

RegistrationManager REST API Description

v 1.0.0.0

Generated by Doxygen 1.8.12

Contents

1	Namespace Index	2
1.1	Namespace List	2
2	Hierarchical Index	2
2.1	Class Hierarchy	2
3	Class Index	3
3.1	Class List	3
4	File Index	4
4.1	File List	4
5	Namespace Documentation	5
5.1	RegistrationManager Namespace Reference	5
5.2	RegistrationManager.Controllers Namespace Reference	5
5.3	RegistrationManager.Controllers.Api Namespace Reference	5
5.4	RegistrationManager.Migrations Namespace Reference	5
5.5	RegistrationManager.Models Namespace Reference	6
6	Class Documentation	6
6.1	RegistrationManager.Migrations.AddingIdentity Class Reference	6
6.1.1	Member Function Documentation	6
6.2	RegistrationManager.Controllers.Api.AuthController Class Reference	7
6.2.1	Detailed Description	7
6.2.2	Constructor & Destructor Documentation	7
6.2.3	Member Function Documentation	8
6.3	RegistrationManager.Models.DbManagerContext Class Reference	9
6.3.1	Detailed Description	9
6.3.2	Constructor & Destructor Documentation	9
6.3.3	Member Function Documentation	10
6.3.4	Property Documentation	10

6.4	RegistrationManager.Migrations.DbManagerContextModelSnapshot Class Reference	10
6.4.1	Member Function Documentation	11
6.5	RegistrationManager.Models.DbSeeding Class Reference	11
6.5.1	Detailed Description	11
6.5.2	Constructor & Destructor Documentation	11
6.5.3	Member Function Documentation	12
6.6	RegistrationManager.Models.DbUserIdentity Class Reference	12
6.6.1	Detailed Description	12
6.6.2	Property Documentation	12
6.7	RegistrationManager.Migrations.IdentityClasses Class Reference	13
6.7.1	Member Function Documentation	13
6.8	RegistrationManager.Controllers.IndexController Class Reference	13
6.8.1	Detailed Description	14
6.8.2	Constructor & Destructor Documentation	14
6.8.3	Member Function Documentation	14
6.9	RegistrationManager.Models.IRegistrationsRepository Interface Reference	14
6.9.1	Detailed Description	15
6.9.2	Member Function Documentation	15
6.10	RegistrationManager.Models.Login Class Reference	15
6.10.1	Detailed Description	16
6.10.2	Property Documentation	16
6.11	RegistrationManager.Program Class Reference	16
6.11.1	Detailed Description	16
6.11.2	Member Function Documentation	16
6.12	RegistrationManager.Models.Register Class Reference	17
6.12.1	Detailed Description	17
6.12.2	Property Documentation	17
6.13	RegistrationManager.Models.RegistrationsRepository Class Reference	18
6.13.1	Detailed Description	18
6.13.2	Constructor & Destructor Documentation	18
6.13.3	Member Function Documentation	18
6.14	RegistrationManager.Startup Class Reference	19
6.14.1	Detailed Description	19
6.14.2	Constructor & Destructor Documentation	19
6.14.3	Member Function Documentation	20
6.14.4	Property Documentation	20

7	File Documentation	21
7.1	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/bin/src/↔ RegistrationManager/obj/Debug/netcoreapp1.0/dotnet-compile.assemblyinfo.cs File Reference . . .	21
7.2	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Debug/netcoreapp1.0/dotnet- compile.assemblyinfo.cs File Reference	21
7.3	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Release/netcoreapp1.0/dotnet- compile.assemblyinfo.cs File Reference	21
7.4	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/↔ Api/AuthController.cs File Reference	21
7.5	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/↔ IndexController.cs File Reference	21
7.6	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Db↔ ManagerContext.cs File Reference	21
7.7	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DB↔ Seeding.cs File Reference	22
7.8	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Db↔ UserIdentity.cs File Reference	22
7.9	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/I↔ RegistrationsRepository.cs File Reference	22
7.10	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/20160906033719_AddingIdentity.cs File Reference	22
7.11	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/20160906033719_AddingIdentity.Designer.cs File Reference	23
7.12	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/20160906033813_IdentityClasses.cs File Reference	23
7.13	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/20160906033813_IdentityClasses.Designer.cs File Reference	23
7.14	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/DbManagerContextModelSnapshot.cs File Reference	23
7.15	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ RegistrationsRepository.cs File Reference	24
7.16	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/↔ Login.cs File Reference	24
7.17	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/↔ Register.cs File Reference	24
7.18	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Program.cs File Reference	24
7.19	C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Startup.cs File Reference	25

Index	27
-----------------------	----

1 Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

RegistrationManager	5
RegistrationManager.Controllers	5
RegistrationManager.Controllers.Api	5
RegistrationManager.Migrations	5
RegistrationManager.Models	6

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Controller

RegistrationManager.Controllers.Api.AuthController	7
RegistrationManager.Controllers.IndexController	13
RegistrationManager.Models.DbSeeding	11
IdentityDbContext	
RegistrationManager.Models.DbManagerContext	9
IdentityUser	
RegistrationManager.Models.DbUserIdentity	12
RegistrationManager.Models.IRegistrationsRepository	14
RegistrationManager.Models.RegistrationsRepository	18
RegistrationManager.Models.Login	15
RegistrationManager.Models.Register	17
Migration	
RegistrationManager.Migrations.AddingIdentity	6
RegistrationManager.Migrations.IdentityClasses	13
ModelSnapshot	
RegistrationManager.Migrations.DbManagerContextModelSnapshot	10

RegistrationManager.Program	16
RegistrationManager.Startup	19

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

RegistrationManager.Migrations.AddingIdentity	6
RegistrationManager.Controllers.Api.AuthController The class manage he authorization and authentication of an user. All responded and response will be done by means of json format. Moreover, the descrition of the resources are also being used by the Swashbuckle (Swagger) framework.	7
RegistrationManager.Models.DbManagerContext Represents all types of data that will be stored in the database.	9
RegistrationManager.Migrations.DbManagerContextModelSnapshot	10
RegistrationManager.Models.DbSeeding Create a seed entry in the databse, when it was empty before.	11
RegistrationManager.Models.DbUserIdentity Extends the database, where the user information are stored with additional properties.	12
RegistrationManager.Migrations.IdentityClasses	13
RegistrationManager.Controllers.IndexController Manage the start page of the API in order to find out how to use the service.	13
RegistrationManager.Models.IRegistrationsRepository Repository pattern interface, in order to manage the database.	14
RegistrationManager.Models.Login Container in order to check the login details of an user.	15
RegistrationManager.Program Contains the main class and the host configuration.	16
RegistrationManager.Models.Register Container which inherents Login properties and extends them with the ConformPassword attribute.	17
RegistrationManager.Models.RegistrationsRepository DB queries will be implemented in the repository call (pattern).	18
RegistrationManager.Startup This is class contains all configuration methods and properties. The class is divided into three parts: Firstly, the constructor Secondly, the ConfigureServices method, to specify the service itself Thirdly, the Configure method, to specify the frameworks and to start the database seeding	19

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/ Program.cs	24
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/ Startup.cs	25
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/bin/src/↔ RegistrationManager/obj/Debug/netcoreapp1.0/ dotnet-compile.assemblyinfo.cs	21
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/ Index↔ Controller.cs	21
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/↔ Api/ AuthController.cs	21
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/ Db↔ ManagerContext.cs	21
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/ DB↔ Seeding.cs	22
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/ Db↔ UserIdentity.cs	22
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/ I↔ RegistrationsRepository.cs	22
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/ Registrations↔ Repository.cs	24
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/ 20160906033719_AddingIdentity.cs	22
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/ 20160906033719_AddingIdentity.Designer.cs	23
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/ 20160906033813_IdentityClasses.cs	23
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/ 20160906033813_IdentityClasses.Designer.cs	23
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/↔ Migrations/ DbManagerContextModelSnapshot.cs	23
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/ Login.↔ cs	24
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/ Register.↔ cs	24
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Debug/netcoreapp1.↔ 0/ dotnet-compile.assemblyinfo.cs	21
C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Release/netcoreapp1.↔ 0/ dotnet-compile.assemblyinfo.cs	21

5 Namespace Documentation

5.1 RegistrationManager Namespace Reference

Namespaces

- namespace [Controllers](#)
- namespace [Migrations](#)
- namespace [Models](#)

Classes

- class [Program](#)
Contains the main class and the host configuration.
- class [Startup](#)
This is class contains all configuration methods and properties. The class is divided into three parts: Firstly, the constructor Secondly, the ConfigureServices method, to specify the service itself Thirdly, the Configure method, to specify the frameworks and to start the database seeding

5.2 RegistrationManager.Controllers Namespace Reference

Namespaces

- namespace [Api](#)

Classes

- class [IndexController](#)
Manage the start page of the API in order to find out how to use the service.

5.3 RegistrationManager.Controllers.Api Namespace Reference

Classes

- class [AuthController](#)
The class manage he authorization and authentication of an user. All responded and response will be done by means of json format. Moreover, the descrition of the resources are also being used by the Swashbuckle (Swagger) framework.

5.4 RegistrationManager.Migrations Namespace Reference

Classes

- class [AddingIdentity](#)
- class [DbManagerContextModelSnapshot](#)
- class [IdentityClasses](#)

5.5 RegistrationManager.Models Namespace Reference

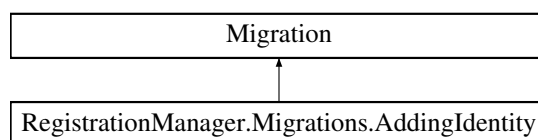
Classes

- class [DbManagerContext](#)
Represents all types of data that will be stored in the database.
- class [DbSeeding](#)
Create a seed entry in the database, when it was empty before.
- class [DbUserIdentity](#)
Extends the database, where the user information are stored with additional properties.
- interface [IRegistrationsRepository](#)
Repository pattern interface, in order to manage the database.
- class [Login](#)
Container in order to check the login details of an user.
- class [Register](#)
Container which inherits [Login](#) properties and extends them with the ConformPassword attribute.
- class [RegistrationsRepository](#)
DB queries will be implemented in the repository call (pattern).

6 Class Documentation

6.1 RegistrationManager.Migrations.AddingIdentity Class Reference

Inheritance diagram for RegistrationManager.Migrations.AddingIdentity:



Protected Member Functions

- override void [Up](#) (MigrationBuilder migrationBuilder)
- override void [Down](#) (MigrationBuilder migrationBuilder)
- override void [BuildTargetModel](#) (ModelBuilder modelBuilder)

6.1.1 Member Function Documentation

6.1.1.1 BuildTargetModel()

```

override void RegistrationManager.Migrations.AddingIdentity.BuildTargetModel (
    ModelBuilder modelBuilder ) [inline], [protected]
  
```

6.1.1.2 Down()

```

override void RegistrationManager.Migrations.AddingIdentity.Down (
    MigrationBuilder migrationBuilder ) [inline], [protected]
  
```

6.1.1.3 Up()

```
override void RegistrationManager.Migrations.AddingIdentity.Up (
    MigrationBuilder migrationBuilder ) [inline], [protected]
```

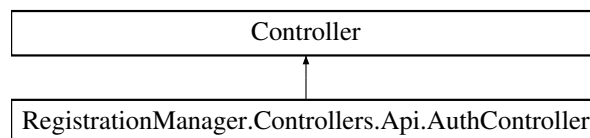
The documentation for this class was generated from the following files:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719↵
_AddingIdentity.cs
- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719↵
_AddingIdentity.Designer.cs

6.2 RegistrationManager.Controllers.Api.AuthController Class Reference

The class manage he authorization and authentication of an user. All responded and response will be done by means of json format. Moreover, the descrition of the resources are also being used by the Swashbuckle (Swagger) framework.

Inheritance diagram for RegistrationManager.Controllers.Api.AuthController:



Public Member Functions

- [AuthController](#) (ILogger< [AuthController](#) > loggerFactory, [IRegistrationsRepository](#) registrationRepository, SignInManager< [DbUserIdentity](#) > signInManager)
The authorization and authentication controller. Check whether a certain user is registered in the database. If not the service provides the ressource in order to store the user in the database. Appropriate html status codes will be returned.
- async Task< IActionResult > [LoginPost](#) ([FromBody, Required] [Login](#) login)
Check whether the user already exist in the database. This is required to use further services.
- async Task< IActionResult > [RegistrationPost](#) ([FromBody, Required] [Register](#) registration)
Registration service adds a new user to the database. If the registration failed a appropriate html status code will be returned. Keep in mind the username and email are the same, currently they are not seperated. Therefore, please use the email attribute only!

6.2.1 Detailed Description

The class manage he authorization and authentication of an user. All responded and response will be done by means of json format. Moreover, the descrition of the resources are also being used by the Swashbuckle (Swagger) framework.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 AuthController()

```
RegistrationManager.Controllers.Api.AuthController.AuthController (
    ILogger< AuthController > loggerFactory,
    IRegistrationsRepository registrationRepository,
    SignInManager< DbUserIdentity > signInManager ) [inline]
```

The authorization and authentication controller. Check whether a certain user is registered in the database. If not the service provides the ressource in order to store the user in the database. Appropriate html status codes will be returned.

Parameters

<i>loggerFactory</i>	Logs all steps in case of an error.
<i>registrationRepository</i>	Repository pattern which handles all database call.
<i>signInManager</i>	Sign in manager stores and check the user credentials by means of the database.

6.2.3 Member Function Documentation**6.2.3.1 LoginPost()**

```
async Task<IActionResult> RegistrationManager.Controllers.Api.AuthController.LoginPost (
    [FromBody, Required] Login login ) [inline]
```

Check whether the user already exist in the database. This is required to use further services.

Parameters

<i>login</i>	Login informatio about the user, which need to be comform.
--------------	--

Returns

Task is a asynchronous param which binds the IActionResult. IActionResult represent the current state of the service.

<response code="200">User had been added to the database.</response> <response code="507">Database request failed</response> <response code="400">Probably the credentials are not valid.</response>

6.2.3.2 RegistrationPost()

```
async Task<IActionResult> RegistrationManager.Controllers.Api.AuthController.RegistrationPost
(
    [FromBody, Required] Register registration ) [inline]
```

Registration service adds a new user to the database. If the registration failed a appropriate html status code will be returned. Keep in mind the username and email are the same, currently they are not seperated. Therefore, please use the email attribute only!

Parameters

<i>registration</i>	Registration informatio about the user.
---------------------	---

Returns

Task is a asynchronous param which binds the IActionResult. IActionResult represent the current state of the service. In case of an exception an interal server error will be returned.

<response code="201">A new user had been added to the database.</response> <response code="403">User could not be added to the database</response> <response code="300">User is already stored in

the database.</response> <response code="500">Internal server error, most likely because of the database.</response> <response code="400">Probably the credentials are not valid.</response>

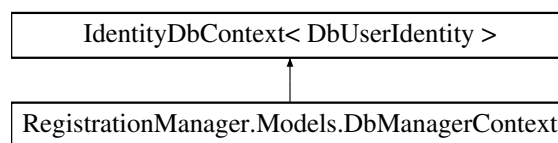
The documentation for this class was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/Api/Auth↔
Controller.cs

6.3 RegistrationManager.Models.DbManagerContext Class Reference

Represents all types of data that will be stored in the database.

Inheritance diagram for RegistrationManager.Models.DbManagerContext:



Public Member Functions

- [DbManagerContext](#) (IConfigurationRoot configurationRoot, DbContextOptions options)
Constructor where the classes and frameworks are injected, in order to manage the database connection.

Protected Member Functions

- override void [OnConfiguring](#) (DbContextOptionsBuilder optionsBuilder)
Establish the connection to the database and it passes them to the base class.

Properties

- DbSet< [DbUserIdentity](#) > [DbUserIdentity](#) [get, set]
Database extension with additional user information.

6.3.1 Detailed Description

Represents all types of data that will be stored in the database.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 DbManagerContext()

```

RegistrationManager.Models.DbManagerContext.DbManagerContext (
    IConfigurationRoot configurationRoot,
    DbContextOptions options ) [inline]
  
```

Constructor where the classes and frameworks are injected, in order to manage the database connection.

Parameters

<i>configurationRoot</i>	Configuration of the database, e.g. establish the database connection.
<i>options</i>	Database option can be specified here, which then will be passed to the base class.

6.3.3 Member Function Documentation**6.3.3.1 OnConfiguring()**

```
override void RegistrationManager.Models.DbManagerContext.OnConfiguring (
    DbContextOptionsBuilder optionsBuilder ) [inline], [protected]
```

Establish the connection to the database and it passes them to the base class.

Parameters

<i>optionsBuilder</i>	The option builder gets the connection string from the config.json file.
-----------------------	--

6.3.4 Property Documentation**6.3.4.1 DbUserIdentity**

```
DbSet<DbUserIdentity> RegistrationManager.Models.DbManagerContext.DbUserIdentity [get], [set]
```

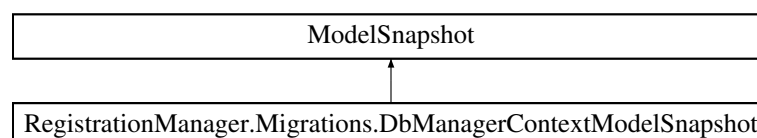
Database extension with additional user information.

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/[DbManagerContext.cs](#)

6.4 RegistrationManager.Migrations.DbManagerContextModelSnapshot Class Reference

Inheritance diagram for RegistrationManager.Migrations.DbManagerContextModelSnapshot:

**Protected Member Functions**

- override void [BuildModel](#) (ModelBuilder modelBuilder)

6.4.1 Member Function Documentation

6.4.1.1 BuildModel()

```
override void RegistrationManager.Migrations.DbManagerContextModelSnapshot.BuildModel (
    ModelBuilder modelBuilder ) [inline], [protected]
```

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/[DbManagerContextModelSnapshot.cs](#)

6.5 RegistrationManager.Models.DbSeeding Class Reference

Create a seed entry in the database, when it was empty before.

Public Member Functions

- [DbSeeding](#) ([IRegistrationsRepository](#) registrationRepository, UserManager< [DbUserIdentity](#) > userManagerIdentity)

Creates seeding information in the database, in case it had been empty before.
- async Task [EnsureSeedData](#) ()

Seed the database, when it was empty before.

6.5.1 Detailed Description

Create a seed entry in the database, when it was empty before.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 DbSeeding()

```
RegistrationManager.Models.DbSeeding.DbSeeding (
    IRegistrationsRepository registrationRepository,
    UserManager< DbUserIdentity > userManagerIdentity ) [inline]
```

Creates seeding information in the database, in case it had been empty before.

Parameters

<i>registrationRepository</i>	Constructor injection of the repository pattern, in order to manage a new seeding entry in the database.
<i>userManagerIdentity</i>	Constructor injection of the user manager, in order to retrieve information from the database.

6.5.3 Member Function Documentation

6.5.3.1 EnsureSeedData()

```
async Task RegistrationManager.Models.DbSeeding.EnsureSeedData ( ) [inline]
```

Seed the database, when it was empty before.

Returns

The Task class is required for the Wait() operator while seeding the database.

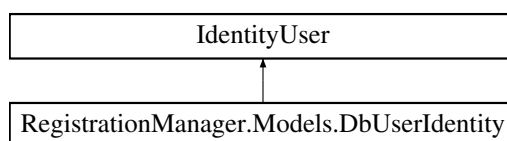
The documentation for this class was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/[DBSeeding.cs](#)

6.6 RegistrationManager.Models.DbUserIdentity Class Reference

Extends the database, where the user information are stored with additional properties.

Inheritance diagram for RegistrationManager.Models.DbUserIdentity:



Properties

- DateTime [Registration](#) [get, set]
Additional stored column in the database and the column type is DateTime.

6.6.1 Detailed Description

Extends the database, where the user information are stored with additional properties.

6.6.2 Property Documentation

6.6.2.1 Registration

```
DateTime RegistrationManager.Models.DbUserIdentity.Registration [get], [set]
```

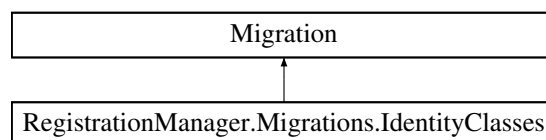
Additional stored column in the database and the column type is DateTime.

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/[DbUserIdentity.cs](#)

6.7 RegistrationManager.Migrations.IdentityClasses Class Reference

Inheritance diagram for RegistrationManager.Migrations.IdentityClasses:



Protected Member Functions

- override void [Up](#) (MigrationBuilder migrationBuilder)
- override void [Down](#) (MigrationBuilder migrationBuilder)
- override void [BuildTargetModel](#) (ModelBuilder modelBuilder)

6.7.1 Member Function Documentation

6.7.1.1 BuildTargetModel()

```
override void RegistrationManager.Migrations.IdentityClasses.BuildTargetModel (
    modelBuilder modelBuilder ) [inline], [protected]
```

6.7.1.2 Down()

```
override void RegistrationManager.Migrations.IdentityClasses.Down (
    MigrationBuilder migrationBuilder ) [inline], [protected]
```

6.7.1.3 Up()

```
override void RegistrationManager.Migrations.IdentityClasses.Up (
    MigrationBuilder migrationBuilder ) [inline], [protected]
```

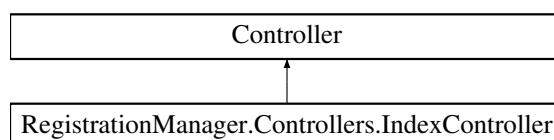
The documentation for this class was generated from the following files:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/[20160906033813↔_IdentityClasses.cs](#)
- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/[20160906033813↔_IdentityClasses.Designer.cs](#)

6.8 RegistrationManager.Controllers.IndexController Class Reference

Manage the start page of the API in order to find out how to use the service.

Inheritance diagram for RegistrationManager.Controllers.IndexController:



Public Member Functions

- [IndexController](#) (ILogger< [IndexController](#) > loggerFactory)
Constructor to inject the required classes and frameworks like the logger.
- IActionResult [API](#) ()
Index/start page of the service, in order to find out how to use the API.

6.8.1 Detailed Description

Manage the start page of the API in order to find out how to use the service.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 IndexController()

```
RegistrationManager.Controllers.IndexController.IndexController (
    ILogger< IndexController > loggerFactory ) [inline]
```

Constructor to inject the required classes and frameworks like the logger.

Parameters

<i>loggerFactory</i>	Logs all steps in case of an error.
----------------------	-------------------------------------

6.8.3 Member Function Documentation

6.8.3.1 API()

```
IActionResult RegistrationManager.Controllers.IndexController.API ( ) [inline]
```

Index/start page of the service, in order to find out how to use the API.

Returns

Returns the strat page which is managed via the swagger (Swashbuckle) framework

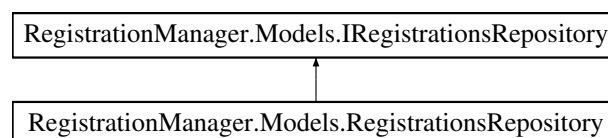
The documentation for this class was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/[IndexController.cs](#)

6.9 RegistrationManager.Models.IRegistrationsRepository Interface Reference

Repository pattern interface, in order to manage the database.

Inheritance diagram for RegistrationManager.Models.IRegistrationsRepository:



Public Member Functions

- Task< Boolean > [CreateEntry](#) ([Register](#) newRegistration)
Create a new entry (row) in the database by means of the entity framework.

6.9.1 Detailed Description

Repository pattern interface, in order to manage the database.

6.9.2 Member Function Documentation

6.9.2.1 CreateEntry()

```
Task<Boolean> RegistrationManager.Models.IRegistrationsRepository.CreateEntry (
    Register newRegistration )
```

Create a new entry (row) in the database by means of the entity framework.

Parameters

<i>newRegistration</i>	Required information about the user, e.g. email, password, etc.
------------------------	---

Returns

Retrurns a true when the user was sored in the databse, otherwise false.

Implemented in [RegistrationManager.Models.RegistrationsRepository](#).

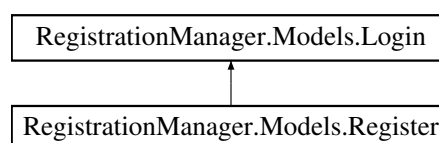
The documentation for this interface was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/[IRegistrationsRepository.cs](#)

6.10 RegistrationManager.Models.Login Class Reference

Container in order to check the login details of an user.

Inheritance diagram for RegistrationManager.Models.Login:



Properties

- string [Email](#) [get, set]
Attribute in order to store the email detail of an user.
- string [Password](#) [get, set]
Attribute in order to store the password detail of an user.

6.10.1 Detailed Description

Container in order to check the login details of an user.

6.10.2 Property Documentation

6.10.2.1 Email

```
string RegistrationManager.Models.Login.Email [get], [set]
```

Attribute in order to store the email detail of an user.

6.10.2.2 Password

```
string RegistrationManager.Models.Login.Password [get], [set]
```

Attribute in order to store the password detail of an user.

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/[Login.cs](#)

6.11 RegistrationManager.Program Class Reference

Contains the main class and the host configuration.

Static Public Member Functions

- static void [Main](#) (string[] args)
Main file as well as configuration file about the host environment. Moreover, one can define the directories and the [Startup.cs](#) file.

6.11.1 Detailed Description

Contains the main class and the host configuration.

6.11.2 Member Function Documentation

6.11.2.1 Main()

```
static void RegistrationManager.Program.Main (  
    string [] args ) [inline], [static]
```

Main file as well as configuration file about the host environment. Moreover, one can define the directories and the [Startup.cs](#) file.

Parameters

<i>args</i>	In order to pass additional external parameters, which is currently not used.
-------------	---

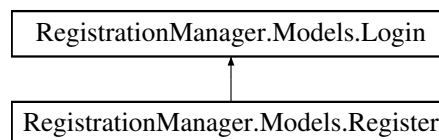
The documentation for this class was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/[Program.cs](#)

6.12 RegistrationManager.Models.Register Class Reference

Container which inherents [Login](#) properties and extends them with the ConformPassword attribute.

Inheritance diagram for RegistrationManager.Models.Register:



Properties

- string [ConfirmPassword](#) [get, set]
Attribure in order to store the confirm password detail of an user.

6.12.1 Detailed Description

Container which inherents [Login](#) properties and extends them with the ConformPassword attribute.

6.12.2 Property Documentation

6.12.2.1 ConfirmPassword

```
string RegistrationManager.Models.Register.ConfirmPassword [get], [set]
```

Attribure in order to store the confirm password detail of an user.

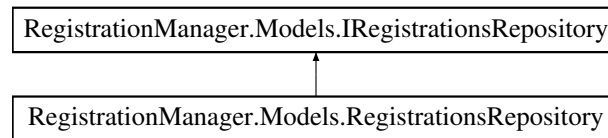
The documentation for this class was generated from the following file:

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/[Register.cs](#)

6.13 RegistrationManager.Models.RegistrationsRepository Class Reference

DB queries will be implemented in the repository call (pattern).

Inheritance diagram for RegistrationManager.Models.RegistrationsRepository:



Public Member Functions

- [RegistrationsRepository](#) ([DbManagerContext](#) dbManagerContext, UserManager< [DbUserIdentity](#) > userManagerIdentity)
Repository pattern in order to manage everything that is related to the database.
- async Task< Boolean > [CreateEntry](#) ([Register](#) newRegistration)
Creates a new entry in the database. The User Manager is required in order to store all important information about the user.

6.13.1 Detailed Description

DB queries will be implemented in the repository call (pattern).

6.13.2 Constructor & Destructor Documentation

6.13.2.1 RegistrationsRepository()

```

RegistrationManager.Models.RegistrationsRepository.RegistrationsRepository (
    DbManagerContext dbManagerContext,
    UserManager< DbUserIdentity > userManagerIdentity ) [inline]
  
```

Repository pattern in order to manage everything that is related to the database.

Parameters

<i>dbManagerContext</i>	Constructore injection of the database management context.
<i>userManagerIdentity</i>	Constructor injection of the user management identity.

6.13.3 Member Function Documentation

6.13.3.1 CreateEntry()

```

async Task<Boolean> RegistrationManager.Models.RegistrationsRepository.CreateEntry (
    Register newRegistration ) [inline]
  
```

Creates a new entry in the database. The User Manager is required in order to store all important information about the user.

Parameters

<code>newRegistration</code>	Required information about the user, e.g. email, password, etc.
------------------------------	---

Returns

Retrurns a true when the user was stored in the databse, otherwies false.

Implements [RegistrationManager.Models.IRegistrationsRepository](#).

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/[RegistrationsRepository.cs](#)

6.14 RegistrationManager.Startup Class Reference

This is class contains all configuration methods and properties. The class is divided into three parts: Firstly, the constructor Secondly, the ConfigureServices method, to specify the service itself Thirdly, the Configure method, to specify the frameworks and to start the database seeding

Public Member Functions

- [Startup](#) (IHostingEnvironment env)
Constructor of [startup.cs](#) class, in oder to specify the hosting environment.
- void [ConfigureServices](#) (IServiceCollection services)
This method gets called by the runtime. Use this method to add services to the container.
- void [Configure](#) (IApplicationBuilder app, [DbSeeding](#) seeder, ILoggerFactory loggerFactory, IHostingEnvironment environment)
This method gets called by the runtime. Use this method to configure the HTTP request pipeline. Note: the order matters to ensure which framework will be called first.

Properties

- IConfigurationRoot [Configuration](#) [get]
Returns the configuration property about the service. It is not beeing used within the service itself.

6.14.1 Detailed Description

This is class contains all configuration methods and properties. The class is divided into three parts: Firstly, the constructor Secondly, the ConfigureServices method, to specify the service itself Thirdly, the Configure method, to specify the frameworks and to start the database seeding

6.14.2 Constructor & Destructor Documentation

6.14.2.1 Startup()

```
RegistrationManager.Startup.Startup (
    IHostingEnvironment env ) [inline]
```

Constructor of [startup.cs](#) class, in oder to specify the hosting environment.

Parameters

<i>env</i>	Hosting environment in which the service will be held, as well as to distinguish between developer and hosting enviroment.
------------	--

6.14.3 Member Function Documentation**6.14.3.1 Configure()**

```
void RegistrationManager.Startup.Configure (
    IApplicationBuilder app,
    DbSeeding seeder,
    ILoggerFactory loggerFactory,
    IHostingEnvironment environment ) [inline]
```

This method gets called by the runtime. Use this method to configure the HTTP request pipeline. Note: the order matters to ensure which framework will be called first.

Parameters

<i>app</i>	The service which is provided
<i>seeder</i>	Initialize the database, in case no entry exists.
<i>loggerFactory</i>	Logs the process of the service, in case of an error.
<i>environment</i>	Environment variable in order to ensure whether it is a development environment or a runtime environment.

6.14.3.2 ConfigureServices()

```
void RegistrationManager.Startup.ConfigureServices (
    IServiceCollection services ) [inline]
```

This method gets called by the runtime. Use this method to add services to the container.

Parameters

<i>services</i>	Specifiction of how the injection should be called, as well as the configuration of the injections.
-----------------	---

6.14.4 Property Documentation**6.14.4.1 Configuration**

```
IConfigurationRoot RegistrationManager.Startup.Configuration [get]
```

Returns the configuration property about the service. It is not beeing used within the service itself.

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

- C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/[Startup.cs](#)

7 File Documentation

- 7.1 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/bin/src/RegistrationManager/obj/Debug/netcoreapp1.0/dotnet-compile.assemblyinfo.cs](#) File Reference ↩
- 7.2 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Debug/netcoreapp1.0/dotnet-compile.assemblyinfo.cs](#) File Reference
- 7.3 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Release/netcoreapp1.0/dotnet-compile.assemblyinfo.cs](#) File Reference
- 7.4 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/Api/AuthController.cs](#) File Reference ↩

Classes

- class [RegistrationManager.Controllers.Api.AuthController](#)
The class manage he authorization and authentication of an user. All responded and response will be done by means of json format. Moreover, the descriton of the resources are also being used by the Swashbuckle (Swagger) framework.

Namespaces

- namespace [RegistrationManager.Controllers.Api](#)

- 7.5 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/IndexController.cs](#) File Reference ↩

Classes

- class [RegistrationManager.Controllers.IndexController](#)
Manage the start page of the API in order to find out how to use the service.

Namespaces

- namespace [RegistrationManager.Controllers](#)

- 7.6 [C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DbManagerContext.cs](#) File Reference ↩

Classes

- class [RegistrationManager.Models.DbManagerContext](#)
Represents all types of data that will be stored in the database.

Namespaces

- namespace [RegistrationManager.Models](#)

7.7 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DBSeeding.cs File Reference

Classes

- class [RegistrationManager.Models.DbSeeding](#)
Create a seed entry in the database, when it was empty before.

Namespaces

- namespace [RegistrationManager.Models](#)

7.8 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DbUserIdentity.cs File Reference

Classes

- class [RegistrationManager.Models.DbUserIdentity](#)
Extends the database, where the user information are stored with additional properties.

Namespaces

- namespace [RegistrationManager.Models](#)

7.9 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/IRegistrationsRepository.cs File Reference

Classes

- interface [RegistrationManager.Models.IRegistrationsRepository](#)
Repository pattern interface, in order to manage the database.

Namespaces

- namespace [RegistrationManager.Models](#)

7.10 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719_AddingIdentity.cs File Reference

Classes

- class [RegistrationManager.Migrations.AddingIdentity](#)

Namespaces

- namespace [RegistrationManager.Migrations](#)

7.11 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719_↵
_AddingIdentity.Designer.cs File Reference

Classes

- class [RegistrationManager.Migrations.AddingIdentity](#)

Namespaces

- namespace [RegistrationManager.Migrations](#)

7.12 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033813_↵
_IdentityClasses.cs File Reference

Classes

- class [RegistrationManager.Migrations.IdentityClasses](#)

Namespaces

- namespace [RegistrationManager.Migrations](#)

7.13 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033813_↵
_IdentityClasses.Designer.cs File Reference

Classes

- class [RegistrationManager.Migrations.IdentityClasses](#)

Namespaces

- namespace [RegistrationManager.Migrations](#)

7.14 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/↵
DbManagerContextModelSnapshot.cs File Reference

Classes

- class [RegistrationManager.Migrations.DbManagerContextModelSnapshot](#)

Namespaces

- namespace [RegistrationManager.Migrations](#)

7.15 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/RegistrationsRepository.cs File Reference

Classes

- class [RegistrationManager.Models.RegistrationsRepository](#)
DB queries will be implemented in the repository call (pattern).

Namespaces

- namespace [RegistrationManager.Models](#)

7.16 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/Login.cs File Reference

Classes

- class [RegistrationManager.Models.Login](#)
Container in order to check the login details of an user.

Namespaces

- namespace [RegistrationManager.Models](#)

7.17 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/Register.cs File Reference

Classes

- class [RegistrationManager.Models.Register](#)
Container which inherits [Login](#) properties and extends them with the ConformPassword attribute.

Namespaces

- namespace [RegistrationManager.Models](#)

7.18 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Program.cs File Reference

Classes

- class [RegistrationManager.Program](#)
Contains the main class and the host configuration.

Namespaces

- namespace [RegistrationManager](#)

7.19 C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Startup.cs File Reference

Classes

- class [RegistrationManager.Startup](#)

This is class contains all configuration methods and properties. The class is divided into three parts: Firstly, the constructor Secondly, the ConfigureServices method, to specify the service itself Thirdly, the Configure method, to specify the frameworks and to start the database seeding

Namespaces

- namespace [RegistrationManager](#)

Index

API

RegistrationManager::Controllers::IndexController, 14

AuthController

RegistrationManager::Controllers::Api::AuthController, 7

BuildModel

RegistrationManager::Migrations::DbManagerContextModelSnapshot, 11

BuildTargetModel

RegistrationManager::Migrations::AddingIdentity, 6
RegistrationManager::Migrations::IdentityClasses, 13

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/Api/AuthController.cs, 21

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Controllers/IndexController.cs, 21

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DBSeeding.cs, 22

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DbManagerContext.cs, 21

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/DbUserIdentity.cs, 22

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/IRegistrationsRepository.cs, 22

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719_AddingIdentity.cs, 22

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033719_AddingIdentity.Designer.cs, 23

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033813_IdentityClasses.cs, 23

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/20160906033813_IdentityClasses.Designer.cs, 23

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/Migrations/DbManagerContextModelSnapshot.cs, 23

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Data/RegistrationsRepository.cs, 24

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/Login.cs, 24

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Models/Register.cs, 24

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Program.cs, 24

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/Startup.cs, 25

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/bin/src/RegistrationManager/obj/Debug/netcoreapp1.0/dotnet-compile.assemblyinfo.cs, 21

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Debug/netcoreapp1.0/dotnet-compile.assemblyinfo.cs, 21

C:/Development/Panna/RegisByHand/RegistrationManager/src/RegistrationManager/obj/Release/netcoreapp1.0/dotnet-compile.assemblyinfo.cs, 21

Configuration

RegistrationManager::Startup, 20

Configure

RegistrationManager::Startup, 20

ConfigureServices

RegistrationManager::Startup, 20

ConfirmPassword

RegistrationManager::Models::Register, 17

CreateEntry

RegistrationManager::Models::IRegistrationsRepository, 15

RegistrationManager::Models::RegistrationsRepository, 18

DbManagerContext

RegistrationManager::Models::DbManagerContext, 9

DbSeeding

RegistrationManager::Models::DbSeeding, 11

DbUserIdentity

RegistrationManager::Models::DbManagerContext, 10

Down

RegistrationManager::Migrations::AddingIdentity, 6
RegistrationManager::Migrations::IdentityClasses, 13

Email

RegistrationManager::Models::Login, 16

EnsureSeedData

RegistrationManager::Models::DbSeeding, 12

- IndexController
 - RegistrationManager::Controllers::IndexController, 14
- LoginPost
 - RegistrationManager::Controllers::Api::AuthController, 8
- Main
 - RegistrationManager::Program, 16
- OnConfiguring
 - RegistrationManager::Models::DbManagerContext, 10
- Password
 - RegistrationManager::Models::Login, 16
- Registration
 - RegistrationManager::Models::DbUserIdentity, 12
- RegistrationManager, 5
 - RegistrationManager.Controllers, 5
 - RegistrationManager.Controllers.Api, 5
 - RegistrationManager.Controllers.Api.AuthController, 7
 - RegistrationManager.Controllers.IndexController, 13
 - RegistrationManager.Migrations, 5
 - RegistrationManager.Migrations.AddingIdentity, 6
 - RegistrationManager.Migrations.DbManagerContextModelSnapshot, 10
 - RegistrationManager.Migrations.IdentityClasses, 13
 - RegistrationManager.Models, 6
 - RegistrationManager.Models.DbManagerContext, 9
 - RegistrationManager.Models.DbSeeding, 11
 - RegistrationManager.Models.DbUserIdentity, 12
 - RegistrationManager.Models.IRegistrationsRepository, 14
 - RegistrationManager.Models.Login, 15
 - RegistrationManager.Models.Register, 17
 - RegistrationManager.Models.RegistrationsRepository, 18
 - RegistrationManager.Program, 16
 - RegistrationManager.Startup, 19
 - RegistrationManager::Controllers::Api::AuthController
 - AuthController, 7
 - LoginPost, 8
 - RegistrationPost, 8
 - RegistrationManager::Controllers::IndexController
 - API, 14
 - IndexController, 14
 - RegistrationManager::Migrations::AddingIdentity
 - BuildTargetModel, 6
 - Down, 6
 - Up, 6
 - RegistrationManager::Migrations::DbManagerContextModelSnapshot
 - BuildModel, 11
 - RegistrationManager::Migrations::IdentityClasses
 - BuildTargetModel, 13
 - Down, 13
 - Up, 13
 - RegistrationManager::Models::DbManagerContext
 - DbManagerContext, 9
 - DbUserIdentity, 10
 - OnConfiguring, 10
 - RegistrationManager::Models::DbSeeding
 - DbSeeding, 11
 - EnsureSeedData, 12
 - RegistrationManager::Models::DbUserIdentity
 - Registration, 12
 - RegistrationManager::Models::IRegistrationsRepository
 - CreateEntry, 15
 - RegistrationManager::Models::Login
 - Email, 16
 - Password, 16
 - RegistrationManager::Models::Register
 - ConfirmPassword, 17
 - RegistrationManager::Models::RegistrationsRepository
 - CreateEntry, 18
 - RegistrationsRepository, 18
 - RegistrationManager::Program
 - Main, 16
 - RegistrationManager::Startup
 - Configuration, 20
 - Configure, 20
 - ConfigureServices, 20
 - Startup, 19
 - RegistrationPost
 - RegistrationManager::Controllers::Api::AuthController, 8
 - RegistrationsRepository
 - RegistrationManager::Models::RegistrationsRepository, 18
 - Startup
 - RegistrationManager::Startup, 19
 - Up