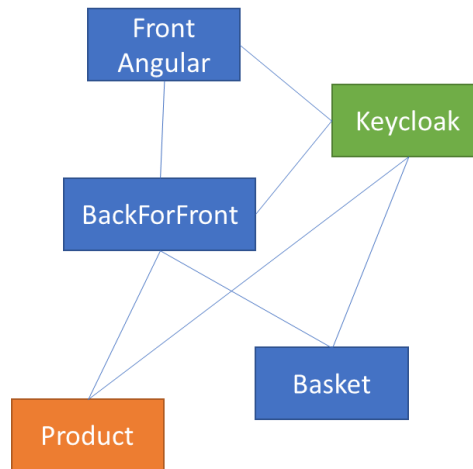


Bonjour jeune développeur et bienvenue chez MiageCorp pour participer à notre super projet.

Notre application est composée de microservices avec l'architecture suivante :



Tu fais désormais partie de la Team Product et ta mission est de créer ce microservice, le conteneuriser et le sécuriser pour l'authentification Keycloak.

A ta disposition tu trouveras :

- Les sources des autres briques dont tu peux t'inspirer
- Un docker-compose à exécuter en amont qui va t'embarquer toutes les briques dont tu as besoin pour développer ton composant
- Le CM de la dernière fois (même si cela ne te servira à rien 😊)

Mission #1 : Création de l'API

Votre 1^{ère} mission est de créer une nouvelle application, qui va permettre d'héberger vos produit.

Les contraintes :

- L'API devra respecter la norme REST sur les gestion des produits
- Le Controller devra faire appel à une couche Service
- Il faut utiliser le système d'injection de dépendance
- Votre couche service utilise la DAL suivante :

```

private static List<Product> _products = new List<Product>()
{
    new Product() { Id = Guid.NewGuid().ToString(), Label = "Produit 1", Description =
"Lorem ipsum dolor sit amet, consectetur adipiscing elit", Price = 20d },
    new Product() { Id = Guid.NewGuid().ToString(), Label = "Produit 2", Description =
"Lorem ipsum dolor sit amet, consectetur adipiscing elit", Price = 12d },
    new Product() { Id = Guid.NewGuid().ToString(), Label = "Produit 3", Description =
"Lorem ipsum dolor sit amet, consectetur adipiscing elit", Price = 99.99d },
    new Product() { Id = Guid.NewGuid().ToString(), Label = "Produit 4", Description =
"Lorem ipsum dolor sit amet, consectetur adipiscing elit", Price = 5d },
    new Product() { Id = Guid.NewGuid().ToString(), Label = "Produit 5", Description =
"Lorem ipsum dolor sit amet, consectetur adipiscing elit", Price = 55.55d },
};
  
```

Mission #2 : Conteneurisation de l'application

Deux objectifs :

- Builder l'image du conteneur avec le tag **product :1.0**
- Ajout de votre image au fichier docker-compose et modifications pour que le backForFront le vois

Mission #3 : Sécurisation de l'application

Ajouter la sécurisation oAuth2 par Keycloak sur votre API. Pensez à reconteneuriser votre application en version 1.1 et modifier votre docker-compose.

Code à ajouter pour gérer le JWT dans Program:

```

        services.AddAuthentication(options =>
        {
            options.DefaultAuthenticateScheme =
JwtBearerDefaults.AuthenticationScheme;
            options.DefaultChallengeScheme =
JwtBearerDefaults.AuthenticationScheme;
        }).AddJwtBearer(o =>
        {
            o.Authority = Configuration["Keycloak:Authority"];
            o.Audience = Configuration["Keycloak:Audience"];
            o.RequireHttpsMetadata = !Environment.IsDevelopment();

            o.Events = new JwtBearerEvents()
            {
                OnAuthenticationFailed = c =>
                {
                    c.NoResult();

                    c.Response.StatusCode = 500;
                    c.Response.ContentType = "text/plain";
                    if (Environment.IsDevelopment())
                    {
                        return
c.Response.WriteAsync(c.Exception.ToString());
                    }
                    return c.Response.WriteAsync("An error occurred
processing your authentication.");
                }
            };
        });

...

app.UseAuthentication();
app.UseAuthorization();

```