

Vue → <form action=""

Par défaut

Home Ctrl

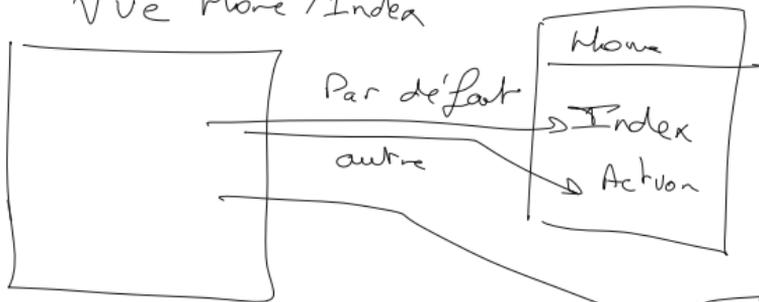
Home/ Index

Index

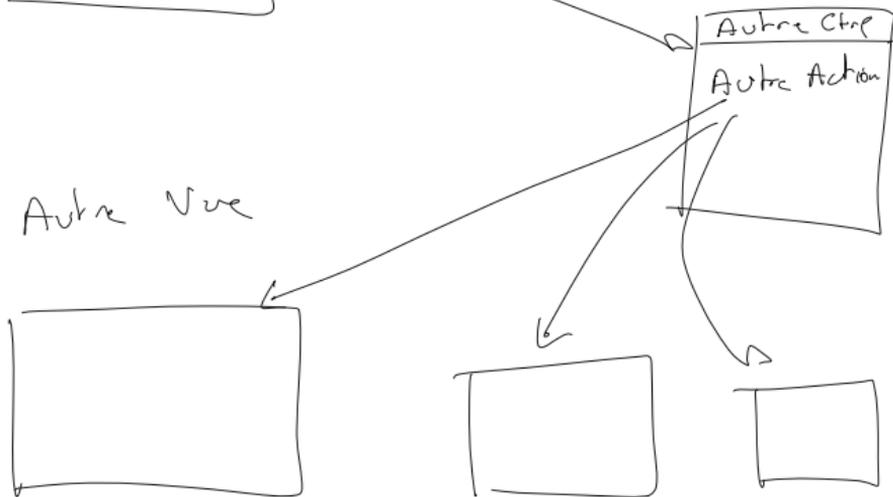
Documentation de référence :

<https://docs.microsoft.com/fr-fr/aspnet/>

Vue Home / Index



Autre Vue



Exercice de mise en oeuvre d'une application ASP.Net MVC Core

Objectif : constituer une application qui mette en oeuvre un maximum d'éléments prévus pour le passage de la certification
Il pourrait s'agir d'une application qui serve de modèle.

- Authentification
- ActionFilters
- WebAPI
- Custom attributes
- Vues et vues partielles
- Extensions de helpers
- Mise en cache

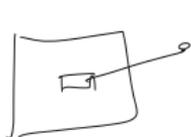
Objectif : Développez un carnet d'adresse

Etapas

- constituez une table "Adresses" dans un fichier de base de données .mdf local à votre application (utilisez l'éditeur de fichier sql server fourni avec visual studio)
- Développez une couche ORM (Entity Framework) vous permettant d'obtenir un contexte EF pour votre application
- Développez une couche API fournissant les opérations crud sur cette entité.
 - La couche API devra fournir de l'authentification :
 - les anonymes pourront consulter la liste des fiches, mais uniquement le nom et prénom des personnes , pas le détail
 - les membres du groupe READER pourront lire le détail d'une fiche (adresse, email, téléphone)
 - les membres du groupe WRITER pourront modifier, créer et supprimer une fiche.
- Développez un

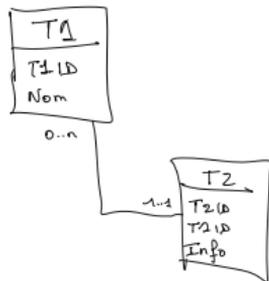
Pattern Object Relational Mapping

Form



"Insert into" ...

BDD:



↓ ORN (à la main, bcp de boulot)

↳ pour peu de Tables

↓ Utiliser des générateurs (Borland Together)

obsoleto

Sybase Power Designer

↓ Utiliser un Framework

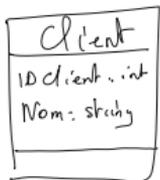
EF en .net / Projets d'envergure

Pattern ORN

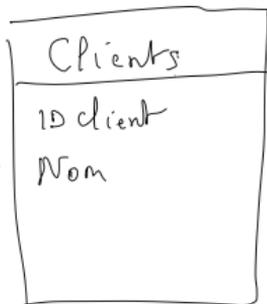
DAO (Data Access Objects)



DTO (Data Transfer Objects)



BDD:



Front:

Client c = new
client();
c.Nom = "toto";

clientsManager.
create(c);

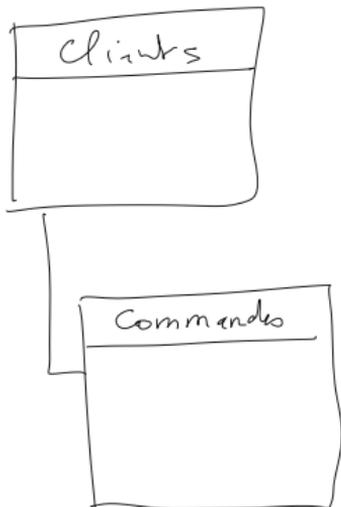
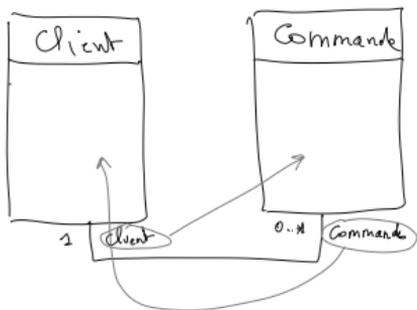
```
public static Create (client c)
{
```

```
    SqlCommand cmd = .... ("Insert into Clients Values (@id, @nom)
    cmd.Parameters.Add("nom", value)
    cmd.ExecuteNonQuery();
```

Paramètres

}

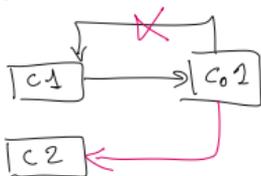
° RN:



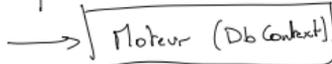
```
Client c1 = new Client();
Client c2 = new Client();
co1 = new Commande();
c1.Commandes.Add(co1);
```

```
c1.Commandes[0].Client = c1
```

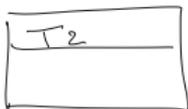
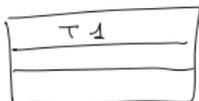
```
co1.Client = c2;
```



Requêtes



DBO



Voir le pattern UnitOfWork !

Design pattern Unit of work !! :

<https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application>

