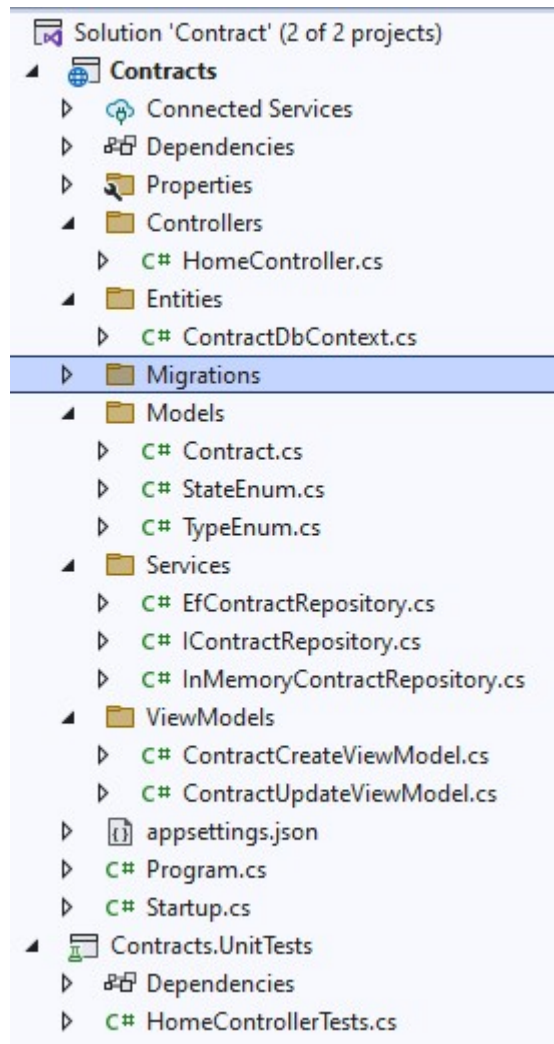


Oefening Contracts

We bouwen een API voor het beheren van contracten. Bouw in Visual Studio 2022 een solution met onderstaande structuur. (Het unittests project is optioneel)



De Models folder bevat onderstaande klasse en enums:

```
public class Contract
{
    3 references | 0 exceptions
    public int Id { get; set; }
    1 reference | 0 exceptions
    public string Code { get; set; }
    2 references | 0 exceptions
    public string Description { get; set; }
    1 reference | 0 exceptions
    public TypeEnum Type { get; set; }
    2 references | 0 exceptions
    public StateEnum State { get; set; }
}

public enum TypeEnum
{
    Managed,
    Trading
}

public enum StateEnum
{
    Preparing,
    Active,
    Cancelled
}
```

De IContractRepository onder services voorziet onderstaande methods:

```
public interface IContractRepository
{
    2 references | 0 exceptions
    IEnumerable<Contract> GetAll();
    5 references | 0 exceptions
    Contract Get(int id);
    2 references | 0 exceptions
    void Add(Contract contract);
    3 references | 0 exceptions
    void Delete(Contract contract);
    2 references | 0 exceptions
    void Update(Contract contract);
}
```

Als je de applicatie opstart krijg je onderstaande swagger te zien:

Home

GET /Home

PUT /Home

POST /Home

DELETE /Home

GET /Home/id

Models

```
ContractCreateViewModel {
  code*           string
  description*    string
  type           integer($int32)
                Enum:
                > Array [ 2 ]
  state          integer($int32)
                Enum:
                > Array [ 3 ]
}
```

```
ContractUpdateViewModel {
  description*    string
  state          integer($int32)
                Enum:
                > Array [ 3 ]
}
```

Via deze swagger kan je contracten toevoegen. Zorg ervoor dat code en description verplicht op te geven zijn:

POST /Home

Parameters Try it out

Name	Description
ContractCreateViewModel <i>(body)</i>	Example Value Model <pre>{ "code": "string", "description": "string", "type": 0, "state": 0 }</pre> Parameter content type <input type="text" value="application/json-patch+json"/>

Verder kan je via de swagger al de contracten opvragen die in het systeem zitten. Je kan ook de details opvragen van een specifiek contract op basis van id:

GET /Home/id

Parameters Cancel

Name	Description
id integer(\$int32) <i>(query)</i>	<input type="text" value="id"/>

Execute

Responses Response content type

Code	Description
------	-------------

Verder kan je description en state aanpassen voor een bepaalde id:

PUT /Home

Parameters Cancel

Name	Description
id integer(\$int32) (query)	id
contractUpdateViewModel (body)	Edit Value Model <pre>{ "description": "string", "state": 0 }</pre>

En je kan ook nog items verwijderen op basis van id:

DELETE /Home

Parameters Cancel

Name	Description
id integer(\$int32) (query)	id

Execute

Voeg optioneel een unit test project toe. Hier test je de delete method op de home controller. Je test of de controller correct omgaat met het proberen verwijderen van items die niet bestaan. Dit doe je door de IContractRepository uit te mocken met Moq.