# BakeSchoolAdmin

*Umgeher Alexander*
*Version Version 1.0*
*Sun May 29 2022*

# Inhaltsverzeichnis

# BakeSchoolAdmin

## Anleitung und Dokumentation

Diese befinden sich im Dokumentationsordner.

## Installation

Um das Programm zu erstellen, müssen Sie Docker lokal hosten. Sie sollten das offizielle MongoDb-Image pushen. docker pull mongo Laden Sie das Image in Ihr eigenes Docker und führen Sie es aus.

Beim ersten Start des Programms wird eine Datenbank "LearnBakeDb" mit zwei Sammlungen erstellt: "CategoryCollection" "RecipeCollectin"

## Bei Störungen und Fehlern

Wenn Fehler in unserem Programm auftreten, versuchen Sie bitte, das Programm neu zu starten und stellen Sie sicher, dass die Datenbank im Hintergrund läuft. Wenn der Fehler weiterhin auftritt, wenden Sie sich bitte an unseren Support. Wir werden so schnell wie möglich ein neues Update an das Repository senden.

#Email: xyc@gmail.com

## Programmfunktion

Der User kann auf der Startseite des Programm auswählen, ob er in den Seite mit den Kategorien oder mit den Rezepten haben will. Dies kann der User per Button entscheiden.

### Bereichseite (Category)

Hier kann der User die aktuellen Bereiche, die in einer MongoDb abgespeichert sind, ansehen und per Mausklick bekommt der User eine paar Details, wie beispielsweise der Name und die Beschreibung für den jeweiligen Bereich. Dort kann er direkt eine Änderung durchführen und mit dem Button „Ändern“, wird diese Daten in der Datenbank geändert. Mit dem Button „Delete“ kann der User einen Bereich in der Datenbank löschen. Zuletzt kann der User mit dem Button „Erstellen“, dieser befindet sich unten der Auflistung der Bereiche, einen Bereich erstellen.

### Rezeptseite (recipes)

Wie auf der obigen Seite, bekommt der User eine Auflistung von den ganzen Rezepten, die in der Datenbank existieren. Mit einem Mausklick auf ein Rezept öffnet sich ein Fenster und gibt den User alle benötigen Details, wie zum Beispiel: Zutaten, Beschreibungstext und Beschreibungsbild. Die Beschreibung- Schritte kann der Nutzer mit den nummerierten Buttons (1,2,3) anklicken und sich Schritt für Schritt durchklicken. Das Rezept kann man nicht

verändern oder löschen, weil diese Daten werden, parallel auf der Webseite von BakeSchool angezeigt.

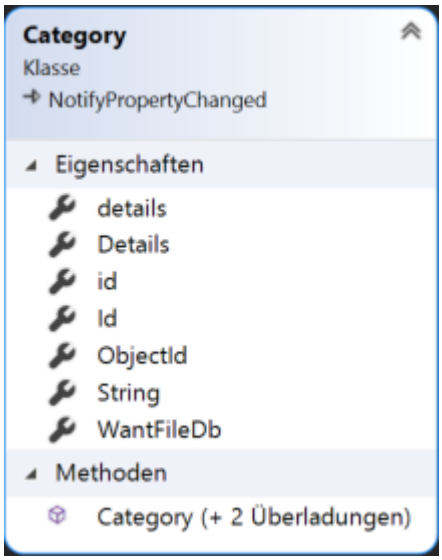
Das Erstellen von einem Rezept wird mit dem Button „Erstellen“ beginnen, dort kann der User in den verschieden Details, wie beispielsweiße: Zutat, Beschreibung mit Bild, hinzufügen.

Erst wenn eine Zutat und eine Beschreibung in den angeordneten Felder hineingeschrieben worden ist, kann der User auf dem Button „Check“ die Daten überprüfen bevor diese in der Datenbank abgespeichert werden. Mit dem Button „Home“ wird der User wieder auf die Startseite zurück kommen.

## Klassendiagram grobe Übersicht

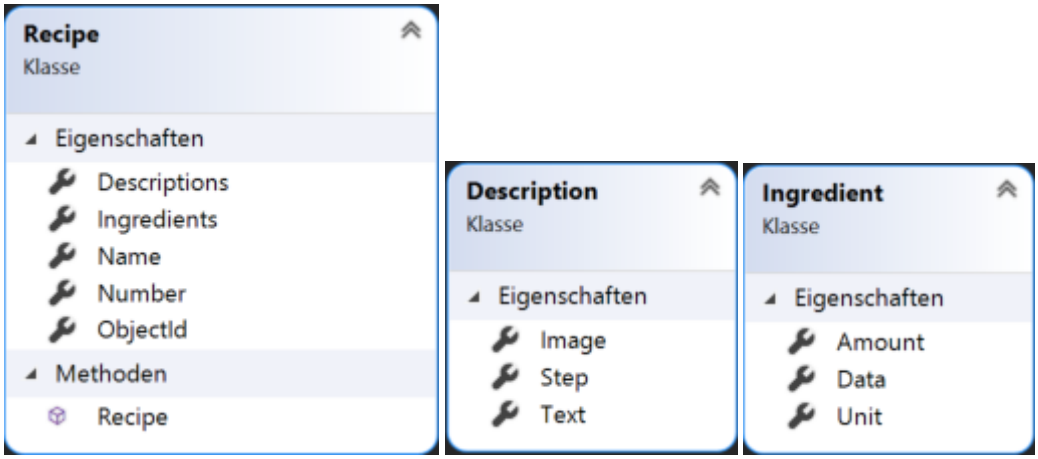
Es wurde für das Programm zwei Hauptklassen (Bereich klasse und Rezeptklasse) erstellt. Die Kategorie besteht aus dem Name des Bereichs und mehrere Details, so wie der Beschreibungstext oder die Schwierigkeitsstufe. Die Rezeptklasse ist mit zwei Unterklassen (Klasse „Zutat“ und Klasse „Beschreibung“) zusammengebaut.

### Category Class (Bereich klasse)





### Recipe Class (Rezeptklasse)


Recipe
Klasse
Eigenschaften
Descriptions
Ingredients
Name
Number
ObjectId
Methoden
Recipe
Description
Klasse
Eigenschaften
Image
Step
Text
Ingredient
Klasse
Eigenschaften
Amount
Data
Unit

# Datenbankabbildung

## Kategorie Datenbank:

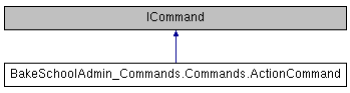

```
categoryId: 1
details: Object
    name: "Brot"
    img: "ffgs"
    text: "Das ist ein Brot"
    difficultyLevel: 2
```


## Rezeptdatenbank:


```
number: 1
ingredients: Array
  0: Object
      data: "Zucker"
      amount: 10
      unit: "g"
  1: Object
description: Array
  0: Object
      step: 1
      text: "Das ist eine Beschreibung"
      image: "C:\Users\Umgeher\Downloads\BakeSchoolAdmin\BakeSchoolAdmin\bin\Debug\i..."
```


# Commands.ActionCommand Class Reference

Class for the action commands Derives from ICommand
Inheritance diagram for BakeSchoolAdmin_Commands.Commands.ActionCommand:




## Public Member Functions

**ActionCommand** (Action< object > execute, Func< object, bool > canExecute)

*Initializes a new instance of the **ActionCommand** class*

bool **CanExecute** (object parameter)

*Defines the method that determines if the command can execute in this current state*

void **Execute** (object parameter)

*Defines the method to be called when the command is invoked*

## Properties

EventHandler **CanExecuteChanged**

*Gets executed when changes happen which affect whether or not the command should execute*

---

## Detailed Description

Class for the action commands Derives from ICommand

---

## Constructor & Destructor Documentation

### ActionCommand (Action< object > *execute*, Func< object, bool > *canExecute*)

Initializes a new instance of the **ActionCommand** class

**Parameters**

| | |
|---|---|
| *execute* | this method called when execute is invoked. |
| *canExecute* | the method called when **CanExecute()** is invoked |

---

## Member Function Documentation

### ActionCommand.CanExecute (object *parameter*)

Defines the method that determines if the command can execute in this current state

**Parameters**

| | |
|---|---|
| *parameter* | Data used by the command |

**Returns**

`true` if the command can execute, otherwise `false`

### void ActionCommand.Execute (object *parameter*)

Defines the method to be called when the command is invoked

**Parameters**

| | |
|---|---|
| *parameter* | Data used by the command |

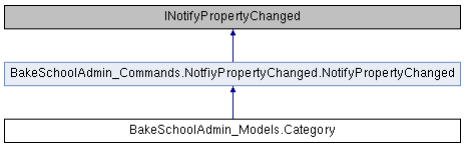
---

### The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Commands/Commands/ActionCommand.cs

# Category Class Reference

Defines the **Category**.
Inheritance diagram for BakeSchoolAdmin_Models.Category:




## Public Member Functions

**Category** ()
*Initializes a new instance of the **Category** class.*

**Category** (int **id**, **CategoryDetails details**)
*Initializes a new instance of the **Category** class.*

**Category** (int **id**, **CategoryDetails details**, bool wantFileDb)
*Initializes a new instance of the **Category** class.*

## Properties

ObjectId **ObjectId** `[get, set]`
*Gets or sets the ObjectId.*

string **String** `[get, set]`
*Gets or sets the String.*

int **Id** `[get, set]`
*Gets or sets the Id.*

**Detail Details** `[get, set]`
bool **WantFileDb** `[get, set]`
*Gets or sets a value indicating whether the user want to save the database into a file.*

int **id** `[get, set]`
*Gets or sets the id.*

**CategoryDetails details** `[get, set]`
*Gets or sets the details.*

## Additional Inherited Members

## Detailed Description

Defines the **Category**.

---

## Constructor & Destructor Documentation

### Category.Category (int *id*, CategoryDetails *details*)

Initializes a new instance of the **Category** class.

#### Parameters

| | |
|---|---|
| *id* | the id from the category |
| *details* | the category details |

### Category.Category (int *id*, CategoryDetails *details*, bool *wantFileDb*)

Initializes a new instance of the **Category** class.

#### Parameters

| | |
|---|---|
| *id* | category id. |
| *details* | category details |
| *wantFileDb* | check the desire of saving |

**Category**

---

### The documentation for this class was generated from the following files:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Modals/Category/Category.cs

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Models/Modals/Category/Category.cs

# CategoryData Class Reference

Defines the **CategoryData**.

## Properties

ObjectId **ObjectId** `[get, set]`

*Gets or sets the ObjectId.*

string **String** `[get, set]`

*Gets or sets the String.*

int **Id** `[get, set]`

*Gets or sets the Id.*

**Detail Details** `[get, set]`

*Gets or sets the Details.*

---

## Detailed Description

Defines the **CategoryData**.

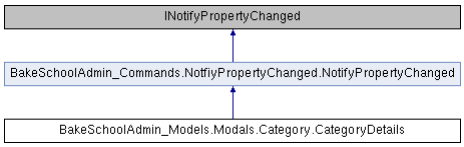
---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Models/CategoryData.cs

# CategoryDetails Class Reference

Defines the **CategoryDetails**.
Inheritance diagram for CategoryDetails:




## Properties

string **Name** `[get, set]`

*Gets or sets the Name.*

string **Image** `[get, set]`

*Gets or sets the Image.*

string **Text** `[get, set]`

*Gets or sets the Text.*

int **Level** `[get, set]`

*Gets or sets the Level.*

## Additional Inherited Members

---

## Detailed Description

Defines the **CategoryDetails**.

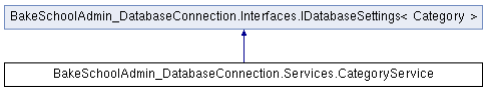
---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Models/Modals/Category/CategoryDetails.cs

# CategoryService Class Reference

Defines the **CategoryService**.
Inheritance diagram for Services.CategoryService:




## Public Member Functions

**CategoryService** ()
*Initializes a new instance of the **CategoryService** class.*

bool **init** ()
*The initialize .*

bool **IsConnection** ()
*Check whether database is connected.*

**Category GetCategory** ()
*The GetCategory.*

**Category ReadData** (int id)
*The ReadData.*

IList< **Category** > **ReadData** ()
*The ReadData.*

bool **DeleteData** (int id)
*The DeleteData.*

bool **WriteData** (**Category** data)
*The WriteDataRecipe.*

bool **PrettyWrite** (object obj, string fileName)
*Write the object into a file pretty*

bool **UpdateData** (**Category** data)
*Update the data in the database*

void **WriteDataRecipe** (IList< **Category** > data)
*The WriteDataRecipe.*

ObservableCollection< **Category** > **GetCategoryObserv** (IList< **Category** > categoriesList)

*The Category observableCollection.*

## Properties

IMongoCollection< **CategoryData** > **categoriesdata** `[get, set]`

*Gets or sets the categories data.*

---

## Detailed Description

Defines the **CategoryService**.

---

## Member Function Documentation

### CategoryService.DeleteData (int *id*)

The DeleteData.

#### Parameters

| | |
|---|---|
| *id* | The idint. |

#### Returns

The bool.

### Category CategoryService.GetCategory ()

The GetCategory.

#### Returns

The Category.

### ObservableCollection< Category > CategoryService.GetCategoryObserv (IList< Category > *categoriesList*)

The Category observableCollection.

#### Parameters

| | |
|---|---|
| *categoriesList* | The categoriesListIList<Category>. |

#### Returns

The ObservableCollection<Category>.

### bool CategoryService.init ()

The initialize .

#### Returns

The bool.

### bool CategoryService.IsConnection ()

Check whether database is connected.

#### Returns

the value of right to connection

### bool CategoryService.PrettyWrite (object *obj*, string *fileName*)

Write the object into a file pretty

#### Parameters

| | |
|---|---|
| *obj* | JSON object |
| *fileName* | name of the file |

#### Returns

the value of update state

### IList< Category > CategoryService.ReadData ()

The read Data from the Database

#### Returns

The IList<Category>.

### Category CategoryService.ReadData (int *id*)

The ReadData.

#### Parameters

| | |
|---|---|
| *id* | The id int. |

#### Returns

The Category.

### bool CategoryService.UpdateData (Category *data*)

Update the data in the database

**Parameters**

| | |
|---|---|
| *data* | category data |

**Returns**

the value of update state

### bool CategoryService.WriteData (Category *data*)

The WriteDataRecipe.

**Parameters**

| | |
|---|---|
| *data* | The dataCategory. |

**Returns**

The boolwrite data query

### void CategoryService.WriteDataRecipe (IList< Category > *data*)

The WriteDataRecipe.

**Parameters**

| | |
|---|---|
| *data* | The dataIList<Category>. |

---

**The documentation for this class was generated from the following file:**

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Services/CategoryService.cs

## CategoryService Class Reference

### Public Member Functions

**CategoryService** (IOptions< **DatabaseSettings** > categoryOptions)

*added the options from DatabaseSettings*

async Task< List< **Category** > > **GetAsync** ()

---

### Constructor & Destructor Documentation

### BakeSchoolAdmin_DataConnection.CategoryService.CategoryService (IOptions< DatabaseSettings > *categoryOptions*)

added the options from DatabaseSettings

**Parameters**

| | |
|---|---|
| *categoryOptions* | |

---

**The documentation for this class was generated from the following file:**

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Services/CategoryService.cs

# DatabaseSettings Class Reference

Defines the **DatabaseSettings**.

## Properties

string **DatabaseConnection** `[get, set]`

*Gets or sets the databaseConnection.*

string **DatabaseName** `[get, set]`

*Gets or sets the databaseName.*

string **CategoryCollectionName** `[get, set]`

*Gets or sets the categoryCollectionName.*

---

## Detailed Description

Defines the **DatabaseSettings**.

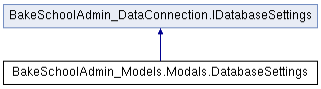
---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Services/DatabaseSettings.cs

## DatabaseSettings Class Reference

Inheritance diagram for BakeSchoolAdmin_Models.Modals.DatabaseSettings:




### Properties

string **DatabaseConnection** `[get, set]`
string **DatabaseName** `[get, set]`
string **CategoryCollectionName** `[get, set]`
string **RecipesCollectionName** `[get, set]`

## Description Class Reference

the description for the repices step by step

### Properties

int **Step** `[get, set]`
string **Text** `[get, set]`
List< **Hint** > **Hints** `[get, set]`
string **Image** `[get, set]`

---

### Detailed Description

the description for the repices step by step

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Modals/Recipe/Recipe.cs

## Recipe.Description Class Reference

the description for the recipes step by step.

### Properties

int **Step** `[get, set]`

*Gets or sets the Step.*

string **Text** `[get, set]`

*Gets or sets the Text.*

string **Image** `[get, set]`

*Gets or sets the Image.*

### Detailed Description

the description for the recipes step by step.

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Models/Modals/Recipe/Description.cs

## DescriptionData Class Reference

the description for the recipes step by step.

### Properties

int **Step** `[get, set]`
*Gets or sets the Step.*

string **Text** `[get, set]`
*Gets or sets the Text.*

List< **Hint** > **Hints** `[get, set]`
*Gets or sets the Hints.*

string **Image** `[get, set]`
*Gets or sets the Image.*

---

### Detailed Description

the description for the recipes step by step.

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Models/RecipeData.cs

# DatabaseConnection.Models.Detail Class Reference

The class is dependency at the parent class! Contain the details of the Category.

## Properties

string **Name** `[get, set]`

*Gets or sets the Name.*

string **Image** `[get, set]`

*Gets or sets the Image.*

string **Text** `[get, set]`

*Gets or sets the Text.*

int **Level** `[get, set]`

*Gets or sets the Level.*

---

## Detailed Description

The class is dependency at the parent class! Contain the details of the Category.

---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Models/CategoryData.cs

# BakeSchoolAdmin_Models.Detail Class Reference

The class is dependency at the parent class! Contain the details of the **Category**

## Properties

string **Name** `[get, set]`
string **Image** `[get, set]`
string **Text** `[get, set]`
int **Level** `[get, set]`

## Detailed Description

The class is dependency at the parent class! Contain the details of the **Category**

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnectio/Modals/Categoy/Category.cs

# Interfaces.IDatabaseSettings< T > Interface Template Reference

Defines the IDatabaseSettings<T>.

## Public Member Functions

bool **init** ()
*check the initialize for the database.*

T **ReadData** (int id)
*Read single dataset from service.*

IList< T > **ReadData** ()
*Read all datasets form service.*

void **WriteDataRecipe** (T data)
*Write single dataset to service.*

void **WriteDataRecipe** (IList< T > data)
*Write all datasets to service.*

---

## Detailed Description

Defines the IDatabaseSettings<T>.

### Template Parameters

| | |
|---|---|
| *T* | Only the database interface |

**Type Constraints**

*T* : *class*

---

## Member Function Documentation

### bool Interfaces.IDatabaseSettings< T >.init ()

check the initialize for the database.

#### Returns

the value true/false

### IList< T > Interfaces.IDatabaseSettings< T >.ReadData ()

Read all datasets form service.

#### Returns

The IList of a generic class

### Interfaces.IDatabaseSettings< T >.ReadData (int *id*)

Read single dataset from service.

#### Parameters

| | |
|---|---|
| *id* | the data id from the database |

#### Returns

the id data

### void Interfaces.IDatabaseSettings< T >.WriteDataRecipe (IList< T > *data*)

Write all datasets to service.

#### Parameters

| | |
|---|---|
| *data* | Context from the database |

### void IDatabaseSettings< T >.WriteDataRecipe (T *data*)

Write single dataset to service.

#### Parameters

| | |
|---|---|
| *data* | Context from the database |

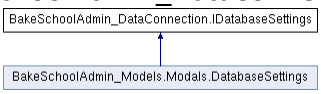
---

### The documentation for this interface was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Interfaces/IDatabaseSettings.cs

## IDatabaseSettings Interface Reference

Inheritance diagram for BakeSchoolAdmin_DataConnection.IDatabaseSettings:




### Properties

string **DatabaseConnection** `[get, set]`
string **DatabaseName** `[get, set]`
string **CategoryCollectionName** `[get, set]`

---

The documentation for this interface was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Interfaces/IDatabaseSettings.cs

## Ingredient Class Reference

The class is dependency at the parent class! Contain the details of the Recipes

### Properties

string **Data** `[get, set]`
double **Amount** `[get, set]`
string **Unit** `[get, set]`

---

### Detailed Description

The class is dependency at the parent class! Contain the details of the Recipes

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Modals/Recipe/Recipe.cs

## Recipe.Ingredient Class Reference

The class is dependency at the parent class! Contain the details of the Recipes.

### Properties

string **Data** `[get, set]`

*Gets or sets the Data.*

double **Amount** `[get, set]`

*Gets or sets the Amount.*

string **Unit** `[get, set]`

*Gets or sets the Unit.*

---

### Detailed Description

The class is dependency at the parent class! Contain the details of the Recipes.

---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Models/Modals/Recipe/Ingredient.cs

## IngredientData Class Reference

The class is dependency at the parent class! Contain the details of the Recipes.

### Properties

string **Data** `[get, set]`

*Gets or sets the Data.*

double **Amount** `[get, set]`

*Gets or sets the Amount.*

string **Unit** `[get, set]`

*Gets or sets the Unit.*

---

### Detailed Description

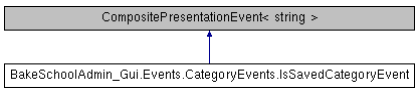
The class is dependency at the parent class! Contain the details of the Recipes.

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Models/RecipeData.cs

## CategoryEvents.IsSavedCategoryEvent Class Reference

Event for the state of saving by the category
Inheritance diagram for BakeSchoolAdmin_Gui.Events.CategoryEvents.IsSavedCategoryEvent:




---

### Detailed Description

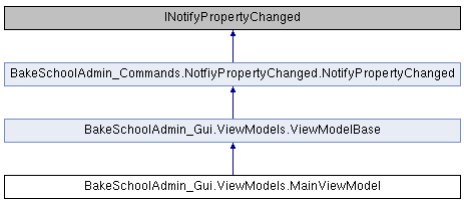
Event for the state of saving by the category

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Gui/Events/CategoryEvents/IsSavedCategoryEvent.cs

## MainViewModel Class Reference

Defines the **MainViewModel**.
Inheritance diagram for BakeSchoolAdmin_Gui.ViewModels.MainViewModel:




### Public Member Functions

**MainViewModel** (IEventAggregator eventAggregator)
*Initializes a new instance of the **MainViewModel** class.*

### Properties

UserControl **CurrentMainView** `[get, set]`
*Gets or sets the view that is currently bound to the main ContentControl..*

ICommand **MainViewCommand** `[get]`
*Gets the MainViewCommand.*

ICommand **CategoryViewCommand** `[get]`

*Gets the CategoryViewCommand.*

ICommand **RecipesViewCommand** `[get]`

*Gets the RecipesViewCommand.*

ICommand **TestRecipeCommand** `[get]`

*Gets the RecipesViewCommand.*

## Additional Inherited Members

---

## Detailed Description

Defines the **MainViewModel**.

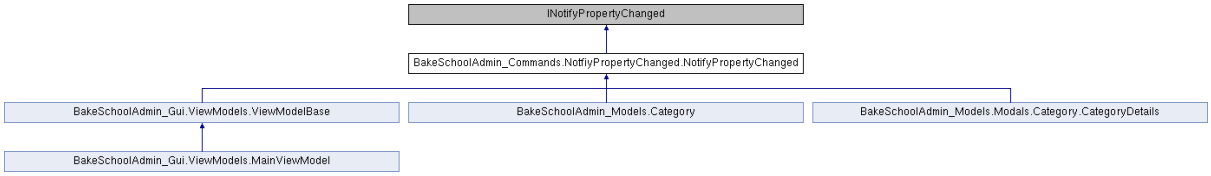
---

## Constructor & Destructor Documentation

### BakeSchoolAdmin_Gui.ViewModels.MainViewModel.MainViewModel (IEventAggregator *eventAggregator*)

# NotfiyPropertyChanged.NotifyPropertyChanged Class Reference

Defines the **NotifyPropertyChanged**.
Inheritance diagram for BakeSchoolAdmin_Commands.NotfiyPropertyChanged.NotifyPropertyChanged:




## Public Member Functions

void **OnPropertyChanged** (string property)

*Notifies a listener that a property value has changed. Name of the property to used to notify the listener.*

## Events

PropertyChangedEventHandler **PropertyChanged**

*Multicast event for property change notifications.*

---

## Detailed Description

Defines the **NotifyPropertyChanged**.

---

## Member Function Documentation

### void BakeSchoolAdmin_Commands.NotfiyPropertyChanged.NotifyPropertyChanged.OnPropertyChanged (string *property*)

Notifies a listener that a property value has changed. Name of the property to used to notify the listener.

#### Parameters

| | |
|---|---|
| *property* | Name of the property. |

---

### The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Commands/NotfiyPropertyChanged/NotifyPropertyChanged.cs

# Recipe Class Reference

Defines the **Recipe**.

## Public Member Functions

**Recipe** ()

*Initializes a new instance of the **Recipe** class.*

## Properties

ObjectId **ObjectId** `[get, set]`

*Gets or sets the ObjectId.*

string **Name** `[get, set]`

*Gets or sets the Name.*

int **Number** `[get, set]`

*Gets or sets the Number.*

List< **Ingredient** > **Ingredients** `[get, set]`

*Gets or sets the Ingredients.*

List< **Description** > **Descriptions** `[get, set]`

*Gets or sets the Descriptions.*

---

## Detailed Description

Defines the **Recipe**.

---

The documentation for this class was generated from the following files:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Modals/Recipe/Recipe.cs

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Models/Modals/Recipe/Recipe.cs

# RecipeData Class Reference

Defines the **RecipeData**.

## Properties

ObjectId **ObjectId** `[get, set]`

*Gets or sets the ObjectId.*

string **Name** `[get, set]`

*Gets or sets the Name.*

int **Number** `[get, set]`

*Gets or sets the Number.*

**IngredientData**[] **Ingredients** `[get, set]`

*Gets or sets the Ingredients.*

**DescriptionData**[] **Descriptions** `[get, set]`

*Gets or sets the Descriptions.*

---

## Detailed Description

Defines the **RecipeData**.

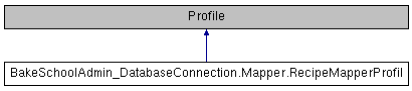
---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Models/RecipeData.cs

# Mapper.RecipeMapperProfil Class Reference

Defines the **RecipeMapperProfil**.
Inheritance diagram for
BakeSchoolAdmin_DatabaseConnection.Mapper.RecipeMapperProfil:




## Public Member Functions

**RecipeMapperProfil** ()

*Initializes a new instance of the **RecipeMapperProfil** class.*

---

## Detailed Description

Defines the **RecipeMapperProfil**.

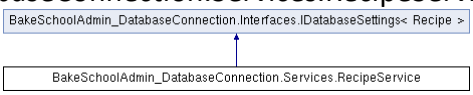
---

The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Mapper/RecipeMapperProfil.cs

# Services.RecipeService Class Reference

Defines the **RecipeService**.
Inheritance diagram for
BakeSchoolAdmin_DatabaseConnection.Services.RecipeService:




## Public Member Functions

bool **init** ()
*The initialize.*

bool **IsConnection** ()
*Check whether database is connected.*

**Recipe ReadData** (int id)
*The ReadData.*

IList< **Recipe** > **ReadData** ()
*The ReadData.*

void **WriteDataRecipe** (**Recipe** data)
*The WriteDataRecipe.*

ObservableCollection< **Recipe** > **GetRecipesObs** (IList< **Recipe** > recipeList)
*The recipes observableCollection.*

void **WriteDataRecipe** (IList< **Recipe** > data)
*Write data into a list*

## Properties

IMongoCollection< **RecipeData** > **recipesdata** `[get, set]`
*Gets or sets the recipes data.*

---

## Detailed Description

Defines the **RecipeService**.

---

## Member Function Documentation

### ObservableCollection< Recipe > RecipeService.GetRecipesObs (IList< Recipe > *recipeList*)

The recipes observableCollection.

**Parameters**

| | |
|---|---|
| *recipeList* | The recipe listIList<Recipe>. |

**Returns**

The ObservableCollection<Recipe>.

### bool RecipeService.init ()

The initialize.

**Returns**

The bool.

### bool RecipeService.IsConnection ()

Check whether database is connected.

**Returns**

value of connection state

### IList< Recipe > RecipeService.ReadData ()

The ReadData.

**Returns**

The IList<Recipe>.

### Recipe ReadData (int *id*)

The ReadData.

**Parameters**

| | |
|---|---|
| *id* | The idint. |

**Returns**

The Recipe.

### void RecipeService.WriteDataRecipe (IList< Recipe > *data*)

Write data into a list

**Parameters**

| | |
|---|---|
| *data* | recipe list |

### void WriteDataRecipe (Recipe *data*)

The WriteDataRecipe.

**Parameters**

| | |
|---|---|
| *data* | The dataRecipe. |

---

### The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DatabaseConnection/Services/RecipeService.cs

## RecipeServices Class Reference

### Public Member Functions

**RecipeServices** (IOptions< **DatabaseSettings** > options)
*added the options from DatabaseSettings*

async Task< List< **Recipe** > > **GetAsync** ()
async Task< **Recipe** > **GetAsync** (int id)

---

### Constructor & Destructor Documentation

### BakeSchoolAdmin_DataConnection.RecipeServices.RecipeServices (IOptions< DatabaseSettings > *options*)

added the options from DatabaseSettings

**Parameters**

| | |
|---|---|
| *categoryOptions* | |

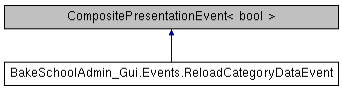
---

### The documentation for this class was generated from the following file:

C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_DataConnection/Services/RecipeServices.cs

## Events.ReloadCategoryDataEvent Class Reference

Reload the data of category on the main page
Inheritance diagram for BakeSchoolAdmin_Gui.Events.ReloadCategoryDataEvent:




---

### Detailed Description

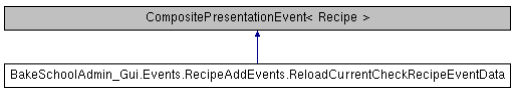
Reload the data of category on the main page

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Gu/Events/CategoryEvents/ReloadCategoryDataEvent.cs

## Events.RecipeAddEvents.ReloadCurrentCheckRecipeEventData Class Reference

Reload the recent recipe on the main page
Inheritance diagram for BakeSchoolAdmin_Gui.Events.RecipeAddEvents.ReloadCurrentCheckRecipeEventData:




---

### Detailed Description

Reload the recent recipe on the main page

---

The documentation for this class was generated from the following file:
C:/Users/Umgeher/Downloads/BakeSchoolAdmin/BakeSchoolAdmin/BakeSchoolAdmin_Gui/Events/RecipeAddEvents/ReloadCurrentCheckRecipeEventData.cs

# Wichtige Funktionenbeschreibung

## Ausgewählte Rezepte anzeigen

```
/// <summary>
/// Event handler to notice changes in the current category data.
/// </summary>
/// <param name="recipe">Reference to the sent student data.</param>
1 Verweis
private void SelectedRecipe(Recipe recipe)
{
    this.IsSelectedColor = "Black";
    this.Name = recipe.Name;

    // set the ingredients into the view
    ObservableCollection<Ingredient> ingredientsNew = new ObservableCollection<Ingredient>();
    foreach (var item in recipe.Ingredients)
    {
        ingredientsNew.Add(item);
    }

    this.Ingredients = ingredientsNew;
    this.OnPropertyChanged(nameof(this.Ingredients));

    // set the Descritpion into the view
    ObservableCollection<Description> descriptionNew = new ObservableCollection<Description>();
    foreach (Description item in recipe.Descriptions)
    {
        descriptionNew.Add(item);
    }

    this.Descriptions = descriptionNew;
    this.OnPropertyChanged(nameof(this.Descriptions));

    this.StepCount = descriptionNew.Count;
    this.OnPropertyChanged(nameof(this.StepCount));
}
```

### Beschreibung:

Hier wird das ausgewählte Rezept mit einem Klick in den verschiedenen Collection reingespeichert, da die descritptionNew und ingredientsNew in der View angezeigt wird. Für eine bessere Übersicht wird das Rezept schwarz markiert und der Rezept Name taucht in der ListViewRecipe auf.

## Jsonerstellung für die Rezepte

```
public void WriteDataRecipe(Recipe data)
{
    string name = data.Name;
    int number = data.Number;
    List<Ingredient> ingredients = new List<Ingredient>();
    List<Description> descriptions = new List<Description>();
    ingredients = data.Ingredients;
    descriptions = data.Descriptions;

    // Recipe to BsonObject
    BsonArray bsonArray = new BsonArray();
    foreach (var item in ingredients)
    {
        BsonDocument docu = new BsonDocument { { "data", item.Data }, { "amount", item.Amount }, { "unit", item.Unit } };

        bsonArray.Add(docu);
    }

    BsonArray bsonArrayDes = new BsonArray();
    foreach (var item in descriptions)
    {
        BsonDocument docu = new BsonDocument { { "step", item.Step }, { "text", item.Text }, { "image", item.Image } };

        bsonArrayDes.Add(docu);
    }

    try
    {
        var document = new BsonDocument
    {
        { "name",  name },
        { "number",  number },
        { "ingredients", bsonArray },
        { "description", bsonArrayDes }
    };
        var database = this.mongoclient.GetDatabase(this.databaseName);
        var collection = database.GetCollection<BsonDocument>(this.recipeCollectionName);
        collection.InsertOne(document);
    }
```

## Beschreibung:

Hier wird das Rezept, das in einer ObservableList gespeichert ist, in einer List<Ingredient> und List<Description>. Danach wird ein BsonArray erstellt und danach mit einer foreach Schleife wird ein Bson Document erstellt. Dies is nötig für die Datanbankspeicherung in meiner Datenbank MONGODB.
Nach der Erstellung von mehreren BsoArrays wird ein Bson Dokument erstellt.
Das beinhaltet dann alle notwendige Daten, die in der Datenbank gespeichert werden muss.

## Ausgewählt Id finden

```
foreach (Category category in this.Categories)
{
    if (category.Id == (int)parameter)
    {
        this.SelectedCategory = category;

        this.EventAggregator.GetEvent<SelectedCategoryDataEvent>().Publish(this.SelectedCategory);
    }
}
```

### Beschreibung

Mit der Funktion wird die geklickte Category gefunden. Dies is notwendig für die weitere Arbeit in der anderen ViewModel. Es ist essential, dass man eine Id bekommt und danach schießt ein Event die Id mit der ganzen Category Data in die gewünschte ViewModel.