

PathTalesBack

1.0.0

Généré le Mardi 11 Novembre 2025 10:00:05 pour PathTalesBack par Doxygen 1.9.8

Mardi 11 Novembre 2025 10:00:05

1 BlogApi - Documentation technique complète

1.1 Présentation du projet

BlogApi est une API REST développée en **ASP.NET Core 8.0** pour la gestion complète d'articles de blog. Ce projet implémente les bonnes pratiques de développement d'API modernes avec une architecture en couches claire et maintenable.

1.1.1 Informations générales

Propriété	Valeur
Nom	BlogApi
Version	1.0.0
Framework	.NET 8.0 (ASP.NET Core)
Base de données	MariaDB 10.11
ORM	Entity Framework Core 9.0
Auteur	[Votre nom - À personnaliser]
Date	[Date - À personnaliser]

1.2 Objectifs du projet

Ce projet a été développé dans le cadre du cours **APROG (Programmation Avancée)** et démontre les compétences suivantes :

1. **Architecture en couches** : Séparation claire des responsabilités (Controllers, Models, DTOs, Repositories, Persistence)
2. **Patterns de conception** : Repository Pattern, DTO Pattern
3. **Entity Framework Core** : ORM pour l'accès aux données avec migrations
4. **API RESTful** : Respect des conventions REST (GET, POST, PUT, DELETE)
5. **Validation des données** : Data Annotations pour garantir l'intégrité
6. **Documentation API** : Swagger/OpenAPI pour la documentation interactive
7. **Conteneurisation** : Docker Compose pour MariaDB et phpMyAdmin

1.3 Architecture

Le projet suit une architecture en couches qui sépare les préoccupations :

Controllers (API Layer)	← Points d'entrée HTTP
- PostsController	
- VersionController	
DTOs Layer	← Objets de transfert
- PostDto	
- PostCreateDto	
- PostUpdateDto	
Repository Layer	← Logique d'accès aux données
- IPostRepository (Interface)	
- PostRepository (Implémentation)	
Persistence Layer	← Configuration EF Core
- BlogDbContext	
- Migrations	
Models Layer	← Entités du domaine
- Post	

1.3.1 Diagrammes UML

Pour une meilleure compréhension de l'architecture, voici les diagrammes techniques :

1. Diagramme de classes

FIGURE 1 Diagramme de classes complet

Ce diagramme montre toutes les classes de l'application et leurs relations.

2. Diagramme d'architecture (Composants)

FIGURE 2 Architecture du système

Ce diagramme illustre l'architecture globale du système avec Docker Compose.

3. Diagramme de paquetages

FIGURE 3 Organisation en paquetages

Ce diagramme présente l'organisation en couches et les dépendances entre namespaces.

1.4 Fonctionnalités principales

1.4.1 Opérations CRUD complètes

L'API expose les endpoints suivants :

Méthode	Endpoint	Description
GET	/api/posts	Liste tous les articles
GET	/api/posts/{id}	Récupère un article spécifique
POST	/api/posts	Crée un nouvel article
PUT	/api/posts/{id}	Met à jour un article existant
DELETE	/api/posts/{id}	Supprime un article
GET	/health	Health check de l'API

1.4.2 Diagrammes de cas d'utilisation

Cas d'utilisation principaux

FIGURE 4 Diagramme de cas d'utilisation

Ce diagramme montre les interactions entre les utilisateurs et le système.

1.4.3 Flux de création d'un article

Diagramme de séquence - POST /api/posts

FIGURE 5 Flux de création d'un article

Ce diagramme détaille le flux complet de création d'un article, de la requête HTTP jusqu'à la persistance en base.

1.4.4 Flux de mise à jour d'un article

Diagramme de communication - PUT /api/posts/{id}

FIGURE 6 Interactions lors d'une mise à jour

Ce diagramme montre les interactions entre objets lors de la mise à jour d'un article.

1.4.5 Validation automatique

Toutes les données entrantes sont validées automatiquement avec **Data Annotations** :

- **Titre** : 3-150 caractères, requis
- **Contenu** : minimum 3 caractères, requis
- **CreatedAt** : généré automatiquement (UTC)

1.4.6 Support CORS

L'API est configurée pour accepter les requêtes du frontend Blazor (<http://localhost:5170>).

1.4.7 Migrations automatiques

Les migrations Entity Framework Core sont appliquées automatiquement au démarrage de l'application.

1.5 Modèle de données

1.5.1 Entité Post

L'entité principale **Post** représente un article de blog :

Propriété	Type	Description
Id	int	Clé primaire (auto-incrémentée)
Title	string	Titre de l'article
Content	string	Contenu complet de l'article
CreatedAt	DateTime	Date et heure de création (UTC)

Contraintes de la base de données :

- **Id** : PRIMARY KEY, AUTO_INCREMENT
- **Title** : NOT NULL, LONGTEXT
- **Content** : NOT NULL, LONGTEXT
- **CreatedAt** : NOT NULL, DATETIME(6)

1.6 Technologies utilisées

1.6.1 Backend

- **ASP.NET Core 8.0** : Framework web moderne et performant
- **Entity Framework Core 9.0** : ORM pour .NET
- **Pomelo.EntityFrameworkCore.MySql 9.0** : Provider MySQL/MariaDB

1.6.2 Base de données

- **MariaDB 10.11** : Système de gestion de base de données relationnel
- **phpMyAdmin** : Interface web pour la gestion de MariaDB

1.6.3 Documentation

- **Swashbuckle (Swagger)** : Génération de documentation OpenAPI
- **Doxygen** : Génération de documentation technique avec code source

1.6.4 Outils de développement

- **Docker Compose** : Orchestration des conteneurs
 - **Visual Studio 2022 / VS Code** : Environnements de développement
-

1.7 Configuration

1.7.1 Chaîne de connexion

La connexion à la base de données est configurée dans `appsettings.json` :

```
{  "ConnectionStrings": {    "Default": "Server=localhost;Port=3306;Database=blogdb;User=root;Password=example"  }}
```

1.7.2 CORS

Le CORS est configuré pour accepter les requêtes depuis :

- <http://localhost:5170> (Frontend Blazor)

1.7.3 Swagger

L'interface Swagger est accessible en mode développement à :

- <https://localhost:5001/swagger> (ou le port configuré)
-

1.8 Structure du code source

Ce document contient le code source complet de tous les fichiers du projet, organisé par namespace :

1. **BlogApi.Controllers** : Contrôleurs API
2. **BlogApi.Models** : Entités du domaine
3. **BlogApi.DTOs** : Objets de transfert de données
4. **BlogApi.Repositories** : Interfaces et implémentations des repositories
5. **BlogApi.Persistence** : Configuration Entity Framework Core

Chaque classe inclut :

- Le code source complet avec coloration syntaxique
 - Les commentaires XML pour la documentation
 - Les dépendances et références
-

1.9 Contexte pédagogique

Ce projet a été réalisé dans le cadre des ateliers progressifs du cours APROG :

- **Ateliers 1-4** : Développement initial avec SQLite
 - **Atelier 5** : Migration vers MariaDB avec Docker
 - **Ateliers suivants** : Fonctionnalités avancées
-

1.10 Contact et support

Pour toute question concernant ce projet :

- **Enseignant** : [Nom de l'enseignant]
- **Email** : [Email]
- **Année** : 2025-2026

Note

Cette documentation a été générée automatiquement avec **Doxygen** pour le TPI (Travail Pratique Individuel).

Auteur

[Votre nom - À personnaliser]

Date

[Date - À personnaliser]

Version

1.0.0

2 Index des espaces de nommage

2.1 Liste des paquetages

Liste des paquetages avec une brève description (si disponible) :

PathTalesBack	??
PathTalesBack.Controllers	??
PathTalesBack.Data	??
PathTalesBack.Entities	??

3 Index hiérarchique

3.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

ControllerBase	??
PathTalesBack.Controllers.UserController	??
PathTalesBack.Entities.Customer	??
PathTalesBack.Data.MongoDbService	??
PathTalesBack.Entities.User	??

4 Index des classes

4.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

ControllerBase	??
PathTalesBack.Entities.Customer	??

PathTalesBack.Data.MongoDbService	??
PathTalesBack.Entities.User	??
PathTalesBack.Controllers.UserController	??

5 Index des fichiers

5.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

PathTalesBack/PathTalesBack/Program.cs	??
PathTalesBack/PathTalesBack/Controllers/UserController.cs	??
PathTalesBack/PathTalesBack/Data/MongoDbService.cs	??
PathTalesBack/PathTalesBack/Entities/Customer.cs	??
PathTalesBack/PathTalesBack/Entities/User.cs	??

6 Documentation des espaces de nommage

6.1 Référence de l'espace de nommage PathTalesBack

Espaces de nommage

- namespace [Controllers](#)
- namespace [Data](#)
- namespace [Entities](#)

6.2 Référence de l'espace de nommage PathTalesBack.Controllers

Classes

- class [UserController](#)

6.3 Référence de l'espace de nommage PathTalesBack.Data

Classes

- class [MongoDbService](#)

6.4 Référence de l'espace de nommage PathTalesBack.Entities

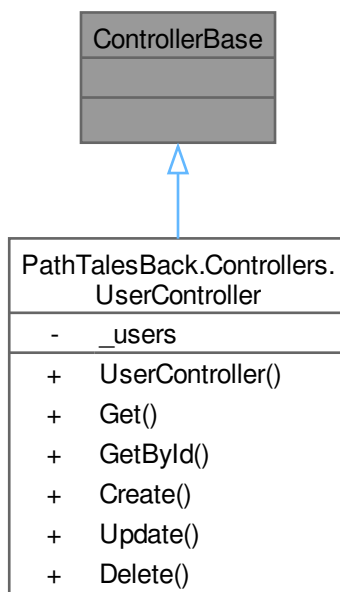
Classes

- class [Customer](#)
- class [User](#)

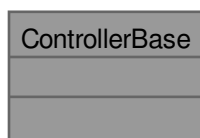
7 Documentation des classes

7.1 Référence de la classe ControllerBase

Graphe d'héritage de ControllerBase:



Graphe de collaboration de ControllerBase:



La documentation de cette classe a été générée à partir du fichier suivant :

— [PathTalesBack/PathTalesBack/Controllers/UserController.cs](#)

7.2 Référence de la classe PathTalesBack.Entities.Customer

Graphe de collaboration de PathTalesBack.Entities.Customer:

PathTalesBack.Entities.Customer	
+	Id
+	UserName
+	Email

Propriétés

- string? `Id` [get, set]
- string? `UserName` [get, set]
- string? `Email` [get, set]

7.2.1 Description détaillée

Définition à la ligne 6 du fichier [Customer.cs](#).

7.2.2 Documentation des propriétés

Email

```
string? PathTalesBack.Entities.Customer.Email [get], [set]
```

Définition à la ligne 15 du fichier [Customer.cs](#).

Id

```
string? PathTalesBack.Entities.Customer.Id [get], [set]
```

Définition à la ligne 10 du fichier [Customer.cs](#).

UserName

```
string? PathTalesBack.Entities.Customer.UserName [get], [set]
```

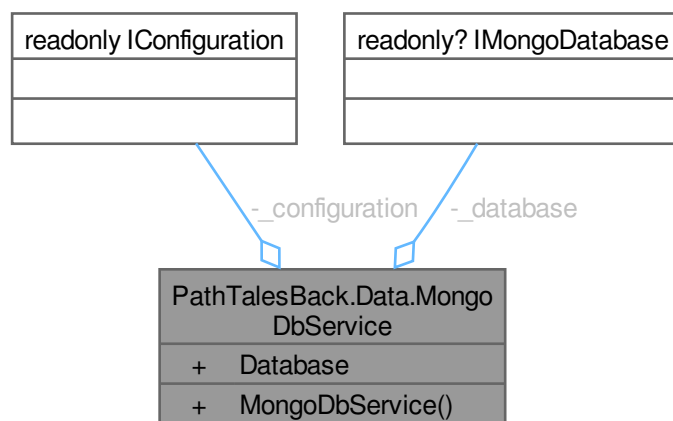
Définition à la ligne 13 du fichier [Customer.cs](#).

La documentation de cette classe a été générée à partir du fichier suivant :

- [PathTalesBack/PathTalesBack/Entities/Customer.cs](#)

7.3 Référence de la classe PathTalesBack.Data.MongoDbService

Grphe de collaboration de PathTalesBack.Data.MongoDbService:



Fonctions membres publiques

- [MongoDbService](#) ([IConfiguration](#) configuration)

Propriétés

- [IMongoDatabase? Database](#) [get]

Attributs privés

- [readonly IConfiguration _configuration](#)
- [readonly? IMongoDatabase _database](#)

7.3.1 Description détaillée

Définition à la ligne 5 du fichier [MongoDbService.cs](#).

7.3.2 Documentation des constructeurs et destructeur

MongoDbService()

```
PathTalesBack.Data.MongoDbService.MongoDbService (
    IConfiguration configuration )
```

Définition à la ligne 11 du fichier [MongoDbService.cs](#).

Références [PathTalesBack.Data.MongoDbService._configuration](#), et [PathTalesBack.Data.MongoDbService._database](#).

7.3.3 Documentation des données membres

_configuration

```
readonly IConfiguration PathTalesBack.Data.MongoDbService._configuration [private]
```

Définition à la ligne 7 du fichier [MongoDbService.cs](#).

Référencé par [PathTalesBack.Data.MongoDbService.MongoDbService\(\)](#).

`_database`

`readonly? IMongoDatabase PathTalesBack.Data.MongoDbService._database [private]`

Définition à la ligne 8 du fichier [MongoDbService.cs](#).

Référencé par [PathTalesBack.Data.MongoDbService.MongoDbService\(\)](#).

7.3.4 Documentation des propriétés

Database

`IMongoDatabase? PathTalesBack.Data.MongoDbService.Database [get]`

Définition à la ligne 21 du fichier [MongoDbService.cs](#).

Référencé par [PathTalesBack.Controllers.UserController.UserController\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

— [PathTalesBack/PathTalesBack/Data/MongoDbService.cs](#)

7.4 Référence de la classe PathTalesBack.Entities.User

Graphe de collaboration de `PathTalesBack.Entities.User`:

PathTalesBack.Entities.User	
+	Id
+	Name
+	Email
+	Token
+	Role
+	IsBlocked
+	BlockReason

Propriétés

- `string? Id` [get, set]
- `string? Name` [get, set]
- `string? Email` [get, set]
- `string? Token = null` [get, set]
- `string? Role` [get, set]
- `Boolean IsBlocked` [get, set]
- `string? BlockReason` [get, set]

7.4.1 Description détaillée

Définition à la ligne 6 du fichier [User.cs](#).

7.4.2 Documentation des propriétés

BlockReason

`string? PathTalesBack.Entities.User.BlockReason [get], [set]`

Définition à la ligne 34 du fichier [User.cs](#).

Email

```
string? PathTalesBack.Entities.User.Email [get], [set]
```

Définition à la ligne 18 du fichier [User.cs](#).

Id

```
string? PathTalesBack.Entities.User.Id [get], [set]
```

Définition à la ligne 10 du fichier [User.cs](#).

Référencé par [PathTalesBack.Controllers.UserController.Update\(\)](#).

IsBlocked

```
Boolean PathTalesBack.Entities.User.IsBlocked [get], [set]
```

Définition à la ligne 30 du fichier [User.cs](#).

Name

```
string? PathTalesBack.Entities.User.Name [get], [set]
```

Définition à la ligne 14 du fichier [User.cs](#).

Role

```
string? PathTalesBack.Entities.User.Role [get], [set]
```

Définition à la ligne 26 du fichier [User.cs](#).

Token

```
string? PathTalesBack.Entities.User.Token = null [get], [set]
```

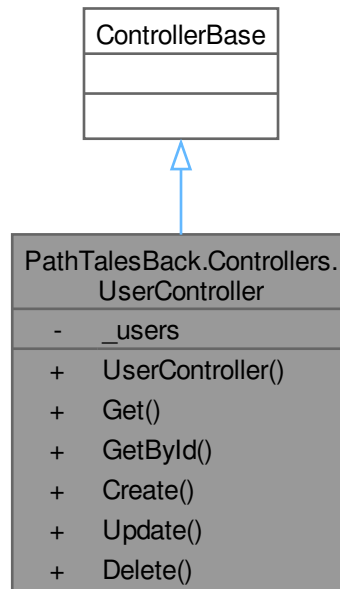
Définition à la ligne 22 du fichier [User.cs](#).

La documentation de cette classe a été générée à partir du fichier suivant :

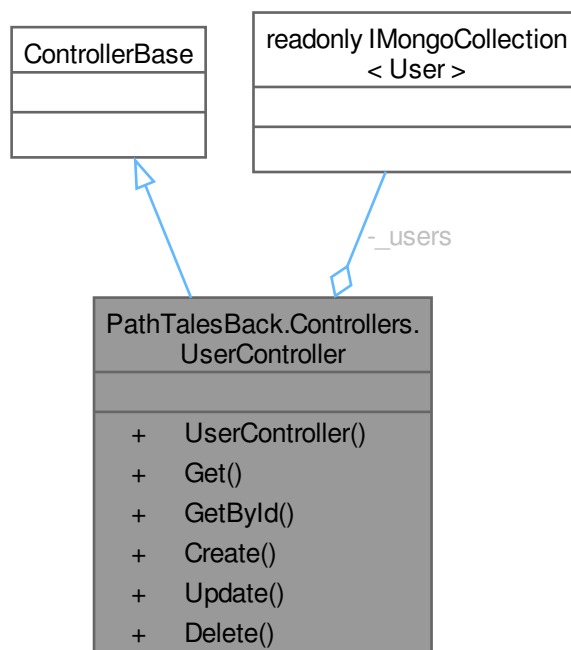
— [PathTalesBack/PathTalesBack/Entities/User.cs](#)

7.5 Référence de la classe PathTalesBack.Controllers.UserController

Graphe d'héritage de PathTalesBack.Controllers.UserController:



Graphe de collaboration de PathTalesBack.Controllers.UserController:



Fonctions membres publiques

- `UserController` (`MongoDbService` mongoDbService)
- `async Task< IEnumerable< User > > Get ()`
- `async Task< ActionResult< User?> > GetById` (string id)
- `async Task< ActionResult > Create` (`User` user)
- `async Task< ActionResult > Update` (`User` user)
- `async Task< ActionResult > Delete` (string id)

Attributs privés

- `readonly IMongoCollection< User > _users`

7.5.1 Description détaillée

Définition à la ligne 13 du fichier `UserController.cs`.

7.5.2 Documentation des constructeurs et destructeur

`UserController()`

```
PathTalesBack.Controllers.UserController.UserController (
    MongoDbService mongoDbService )
```

Définition à la ligne 17 du fichier `UserController.cs`.

Références `PathTalesBack.Controllers.UserController._users`, et `PathTalesBack.Data.MongoDbService.Database`.

7.5.3 Documentation des fonctions membres

Create()

```
async Task< ActionResult > PathTalesBack.Controllers.UserController.Create (
    User user )
```

Définition à la ligne 41 du fichier [UserController.cs](#).

Références [PathTalesBack.Controllers.UserController._users](#), et [PathTalesBack.Controllers.UserController.GetById\(\)](#).

Voici le graphe d'appel pour cette fonction :



Delete()

```
async Task< ActionResult > PathTalesBack.Controllers.UserController.Delete (
    string id )
```

Définition à la ligne 57 du fichier [UserController.cs](#).

Références [PathTalesBack.Controllers.UserController._users](#).

Get()

```
async Task< IEnumerable< User > > PathTalesBack.Controllers.UserController.Get ( )
```

Renvoie

Définition à la ligne 27 du fichier [UserController.cs](#).

Références [PathTalesBack.Controllers.UserController._users](#).

GetById()

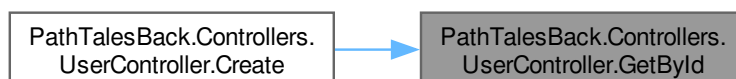
```
async Task< ActionResult< User? > > PathTalesBack.Controllers.UserController.GetById (
    string id )
```

Définition à la ligne 33 du fichier [UserController.cs](#).

Références [PathTalesBack.Controllers.UserController._users](#).

Référencé par [PathTalesBack.Controllers.UserController.Create\(\)](#).

Voici le graphe des appelants de cette fonction :



Update()

```
async Task< ActionResult > PathTalesBack.Controllers.UserController.Update (
    User user )
```

Définition à la ligne 48 du fichier [UserController.cs](#).

Références [PathTalesBack.Controllers.UserController._users](#), et [PathTalesBack.Entities.User.Id](#).

7.5.4 Documentation des données membres

_users

```
readonly IMongoCollection<User> PathTalesBack.Controllers.UserController._users [private]
```

Définition à la ligne 15 du fichier [UserController.cs](#).

Référencé par [PathTalesBack.Controllers.UserController.Create\(\)](#), [PathTalesBack.Controllers.UserController.Delete\(\)](#), [PathTalesBack.Controllers.UserController.Get\(\)](#), [PathTalesBack.Controllers.UserController.GetById\(\)](#), [PathTalesBack.Controllers.UserController.GetUser\(\)](#), et [PathTalesBack.Controllers.UserController.Update\(\)](#).

La documentation de cette classe a été générée à partir du fichier suivant :

— [PathTalesBack/PathTalesBack/Controllers/UserController.cs](#)

8 Documentation des fichiers

8.1 Référence du fichier docs/doxygen/MAINPAGE.md

8.2 Référence du fichier PathTalesBack/PathTalesBack/Controllers/UserController.cs

Classes

— class [PathTalesBack.Controllers.UserController](#)

Espaces de nommage

— namespace [PathTalesBack](#)
— namespace [PathTalesBack.Controllers](#)

8.3 UserController.cs

[Aller à la documentation de ce fichier.](#)

```
00001 using Microsoft.AspNetCore.Http;
00002 using Microsoft.AspNetCore.Http.HttpResults;
00003 using Microsoft.AspNetCore.Mvc;
00004 using MongoDB.Driver;
00005 using PathTalesBack;
00006 using PathTalesBack.Data;
00007 using PathTalesBack.Entities;
00008
00009 namespace PathTalesBack.Controllers
00010 {
00011     [Route("api/[controller]")]
00012     [ApiController]
00013     public class UserController : ControllerBase
00014     {
00015         private readonly IMongoCollection<User> _users;
00016
00017         public UserController(MongoDbService mongoDbService)
00018         {
00019             _users = mongoDbService.Database?.GetCollection<User>("test");
00020         }
00021
00022         /// <summary>
00023         ///
00024         /// </summary>
00025         /// <returns></returns>
00026         [HttpGet]
00027         public async Task<IEnumerable<User>> Get()
00028         {
00029             return await _users.Find(FilterDefinition<User>.Empty).ToListAsync();
00030         }
00031
00032         [HttpGet("{id}")]
00033         public async Task<ActionResult<User?>> GetById(string id)
00034         {
```



```

00035         var filter = Builders<User>.Filter.Eq(x => x.Id, id);
00036         var user = _users.Find(filter).FirstOrDefault();
00037         return user is not null ? Ok(user) : NotFound();
00038     }
00039
00040     [HttpPost]
00041     public async Task<ActionResult> Create(User user)
00042     {
00043         await _users.InsertOneAsync(user);
00044         return CreatedAtAction(nameof(GetById), new { id = user.Id }, user);
00045     }
00046
00047     [HttpPatch]
00048     public async Task<ActionResult> Update(User user)
00049     {
00050         var filter = Builders<User>.Filter.Eq(x => x.Id, user.Id);
00051
00052         await _users.ReplaceOneAsync(filter, user);
00053         return Ok();
00054     }
00055
00056     [HttpDelete("{id}")]
00057     public async Task<ActionResult> Delete(string id)
00058     {
00059         var filter = Builders<User>.Filter.Eq(x => x.Id, id);
00060         await _users.DeleteOneAsync(filter);
00061         return Ok();
00062     }
00063 }
00064 }

```

8.4 Référence du fichier PathTalesBack/PathTalesBack/Data/MongoDbService.cs

Classes

— class [PathTalesBack.Data.MongoDbService](#)

Espaces de nommage

— namespace [PathTalesBack](#)
 — namespace [PathTalesBack.Data](#)

8.5 MongoDbService.cs

[Aller à la documentation de ce fichier.](#)

```

00001 using MongoDB.Driver;
00002
00003 namespace PathTalesBack.Data
00004 {
00005     public class MongoDbService
00006     {
00007         private readonly IConfiguration _configuration;
00008         private readonly IMongoDatabase? _database;
00009
00010
00011         public MongoDbService(IConfiguration configuration)
00012         {
00013             _configuration = configuration;
00014
00015             var connectionString = _configuration.GetConnectionString("DbConnection");
00016             var mongoUrl = MongoUrl.Create(connectionString);
00017             var mongoClient = new MongoClient(mongoUrl);
00018             _database = mongoClient.GetDatabase(mongoUrl.DatabaseName);
00019         }
00020
00021         public IMongoDatabase? Database => _database;
00022     }
00023 }
00024 }

```

8.6 Référence du fichier PathTalesBack/PathTalesBack/Entities/Customer.cs

Classes

— class [PathTalesBack.Entities.Customer](#)

Espaces de nommage

- namespace [PathTalesBack](#)
- namespace [PathTalesBack.Entities](#)

8.7 Customer.cs

[Aller à la documentation de ce fichier.](#)

```
00001 using MongoDB.Bson.Serialization.Attributes;
00002 using MongoDB.Bson;
00003
00004 namespace PathTalesBack.Entities
00005 {
00006     public class Customer
00007     {
00008         [BsonId]
00009         [BsonElement("_id"), BsonRepresentation(BsonType.ObjectId)]
00010         public string? Id { get; set; }
00011
00012         [BsonElement("customer_name"), BsonRepresentation(BsonType.String)]
00013         public string? UserName { get; set; }
00014         [BsonElement("email"), BsonRepresentation(BsonType.String)]
00015         public string? Email { get; set; }
00016     }
00017 }
00018 }
```

8.8 Référence du fichier PathTalesBack/PathTalesBack/Entities/User.cs**Classes**

- class [PathTalesBack.Entities.User](#)

Espaces de nommage

- namespace [PathTalesBack](#)
- namespace [PathTalesBack.Entities](#)

8.9 User.cs

[Aller à la documentation de ce fichier.](#)

```
00001 using MongoDB.Bson.Serialization.Attributes;
00002 using MongoDB.Bson;
00003
00004 namespace PathTalesBack.Entities
00005 {
00006     public class User
00007     {
00008         [BsonId]
00009         [BsonElement("_id"), BsonRepresentation(BsonType.ObjectId)]
00010         public string? Id { get; set; }
00011
00012         // /\ dans la base -> int
00013         [BsonElement("name"), BsonRepresentation(BsonType.String)]
00014         public string? Name { get; set; }
00015
00016         //password
00017         [BsonElement("password"), BsonRepresentation(BsonType.String)]
00018         public string? Email { get; set; }
00019
00020         //token
00021         [BsonElement("token"), BsonRepresentation(BsonType.String)]
00022         public string? Token { get; set; } = null;
00023
00024         //role
00025         [BsonElement("role"), BsonRepresentation(BsonType.String)]
00026         public string? Role { get; set; }
00027
00028         //isBlocked
00029         [BsonElement("isBlocked"), BsonRepresentation(BsonType.Boolean)]
00030         public Boolean IsBlocked { get; set; }
00031
00032         //blockReason
00033         [BsonElement("blockReason"), BsonRepresentation(BsonType.String)]
00034         public string? BlockReason { get; set; }
00035     }
00036 }
```

8.10 Référence du fichier PathTalesBack/PathTalesBack/Program.cs

Fonctions

- builder.Services. [AddControllers](#) ()
- builder.Services. [AddEndpointsApiExplorer](#) ()
- builder.Services. [AddSwaggerGen](#) ()
- builder.Services. [AddSingleton< MongoDBService >](#) ()
- if (app.Environment.IsDevelopment())
- app. [UseHttpsRedirection](#) ()
- app. [UseAuthorization](#) ()
- app. [MapControllers](#) ()
- app. [Run](#) ()

Variables

- var [builder](#) = WebApplication.CreateBuilder(args)
- var [app](#) = builder.Build()

8.10.1 Documentation des fonctions

AddControllers()

```
builder.Services. AddControllers ( )
```

AddEndpointsApiExplorer()

```
builder.Services. AddEndpointsApiExplorer ( )
```

AddSingleton< MongoDBService >()

```
builder.Services. AddSingleton< MongoDbService > ( )
```

AddSwaggerGen()

```
builder.Services. AddSwaggerGen ( )
```

if()

```
if (
    app::Environment::IsDevelopment() )
```

Définition à la ligne [16](#) du fichier [Program.cs](#).

Références [app](#).

MapControllers()

```
app. MapControllers ( )
```

Run()

```
app. Run ( )
```

UseAuthorization()

```
app. UseAuthorization ( )
```

UseHttpsRedirection()

```
app. UseHttpsRedirection ( )
```

8.10.2 Documentation des variables

app

`var app = builder.Build()`

Définition à la ligne 13 du fichier [Program.cs](#).

Référencé par `if()`.

builder

`var builder = WebApplication.CreateBuilder(args)`

Définition à la ligne 3 du fichier [Program.cs](#).

8.11 Program.cs

[Aller à la documentation de ce fichier.](#)

```
00001 using PathTalesBack.Data;
00002
00003 var builder = WebApplication.CreateBuilder(args);
00004
00005 // Add services to the container.
00006
00007 builder.Services.AddControllers();
00008 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
00009 builder.Services.AddEndpointsApiExplorer();
00010 builder.Services.AddSwaggerGen();
00011 builder.Services.AddSingleton<MongoDbService>();
00012
00013 var app = builder.Build();
00014
00015 // Configure the HTTP request pipeline.
00016 if (app.Environment.IsDevelopment())
00017 {
00018     app.UseSwagger();
00019     app.UseSwaggerUI();
00020 }
00021
00022 app.UseHttpsRedirection();
00023
00024 app.UseAuthorization();
00025
00026 app.MapControllers();
00027
00028 app.Run();
```

