

# Namespace APIFouBassan

## Classes

### [Program](#)

La classe du programme, le début de tout

# Class Program

Namespace: [APIFouBassan](#)

Assembly: APIFouBassan.dll

La classe du programme, le début de tout

```
public class Program
```

## Inheritance

[object](#)  ← Program

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Methods

### Main(string[])

Méthode qui va appeler le code entier, c'est par ici que le programme de l'API commence

```
public static void Main(string[] args)
```

## Parameters

args [string](#)  []

Les arguments d'entrées pour le Main

# Namespace APIFouBassan.Controllers

## Classes

### [FouBassanController](#)

C'est le controller de mon API, il fait les interactions concernant les fou de bassan

# Class FouBassanController

Namespace: [APIFouBassan.Controllers](#)

Assembly: APIFouBassan.dll

C'est le controller de mon API, il fait les interactions concernant les fou de bassan

[ApiController]

```
public class FouBassanController : Controller, IActionFilter, IAsyncActionFilter,
IFilterMetadata, IDisposable
```

## Inheritance

[object](#) ← [ControllerBase](#) ← [Controller](#) ← FouBassanController

## Implements

[IActionFilter](#), [IAsyncActionFilter](#), [IFilterMetadata](#), [IDisposable](#)

## Inherited Members

[Controller.View\(\)](#), [Controller.View\(string\)](#), [Controller.View\(object\)](#),  
[Controller.View\(string, object\)](#), [Controller.PartialView\(\)](#), [Controller.PartialView\(string\)](#),  
[Controller.PartialView\(object\)](#), [Controller.PartialView\(string, object\)](#),  
[Controller.ViewComponent\(string\)](#), [Controller.ViewComponent\(Type\)](#),  
[Controller.ViewComponent\(string, object\)](#), [Controller.ViewComponent\(Type, object\)](#),  
[Controller.Json\(object\)](#), [Controller.Json\(object, object\)](#),  
[Controller.OnActionExecuting\(ActionExecutingContext\)](#),  
[Controller.OnActionExecuted\(ActionExecutedContext\)](#),  
[Controller.OnActionExecutionAsync\(ActionExecutingContext, ActionExecutionDelegate\)](#),  
[Controller.Dispose\(\)](#), [Controller.Dispose\(bool\)](#), [Controller.ViewData](#), [Controller.TempData](#),  
[Controller.ViewBag](#), [ControllerBase.StatusCode\(int\)](#), [ControllerBase.StatusCode\(int, object\)](#),  
[ControllerBase.Content\(string\)](#), [ControllerBase.Content\(string, string\)](#),  
[ControllerBase.Content\(string, string, Encoding\)](#),  
[ControllerBase.Content\(string, MediaTypeHeaderValue\)](#), [ControllerBase.NoContent\(\)](#),  
[ControllerBase.Ok\(\)](#), [ControllerBase.Ok\(object\)](#), [ControllerBase.Redirect\(string\)](#),  
[ControllerBase.RedirectPermanent\(string\)](#), [ControllerBase.RedirectPreserveMethod\(string\)](#),  
[ControllerBase.RedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.LocalRedirect\(string\)](#),  
[ControllerBase.LocalRedirectPermanent\(string\)](#), [ControllerBase.LocalRedirectPreserveMethod\(string\)](#),  
[ControllerBase.LocalRedirectPermanentPreserveMethod\(string\)](#), [ControllerBase.RedirectToAction\(\)](#),  
[ControllerBase.RedirectToAction\(string\)](#), [ControllerBase.RedirectToAction\(string, object\)](#),  
[ControllerBase.RedirectToAction\(string, string\)](#),  
[ControllerBase.RedirectToAction\(string, string, object\)](#),

[ControllerBase.RedirectToAction\(string, string, string\)](#),  
[ControllerBase.RedirectToAction\(string, string, object, string\)](#),  
[ControllerBase.RedirectToActionPreserveMethod\(string, string, object, string\)](#),  
[ControllerBase.RedirectToActionPermanent\(string\)](#),  
[ControllerBase.RedirectToActionPermanent\(string, object\)](#),  
[ControllerBase.RedirectToActionPermanent\(string, string\)](#),  
[ControllerBase.RedirectToActionPermanent\(string, string, string\)](#),  
[ControllerBase.RedirectToActionPermanent\(string, string, object\)](#),  
[ControllerBase.RedirectToActionPermanent\(string, string, object, string\)](#),  
[ControllerBase.RedirectToRoute\(string\)](#), [ControllerBase.RedirectToRoute\(object\)](#),  
[ControllerBase.RedirectToRoute\(string, object\)](#), [ControllerBase.RedirectToRoute\(string, string\)](#),  
[ControllerBase.RedirectToRoute\(string, object, string\)](#),  
[ControllerBase.RedirectToRoutePreserveMethod\(string, object, string\)](#),  
[ControllerBase.RedirectToRoutePermanent\(string\)](#),  
[ControllerBase.RedirectToRoutePermanent\(object\)](#),  
[ControllerBase.RedirectToRoutePermanent\(string, object\)](#),  
[ControllerBase.RedirectToRoutePermanent\(string, string\)](#),  
[ControllerBase.RedirectToRoutePermanent\(string, object, string\)](#),  
[ControllerBase.RedirectToRoutePermanentPreserveMethod\(string, object, string\)](#),  
[ControllerBase.RedirectToPage\(string\)](#), [ControllerBase.RedirectToPage\(string, object\)](#),  
[ControllerBase.RedirectToPage\(string, string\)](#), [ControllerBase.RedirectToPage\(string, string, object\)](#),  
[ControllerBase.RedirectToPage\(string, string, string\)](#),  
[ControllerBase.RedirectToPage\(string, string, object, string\)](#),  
[ControllerBase.RedirectToPagePermanent\(string\)](#),  
[ControllerBase.RedirectToPagePermanent\(string, object\)](#),  
[ControllerBase.RedirectToPagePermanent\(string, string\)](#),  
[ControllerBase.RedirectToPagePermanent\(string, string, string\)](#),  
[ControllerBase.RedirectToPagePermanent\(string, string, object, string\)](#),  
[ControllerBase.RedirectToPagePreserveMethod\(string, string, object, string\)](#),  
[ControllerBase.RedirectToPagePermanentPreserveMethod\(string, string, object, string\)](#),  
[ControllerBase.File\(byte\[\], string\)](#), [ControllerBase.File\(byte\[\], string, bool\)](#),  
[ControllerBase.File\(byte\[\], string, string\)](#), [ControllerBase.File\(byte\[\], string, string, bool\)](#),  
[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.File\(byte\[\], string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.File\(byte\[\], string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.File\(Stream, string\)](#), [ControllerBase.File\(Stream, string, bool\)](#),  
[ControllerBase.File\(Stream, string, string\)](#), [ControllerBase.File\(Stream, string, string, bool\)](#),  
[ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue\)](#),

[ControllerBase.File\(Stream, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.File\(Stream, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.File\(string, string\)](#), [ControllerBase.File\(string, string, bool\)](#),  
[ControllerBase.File\(string, string, string\)](#), [ControllerBase.File\(string, string, string, bool\)](#),  
[ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.File\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.File\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.PhysicalFile\(string, string\)](#), [ControllerBase.PhysicalFile\(string, string, bool\)](#),  
[ControllerBase.PhysicalFile\(string, string, string\)](#),  
[ControllerBase.PhysicalFile\(string, string, string, bool\)](#),  
[ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.PhysicalFile\(string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue\)](#),  
[ControllerBase.PhysicalFile\(string, string, string, DateTimeOffset?, EntityTagHeaderValue, bool\)](#),  
[ControllerBase.Unauthorized\(\)](#), [ControllerBase.Unauthorized\(object\)](#), [ControllerBase.NotFound\(\)](#),  
[ControllerBase.NotFound\(object\)](#), [ControllerBase.BadRequest\(\)](#),  
[ControllerBase.BadRequest\(object\)](#), [ControllerBase.BadRequest\(ModelStateDictionary\)](#),  
[ControllerBase.UnprocessableEntity\(\)](#), [ControllerBase.UnprocessableEntity\(object\)](#),  
[ControllerBase.UnprocessableEntity\(ModelStateDictionary\)](#), [ControllerBase.Conflict\(\)](#),  
[ControllerBase.Conflict\(object\)](#), [ControllerBase.Conflict\(ModelStateDictionary\)](#),  
[ControllerBase.Problem\(string, string, int?, string, string\)](#),  
[ControllerBase.ValidationProblem\(ValidationProblemDetails\)](#),  
[ControllerBase.ValidationProblem\(ModelStateDictionary\)](#), [ControllerBase.ValidationProblem\(\)](#),  
[ControllerBase.ValidationProblem\(string, string, int?, string, string, ModelStateDictionary\)](#),  
[ControllerBase.Created\(string, object\)](#), [ControllerBase.Created\(Uri, object\)](#),  
[ControllerBase.CreatedAtAction\(string, object\)](#),  
[ControllerBase.CreatedAtAction\(string, object, object\)](#),  
[ControllerBase.CreatedAtAction\(string, string, object, object\)](#),  
[ControllerBase.CreatedAtRoute\(string, object\)](#), [ControllerBase.CreatedAtRoute\(object, object\)](#),  
[ControllerBase.CreatedAtRoute\(string, object, object\)](#), [ControllerBase.Accepted\(\)](#),  
[ControllerBase.Accepted\(object\)](#), [ControllerBase.Accepted\(Uri\)](#), [ControllerBase.Accepted\(string\)](#),  
[ControllerBase.Accepted\(string, object\)](#), [ControllerBase.Accepted\(Uri, object\)](#),  
[ControllerBase.AcceptedAtAction\(string\)](#), [ControllerBase.AcceptedAtAction\(string, string\)](#),  
[ControllerBase.AcceptedAtAction\(string, object\)](#),  
[ControllerBase.AcceptedAtAction\(string, string, object\)](#),  
[ControllerBase.AcceptedAtAction\(string, object, object\)](#),  
[ControllerBase.AcceptedAtAction\(string, string, object, object\)](#),  
[ControllerBase.AcceptedAtRoute\(object\)](#), [ControllerBase.AcceptedAtRoute\(string\)](#),

[ControllerBase.AcceptedAtRoute\(string, object\).☐](#) , [ControllerBase.AcceptedAtRoute\(object, object\).☐](#) ,  
[ControllerBase.AcceptedAtRoute\(string, object, object\).☐](#) , [ControllerBase.Challenge\(\).☐](#) ,  
[ControllerBase.Challenge\(params string\[\]\).☐](#) , [ControllerBase.Challenge\(AuthenticationProperties\).☐](#) ,  
[ControllerBase.Challenge\(AuthenticationProperties, params string\[\]\).☐](#) , [ControllerBase.Forbid\(\).☐](#) ,  
[ControllerBase.Forbid\(params string\[\]\).☐](#) , [ControllerBase.Forbid\(AuthenticationProperties\).☐](#) ,  
[ControllerBase.Forbid\(AuthenticationProperties, params string\[\]\).☐](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal\).☐](#) , [ControllerBase.SignIn\(ClaimsPrincipal, string\).☐](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties\).☐](#) ,  
[ControllerBase.SignIn\(ClaimsPrincipal, AuthenticationProperties, string\).☐](#) , [ControllerBase.SignOut\(\).☐](#) ,  
[ControllerBase.SignOut\(AuthenticationProperties\).☐](#) , [ControllerBase.SignOut\(params string\[\]\).☐](#) ,  
[ControllerBase.SignOut\(AuthenticationProperties, params string\[\]\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, params Expression<Func<TModel, object>>\[\]\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, Func<ModelMetadata, bool>\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, params Expression<Func<TModel, object>>\[\]\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync<TModel>\(TModel, string, IValueProvider, Func<ModelMetadata, bool>\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync\(object, Type, string\).☐](#) ,  
[ControllerBase.TryUpdateModelAsync\(object, Type, string, IValueProvider, Func<ModelMetadata, bool>\).☐](#) ,  
[☐](#) ,  
[ControllerBase.TryValidateModel\(object\).☐](#) , [ControllerBase.TryValidateModel\(object, string\).☐](#) ,  
[ControllerBase.HttpContext☐](#) , [ControllerBase.Request☐](#) , [ControllerBase.Response☐](#) ,  
[ControllerBase.RouteData☐](#) , [ControllerBase.ModelState☐](#) , [ControllerBase.ControllerContext☐](#) ,  
[ControllerBase.MetadataProvider☐](#) , [ControllerBase.ModelBinderFactory☐](#) , [ControllerBase.Url☐](#) ,  
[ControllerBase.ObjectValidator☐](#) , [ControllerBase.ProblemDetailsFactory☐](#) , [ControllerBase.User☐](#) ,  
[ControllerBase.Empty☐](#) , [object.Equals\(object\).☐](#) , [object.Equals\(object, object\).☐](#) ,  
[object.GetHashCode\(\).☐](#) , [object.GetType\(\).☐](#) , [object.MemberwiseClone\(\).☐](#) ,  
[object.ReferenceEquals\(object, object\).☐](#) , [object.ToString\(\).☐](#)

## Constructors

### FouBassanController(EspDbContext)

Constructeur du fou de bassan pour faire la liaison vers la base de données

```
public FouBassanController(EspDbContext bd)
```

## Parameters

bd [EspDbContext](#)

# Methods

## Delete(string)

Méthode pour enlever une donnée sur la base de données. Il va prendre le tag de l'oiseau et enlever toutes les données concernant cet oiseau

```
[HttpDelete("{tag}")]
[ProducesResponseType(202, Type = typeof(AcceptedResult))]
[ProducesResponseType(400, Type = typeof(BadRequestResult))]
public IActionResult Delete(string tag)
```

## Parameters

tag [string](#)

## Returns

[IActionResult](#)

## Get(string, DateTime, DateTime)

Méthode qui sert à aller chercher les données voulu pour faire l'affichage

```
[HttpGet("{tag}")]
[ProducesResponseType(202, Type = typeof(FouBassanDTO))]
[ProducesResponseType(400, Type = typeof(BadRequestResult))]
public IActionResult Get(string tag, DateTime dateDebut, DateTime dateFin)
```

## Parameters



tag [string](#)

Prends le tag de l'oiseau qu'on veut récupérer

dateDebut [DateTime](#)

Prends la date et heure de début du rescencement, maximum de 5h

dateFin [DateTime](#)

Prend la date et l'heure de fin du rescencement, maximum de 5h

## Returns

[IActionResult](#)

Retourne un objet FouBassanDTO, qui possède une liste de position et une liste de profondeurs entre les temps demandés

## Post(PositionTempsDTO)

Méthode pour publier les données de position sur la base de donnée, il va créer un nouveau fou si le tag n'existe pas, et il va donner sa valeur de position sur le temps, ainsi que sa valeur de température.

```
[HttpPost]
[Route("ajouterPosition")]
[ProducesResponseType(201, Type = typeof(PositionTemps))]
[ProducesResponseType(400, Type = typeof(BadRequestResult))]
[ProducesResponseType(409, Type = typeof(ConflictResult))]
public IActionResult Post(PositionTempsDTO position)
```

## Parameters

position [PositionTempsDTO](#)

Prends un objet PositionTempsDTO pour directement l'envoyer dans la base de données

## Returns

[IActionResult](#)

Retourne un succès s'il réussi à installer les données sur la base de données, sinon il va indiquer s'il y a un conflit, ou s'il n'a pas réussi à mettre la donnée

## Post(ProfondeurTempsDTO)

C'est une méthode qui permet d'ajouter une donnée de profondeur sur la base de donnée. Il va créer un nouveau fou si le tag inscrit n'existe pas, et il va lui donner sa valeur de profondeur sur le temps, ainsi que son comportement.

```
[HttpPost]
[Route("ajouterProfondeur")]
[ProducesResponseType(201, Type = typeof(ProfondeurTemps))]
[ProducesResponseType(400, Type = typeof(BadRequestResult))]
[ProducesResponseType(409, Type = typeof(ConflictResult))]
public IActionResult Post(ProfondeurTempsDTO profondeur)
```

## Parameters

**profondeur** [ProfondeurTempsDTO](#)

Prends un objet ProfondeurTempsDTO pour que ce soit plus simple entrer les données

## Returns

[IActionResult](#)<sup>↗</sup>

Retourne un succès s'il réussi à installer les données sur la base de données, sinon il va indiquer s'il y a un conflit, ou s'il n'a tout simplement pas réussi à mettre les données

# Namespace APIFouBassan.Models

## Classes

### [EspDbContext](#)

C'est la classe qui s'occupe de faire le liens entre l'API et la base de donnée

### [FouBassan](#)

C'est la classe qui fait les liens vers les autres informations du fou de bassan, la classe mère

### [FouBassanDTO](#)

Ici c'est un Data Transfer Object (DTO) de la classe FouBassan, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet, c'est cette classe qui va être utilisé pour l'affichage des données

### [PositionTemps](#)

C'est la classe qui prends les données nécessaire pour la position, ainsi que le comportement de l'oiseau sur le moment

### [PositionTempsDTO](#)

Ici c'est un Data Transfer Object (DTO) de la classe PositionTemps, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet

### [ProfondeurTemps](#)

C'est la classe qui sert à enregistrer les données de profondeur de l'oiseau

### [ProfondeurTempsDTO](#)

Ici c'est un Data Transfer Object (DTO) de la classe profondeur, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet

# Class EspDbContext

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll

C'est la classe qui s'occupe de faire le liens entre l'API et la base de donnée

```
public class EspDbContext : DbContext, IInfrastructure<IServiceProvider>,
    IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,
    IDisposable, IAsyncDisposable
```

## Inheritance

[object](#) ← [DbContext](#) ← EspDbContext

## Implements

[IInfrastructure](#) <[IServiceProvider](#)>, [IDbContextDependencies](#), [IDbSetCache](#), [IDbContextPoolable](#), [IResettableService](#), [IDisposable](#), [IAsyncDisposable](#)

## Inherited Members

[DbContext.Set<TEntity>\(\)](#), [DbContext.Set<TEntity>\(string\)](#),  
[DbContext.ConfigureConventions\(ModelConfigurationBuilder\)](#), [DbContext.SaveChanges\(\)](#),  
[DbContext.SaveChanges\(bool\)](#), [DbContext.SaveChangesAsync\(Cancellation\\_token\)](#),  
[DbContext.SaveChangesAsync\(bool, Cancellation\\_token\)](#), [DbContext.Dispose\(\)](#),  
[DbContext.DisposeAsync\(\)](#), [DbContext.Entry<TEntity>\(TEntity\)](#), [DbContext.Entry\(object\)](#),  
[DbContext.Add<TEntity>\(TEntity\)](#), [DbContext.AddAsync<TEntity>\(TEntity, Cancellation\\_token\)](#),  
[DbContext.Attach<TEntity>\(TEntity\)](#), [DbContext.Update<TEntity>\(TEntity\)](#),  
[DbContext.Remove<TEntity>\(TEntity\)](#), [DbContext.Add\(object\)](#),  
[DbContext.AddAsync\(object, Cancellation\\_token\)](#), [DbContext.Attach\(object\)](#),  
[DbContext.Update\(object\)](#), [DbContext.Remove\(object\)](#), [DbContext.AddRange\(params object\[\]\)](#),  
[DbContext.AddRangeAsync\(params object\[\]\)](#), [DbContext.AttachRange\(params object\[\]\)](#),  
[DbContext.UpdateRange\(params object\[\]\)](#), [DbContext.RemoveRange\(params object\[\]\)](#),  
[DbContext.AddRange\(IEnumerable<object>\)](#),  
[DbContext.AddRangeAsync\(IEnumerable<object>, Cancellation\\_token\)](#),  
[DbContext.AttachRange\(IEnumerable<object>\)](#), [DbContext.UpdateRange\(IEnumerable<object>\)](#),  
[DbContext.RemoveRange\(IEnumerable<object>\)](#), [DbContext.Find\(Type, params object\[\]\)](#),  
[DbContext.FindAsync\(Type, params object\[\]\)](#),  
[DbContext.FindAsync\(Type, object\[\], Cancellation\\_token\)](#), [DbContext.Find<TEntity>\(params object\[\]\)](#),  
[DbContext.FindAsync<TEntity>\(params object\[\]\)](#),  
[DbContext.FindAsync<TEntity>\(object\[\], Cancellation\\_token\)](#),  
[DbContext.FromExpression<TResult>\(Expression<Func<IQueryable<TResult>>>\)](#),

[DbContext.Database](#) , [DbContext.ChangeTracker](#) , [DbContext.Model](#) , [DbContext.ContextId](#) , [DbContext.SavingChanges](#) , [DbContext.SavedChanges](#) , [DbContext.SaveChangesFailed](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### EspDbContext()

Constructeur vide pour créer un nouveau DbContext avec les valeurs par défaut

```
public EspDbContext()
```

### EspDbContext(DbContextOptions<EspDbContext>)

Constructeur qui prend des options en paramètres, pour plus de configuration

```
public EspDbContext(DbContextOptions<EspDbContext> options)
```

#### Parameters

`options` [DbContextOptions](#) <[EspDbContext](#)>

### EspDbContext(string)

Constructeur avec la chaîne de connexion en entrée, pour vouloir changer la chaîne de connexion si on change de base de données

```
public EspDbContext(string connectionString)
```

#### Parameters

`connectionString` [string](#)

## Properties

# FouBassan

Liste des fous de bassan

```
public virtual DbSet<FouBassan> FouBassan { get; set; }
```

Property Value

[DbSet](#) <[FouBassan](#)>

# PositionTemps

Liste des positions sur le temps

```
public virtual DbSet<PositionTemps> PositionTemps { get; set; }
```

Property Value

[DbSet](#) <[PositionTemps](#)>

# ProfondeurTemps

Liste des profondeurs sur le temps

```
public virtual DbSet<ProfondeurTemps> ProfondeurTemps { get; set; }
```

Property Value

[DbSet](#) <[ProfondeurTemps](#)>

# Methods

## OnConfiguring(DbContextOptionsBuilder)

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

Parameters

optionsBuilder [DbContextOptionsBuilder](#)↗

## OnModelCreating(ModelBuilder)

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

modelBuilder [ModelBuilder](#)↗

# Class FouBassan

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll

C'est la classe qui fait les liens vers les autres informations du fou de bassan, la classe mère

```
public class FouBassan
```

## Inheritance

[object](#)  ← FouBassan

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

# Constructors

## FouBassan()

Constructeur vide pour créer un nouvel objet avec les valeurs par défaut

```
public FouBassan()
```

## FouBassan(string)

Constructeur de fou de bassan en lui donnant un tag précis.

```
public FouBassan(string tag)
```

## Parameters

tag [string](#) 

# Properties



# Id

L'id du fou de bassan, ça sert pour faire les liens entre les objets, il est auto-incrémental

```
public int Id { get; set; }
```

## Property Value

[int](#)

# PositionTemps

Un pointeur vers sa liste de positions

```
public virtual ICollection<PositionTemps> PositionTemps { get; set; }
```

## Property Value

[ICollection](#) <[PositionTemps](#)>

# ProfondeurTemps

Un pointeur vers sa liste de profondeurs

```
public virtual ICollection<ProfondeurTemps> ProfondeurTemps { get; set; }
```

## Property Value

[ICollection](#) <[ProfondeurTemps](#)>

# Tag

C'est l'ID de l'oiseau, est en format texte

```
public string? Tag { get; set; }
```

Property Value

[string](#) 

# Class FouBassanDTO

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll








Ici c'est un Data Transfer Object (DTO) de la classe FouBassan, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet, c'est cette classe qui va être utilisé pour l'affichage des données

```
public class FouBassanDTO
```

## Inheritance

[object](#)  ← FouBassanDTO

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

FouBassanDTO(List<PositionTempsDTO>,  
List<ProfondeurTempsDTO>)

Constructeur du DTO, il prend la liste des positions et la liste des profondeurs pour se créer

```
public FouBassanDTO(List<PositionTempsDTO> positionTempsDTOs, List<ProfondeurTempsDTO>  
profondeurTempsDTOs)
```

## Parameters

positionTempsDTOs [List](#)  <[PositionTempsDTO](#)>

profondeurTempsDTOs [List](#)  <[ProfondeurTempsDTO](#)>

## Properties

# PositionTemps

C'est une liste qui contient plusieurs PositionTemps sur son format de DTO, ça sert pour l'affichage des données de position

```
public List<PositionTempsDTO> PositionTemps { get; set; }
```

## Property Value

[List](#) <[PositionTempsDTO](#)>

# ProfondeurTemps

C'est une liste qui contient plusieurs ProfondeurTemps sur son format de DTO, ça sert pour l'affichage des données de profondeur

```
public List<ProfondeurTempsDTO> ProfondeurTemps { get; set; }
```

## Property Value

[List](#) <[ProfondeurTempsDTO](#)>

# Class PositionTemps

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll








C'est la classe qui prends les données nécessaire pour la position, ainsi que le comportement de l'oiseau sur le moment

```
public class PositionTemps
```

## Inheritance

[object](#)  ← PositionTemps

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

### PositionTemps()

Constructeur vide qui sert pour instancier un nouvel objet avec les valeurs par défaut

```
public PositionTemps()
```

### PositionTemps(PositionTempsDTO, FouBassan)

Constructeur qui prends le DTO en entrée et le fou de bassan pour créer

```
public PositionTemps(PositionTempsDTO dto, FouBassan nav)
```

## Parameters

dto [PositionTempsDTO](#)

nav [FouBassan](#)

# Properties

## Comportement

Le comportement de l'oiseau, s'il est à la colonie, en voyage, à la pêche ou en dérive, peut être null

```
public string? Comportement { get; set; }
```

### Property Value

[string](#)↗

## Date

Paramètre sur le format AAAA-MM-JJTHH:mm:ss.sssZ, où T sépare le jour du temps et Z pour Zero timezone

```
public DateTime Date { get; set; }
```

### Property Value

[DateTime](#)↗

## Id

Id de l'objet, est auto-incrémental donc il s'occupe lui même de sa valeur. Sert pour aller chercher des données et comparer des valeurs.

```
public int Id { get; set; }
```

### Property Value

[int](#)↗

## IdFouBassan

C'est l'ID qui pointe vers le fou de bassan

```
public int IdFouBassan { get; set; }
```

Property Value

[int](#)

## IdFouBassanNavigation

C'est l'objet qui fait la navigation vers le fou de bassan de la donnée

```
public virtual FouBassan? IdFouBassanNavigation { get; set; }
```

Property Value

[FouBassan](#)

## Latitude

Donnée de latitude, peut être null

```
public double? Latitude { get; set; }
```

Property Value

[double](#)

## Longitude

Donnée de longitude, peut être null

```
public double? Longitude { get; set; }
```

Property Value

[double](#)?

## Tag

C'est l'ID de l'oiseau, est en format texte

```
public string Tag { get; set; }
```

## Property Value

[string](#)



# Class PositionTempsDTO

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll








Ici c'est un Data Transfer Object (DTO) de la classe PositionTemps, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet

```
public class PositionTempsDTO
```

## Inheritance

[object](#)  ← PositionTempsDTO

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

### PositionTempsDTO(DateTime, double?, double?, string?, string)

Le constructeur du DTO, il prend les paramètres en entrée nécessaire pour construire l'objet

```
public PositionTempsDTO(DateTime date, double? latitude, double? longitude, string? comportement, string tag)
```

## Parameters

date [DateTime](#) 

latitude [double](#) ?

longitude [double](#) ?

comportement [string](#) 

tag [string](#) 

# Properties

## Comportement

Le comportement de l'oiseau, s'il est à la colonie, en voyage, à la pêche ou en dérive, peut être null

```
public String? Comportement { get; set; }
```

## Property Value

[String](#)

## Date

Paramètre sur le format AAAA-MM-JJTHH:mm:ss.sssZ, où T sépare le jour du temps et Z pour Zero timezone

```
public DateTime Date { get; set; }
```

## Property Value

[DateTime](#)

## Latitude

Donnée de latitude, peut être null

```
public double? Latitude { get; set; }
```

## Property Value

[double](#)

## Longitude

Donnée de longitude, peut être null

```
public double? Longitude { get; set; }
```

Property Value

[double](#)?

## Tag

C'est l'ID de l'oiseau, est en format texte

```
public string Tag { get; set; }
```

Property Value

[string](#)

# Class ProfondeurTemps


Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll

C'est la classe qui sert à enregistrer les données de profondeur de l'oiseau

```
public class ProfondeurTemps
```

## Inheritance

[object](#)  ← ProfondeurTemps

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

### ProfondeurTemps()

Constructeur vide, pour instancier une nouvelle classe avec les données par défaut

```
public ProfondeurTemps()
```

### ProfondeurTemps(ProfondeurTempsDTO, FouBassan)

Constructeur avec l'objet DTO en paramètre et avec le fou de bassan de navigation

```
public ProfondeurTemps(ProfondeurTempsDTO dto, FouBassan nav)
```

## Parameters

dto [ProfondeurTempsDTO](#)

nav [FouBassan](#)

# Properties

## Date

Paramètre sur le format AAAA-MM-JJTHH:mm:ss.sssZ, où T sépare le jour du temps et Z pour Zero timezone

```
public DateTime Date { get; set; }
```

## Property Value

[DateTime](#)

## Id

Id de l'objet, est auto-incrémental donc il s'occupe lui même de sa valeur. Sert pour aller chercher des données et comparer des valeurs.

```
public int Id { get; set; }
```

## Property Value

[int](#)

## IdFouBassan

C'est l'ID qui pointe vers le fou de bassan

```
public int IdFouBassan { get; set; }
```

## Property Value

[int](#)

## IdFouBassanNavigation

C'est l'objet qui fait la navigation vers le fou de bassan de la donnée

```
public virtual FouBassan IdFouBassanNavigation { get; set; }
```

Property Value

[FouBassan](#)

## Profondeur

Donnée de profondeur, peut être null. Est en mètres

```
public double? Profondeur { get; set; }
```

Property Value

[double](#)?

## Tag

C'est l'ID de l'oiseau, est en format texte

```
public string Tag { get; set; }
```

Property Value

[string](#)

## Temperature

Donnée de température de la position actuelle. Est en degré Celcius et peut être null

```
public double? Temperature { get; set; }
```

Property Value

[double](#) ↗?

# Class ProfondeurTempsDTO

Namespace: [APIFouBassan.Models](#)

Assembly: APIFouBassan.dll

Ici c'est un Data Transfer Object (DTO) de la classe profondeur, elle sert à initialiser une classe qui va servir seulement pour le déplacement de l'objet

```
public class ProfondeurTempsDTO
```

## Inheritance

[object](#) ← ProfondeurTempsDTO

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Constructors

### ProfondeurTempsDTO(DateTime, double?, double?, string)

Le constructeur du DTO, elle va être appelé lorsqu'on veut construire un nouvel objet ProfondeurTempsDTO

```
public ProfondeurTempsDTO(DateTime date, double? profondeur, double? temperature, string tag)
```

## Parameters

date [DateTime](#)

profondeur [double](#)?

temperature [double](#)?

tag [string](#)



# Properties

## Date

Paramètre sur le format AAAA-MM-JJTHH:mm:ss.sssZ, où T sépare le jour du temps et Z pour Zero timezone

```
public DateTime Date { get; set; }
```

## Property Value

[DateTime](#)↗

## Profondeur

Donnée de profondeur, peut être null. Est en mètres

```
public double? Profondeur { get; set; }
```

## Property Value

[double](#)↗?

## Tag

C'est l'ID de l'oiseau, est en format texte

```
public string Tag { get; set; }
```

## Property Value

[string](#)↗

## Temperature

Donnée de température de la position actuelle. Est en degré Celcius et peut être null

```
public double? Temperature { get; set; }
```

Property Value

[double](#)<sup>?</sup>?