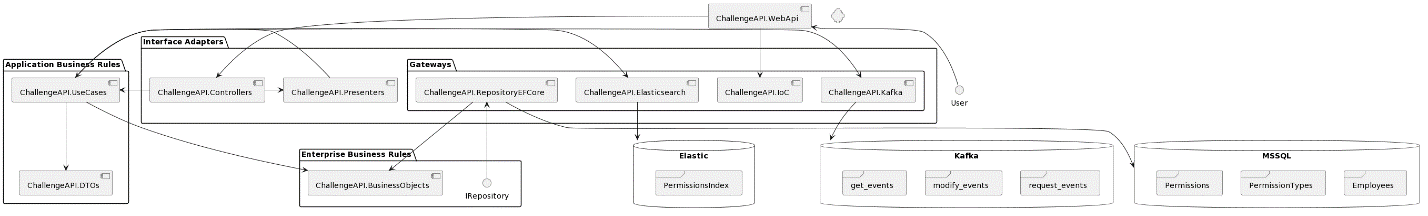
Components Diagram



\*plantuml

Create Permission Test

curl --location 'https://localhost:44367/api/CreatePermission' \

--header 'Content-Type: application/json' \

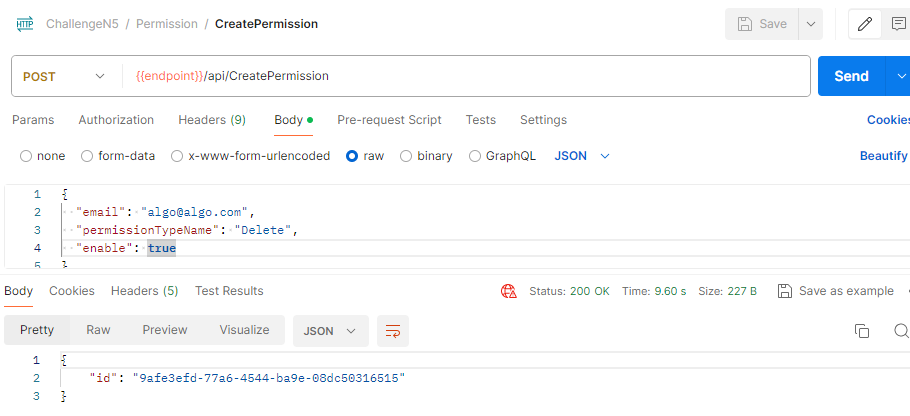
--data-raw '{

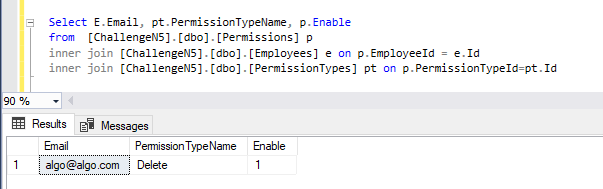
  "email": "algo@algo.com",

  "permissionTypeName": "Delete",

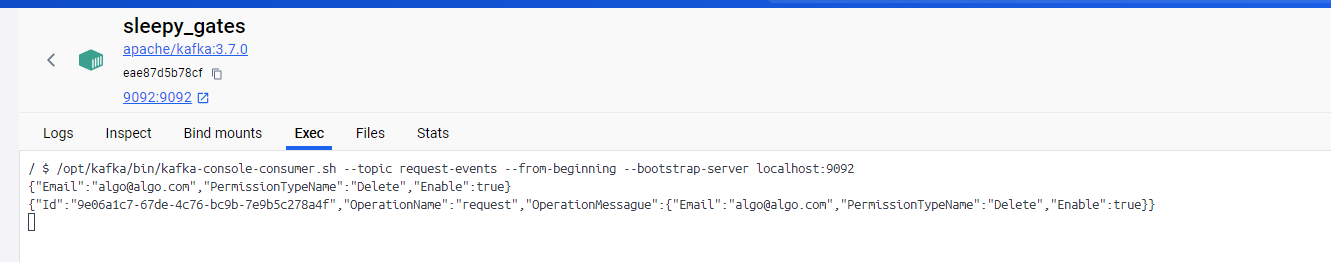
  "enable": true

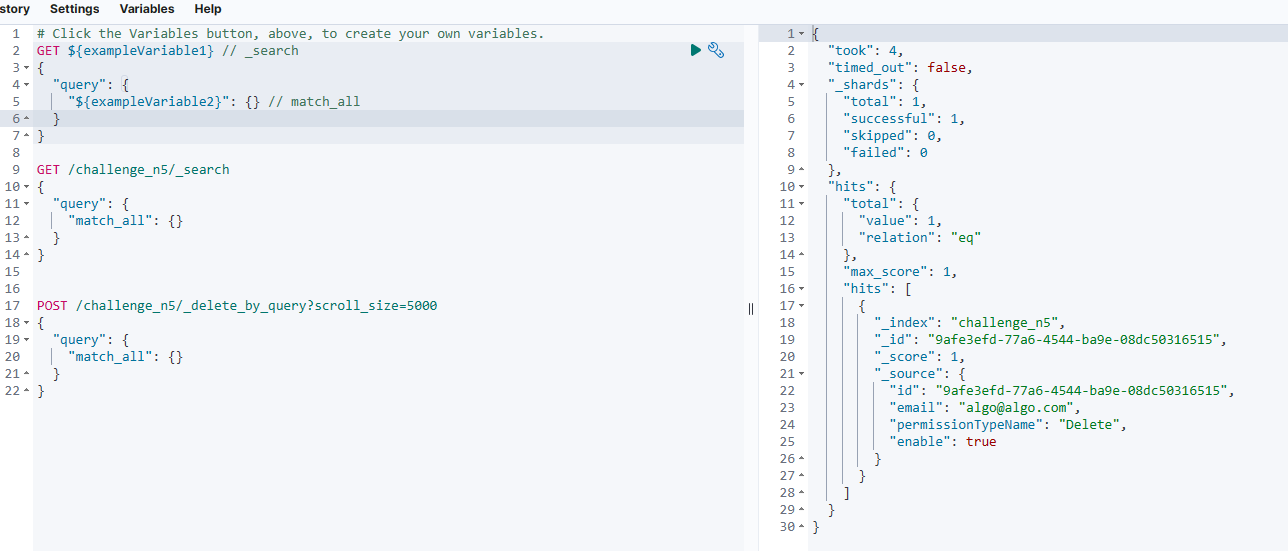
}'





/opt/kafka/bin/kafka-console-consumer.sh --topic request-events --from-beginning --bootstrap-server localhost:9092





Update Permission Test

curl --location --request PUT 'https://localhost:44367/api/UpdatePermission' \

--header 'Content-Type: application/json' \

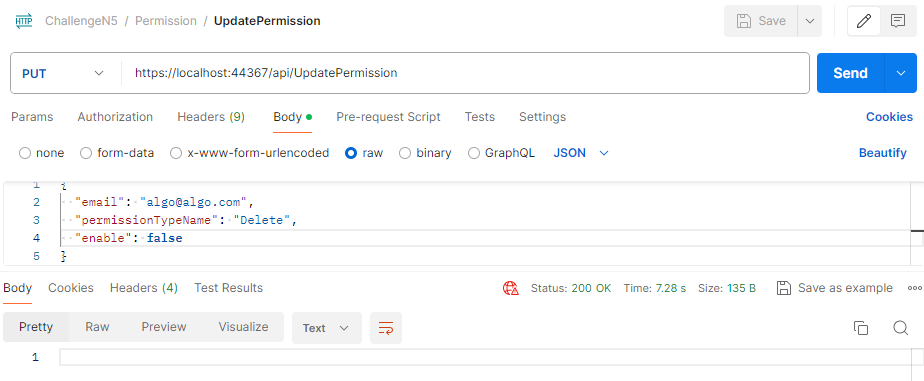
--data-raw '{

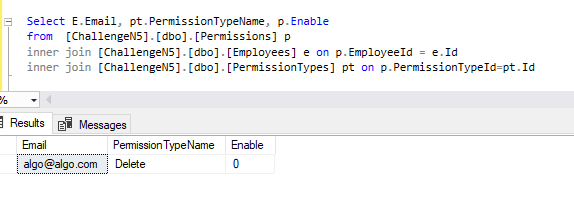
  "email": "algo@algo.com",

  "permissionTypeName": "Delete",

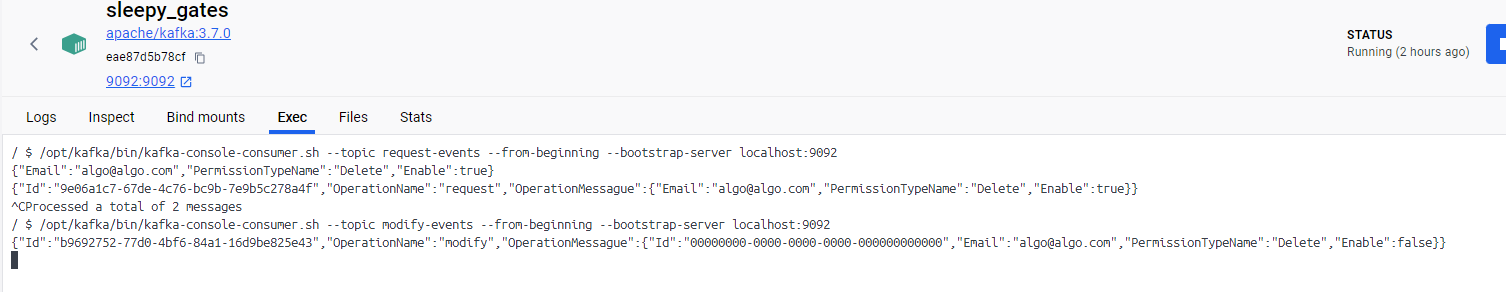
  "enable": false

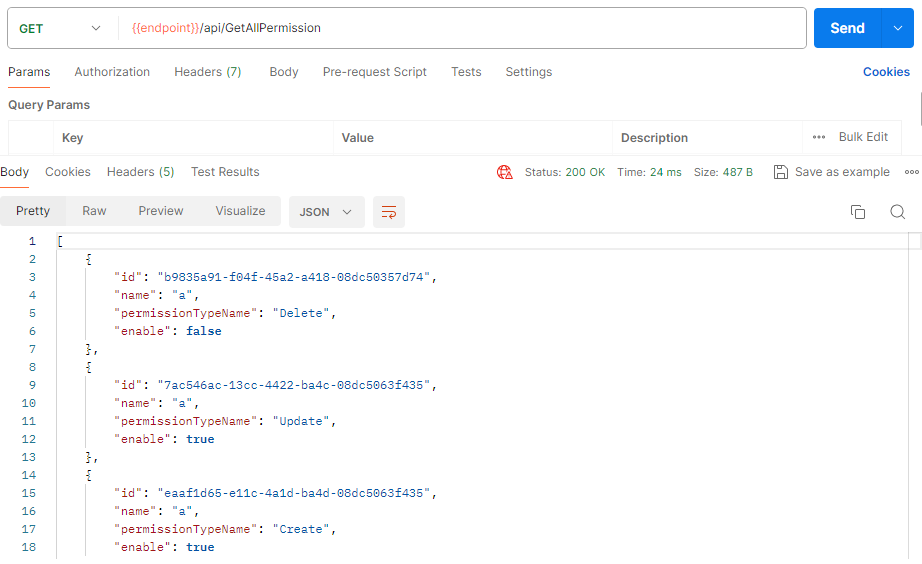
}'





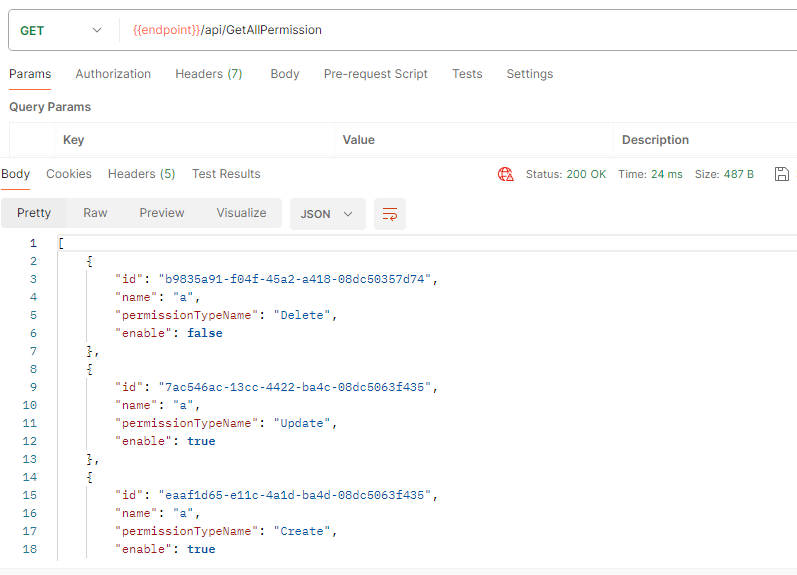
/opt/kafka/bin/kafka-console-consumer.sh --topic modify-events --from-beginning --bootstrap-server localhost:9092





Get All from SQL

curl --location 'https://localhost:44367/api/GetAllPermission'



Get By Email From Elastic

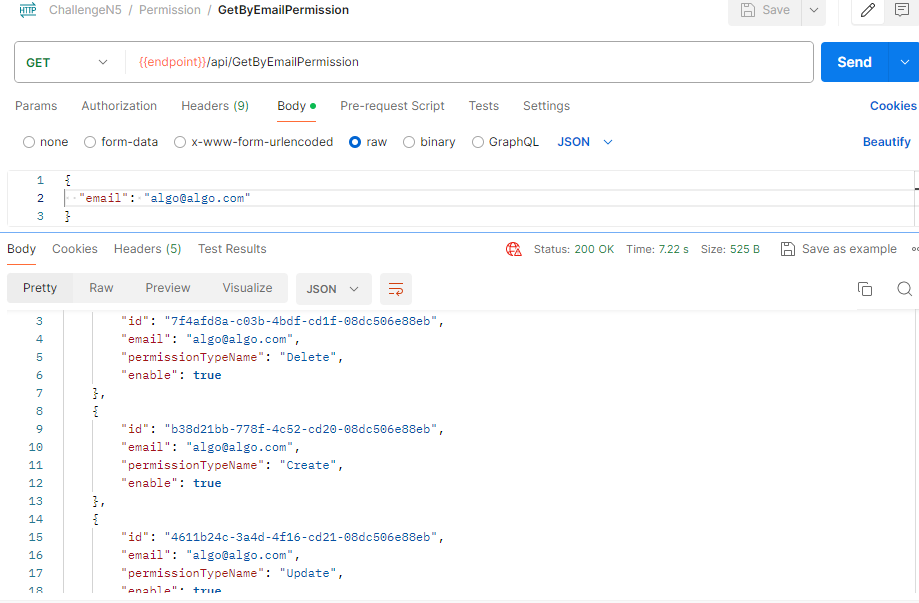
curl --location --request GET 'https://localhost:44367/api/GetByEmailPermission' \

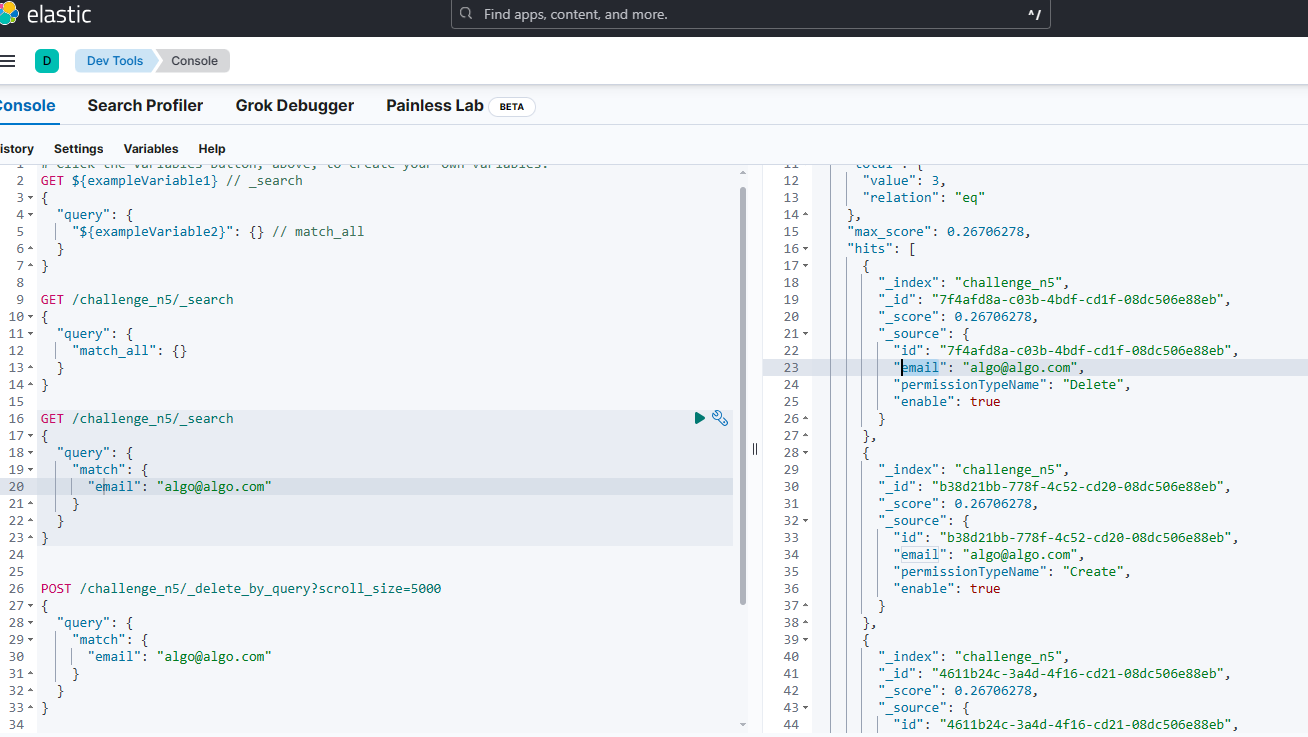
--header 'Content-Type: application/json' \

--data-raw '{

  "email": "algo@algo.com"

}'





-uml

@startuml

package "Enterprise Business Rules" {

[ChallengeAPI.BusinessObjects]

interface "IRepository"

}

package "Application Business Rules" {

[ChallengeAPI.DTOs]

[ChallengeAPI.UseCases]

}

package "Interface Adapters" {

[ChallengeAPI.Controllers] -> [ChallengeAPI.Presenters]

[ChallengeAPI.Presenters]

package "Gateways" {

[ChallengeAPI.Elasticsearch]

[ChallengeAPI.IoC]

[ChallengeAPI.Kafka]

[ChallengeAPI.RepositoryEFCore]

}

}

"IRepository"--> [ChallengeAPI.RepositoryEFCore]

[ChallengeAPI.RepositoryEFCore] -->[ChallengeAPI.BusinessObjects]

[ChallengeAPI.RepositoryEFCore] -->"MSSQL"

[ChallengeAPI.Kafka] -->"Kafka"

[ChallengeAPI.Elasticsearch]-->"Elastic"

[ChallengeAPI.UseCases] --> [ChallengeAPI.Kafka]

[ChallengeAPI.UseCases] --> [ChallengeAPI.Elasticsearch]

[ChallengeAPI.UseCases] --> [ChallengeAPI.DTOs]

[ChallengeAPI.UseCases] --> [ChallengeAPI.BusinessObjects]

[ChallengeAPI.Presenters]-->[ChallengeAPI.UseCases]

[ChallengeAPI.Controllers] -->[ChallengeAPI.UseCases]

[ChallengeAPI.WebApi] -->[ChallengeAPI.Controllers]

[ChallengeAPI.WebApi] -->[ChallengeAPI.IoC]

cloud {

[ChallengeAPI.WebApi]

}

database "MSSQL" {

frame "Employees" {

}

frame "PermissionTypes" {

}

frame "Permissions" {

}

}

database "Elastic" {

frame "PermissionsIndex" { }}

database "Kafka" {

frame "request\_events" { }

frame "modify\_events" { }

frame "get\_events" { }

}

[ChallengeAPI.WebApi]<-- User

@enduml

Reference links:

<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>

[How to Produce and Consume Kafka Messages in .NET 6 | by Vinod Pal | Medium](https://medium.com/@vndpal/how-to-produce-and-consume-kafka-messages-in-net-6-47d52307283f)

[Install Elasticsearch with Docker | Elasticsearch Guide [8.12] | Elastic](https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html)

[CRUD usage examples | Elasticsearch .NET Client [8.9] | Elastic](https://www.elastic.co/guide/en/elasticsearch/client/net-api/current/examples.html)