                                                                                                        0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                                          1.0s
 => [1/5] FROM docker.io/library/python:3.9@sha256:e7625cb3833ea188f4eab2c90ad1d4c64f6de617291092577fe49da97444ce28                    0.0s
```

- Then run
  $ docker-compose up



```
students-backend_1     | student.Student.id: (fields.W122) 'max_length' is ignored when used with IntegerField.
students-backend_1     |        HINT: Remove 'max_length' from field
vaccination-backend_1  | No changes detected
students-backend_1     | No changes detected
students-backend_1     | System check identified some issues:
students-backend_1     |
students-backend_1     | WARNINGS:
students-backend_1     | student.Student.id: (fields.W122) 'max_length' is ignored when used with IntegerField.
students-backend_1     |        HINT: Remove 'max_length' from field
vaccination-backend_1  | System check identified some issues:
vaccination-backend_1  |
vaccination-backend_1  | WARNINGS:
vaccination-backend_1  | studentVaccination.StudentVaccination: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'djan
go.db.models.AutoField'.
vaccination-backend_1  |        HINT: Configure the DEFAULT_AUTO_FIELD setting or the StudentvaccinationConfig.default_auto_field attribute to point to a subclass
 of AutoField, e.g. 'django.db.models.BigAutoField'.
students-backend_1     | Operations to perform:
students-backend_1     |   Apply all migrations: admin, auth, contenttypes, sessions, student
vaccination-backend_1  | Operations to perform:
vaccination-backend_1  |   Apply all migrations: admin, auth, contenttypes, sessions, studentVaccination
vaccination-backend_1  | Running migrations:
vaccination-backend_1  |   No migrations to apply.
students-backend_1     | Running migrations:
students-backend_1     |   No migrations to apply.
vaccination-backend_1  | Watching for file changes with StatReloader
vaccination-backend_1  | Performing system checks...
vaccination-backend_1  |
students-backend_1     | Watching for file changes with StatReloader
students-backend_1     | Performing system checks...
students-backend_1     |
vaccination-backend_1  | System check identified some issues:
vaccination-backend_1  |
vaccination-backend_1  | WARNINGS:
vaccination-backend_1  | studentVaccination.StudentVaccination: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'djan
go.db.models.AutoField'.
vaccination-backend_1  |        HINT: Configure the DEFAULT_AUTO_FIELD setting or the StudentvaccinationConfig.default_auto_field attribute to point to a subclass
 of AutoField, e.g. 'django.db.models.BigAutoField'.
vaccination-backend_1  |
vaccination-backend_1  | System check identified 1 issue (0 silenced).
students-backend_1     | System check identified some issues:
students-backend_1     |
students-backend_1     | WARNINGS:
students-backend_1     | student.Student.id: (fields.W122) 'max_length' is ignored when used with IntegerField.
students-backend_1     |        HINT: Remove 'max_length' from field
students-backend_1     |
students-backend_1     | System check identified 1 issue (0 silenced).
vaccination-backend_1  | May 29, 2022 - 11:21:23
vaccination-backend_1  | Django version 4.0.2, using settings 'vaccination.settings'
vaccination-backend_1  | Starting development server at http://0.0.0.0:8001/
vaccination-backend_1  | Quit the server with CONTROL-C.
students-backend_1     | May 29, 2022 - 11:21:23
students-backend_1     | Django version 4.0.2, using settings 'students.settings'
students-backend_1     | Starting development server at http://0.0.0.0:8000/
students-backend_1     | Quit the server with CONTROL-C.
```