

DotnetIdentity

Generated by Doxygen 1.9.4

1 Todo List	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 Namespace Documentation	11
5.1 DotNetIdentity Namespace Reference	11
5.2 DotNetIdentity.Controllers Namespace Reference	11
5.3 DotNetIdentity.Data Namespace Reference	11
5.4 DotNetIdentity.Helpers Namespace Reference	12
5.5 DotNetIdentity.Helpers.TagHelpers Namespace Reference	12
5.6 DotNetIdentity.IdentitySettings Namespace Reference	12
5.7 DotNetIdentity.IdentitySettings.Requirements Namespace Reference	12
5.8 DotNetIdentity.IdentitySettings.Validators Namespace Reference	13
5.9 DotNetIdentity.Models Namespace Reference	13
5.9.1 Enumeration Type Documentation	13
5.9.1.1 Department	13
5.9.1.2 Gender	13
5.9.1.3 TwoFactorType	15
5.10 DotNetIdentity.Models.BusinessModels Namespace Reference	15
5.11 DotNetIdentity.Models.DataModels Namespace Reference	15
5.12 DotNetIdentity.Models.Identity Namespace Reference	16
5.13 DotNetIdentity.Models.ViewModels Namespace Reference	16
5.14 DotNetIdentity.Services Namespace Reference	17
5.15 DotNetIdentity.Services.SettingsService Namespace Reference	17
6 Class Documentation	19
6.1 DotNetIdentity.Controllers.AdminController Class Reference	19
6.1.1 Detailed Description	20
6.1.2 Constructor & Destructor Documentation	20
6.1.2.1 AdminController()	21
6.1.3 Member Function Documentation	21
6.1.3.1 AuditLogs()	21
6.1.3.2 DeleteRole()	21
6.1.3.3 DeleteUser()	22
6.1.3.4 DisableLdapLogin()	23
6.1.3.5 DisableMfaForce()	23
6.1.3.6 DisableUser()	24

6.1.3.7 EditUser() [1/2]	24
6.1.3.8 EditUser() [2/2]	26
6.1.3.9 EnableLdapLogin()	26
6.1.3.10 EnableMfaForce()	27
6.1.3.11 EnableUser()	27
6.1.3.12 ErrorLogs()	28
6.1.3.13 GetAppLogs()	28
6.1.3.14 GetAuditLogs()	29
6.1.3.15 GetErrorLogs()	29
6.1.3.16 GetUsers()	30
6.1.3.17 Index()	30
6.1.3.18 NewUser() [1/2]	30
6.1.3.19 NewUser() [2/2]	30
6.1.3.20 Roles()	32
6.1.3.21 SystemLogs()	32
6.1.3.22 UpsertRole() [1/2]	32
6.1.3.23 UpsertRole() [2/2]	33
6.1.3.24 Users()	33
6.2 DotNetIdentity.Data.AppDbContext Class Reference	34
6.2.1 Detailed Description	34
6.2.2 Constructor & Destructor Documentation	35
6.2.2.1 AppDbContext()	35
6.2.3 Member Function Documentation	35
6.2.3.1 OnModelCreating()	35
6.2.4 Property Documentation	36
6.2.4.1 AppLogs	36
6.2.4.2 AppSessionCache	37
6.2.4.3 AppSettings	37
6.3 DotNetIdentity.Models.DataModels.ApplicationSettings Class Reference	37
6.3.1 Detailed Description	37
6.3.2 Property Documentation	38
6.3.2.1 Name	38
6.3.2.2 Type	38
6.3.2.3 Value	38
6.4 DotNetIdentity.Models.DataModels.AppLogs Class Reference	38
6.4.1 Detailed Description	39
6.4.2 Property Documentation	39
6.4.2.1 Exception	39
6.4.2.2 id	39
6.4.2.3 Level	40
6.4.2.4 Message	40
6.4.2.5 MessageTemplate	40

6.4.2.6 Properties	40
6.4.2.7 Timestamp	41
6.5 DotNetIdentity.Models.Identity.AppRole Class Reference	41
6.5.1 Detailed Description	41
6.5.2 Property Documentation	41
6.5.2.1 CreatedOn	42
6.6 DotNetIdentity.Services.SettingsService.AppSettingsBase Class Reference	42
6.6.1 Detailed Description	42
6.6.2 Constructor & Destructor Documentation	42
6.6.2.1 AppSettingsBase()	43
6.6.3 Member Function Documentation	43
6.6.3.1 Load()	43
6.6.3.2 Save()	43
6.7 DotNetIdentity.Models.Identity.AppUser Class Reference	44
6.7.1 Detailed Description	45
6.7.2 Property Documentation	45
6.7.2.1 BirthDay	45
6.7.2.2 CreatedOn	46
6.7.2.3 Department	46
6.7.2.4 FirstName	46
6.7.2.5 Gender	46
6.7.2.6 IsEnabled	47
6.7.2.7 IsLdapLogin	47
6.7.2.8 IsMfaForce	47
6.7.2.9 LastName	47
6.7.2.10 ProfilePicture	48
6.7.2.11 RolesCombined	48
6.7.2.12 TwoFactorType	48
6.8 DotNetIdentity.Models.ViewModels.AssignRoleViewModel Class Reference	48
6.8.1 Detailed Description	49
6.8.2 Property Documentation	49
6.8.2.1 IsAssigned	49
6.8.2.2 RoleId	49
6.8.2.3 RoleName	49
6.9 DotNetIdentity.Models.ViewModels.ChangePasswordViewModel Class Reference	50
6.9.1 Detailed Description	50
6.9.2 Property Documentation	50
6.9.2.1 NewPassword	50
6.9.2.2 NewPasswordConfirm	50
6.9.2.3 Password	51
6.10 DotNetIdentity.Controllers.ClaimBasedController Class Reference	51
6.10.1 Detailed Description	52

6.10.2 Member Function Documentation	52
6.10.2.1 Employee()	52
6.10.2.2 FreeTrial()	52
6.10.2.3 HR()	52
6.10.2.4 Software()	53
6.10.2.5 Violence()	53
6.11 DotNetIdentity.IdentitySettings.ClaimsTransformation Class Reference	53
6.11.1 Detailed Description	54
6.11.2 Constructor & Destructor Documentation	54
6.11.2.1 ClaimsTransformation()	54
6.11.3 Member Function Documentation	54
6.11.3.1 TransformAsync()	54
6.12 DotNetIdentity.Controllers.CultureController Class Reference	55
6.12.1 Detailed Description	56
6.12.2 Member Function Documentation	56
6.12.2.1 SetCulture()	56
6.13 DotNetIdentity.Models.ViewModels.EditUserViewModel Class Reference	57
6.13.1 Detailed Description	57
6.13.2 Property Documentation	57
6.13.2.1 Department	58
6.13.2.2 Email	58
6.13.2.3 FirstName	58
6.13.2.4 Gender	58
6.13.2.5 Id	59
6.13.2.6 LastName	59
6.13.2.7 PhoneNumber	59
6.13.2.8 Roles	59
6.13.2.9 UserName	60
6.14 DotNetIdentity.Helpers.EmailHelper Class Reference	60
6.14.1 Detailed Description	60
6.14.2 Constructor & Destructor Documentation	60
6.14.2.1 EmailHelper()	60
6.14.3 Member Function Documentation	61
6.14.3.1 SendAsync()	61
6.15 DotNetIdentity.Models.BusinessModels.EmailModel Class Reference	61
6.15.1 Detailed Description	62
6.15.2 Property Documentation	62
6.15.2.1 Body	62
6.15.2.2 Subject	62
6.15.2.3 To	63
6.16 DotNetIdentity.IdentitySettings.ErrorDescriber Class Reference	63
6.16.1 Detailed Description	63

6.16.2 Member Function Documentation	64
6.16.2.1 PasswordContainsUsername()	64
6.16.2.2 UserNameContainsEmail()	64
6.16.2.3 UserNameLength()	64
6.17 DotNetIdentity.Controllers.ErrorsController Class Reference	65
6.17.1 Detailed Description	65
6.17.2 Member Function Documentation	65
6.17.2.1 ErrorCd()	65
6.17.2.2 ErrorEx()	66
6.18 DotNetIdentity.Models.ViewModels.ErrorViewModel Class Reference	67
6.18.1 Detailed Description	67
6.18.2 Property Documentation	67
6.18.2.1 RequestId	67
6.18.2.2 ShowRequestId	68
6.19 DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel Class Reference	68
6.19.1 Detailed Description	68
6.19.2 Property Documentation	68
6.19.2.1 Email	68
6.20 DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler Class Reference	69
6.20.1 Detailed Description	69
6.20.2 Member Function Documentation	69
6.20.2.1 HandleRequirementAsync()	69
6.21 DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement Class Reference	70
6.21.1 Detailed Description	70
6.22 DotNetIdentity.Models.BusinessModels.GlobalSettings Class Reference	71
6.22.1 Detailed Description	71
6.22.2 Property Documentation	71
6.22.2.1 ApplicationName	72
6.22.2.2 SessionCookieExpiration	72
6.22.2.3 SessionTimeoutRedirAfter	72
6.22.2.4 SessionTimeoutWarnAfter	72
6.23 DotNetIdentity.Controllers.HomeController Class Reference	73
6.23.1 Detailed Description	73
6.23.2 Constructor & Destructor Documentation	73
6.23.2.1 HomeController()	73
6.23.3 Member Function Documentation	74
6.23.3.1 AccessDenied()	74
6.23.3.2 Index()	74
6.24 DotNetIdentity.Services.SettingsService.ISettingsService Interface Reference	74
6.24.1 Detailed Description	75
6.24.2 Member Function Documentation	75
6.24.2.1 Save()	75

6.24.3 Property Documentation	75
6.24.3.1 Global	75
6.24.3.2 Ldap	76
6.24.3.3 Mail	76
6.25 DotNetIdentity.Models.BusinessModels.LdapSettings Class Reference	76
6.25.1 Detailed Description	77
6.25.2 Property Documentation	77
6.25.2.1 LdapBaseDn	77
6.25.2.2 LdapDomainController	77
6.25.2.3 LdapDomainName	78
6.25.2.4 LdapEnabled	78
6.25.2.5 LdapGroup	78
6.26 DotNetIdentity.Models.BusinessModels.MailSettings Class Reference	79
6.26.1 Detailed Description	79
6.26.2 Property Documentation	79
6.26.2.1 Password	80
6.26.2.2 SmtptFromAddress	80
6.26.2.3 SmtptPort	80
6.26.2.4 SmtptServer	80
6.26.2.5 SmtptUseTls	81
6.26.2.6 Username	81
6.27 DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler Class Reference	81
6.27.1 Detailed Description	82
6.27.2 Member Function Documentation	82
6.27.2.1 HandleRequirementAsync()	82
6.28 DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement Class Reference	83
6.28.1 Detailed Description	83
6.28.2 Constructor & Destructor Documentation	83
6.28.2.1 MinimumAgeRequirement()	83
6.28.3 Property Documentation	84
6.28.3.1 MinimumAge	84
6.29 DotNetIdentity.Models.ViewModels.NewUserModel Class Reference	84
6.29.1 Detailed Description	85
6.29.2 Property Documentation	85
6.29.2.1 Department	85
6.29.2.2 Email	85
6.29.2.3 FirstName	85
6.29.2.4 Gender	86
6.29.2.5 IsLdapLogin	86
6.29.2.6 IsMfaForce	86
6.29.2.7 LastName	86
6.29.2.8 Password	87

6.29.2.9 Roles	87
6.29.2.10 UserName	87
6.30 DotNetIdentity.IdentitySettings.Validators.PasswordValidator Class Reference	88
6.30.1 Detailed Description	88
6.30.2 Member Function Documentation	88
6.30.2.1 ValidateAsync()	88
6.31 DotNetIdentity.Models.ViewModels.ResetPasswordViewModel Class Reference	89
6.31.1 Detailed Description	90
6.31.2 Property Documentation	90
6.31.2.1 Password	90
6.31.2.2 Token	90
6.31.2.3 UserId	90
6.32 DotNetIdentity.Controllers.RoleBasedController Class Reference	91
6.32.1 Detailed Description	91
6.32.2 Member Function Documentation	91
6.32.2.1 Articles()	91
6.32.2.2 Dashboard()	92
6.32.2.3 Profile()	92
6.32.2.4 Users()	92
6.33 DotNetIdentity.Models.DataModels.SessionCache Class Reference	92
6.33.1 Detailed Description	93
6.33.2 Property Documentation	93
6.33.2.1 AbsoluteExpiration	93
6.33.2.2 ExpiresAtTime	93
6.33.2.3 id	93
6.33.2.4 SlidingExpirationInSeconds	94
6.33.2.5 Value	94
6.34 DotNetIdentity.Controllers.SettingsController Class Reference	94
6.34.1 Detailed Description	95
6.34.2 Constructor & Destructor Documentation	95
6.34.2.1 SettingsController()	95
6.34.3 Member Function Documentation	96
6.34.3.1 SettingsApp()	96
6.34.3.2 SettingsLdap() [1/2]	96
6.34.3.3 SettingsLdap() [2/2]	96
6.34.3.4 SettingsMail() [1/3]	97
6.34.3.5 SettingsMail() [2/3]	97
6.34.3.6 SettingsMail() [3/3]	98
6.35 DotNetIdentity.Services.SettingsService.SettingsService Class Reference	98
6.35.1 Detailed Description	99
6.35.2 Constructor & Destructor Documentation	99
6.35.2.1 SettingsService()	99

6.35.3 Member Function Documentation	100
6.35.3.1 Save()	100
6.35.4 Property Documentation	100
6.35.4.1 Global	100
6.35.4.2 Ldap	101
6.35.4.3 Mail	101
6.36 DotNetIdentity.Models.ViewModels.SignInViewModel Class Reference	101
6.36.1 Detailed Description	101
6.36.2 Property Documentation	101
6.36.2.1 Password	102
6.36.2.2 RememberMe	102
6.36.2.3 UserName	102
6.37 DotNetIdentity.Models.ViewModels.SignUpViewModel Class Reference	102
6.37.1 Detailed Description	103
6.37.2 Property Documentation	103
6.37.2.1 BirthDay	103
6.37.2.2 Department	104
6.37.2.3 Email	104
6.37.2.4 FirstName	104
6.37.2.5 Gender	104
6.37.2.6 LastName	105
6.37.2.7 Password	105
6.37.2.8 UserName	105
6.38 DotNetIdentity.Models.ViewModels.StatusCodesModel Class Reference	105
6.38.1 Detailed Description	106
6.38.2 Property Documentation	106
6.38.2.1 ErrorMessage	106
6.38.2.2 RouteOfException	106
6.38.2.3 StatusCode	107
6.39 DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService Class Reference	107
6.39.1 Detailed Description	107
6.39.2 Constructor & Destructor Documentation	107
6.39.2.1 TwoFactorAuthenticationService()	107
6.39.3 Member Function Documentation	108
6.39.3.1 GenerateQrCodeUri()	108
6.40 DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel Class Reference	108
6.40.1 Detailed Description	109
6.40.2 Property Documentation	109
6.40.2.1 AuthenticationUri	109
6.40.2.2 SharedKey	109
6.40.2.3 VerificationCode	109
6.41 DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWModel Class Reference	110

6.41.1 Detailed Description	110
6.41.2 Property Documentation	110
6.41.2.1 IsRecoveryCode	110
6.41.2.2 TwoFactorType	110
6.41.2.3 VerificationCode	111
6.42 DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel Class Reference	111
6.42.1 Detailed Description	111
6.42.2 Property Documentation	111
6.42.2.1 TwoFactorType	111
6.43 DotNetIdentity.Models.ViewModels.UpdateProfileViewModel Class Reference	112
6.43.1 Detailed Description	112
6.43.2 Property Documentation	112
6.43.2.1 BirthDay	112
6.43.2.2 Email	113
6.43.2.3 Gender	113
6.43.2.4 PhoneNumber	113
6.43.2.5 ProfilePicture	113
6.43.2.6 UserName	114
6.44 DotNetIdentity.Models.ViewModels.UpsertRoleViewModel Class Reference	114
6.44.1 Detailed Description	114
6.44.2 Property Documentation	114
6.44.2.1 Id	114
6.44.2.2 Name	115
6.45 DotNetIdentity.Controllers.UserController Class Reference	115
6.45.1 Detailed Description	116
6.45.2 Constructor & Destructor Documentation	116
6.45.2.1 UserController()	117
6.45.3 Member Function Documentation	117
6.45.3.1 ChangePassword() [1/2]	117
6.45.3.2 ChangePassword() [2/2]	117
6.45.3.3 ConfirmEmail()	118
6.45.3.4 debugUserInfo()	119
6.45.3.5 EnforceTwoFactorAuthenticator() [1/2]	119
6.45.3.6 EnforceTwoFactorAuthenticator() [2/2]	120
6.45.3.7 EnforceTwoFactorResult()	120
6.45.3.8 ForgotPassword() [1/2]	121
6.45.3.9 ForgotPassword() [2/2]	121
6.45.3.10 Login() [1/2]	122
6.45.3.11 Login() [2/2]	123
6.45.3.12 Logout()	123
6.45.3.13 ping()	124
6.45.3.14 Profile() [1/2]	124

6.45.3.15 Profile() [2/2]	124
6.45.3.16 Register() [1/2]	125
6.45.3.17 Register() [2/2]	126
6.45.3.18 ResetPassword() [1/2]	127
6.45.3.19 ResetPassword() [2/2]	127
6.45.3.20 TwoFactorAuthenticator() [1/2]	128
6.45.3.21 TwoFactorAuthenticator() [2/2]	128
6.45.3.22 TwoFactorLogin() [1/2]	129
6.45.3.23 TwoFactorLogin() [2/2]	130
6.45.3.24 TwoFactorType() [1/2]	131
6.45.3.25 TwoFactorType() [2/2]	131
6.46 DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper Class Reference	132
6.46.1 Detailed Description	132
6.46.2 Constructor & Destructor Documentation	132
6.46.2.1 UserProfileImageTagHelper()	132
6.46.3 Member Function Documentation	133
6.46.3.1 ProcessAsync()	133
6.46.4 Property Documentation	133
6.46.4.1 UserId	134
6.47 DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper Class Reference	134
6.47.1 Detailed Description	134
6.47.2 Constructor & Destructor Documentation	135
6.47.2.1 UserRolesTagHelper()	135
6.47.3 Member Function Documentation	135
6.47.3.1 ProcessAsync()	135
6.47.4 Property Documentation	136
6.47.4.1 UserId	136
6.48 DotNetIdentity.IdentitySettings.Validators.UserValidator Class Reference	136
6.48.1 Detailed Description	137
6.48.2 Member Function Documentation	137
6.48.2.1 ValidateAsync()	137

Chapter 1

Todo List

Class [DotNetIdentity.Controllers.ErrorsController](#)

add translation

Class [DotNetIdentity.Controllers.SettingsController](#)

add translation

Member [DotNetIdentity.Controllers.UserController.Login](#) ([SignInViewModel](#) viewModel)

Implement Ldap Login

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

DotNetIdentity	11
DotNetIdentity.Controllers	11
DotNetIdentity.Data	11
DotNetIdentity.Helpers	12
DotNetIdentity.Helpers.TagHelpers	12
DotNetIdentity.IdentitySettings	12
DotNetIdentity.IdentitySettings.Requirements	12
DotNetIdentity.IdentitySettings.Validators	13
DotNetIdentity.Models	13
DotNetIdentity.Models.BusinessModels	15
DotNetIdentity.Models.DataModels	15
DotNetIdentity.Models.Identity	16
DotNetIdentity.Models.ViewModels	16
DotNetIdentity.Services	17
DotNetIdentity.Services.SettingsService	17

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

DotNetIdentity.Models.DataModels.ApplicationSettings	37
DotNetIdentity.Models.DataModels.AppLogs	38
AppSettingsBase	
DotNetIdentity.Models.BusinessModels.LdapSettings	76
DotNetIdentity.Services.SettingsService.AppSettingsBase	42
DotNetIdentity.Models.BusinessModels.GlobalSettings	71
DotNetIdentity.Models.BusinessModels.MailSettings	79
DotNetIdentity.Models.ViewModels.AssignRoleViewModel	48
AuthorizationHandler	
DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler	69
DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler	81
DotNetIdentity.Models.ViewModels.ChangePasswordViewModel	50
Controller	
DotNetIdentity.Controllers.AdminController	19
DotNetIdentity.Controllers.ClaimBasedController	51
DotNetIdentity.Controllers.CultureController	55
DotNetIdentity.Controllers.ErrorsController	65
DotNetIdentity.Controllers.HomeController	73
DotNetIdentity.Controllers.RoleBasedController	91
DotNetIdentity.Controllers.SettingsController	94
DotNetIdentity.Controllers.UserController	115
DotNetIdentity.Models.ViewModels.EditUserViewModel	57
DotNetIdentity.Helpers.EmailHelper	60
DotNetIdentity.Models.BusinessModels.EmailModel	61
DotNetIdentity.Models.ViewModels.ErrorViewModel	67
DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel	68
IAuthorizationRequirement	
DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement	70
DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement	83
IClaimsTransformation	
DotNetIdentity.IdentitySettings.ClaimsTransformation	53
IdentityDbContext	
DotNetIdentity.Data.AppDbContext	34
IdentityErrorDescriber	
DotNetIdentity.IdentitySettings.ErrorDescriber	63

IdentityRole	
DotNetIdentity.Models.Identity.AppRole	41
IdentityUser	
DotNetIdentity.Models.Identity.AppUser	44
IPasswordValidator	
DotNetIdentity.IdentitySettings.Validators.PasswordValidator	88
DotNetIdentity.Services.SettingsService.ISettingsService	74
DotNetIdentity.Services.SettingsService.SettingsService	98
IUserValidator	
DotNetIdentity.IdentitySettings.Validators.UserValidator	136
DotNetIdentity.Models.ViewModels.NewUserModel	84
DotNetIdentity.Models.ViewModels.ResetPasswordViewModel	89
DotNetIdentity.Models.DataModels.SessionCache	92
DotNetIdentity.Models.ViewModels.SignInViewModel	101
DotNetIdentity.Models.ViewModels.SignUpViewModel	102
DotNetIdentity.Models.ViewModels.StatusCodesModel	105
TagHelper	
DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper	132
DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper	134
DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService	107
DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel	108
DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWModel	110
DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel	111
DotNetIdentity.Models.ViewModels.UpdateProfileViewModel	112
DotNetIdentity.Models.ViewModels.UpsertRoleViewModel	114

Chapter 4

Class Index

4.1 Class List

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

DotNetIdentity.Controllers.AdminController	
Controller Class for Admin views	19
DotNetIdentity.Data.AppDbContext	
Application Database context class	34
DotNetIdentity.Models.DataModels.ApplicationSettings	
class define an Settings object	37
DotNetIdentity.Models.DataModels.AppLogs	
class to define ann Log object	38
DotNetIdentity.Models.Identity.AppRole	
class to define a role object	41
DotNetIdentity.Services.SettingsService.AppSettingsBase	
base class for Settings classes	42
DotNetIdentity.Models.Identity.AppUser	
class to define a user object	44
DotNetIdentity.Models.ViewModels.AssignRoleViewModel	
model for UserEditView	48
DotNetIdentity.Models.ViewModels.ChangePasswordViewModel	
model for ChangePassword view	50
DotNetIdentity.Controllers.ClaimBasedController	
Controller to test ClaimBased Authorization	51
DotNetIdentity.IdentitySettings.ClaimsTransformation	
class to define own claims	53
DotNetIdentity.Controllers.CultureController	
controller for culture settings	55
DotNetIdentity.Models.ViewModels.EditUserViewModel	
model for EditUser view	57
DotNetIdentity.Helpers.EmailHelper	
class for email methods	60
DotNetIdentity.Models.BusinessModels.EmailModel	
class to define a email	61
DotNetIdentity.IdentitySettings.ErrorDescriber	
class to override IdentityErrorDescriber IdentotyError messages	63
DotNetIdentity.Controllers.ErrorsController	
Controller for error views	65
DotNetIdentity.Models.ViewModels.ErrorViewModel	
model for displaying errors	67

DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel	
model for ForgotPassword view	68
DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler	
class to define own Identity requirements type of FreeTrialExpireRequirement	69
DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement	
class to define own Identity requirements type of IAuthorizationRequirement	70
DotNetIdentity.Models.BusinessModels.GlobalSettings	
class to define a global settings object	71
DotNetIdentity.Controllers.HomeController	
Controller for Home views	73
DotNetIdentity.Services.SettingsService.ISettingsService	
Interface class for SettingsService	74
DotNetIdentity.Models.BusinessModels.LdapSettings	
class to define ldap settings	76
DotNetIdentity.Models.BusinessModels.MailSettings	
class to define a mail settings object	79
DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler	
class implementing AuthorizationHandler of type MinimumAgeRequirement	81
DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement	
class to define own Identity requirements type of IAuthorizationRequirement	83
DotNetIdentity.Models.ViewModels.NewUserModel	
model for NewUser view	84
DotNetIdentity.IdentitySettings.Validators.PasswordValidator	
class to validate a password	88
DotNetIdentity.Models.ViewModels.ResetPasswordViewModel	
model for ResetPassword view	89
DotNetIdentity.Controllers.RoleBasedController	
Controller to test RoleBased Authorization	91
DotNetIdentity.Models.DataModels.SessionCache	
class to define a SessionCache object	92
DotNetIdentity.Controllers.SettingsController	
controller class for settings views	94
DotNetIdentity.Services.SettingsService.SettingsService	
class for settings service	98
DotNetIdentity.Models.ViewModels.SignInViewModel	
model for Login view	101
DotNetIdentity.Models.ViewModels.SignUpViewModel	
model for Register view	102
DotNetIdentity.Models.ViewModels.StatusCodesModel	
model for Error Statuscodes	105
DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService	
class providing 2fa auth services	107
DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel	
model for TwoFactorAuthenticator view	108
DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWModel	
the model for TwoFactorLogin view	110
DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel	
model for TwoFactorType view	111
DotNetIdentity.Models.ViewModels.UpdateProfileViewModel	
model for UpdateProfile view	112
DotNetIdentity.Models.ViewModels.UpsertRoleViewModel	
model for the UpsertRole view	114
DotNetIdentity.Controllers.UserController	
Controller Class for User views	115
DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper	
html taghelper class	132
DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper	
html tag helper class	134

DotNetIdentity.IdentitySettings.Validators.UserValidator	
class to validate IdentityUser	136

Chapter 5

Namespace Documentation

5.1 DotNetIdentity Namespace Reference

5.2 DotNetIdentity.Controllers Namespace Reference

Classes

- class [AdminController](#)
Controller Class for Admin views
- class [ClaimBasedController](#)
Controller to test ClaimBased Authorization
- class [CultureController](#)
controller for culture settings
- class [ErrorsController](#)
Controller for error views
- class [HomeController](#)
Controller for Home views
- class [RoleBasedController](#)
Controller to test RoleBased Authorization
- class [SettingsController](#)
controller class for settings views
- class [UserController](#)
Controller Class for User views

5.3 DotNetIdentity.Data Namespace Reference

Classes

- class [AppDbContext](#)
Application Database context class

5.4 DotNetIdentity.Helpers Namespace Reference

Classes

- class [EmailHelper](#)
class for email methods

5.5 DotNetIdentity.Helpers.TagHelpers Namespace Reference

Classes

- class [UserProfileImageTagHelper](#)
html taghelper class
- class [UserRolesTagHelper](#)
html tag helper class

5.6 DotNetIdentity.IdentitySettings Namespace Reference

Classes

- class [ClaimsTransformation](#)
class to define own claims
- class [ErrorDescriber](#)
class to override IdentityErrorDescriber IdentityError messages
- class [TwoFactorAuthenticationService](#)
class providing 2fa auth services

5.7 DotNetIdentity.IdentitySettings.Requirements Namespace Reference

Classes

- class [FreeTrialExpireHandler](#)
class to define own Identity requirements type of [FreeTrialExpireRequirement](#)
- class [FreeTrialExpireRequirement](#)
class to define own Identity requirements type of [IAuthorizationRequirement](#)
- class [MinimumAgeHandler](#)
class implementing AuthorizationHandler of type [MinimumAgeRequirement](#)
- class [MinimumAgeRequirement](#)
class to define own Identity requirements type of [IAuthorizationRequirement](#)

5.8 DotNetIdentity.IdentitySettings.Validators Namespace Reference

Classes

- class [PasswordValidator](#)
class to validate a password
- class [UserValidator](#)
class to validate IdentityUser

5.9 DotNetIdentity.Models Namespace Reference

Enumerations

- enum [Gender](#) { [Unknown](#) , [Male](#) , **Female** }
Gender enum
- enum [Department](#) { [Software](#) , [HR](#) , [Accounting](#) , **Management** }
Department enum
- enum [TwoFactorType](#) { [None](#) , [Email](#) , **Authenticator** }
TwoFactorType enum

5.9.1 Enumeration Type Documentation

5.9.1.1 Department

```
enum DotNetIdentity.Models.Department
```

Department enum

Enumerator

Software	summary>Department Softwaresummary>Department HR
HR	summary>Department Accounting
Accounting	summary>Department Management

Definition at line 16 of file [Enums.cs](#).

```
00016      {  
00018          Software,  
00020          HR,  
00022          Accounting,  
00024          Management }  
00024      }
```

5.9.1.2 Gender

```
enum DotNetIdentity.Models.Gender
```

Gender enum

Enumerator

Unknown	summary>gender unknownsummary>gender male
Male	summary>gender female

Definition at line 6 of file [Enums.cs](#).

```
00006      {
00008          Unknown,
00010          Male,
00012          Female }
```

5.9.1.3 TwoFactorType

```
enum DotNetIdentity.Models.TwoFactorType
```

TwoFactorType enum

Enumerator

None	summary>2fa type nonesummary>2fa type email
Email	summary>2fa type Authenticator app

Definition at line 28 of file [Enums.cs](#).

```
00028      {
00030          None,
00032          Email,
00034          Authenticator
00035      }
```

5.10 DotNetIdentity.Models.BusinessModels Namespace Reference

Classes

- class [EmailModel](#)
class to define a email
- class [GlobalSettings](#)
class to define a global settings object
- class [LdapSettings](#)
class to define ldap settings
- class [MailSettings](#)
class to define a ail settings object

5.11 DotNetIdentity.Models.DataModels Namespace Reference

Classes

- class [ApplicationSettings](#)
class define an Settings object
- class [AppLogs](#)
class to define ann Log object
- class [SessionCache](#)
class to define a [SessionCache](#) object

5.12 DotNetIdentity.Models.Identity Namespace Reference

Classes

- class [AppRole](#)
class to define a role object
- class [AppUser](#)
class to define a user object

5.13 DotNetIdentity.Models.ViewModels Namespace Reference

Classes

- class [AssignRoleViewModel](#)
model for UserEditView
- class [ChangePasswordViewModel](#)
model for ChangePassword view
- class [EditUserViewModel](#)
model for EditUser view
- class [ErrorViewModel](#)
model for displaying errors
- class [ForgotPasswordViewModel](#)
model for ForgotPassword view
- class [NewUserModel](#)
model for NewUser view
- class [ResetPasswordViewModel](#)
model for ResetPassword view
- class [SignInViewModel](#)
model for Login view
- class [SignUpViewModel](#)
model for Register view
- class [StatusCodesModel](#)
model for Error Statuscodes
- class [TwoFactorAuthenticatorViewModel](#)
model for TwoFactorAuthenticator view
- class [TwoFactorLoginViewWModel](#)
the model for TwoFactorLogin view
- class [TwoFactorTypeViewModel](#)
model for TwoFactorType view
- class [UpdateProfileViewModel](#)
model for UpdateProfile view
- class [UpsertRoleViewModel](#)
model for the UpsertRole view

5.14 DotNetIdentity.Services Namespace Reference

5.15 DotNetIdentity.Services.SettingsService Namespace Reference

Classes

- class [AppSettingsBase](#)
base class for Settings classes
- interface [ISettingsService](#)
Interface class for [SettingsService](#)
- class [SettingsService](#)
class for settings service

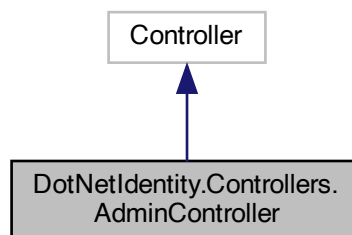
Chapter 6

Class Documentation

6.1 DotNetIdentity.Controllers.AdminController Class Reference

Controller Class for Admin views

Inheritance diagram for DotNetIdentity.Controllers.AdminController:



Public Member Functions

- [AdminController](#) (IStringLocalizer< [UserController](#) > localizer, [AppDbContext](#) db, ILogger< [AdminController](#) > log, IConfiguration conf, UserManager< [AppUser](#) > userManager, RoleManager< [AppRole](#) > roleManager, SignInManager< [AppUser](#) > signInManager)
Controller Class constructor
- IActionResult [Index](#) ()
Controller Action for Index
- IActionResult [SystemLogs](#) ()
Controller Action for SystemLogs
- IActionResult [AuditLogs](#) ()
Controller Action for AuditLogs
- IActionResult [ErrorLogs](#) ()
Controller Action for ErrorLogs

- async Task< IActionResult > [Users](#) ()
Controller Action for Users
- async Task< IActionResult > [Roles](#) ()
Controller Action for Roles
- IActionResult [GetErrorLogs](#) (DataRequest request)
Controller Post Method to fetch ErrorLogs
- IActionResult [GetAppLogs](#) (DataRequest request)
Controller Post Method to fetch Applicationlogs
- IActionResult [GetAuditLogs](#) (DataRequest request)
Controller Post Method to fetch AuditLogs
- JsonResult [GetUsers](#) (DataRequest request)
Controller Post Method to fetch Users
- async Task< IActionResult > [NewUser](#) ()
Controller Action for NewUser view
- async Task< ActionResult > [NewUser](#) ([NewUserModel](#) viewModel)
controller Post Method to save a new User
- async Task< IActionResult > [UpsertRole](#) (string? id)
Controller Action for view UpsertRole
- async Task< ActionResult > [DisableUser](#) (string UserId)
Controller GET Method to disable a user account
- async Task< ActionResult > [EnableUser](#) (string UserId)
Controller GET Method to enable a user account
- async Task< ActionResult > [EnableLdapLogin](#) (string UserId)
Controller GET Method to enable ldap login for a user account
- async Task< ActionResult > [DisableLdapLogin](#) (string UserId)
Controller GET Method to disable ldap login for a user account
- async Task< ActionResult > [EnableMfaForce](#) (string UserId)
Controller GET Method to enable mfa enforcement for a user account
- async Task< ActionResult > [DisableMfaForce](#) (string UserId)
Controller GET Method to disable mfa enforcement for a user account
- async Task< IActionResult > [UpsertRole](#) ([UpsertRoleViewModel](#) viewModel)
Controller POST Method to enable a user account
- async Task< IActionResult > [DeleteUser](#) (string id)
Controller POST Method to delete a user account
- async Task< IActionResult > [DeleteRole](#) (string id)
Controller POST Method to enable a role
- async Task< IActionResult > [EditUser](#) (string id)
Controller action to serve the EditUserView
- async Task< IActionResult > [EditUser](#) ([EditUserViewModel](#) viewModel)
Controller POST Method to save user changes

6.1.1 Detailed Description

Controller Class for Admin views

Definition at line 24 of file [AdminController.cs](#).

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AdminController()

```
DotNetIdentity.Controllers.AdminController.AdminController (
    IStringLocalizer< UserController > localizer,
    AppDbContext db,
    ILogger< AdminController > log,
    IConfiguration conf,
    UserManager< AppUser > userManager,
    RoleManager< AppRole > roleManager,
    SignInManager< AppUser > signInManager ) [inline]
```

Controller Class constructor

Parameters

<i>localizer</i>	type IStringLocalizer
<i>db</i>	type DotNetIdentity.Data.AppDbContext
<i>log</i>	type Microsoft.Extensions.Logging.ILogger
<i>conf</i>	type Microsoft.Extensions.Configuration.IConfiguration
<i>userManager</i>	type Microsoft.AspNetCore.Identity.UserManager
<i>roleManager</i>	type Microsoft.AspNetCore.Identity.RoleManager
<i>signInManager</i>	type Microsoft.AspNetCore.Identity.SignInManager

Definition at line 65 of file [AdminController.cs](#).

```
00066     {
00067         _userManager = userManager;
00068         _roleManager = roleManager;
00069         _signInManager = signInManager;
00070         _configuration = conf;
00071         _logger = log;
00072         _context = db;
00073         _localizer = localizer;
00074     }
```

6.1.3 Member Function Documentation

6.1.3.1 AuditLogs()

```
IActionResult DotNetIdentity.Controllers.AdminController.AuditLogs ( )
```

Controller Action for AuditLogs

Returns

View of type DotnetIdentity.Views.Admin.AuditLogs

6.1.3.2 DeleteRole()

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.DeleteRole (
    string id ) [inline]
```

Controller POST Method to enable a role

Parameters

<i>id</i>	id of the role
-----------	----------------

Returns

redirect to controller action Roles

Definition at line 423 of file [AdminController.cs](#).

```

00424     {
00425         var role = await _roleManager.FindByIdAsync(id);
00426         if (role != null)
00427         {
00428             var result = await _roleManager.DeleteAsync(role);
00429             if (!result.Succeeded)
00430             {
00431                 result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
f.Description));
00432             }
00433         }
00434         else
00435         {
00436             ModelState.AddModelError(string.Empty, _localizer["3"]);
00437         }
00438         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " deleted role " + id);
00439         return RedirectToAction("Roles");
00440     }

```

6.1.3.3 DeleteUser()

```

async Task< IActionResult > DotNetIdentity.Controllers.AdminController.DeleteUser (
    string id ) [inline]

```

Controller POST Method to delete a user account

Parameters

<i>id</i>	the id of the user
-----------	--------------------

Returns

redirect to controller action Users

Definition at line 403 of file [AdminController.cs](#).

```

00403     {
00404         var user = await _userManager.FindByIdAsync(id);
00405         if (user != null) {
00406             var result = await _userManager.DeleteAsync(user);
00407             if (!result.Succeeded) {
00408                 result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
f.Description));
00409             }
00410             } else {
00411                 ModelState.AddModelError(string.Empty, _localizer["2"]);
00412             }
00413             _logger.LogInformation("AUDIT: " + User.Identity!.Name + " deleted user " + id);
00414             return RedirectToAction("Users");
00415         }

```

6.1.3.4 DisableLdapLogin()

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.DisableLdapLogin (
    string UserId ) [inline]
```

Controller GET Method to disable ldap login for a user account

Parameters

<i>UserId</i>	the id of the user
---------------	--------------------

Returns

json object (true / false)

Definition at line 314 of file [AdminController.cs](#).

```
00314 {
00315     var user = await _userManager.FindByIdAsync(UserId);
00316     if(user==null) {
00317         return Json(new { success = false });
00318     }
00319     user.IsLdapLogin = false;
00320     await _userManager.UpdateAsync(user);
00321     _logger.LogInformation("AUDIT: " + User.Identity!.Name + " disabled ldap login for user "
+ user.UserName);
00322     return Json(new { success = true });
00323 }
```

6.1.3.5 DisableMfaForce()

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.DisableMfaForce (
    string UserId ) [inline]
```

Controller GET Method to disable mfa enforcement for a user account

Parameters

<i>UserId</i>	the id of the user
---------------	--------------------

Returns

json object (true / false)

Definition at line 348 of file [AdminController.cs](#).

```
00348 {
00349     var user = await _userManager.FindByIdAsync(UserId);
00350     if(user==null) {
00351         return Json(new { success = false });
00352     }
00353     user.IsMfaForce = false;
00354     await _userManager.UpdateAsync(user);
00355     _logger.LogInformation("AUDIT: " + User.Identity!.Name + " disabled mfa enforce for user "
+ user.UserName);
00356     return Json(new { success = true });
00357 }
```

6.1.3.6 DisableUser()

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.DisableUser (
    string UserId ) [inline]
```

Controller GET Method to disable a user account

Parameters

<i>UserId</i>	the id of the user
---------------	--------------------

Returns

json object (true / false)

Definition at line 263 of file [AdminController.cs](#).

```
00263     {
00264         var user = await _userManager.FindByIdAsync(UserId);
00265         if(user==null) {
00266             return Json(new { success = false });
00267         }
00268         user.IsEnabled = false;
00269         await _userManager.UpdateAsync(user);
00270         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " disabled user " +
user.UserName);
00271         return Json(new {success = true});
00272     }
```

6.1.3.7 EditUser() [1/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.EditUser (
    EditUserViewModel viewModel ) [inline]
```

Controller POST Method to save user changes

Parameters

<i>viewModel</i>	a model of type Dotnetidentity.Models.ViewModels.EditUserViewModel
------------------	--

Returns

view of type Dotnetidentity.Views.Admin.EditUser

Definition at line 480 of file [AdminController.cs](#).

```
00481     {
00482         if (ModelState.IsValid)
00483         {
00484             var user = await _userManager.FindByIdAsync(viewModel.Id);
00485             if (user != null)
00486             {
00487                 if (user.PhoneNumber != viewModel.PhoneNumber && _userManager.Users.Any(a =>
a.PhoneNumber == viewModel.PhoneNumber))
00488                 {
00489                     ModelState.AddModelError(string.Empty, _localizer["4"]);
00490                 }
00491                 else
```

```

00492         {
00493             user.FirstName = viewModel.FirstName;
00494             user.LastName = viewModel.LastName;
00495             user.UserName = viewModel.UserName;
00496             user.Email = viewModel.Email;
00497             user.PhoneNumber = viewModel.PhoneNumber;
00498             user.Gender = viewModel.Gender;
00499             user.Department = viewModel.Department.ToString();
00500
00501             var result = await _userManager.UpdateAsync(user);
00502             if (result.Succeeded)
00503             {
00504                 // Roles
00505                 foreach (var item in viewModel.Roles)
00506                 {
00507                     if (item.IsAssigned)
00508                     {
00509                         await _userManager.AddToRoleAsync(user, item.RoleName);
00510                     }
00511                     else
00512                     {
00513                         await _userManager.RemoveFromRoleAsync(user, item.RoleName);
00514                     }
00515                 }
00516
00517                 // roles combined
00518                 var rls = await _userManager.GetRolesAsync(user);
00519                 if (rls != null)
00520                 {
00521                     var rlsComb = "";
00522                     if (rls.Count() > 0 && rls.Count() <= 1)
00523                     {
00524                         rlsComb = rls[0].ToString();
00525                     }
00526                     else
00527                     {
00528                         for (int i = 0; i < rls.Count; i++)
00529                         {
00530                             if (i + 1 != rls.Count) { rlsComb += rls[i] + ","; }
00531                             else { rlsComb += rls[i]; }
00532                         }
00533                     }
00534                     if (rlsComb.Length > 0)
00535                     {
00536                         user.RolesCombined = rlsComb;
00537                         await _userManager.UpdateAsync(user);
00538                     }
00539                 }
00540
00541                 // Claims
00542                 var userClaims = await _userManager.GetClaimsAsync(user);
00543                 var departmentClaim = userClaims.FirstOrDefault(a => a.Type ==
00544 "Department");
00545                 var claimsToAdd = new Claim("Department",
00546 Enum.GetName(viewModel.Department));
00547                 if (departmentClaim != null)
00548                 {
00549                     await _userManager.ReplaceClaimAsync(user, departmentClaim,
00550 claimsToAdd);
00551                 }
00552                 else
00553                 {
00554                     await _userManager.AddClaimAsync(user, claimsToAdd);
00555                 }
00556                 _logger.LogInformation("AUDIT: " + User.Identity!.Name + " modified user "
00557 + user.UserName);
00558                 return RedirectToAction("Users");
00559             }
00560             result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
00561 f.Description));
00562         }
00563     }
00564     else
00565     {
00566         ModelState.AddModelError(string.Empty, _localizer["2"]);
00567     }
00568 }
00569 }
00570 }
00571 }
00572 }
00573 }
00574 }
00575 }
00576 }
00577 }
00578 }
00579 }
00580 }
00581 }
00582 }
00583 }
00584 }
00585 }
00586 }
00587 }
00588 }
00589 }
00590 }
00591 }
00592 }
00593 }
00594 }
00595 }
00596 }
00597 }
00598 }
00599 }
00600 }
00601 }
00602 }
00603 }
00604 }
00605 }
00606 }
00607 }
00608 }
00609 }
00610 }
00611 }
00612 }
00613 }
00614 }
00615 }
00616 }
00617 }
00618 }
00619 }
00620 }
00621 }
00622 }
00623 }
00624 }
00625 }
00626 }
00627 }
00628 }
00629 }
00630 }
00631 }
00632 }
00633 }
00634 }
00635 }
00636 }
00637 }
00638 }
00639 }
00640 }
00641 }
00642 }
00643 }
00644 }
00645 }
00646 }
00647 }
00648 }
00649 }
00650 }
00651 }
00652 }
00653 }
00654 }
00655 }
00656 }
00657 }
00658 }
00659 }
00660 }
00661 }
00662 }
00663 }
00664 }
00665 }
00666 }
00667 }
00668 }
00669 }
00670 }
00671 }
00672 }
00673 }
00674 }
00675 }
00676 }
00677 }
00678 }
00679 }
00680 }
00681 }
00682 }
00683 }
00684 }
00685 }
00686 }
00687 }
00688 }
00689 }
00690 }
00691 }
00692 }
00693 }
00694 }
00695 }
00696 }
00697 }
00698 }
00699 }
00700 }
00701 }
00702 }
00703 }
00704 }
00705 }
00706 }
00707 }
00708 }
00709 }
00710 }
00711 }
00712 }
00713 }
00714 }
00715 }
00716 }
00717 }
00718 }
00719 }
00720 }
00721 }
00722 }
00723 }
00724 }
00725 }
00726 }
00727 }
00728 }
00729 }
00730 }
00731 }
00732 }
00733 }
00734 }
00735 }
00736 }
00737 }
00738 }
00739 }
00740 }
00741 }
00742 }
00743 }
00744 }
00745 }
00746 }
00747 }
00748 }
00749 }
00750 }
00751 }
00752 }
00753 }
00754 }
00755 }
00756 }
00757 }
00758 }
00759 }
00760 }
00761 }
00762 }
00763 }
00764 }
00765 }
00766 }
00767 }
00768 }
00769 }
00770 }
00771 }
00772 }
00773 }
00774 }
00775 }
00776 }
00777 }
00778 }
00779 }
00780 }
00781 }
00782 }
00783 }
00784 }
00785 }
00786 }
00787 }
00788 }
00789 }
00790 }
00791 }
00792 }
00793 }
00794 }
00795 }
00796 }
00797 }
00798 }
00799 }
00800 }
00801 }
00802 }
00803 }
00804 }
00805 }
00806 }
00807 }
00808 }
00809 }
00810 }
00811 }
00812 }
00813 }
00814 }
00815 }
00816 }
00817 }
00818 }
00819 }
00820 }
00821 }
00822 }
00823 }
00824 }
00825 }
00826 }
00827 }
00828 }
00829 }
00830 }
00831 }
00832 }
00833 }
00834 }
00835 }
00836 }
00837 }
00838 }
00839 }
00840 }
00841 }
00842 }
00843 }
00844 }
00845 }
00846 }
00847 }
00848 }
00849 }
00850 }
00851 }
00852 }
00853 }
00854 }
00855 }
00856 }
00857 }
00858 }
00859 }
00860 }
00861 }
00862 }
00863 }
00864 }
00865 }
00866 }
00867 }
00868 }
00869 }
00870 }
00871 }
00872 }
00873 }
00874 }
00875 }
00876 }
00877 }
00878 }
00879 }
00880 }
00881 }
00882 }
00883 }
00884 }
00885 }
00886 }
00887 }
00888 }
00889 }
00890 }
00891 }
00892 }
00893 }
00894 }
00895 }
00896 }
00897 }
00898 }
00899 }
00900 }
00901 }
00902 }
00903 }
00904 }
00905 }
00906 }
00907 }
00908 }
00909 }
00910 }
00911 }
00912 }
00913 }
00914 }
00915 }
00916 }
00917 }
00918 }
00919 }
00920 }
00921 }
00922 }
00923 }
00924 }
00925 }
00926 }
00927 }
00928 }
00929 }
00930 }
00931 }
00932 }
00933 }
00934 }
00935 }
00936 }
00937 }
00938 }
00939 }
00940 }
00941 }
00942 }
00943 }
00944 }
00945 }
00946 }
00947 }
00948 }
00949 }
00950 }
00951 }
00952 }
00953 }
00954 }
00955 }
00956 }
00957 }
00958 }
00959 }
00960 }
00961 }
00962 }
00963 }
00964 }
00965 }
00966 }
00967 }
00968 }
00969 }
00970 }
00971 }
00972 }
00973 }
00974 }
00975 }
00976 }
00977 }
00978 }
00979 }
00980 }
00981 }
00982 }
00983 }
00984 }
00985 }
00986 }
00987 }
00988 }
00989 }
00990 }
00991 }
00992 }
00993 }
00994 }
00995 }
00996 }
00997 }
00998 }
00999 }

```

References [DotNetIdentity.Models.ViewModels.EditUserViewModel.Department](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.FirstName](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.Gender](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.Id](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.LastName](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.PhoneNumber](#), [DotNetIdentity.Models.ViewModels.EditUserViewModel.Roles](#)

and [DotNetIdentity.Models.ViewModels.EditUserViewModel.UserName](#).

6.1.3.8 EditUser() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.EditUser (
    string id ) [inline]
```

Controller action to serve the EditUserView

Parameters

<i>id</i>	the id of the user
-----------	--------------------

Returns

view of type [DotNetIdentity.Models.ViewModels.EditUserViewModel](#)

Definition at line 447 of file [AdminController.cs](#).

```
00448     {
00449         var user = await _userManager.FindByIdAsync(id);
00450         if (user == null)
00451         {
00452             return RedirectToAction("Users");
00453         }
00454         var viewModel = user.Adapt<EditUserViewModel>();
00455
00456         var userRoles = await _userManager.GetRolesAsync(user);
00457         viewModel.Roles = await _roleManager.Roles.Select(s => new AssignRoleViewModel
00458         {
00459             RoleId = s.Id,
00460             RoleName = s.Name,
00461             IsAssigned = userRoles.Any(a => a == s.Name)
00462         }).ToListAsync();
00463
00464         var userClaims = await _userManager.GetClaimsAsync(user);
00465         var departmentClaim = userClaims.FirstOrDefault(f => f.Type == "Department");
00466         if (departmentClaim != null)
00467         {
00468             viewModel.Department = Enum.Parse<Department>(departmentClaim.Value);
00469         }
00470
00471         return View(viewModel);
00472     }
```

References [DotNetIdentity.Models.ViewModels.EditUserViewModel.Roles](#).

6.1.3.9 EnableLdapLogin()

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.EnableLdapLogin (
    string UserId ) [inline]
```

Controller GET Method to enable ldap login for a user account

Parameters

<i>User↔ Id</i>	the id of the user
---------------------	--------------------

Returns

json object (true / false)

Definition at line 297 of file [AdminController.cs](#).

```

00297                                     {
00298         var user = await _userManager.FindByIdAsync(UserId);
00299         if(user==null) {
00300             return Json(new { success = false });
00301         }
00302         user.IsLdapLogin = true;
00303         await _userManager.UpdateAsync(user);
00304         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " enabled LDAP login for user " +
user.UserName);
00305         return Json(new {success = true});
00306     }

```

6.1.3.10 EnableMfaForce()

```

async Task< ActionResult > DotNetIdentity.Controllers.AdminController.EnableMfaForce (
    string UserId ) [inline]

```

Controller GET Method to enable mfa enforcement for a user account

Parameters

<i>User↔ Id</i>	the id of the user
---------------------	--------------------

Returns

json object (true / false)

Definition at line 331 of file [AdminController.cs](#).

```

00331                                     {
00332         var user = await _userManager.FindByIdAsync(UserId);
00333         if(user==null) {
00334             return Json(new { success = false });
00335         }
00336         user.IsMfaForce = true;
00337         await _userManager.UpdateAsync(user);
00338         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " enabled mfa enforce for user "
+ user.UserName);
00339         return Json(new {success = true});
00340     }

```

6.1.3.11 EnableUser()

```

async Task< ActionResult > DotNetIdentity.Controllers.AdminController.EnableUser (
    string UserId ) [inline]

```

Controller GET Method to enable a user account

Parameters

<i>User</i> ↔ <i>Id</i>	the id of the user
----------------------------	--------------------

Returns

json object (true / false)

Definition at line 280 of file [AdminController.cs](#).

```

00280
00281         var user = await _userManager.FindByIdAsync(UserId);
00282         if(user==null) {
00283             return Json(new { success = false });
00284         }
00285         user.IsEnabled = true;
00286         await _userManager.UpdateAsync(user);
00287         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " enabled user " +
user.UserName);
00288         return Json(new {success = true});
00289     }

```

6.1.3.12 ErrorLogs()

`ActionResult DotNetIdentity.Controllers.AdminController.ErrorLogs ()`

Controller Action for ErrorLogs

Returns

View of type DotnetIdentity.Views.Admin.Errororgs

6.1.3.13 GetAppLogs()

`ActionResult DotNetIdentity.Controllers.AdminController.GetAppLogs (
DataRequest request) [inline]`

Controller Post Method to fetch Applicationlogs

Parameters

<i>request</i>	type DatatablesJs.Data.DataRequest
----------------	------------------------------------

Returns

Jason-Array of DatatablesJs.Data.DataResult

Definition at line 125 of file [AdminController.cs](#).

```

00126     {

```



```
00127         DataRow<AppLogs> result =  
_context.AppLogs!.Where(l=>l.Message.ToLower().StartsWith("audit")==false && Convert.ToInt32(l.Level)  
< 4).ToDataRow(request);  
00128         return Json(result);  
00129     }
```

6.1.3.14 GetAuditLogs()

```
IActionResult DotNetIdentity.Controllers.AdminController.GetAuditLogs (  
    DataRow request ) [inline]
```

Controller Post Method to fetch AuditLogs

Parameters

<i>request</i>	type DatatablesJs.Data.DataRequest
----------------	------------------------------------

Returns

Jason-Array of DatatablesJs.Data.DataResult

Definition at line 137 of file [AdminController.cs](#).

```
00138     {  
00139         DataRow<AppLogs> result =  
_context.AppLogs!.Where(l=>l.Message.ToLower().StartsWith("audit")==true).ToDataRow(request);  
00140         return Json(result);  
00141     }
```

6.1.3.15 GetErrorLogs()

```
IActionResult DotNetIdentity.Controllers.AdminController.GetErrorLogs (  
    DataRow request ) [inline]
```

Controller Post Method to fetch ErrorLogs

Parameters

<i>request</i>	type DatatablesJs.Data.DataRequest
----------------	------------------------------------

Returns

Jason-Array of DatatablesJs.Data.DataResult

Definition at line 113 of file [AdminController.cs](#).

```
00114     {  
00115         DataRow<AppLogs> result =  
_context.AppLogs!.Where(l=>l.Message.ToLower().StartsWith("audit")==false && Convert.ToInt32(l.Level)  
== 4).ToDataRow(request);  
00116         return Json(result);  
00117     }
```

6.1.3.16 GetUsers()

```
JsonResult DotNetIdentity.Controllers.AdminController.GetUsers (
    DataRequest request ) [inline]
```

Controller Post Method to fetch Users

Parameters

<i>request</i>	type DatatablesJs.Data.DataRequest
----------------	------------------------------------

Returns

Jason-Array of DatatablesJs.Data.DataResult

Definition at line 149 of file [AdminController.cs](#).

```
00149                                     {
00150         DataRow<AppUser> result = _userManager.Users.ToDataRow(request);
00151         return Json(result);
00152     }
```

6.1.3.17 Index()

```
ActionResult DotNetIdentity.Controllers.AdminController.Index ( )
```

Controller Action for Index

Returns

View of type DotnetIdentity.Views.Admin.Index

6.1.3.18 NewUser() [1/2]

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.NewUser ( ) [inline]
```

Controller Action for NewUser view

Returns

view of type DotnetIdentity.Views.Admin.NewUser

Definition at line 158 of file [AdminController.cs](#).

```
00158                                     {
00159         var mod = new NewUserModel();
00160         mod.Roles = await _roleManager.Roles.Select(s => new AssignRoleViewModel
00161         {
00162             RoleId = s.Id,
00163             RoleName = s.Name,
00164             IsAssigned = false
00165         }).ToListAsync();
00166         return View(mod);
00167     }
00168 }
```

6.1.3.19 NewUser() [2/2]

```
async Task< ActionResult > DotNetIdentity.Controllers.AdminController.NewUser (
    NewUserModel viewModel ) [inline]
```

controler Post Method to save a new User

Parameters

<i>viewModel</i>	type DotnetIdentity.Models.ViewModels.NewUserModel
------------------	--

Returns

view of type DotnetIdentity.Views.Admin.NewUser

Definition at line 176 of file [AdminController.cs](#).

```

00177     {
00178         if (ModelState.IsValid)
00179         {
00180             var user = new AppUser();
00181             user.IsEnabled = true;
00182             user.IsMfaForce = viewModel.IsMfaForce;
00183             user.IsLdapLogin = viewModel.IsLdapLogin;
00184             user.FirstName = viewModel.FirstName;
00185             user.LastName = viewModel.LastName;
00186             user.Email = viewModel.Email;
00187             user.EmailConfirmed = true;
00188             user.Gender = viewModel.Gender;
00189             user.Department = viewModel.Department.ToString();
00190             user.UserName = viewModel.UserName;
00191
00192             var result = await _userManager.CreateAsync(user, viewModel.Password);
00193             if (result.Succeeded)
00194             {
00195                 // roles
00196                 foreach (var item in viewModel.Roles)
00197                 {
00198                     if (item.IsAssigned)
00199                     {
00200                         await _userManager.AddToRoleAsync(user, item.RoleName);
00201                     }
00202                     else
00203                     {
00204                         await _userManager.RemoveFromRoleAsync(user, item.RoleName);
00205                     }
00206                 }
00207             }
00208
00209             var createdUser = await _userManager.FindByNameAsync(user.UserName);
00210             if (createdUser != null) {
00211                 var rls = await _userManager.GetRolesAsync(createdUser);
00212                 if (rls != null)
00213                 {
00214                     var rlsComb = "";
00215                     if (rls.Count() > 0 && rls.Count() <= 1)
00216                     {
00217                         rlsComb = rls[0].ToString();
00218                     }
00219                     else
00220                     {
00221                         for (int i = 0; i < rls.Count; i++)
00222                         {
00223                             if (i + 1 != rls.Count) { rlsComb += rls[i] + ","; }
00224                             else { rlsComb += rls[i]; }
00225                         }
00226                     }
00227                     if (rlsComb.Length > 0)
00228                     {
00229                         createdUser.RolesCombined = rlsComb;
00230                         await _userManager.UpdateAsync(createdUser);
00231                     }
00232                 }
00233             }
00234
00235             ViewData["message"] = _localizer["1"];
00236             _logger.LogInformation("AUDIT: " + User.Identity!.Name + " created new user " +
00237                 user.UserName);
00238             return View(new NewUserModel());
00239         }

```

References [DotNetIdentity.Models.ViewModels.NewUserModel.Department](#), [DotNetIdentity.Models.ViewModels.NewUserModel.Email](#), [DotNetIdentity.Models.ViewModels.NewUserModel.FirstName](#), [DotNetIdentity.Models.ViewModels.NewUserModel.Gender](#), [DotNetIdentity.Models.ViewModels.NewUserModel.IsLdapLogin](#), [DotNetIdentity.Models.ViewModels.NewUserModel.IsMfaForce](#), [DotNetIdentity.Models.ViewModels.NewUserModel.LastName](#), [DotNetIdentity.Models.ViewModels.NewUserModel.Password](#), [DotNetIdentity.Models.ViewModels.NewUserModel.Roles](#), and [DotNetIdentity.Models.ViewModels.NewUserModel.UserName](#).

6.1.3.20 Roles()

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.Roles ( )
```

Controller Action for Roles

Returns

View of type DotnetIdentity.Views.Admin.Users

6.1.3.21 SystemLogs()

```
IActionResult DotNetIdentity.Controllers.AdminController.SystemLogs ( )
```

Controller Action for SystemLogs

Returns

View of type DotnetIdentity.Views.Admin.SystemLogs

6.1.3.22 UpsertRole() [1/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.UpsertRole (
    string? id ) [inline]
```

Controller Action for view UpsertRole

Parameters

<i>id</i>	
-----------	--

Returns

view of type Dotnetidentity.Views.Admin.UpsertRole

Definition at line 246 of file [AdminController.cs](#).

```
00247     {
00248         if (id != null)
00249         {
00250             var role = await _roleManager.FindByIdAsync(id);
00251             return View(role.Adapt<UpsertRoleViewModel>());
00252         }
00253
00254         return View(new UpsertRoleViewModel());
00255     }
```

6.1.3.23 UpsertRole() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.UpsertRole (
    UpsertRoleViewModel viewModel ) [inline]
```

Controller POST Method to enable a user account

Parameters

<i>viewModel</i>	tmodel of type Dotnetidentity.Models.ViewModels.UpsertRoleViewModel
------------------	---

Returns

redirect to controller action Roles

Definition at line 365 of file [AdminController.cs](#).

```
00366     {
00367         if (ModelState.IsValid)
00368         {
00369             var isUpdate = viewModel.Id != null;
00370
00371             var role = isUpdate ? await _roleManager.FindByIdAsync(viewModel.Id) : new AppRole() {
Name = viewModel.Name, CreatedOn = DateTime.Now };
00372
00373             if (isUpdate)
00374             {
00375                 role.Name = viewModel.Name;
00376             }
00377
00378             var result = isUpdate ? await _roleManager.UpdateAsync(role) : await
_roleManager.CreateAsync(role);
00379             if (result.Succeeded)
00380             {
00381                 if(isUpdate) {
00382                     _logger.LogInformation("AUDIT: " + User.Identity!.Name + " changed name of " +
viewModel.Id + " to " + viewModel.Name);
00383                 } else {
00384                     _logger.LogInformation("AUDIT: " + User.Identity!.Name + " created role " +
viewModel.Name);
00385                 }
00386                 return RedirectToAction("Roles");
00387             }
00388             result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
f.Description));
00389             result.Errors.ToList().ForEach(f => TempData["Error"] += f.Description);
00390         }
00391
00392         //return View(viewModel);
00393         //return View("Roles", await _roleManager.Roles.ToListAsync());
00394         return RedirectToAction("Roles");
00395     }
```

References [DotNetIdentity.Models.ViewModels.UpsertRoleViewModel.Id](#), and [DotNetIdentity.Models.ViewModels.UpsertRoleViewMo](#)

6.1.3.24 Users()

```
async Task< IActionResult > DotNetIdentity.Controllers.AdminController.Users ( )
```

Controller Action for Users

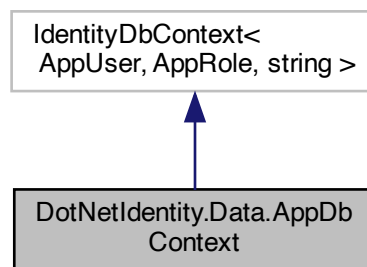
Returns

View of type DotnetIdentity.Views.Admin.Users

6.2 DotNetIdentity.Data.AppDbContext Class Reference

Application Database context class

Inheritance diagram for DotNetIdentity.Data.AppDbContext:



Public Member Functions

- [AppDbContext](#) (`DbContextOptions< AppDbContext > options`)
class constructor

Protected Member Functions

- override void [OnModelCreating](#) (`ModelBuilder builder`)
Overriding OnModelCreation to seed initial data

Properties

- `DbSet< AppLogs >? AppLogs` [get, set]
Property AppLogs
- `DbSet< ApplicationSettings >? AppSettings` [get, set]
property AppSettings
- `DbSet< SessionCache >? AppSessionCache` [get, set]
property AppSessionCache

6.2.1 Detailed Description

Application Database context class

Definition at line 12 of file [AppDbContext.cs](#).

6.2.2 Constructor & Destructor Documentation

6.2.2.1 AppDbContext()

```
DotNetIdentity.Data.AppDbContext.AppDbContext (
    DbContextOptions< AppDbContext > options ) [inline]
```

class constructor

Parameters

<i>options</i>	type DbContextOptions
----------------	-----------------------

Returns

Definition at line 19 of file [AppDbContext.cs](#).

```
00019         : base(options) {
00020     }
```

6.2.3 Member Function Documentation

6.2.3.1 OnModelCreating()

```
override void DotNetIdentity.Data.AppDbContext.OnModelCreating (
    ModelBuilder builder ) [inline], [protected]
```

Overriding OnModelCreation to seed initial data

Parameters

<i>builder</i>	type ModelBuilder
----------------	-------------------

Definition at line 41 of file [AppDbContext.cs](#).

```
00042     {
00043         base.OnModelCreating(builder);
00044
00045         builder.Entity<ApplicationSettings>()
00046             .HasKey(x => new { x.Name, x.Type });
00047
00048         builder.Entity<ApplicationSettings>()
00049             .Property(x => x.Value);
00050
00051         // seeding default settings
00052         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
00053             "ApplicationName", Type = "GlobalSettings", Value = "SecPass"});
00053         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
00053             "SessionTimeoutWarnAfter", Type = "GlobalSettings", Value = "50000"});
```

```

00054         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"SessionTimeoutRedirectAfter", Type = "GlobalSettings", Value = "70000"});
00055         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"SessionCookieExpiration", Type = "GlobalSettings", Value = "7"});
00056         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name = "Username",
Type = "MailSettings", Value = "YOUR_Smtp_Username"});
00057         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name = "Password",
Type = "MailSettings", Value = "YOUR_SmtpPassword"});
00058         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"SmtpServer", Type = "MailSettings", Value = "YOUR_SmtpServer"});
00059         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name = "SmtpPort",
Type = "MailSettings", Value = "587"});
00060         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"SmtpUseTls", Type = "MailSettings", Value = "true"});
00061         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"SmtpFromAddress", Type = "MailSettings", Value = "YOUR_From_Address"});
00062         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"LdapDomainController", Type = "LdapSettings", Value = "YOUR_Domaincontroller_FQDN"});
00063         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"LdapDomainName", Type = "LdapSettings", Value = "YOUR_Domainname"});
00064         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"LdapBaseDn", Type = "LdapSettings", Value = "DC=YOUR,DC=Domain,DC=com"});
00065         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name = "LdapGroup",
Type = "LdapSettings", Value = "YOUR_Ldap_Group"});
00066         builder.Entity<ApplicationSettings>().HasData(new ApplicationSettings {Name =
"LdapEnabled", Type = "LdapSettings", Value = "true"});
00067
00068         //Seeding roles toAspNetRoles table
00069         builder.Entity<AppRole>().HasData(new AppRole { Id =
"dffc6dd5-b145-41e9-a861-c87ff673e9ca", Name = "Admin", NormalizedName = "ADMIN".ToUpper() });
00070         builder.Entity<AppRole>().HasData(new AppRole { Id =
"f8a527ac-d7f6-4d9d-aca6-46b2261b042b", Name = "User", NormalizedName = "USER".ToUpper() });
00071         builder.Entity<AppRole>().HasData(new AppRole { Id =
"g7a527ac-d7t6-4d7z-aca6-45t2261b042b", Name = "Editor", NormalizedName = "EDITOR".ToUpper() });
00072         builder.Entity<AppRole>().HasData(new AppRole { Id =
"p9a527ac-d77w-4d3r-aca6-35b2261b042b", Name = "Moderator", NormalizedName = "MODERATOR".ToUpper()
});
00073
00074
00075
00076         //a hasher to hash the password before seeding the user to the db
00077         var hasher = new PasswordHasher<AppUser>();
00078
00079         //Seeding the Admin User toAspNetUsers table
00080         builder.Entity<AppUser>().HasData(
00081             new AppUser
00082             {
00083                 Id = "6fbfb682-568c-4f5b-a298-85937ca4f7f3", // primary key
00084                 UserName = "super.admin",
00085                 NormalizedUserName = "SUPER.ADMIN",
00086                 PasswordHash = hasher.HashPassword(null!, "Test1000!"),
00087                 Email = "super.admin@local.app",
00088                 NormalizedEmail = "SUPER.ADMIN@LOCAL.APP",
00089                 EmailConfirmed = true,
00090                 FirstName = "Super",
00091                 LastName = "Admin",
00092                 IsMfaForce = false,
00093                 IsLdapLogin = false,
00094                 IsEnabled = true,
00095                 RolesCombined = "Admin"
00096             }
00097         );
00098
00099         //Seeding the relation between our user and role toAspNetUserRoles table
00100         builder.Entity<IdentityUserRole<string>>().HasData(
00101             new IdentityUserRole<string>
00102             {
00103                 RoleId = "dffc6dd5-b145-41e9-a861-c87ff673e9ca",
00104                 UserId = "6fbfb682-568c-4f5b-a298-85937ca4f7f3"
00105             }
00106         );
00107     }

```

6.2.4 Property Documentation

6.2.4.1 AppLogs

DbSet<AppLogs>? DotNetIdentity.Data.AppDbContext.AppLogs [get], [set]

Property AppLogs

Definition at line 25 of file [AppDbContext.cs](#).

```
00025 {get;set;}
```

6.2.4.2 AppSessionCache

```
DbSet<SessionCache>? DotNetIdentity.Data.AppDbContext.AppSessionCache [get], [set]
```

property AppSessionCache

Definition at line 35 of file [AppDbContext.cs](#).

```
00035 {get;set;}
```

6.2.4.3 AppSettings

```
DbSet<ApplicationSettings>? DotNetIdentity.Data.AppDbContext.AppSettings [get], [set]
```

property AppSettings

Definition at line 30 of file [AppDbContext.cs](#).

```
00030 {get;set;}
```

Referenced by [DotNetIdentity.Services.SettingsService.AppSettingsBase.Load\(\)](#), and [DotNetIdentity.Services.SettingsService.AppSettingsBase.Save\(\)](#)

6.3 DotNetIdentity.Models.DataModels.ApplicationSettings Class Reference

class define an Settings object

Properties

- string? [Name](#) [get, set]
property name
- string? [Type](#) [get, set]
property Type
- string? [Value](#) [get, set]
property value

6.3.1 Detailed Description

class define an Settings object

Definition at line 7 of file [ApplicationSettings.cs](#).

6.3.2 Property Documentation

6.3.2.1 Name

```
string? DotNetIdentity.Models.DataModels.ApplicationSettings.Name [get], [set]
```

property name

string

Definition at line 12 of file [ApplicationSettings.cs](#).

```
00012 {get;set;}
```

Referenced by [DotNetIdentity.Services.SettingsService.AppSettingsBase.Save\(\)](#).

6.3.2.2 Type

```
string? DotNetIdentity.Models.DataModels.ApplicationSettings.Type [get], [set]
```

property Type

string

Definition at line 17 of file [ApplicationSettings.cs](#).

```
00017 {get;set;}
```

6.3.2.3 Value

```
string? DotNetIdentity.Models.DataModels.ApplicationSettings.Value [get], [set]
```

property value

string

Definition at line 22 of file [ApplicationSettings.cs](#).

```
00022 {get;set;}
```

6.4 DotNetIdentity.Models.DataModels.AppLogs Class Reference

class to define ann Log object

Properties

- int [id](#) [get, set]
property id
- string? [Exception](#) [get, set]
property Exception
- string [Level](#) = string.Empty [get, set]
property Level
- string [Message](#) = string.Empty [get, set]
property Message
- string [MessageTemplate](#) = string.Empty [get, set]
property MessageTemplate
- string [Properties](#) = string.Empty [get, set]
property Properties
- string [Timestamp](#) = string.Empty [get, set]
property Timestamp

6.4.1 Detailed Description

class to define ann Log object

Definition at line 5 of file [AppLogs.cs](#).

6.4.2 Property Documentation

6.4.2.1 Exception

```
string? DotNetIdentity.Models.DataModels.AppLogs.Exception [get], [set]
```

property Exception

string

Definition at line 15 of file [AppLogs.cs](#).

```
00015 {get;set;}
```

6.4.2.2 id

```
int DotNetIdentity.Models.DataModels.AppLogs.id [get], [set]
```

property id

int

Definition at line 10 of file [AppLogs.cs](#).

```
00010 {get;set;}
```

6.4.2.3 Level

```
string DotNetIdentity.Models.DataModels.AppLogs.Level = string.Empty [get], [set]
```

property Level

string

Definition at line 20 of file [AppLogs.cs](#).

```
00020 {get;set;} = string.Empty;
```

6.4.2.4 Message

```
string DotNetIdentity.Models.DataModels.AppLogs.Message = string.Empty [get], [set]
```

property Message

string

Definition at line 25 of file [AppLogs.cs](#).

```
00025 {get;set;} = string.Empty;
```

6.4.2.5 MessageTemplate

```
string DotNetIdentity.Models.DataModels.AppLogs.MessageTemplate = string.Empty [get], [set]
```

property MessageTemplate

string

Definition at line 30 of file [AppLogs.cs](#).

```
00030 {get;set;} = string.Empty;
```

6.4.2.6 Properties

```
string DotNetIdentity.Models.DataModels.AppLogs.Properties = string.Empty [get], [set]
```

property Properties

string

Definition at line 35 of file [AppLogs.cs](#).

```
00035 {get;set;} = string.Empty;
```

6.4.2.7 Timestamp

```
string DotNetIdentity.Models.DataModels.AppLogs.Timestamp = string.Empty [get], [set]
```

property Timestamp

string

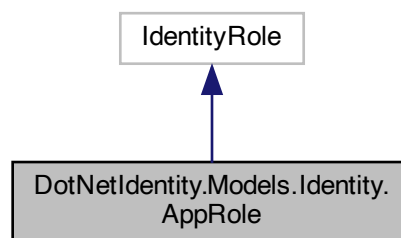
Definition at line 40 of file [AppLogs.cs](#).

```
00040 {get;set;} = string.Empty;
```

6.5 DotNetIdentity.Models.Identity.AppRole Class Reference

class to define a role object

Inheritance diagram for DotNetIdentity.Models.Identity.AppRole:



Properties

- DateTime [CreatedOn](#) [get, set]
property created on

6.5.1 Detailed Description

class to define a role object

Definition at line 8 of file [AppRole.cs](#).

6.5.2 Property Documentation

6.5.2.1 CreatedOn

`DateTime DotNetIdentity.Models.Identity.AppRole.CreatedOn [get], [set]`

property created on

DateTime

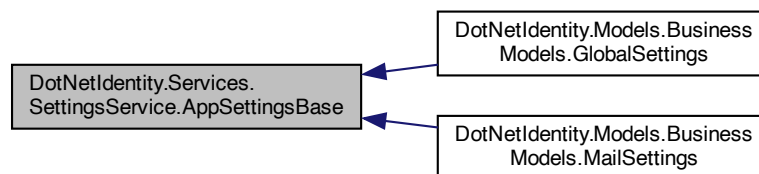
Definition at line 14 of file [AppRole.cs](#).

```
00014 { get; set; }
```

6.6 DotNetIdentity.Services.SettingsService.AppSettingsBase Class Reference

base class for Settings classes

Inheritance diagram for DotNetIdentity.Services.SettingsService.AppSettingsBase:



Public Member Functions

- [AppSettingsBase](#) ()
class constructor
- virtual void [Load](#) ([AppDbContext](#) unitOfWork)
load method
- virtual async Task [Save](#) ([AppDbContext](#) unitOfWork)
save method

6.6.1 Detailed Description

base class for Settings classes

Definition at line 10 of file [AppSettingsBase.cs](#).

6.6.2 Constructor & Destructor Documentation

6.6.2.1 AppSettingsBase()

DotNetIdentity.Services.SettingsService.AppSettingsBase.AppSettingsBase () [inline]

class constructor

Definition at line 24 of file [AppSettingsBase.cs](#).

```
00025     {
00026         var type = this.GetType();
00027         _name = type.Name;
00028         // 2
00029         _properties = type.GetProperties();
00030     }
```

6.6.3 Member Function Documentation

6.6.3.1 Load()

virtual void DotNetIdentity.Services.SettingsService.AppSettingsBase.Load (
 [AppDbContext](#) unitOfWork) [inline], [virtual]

load method

Parameters

<i>unitOfWork</i>	type DbContext
-------------------	----------------

Definition at line 36 of file [AppSettingsBase.cs](#).

```
00037     {
00038         // ARGUMENT CHECKING SKIPPED FOR BREVITY
00039         // 3 get settings for this type name
00040
00041         var settings = unitOfWork.AppSettings!.Where(w => w.Type == _name).ToList();
00042
00043         foreach (var propertyInfo in _properties)
00044         {
00045             // get the setting from the settings list
00046             var setting = settings.SingleOrDefault(s => s.Name == propertyInfo.Name);
00047             if (setting != null)
00048             {
00049                 // 4 assign the setting values to the properties in the type inheriting this class
00050                 propertyInfo.SetValue(this, Convert.ChangeType(setting.Value,
00051                     propertyInfo.PropertyType));
00052             }
00053         }
```

References [DotNetIdentity.Data.AppDbContext.AppSettings](#).

6.6.3.2 Save()

virtual async Task DotNetIdentity.Services.SettingsService.AppSettingsBase.Save (
 [AppDbContext](#) unitOfWork) [inline], [virtual]

save method

Parameters

<i>unitOfWork</i>	type DbContext
-------------------	----------------

Definition at line 59 of file [AppSettingsBase.cs](#).

```

00060     {
00061         // 5 load existing settings for this type
00062         var settings = unitOfWork.AppSettings!.Where(w => w.Type == _name).ToList();
00063
00064         foreach (var propertyInfo in _properties)
00065         {
00066             object propertyValue = propertyInfo.GetValue(this, null)!;
00067             string value = (propertyValue == null) ? null! : propertyValue.ToString();
00068
00069             var setting = settings.SingleOrDefault(s => s.Name == propertyInfo.Name);
00070             if (setting != null)
00071             {
00072                 // 6 update existing value
00073                 setting.Value = value;
00074             }
00075             else
00076             {
00077                 // 7 create new setting
00078                 var newSetting = new ApplicationSettings()
00079                 {
00080                     Name = propertyInfo.Name,
00081                     Type = _name,
00082                     Value = value,
00083                 };
00084                 await unitOfWork.AppSettings!.AddAsync(newSetting);
00085             }
00086         }
00087     }

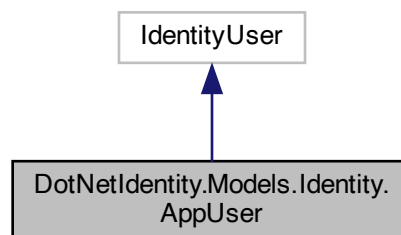
```

References [DotNetIdentity.Data.AppDbContext.AppSettings](#), and [DotNetIdentity.Models.DataModels.ApplicationSettings.Name](#).

6.7 DotNetIdentity.Models.Identity.AppUser Class Reference

class to define a user object

Inheritance diagram for DotNetIdentity.Models.Identity.AppUser:



Properties

- [TwoFactorType](#) [TwoFactorType](#) [get, set]
2fa type property

- DateTime [BirthDay](#) [get, set]
birthday property
- [Gender](#) [Gender](#) [get, set]
gender property
- DateTime [CreatedOn](#) [get, set]
birthday CreatedOn
- String? [FirstName](#) = string.Empty [get, set]
firstname property
- String? [LastName](#) = string.Empty [get, set]
lastname property
- bool [IsMfaForce](#) [get, set]
2fa force property
- bool [IsLdapLogin](#) [get, set]
ldap login property
- string? [Department](#) = string.Empty [get, set]
department property
- string? [ProfilePicture](#) [get, set]
profile picture proerty
- bool [IsEnabled](#) [get, set]
account enbaled property
- string? [RolesCombined](#) = string.Empty [get, set]
roles combined property

6.7.1 Detailed Description

class to define a user object

Definition at line 8 of file [AppUser.cs](#).

6.7.2 Property Documentation

6.7.2.1 BirthDay

```
DateTime DotNetIdentity.Models.Identity.AppUser.BirthDay [get], [set]
```

birthday property

DateTime

Definition at line 19 of file [AppUser.cs](#).

```
00019 { get; set; }
```

6.7.2.2 CreatedOn

`DateTime DotNetIdentity.Models.Identity.AppUser.CreatedOn [get], [set]`

birthday CreatedOn

DateTime

Definition at line 29 of file [AppUser.cs](#).

```
00029 { get; set; }
```

6.7.2.3 Department

`string? DotNetIdentity.Models.Identity.AppUser.Department = string.Empty [get], [set]`

department property

Department

Definition at line 54 of file [AppUser.cs](#).

```
00054 { get; set; } = string.Empty;
```

6.7.2.4 FirstName

`String? DotNetIdentity.Models.Identity.AppUser.FirstName = string.Empty [get], [set]`

firstname property

string

Definition at line 34 of file [AppUser.cs](#).

```
00034 { get; set; } = string.Empty;
```

6.7.2.5 Gender

`Gender DotNetIdentity.Models.Identity.AppUser.Gender [get], [set]`

gender property

Gender

Definition at line 24 of file [AppUser.cs](#).

```
00024 { get; set; }
```

6.7.2.6 IsEnabled

```
bool DotNetIdentity.Models.Identity.AppUser.IsEnabled [get], [set]
```

account enbaled property

bool

Definition at line 64 of file [AppUser.cs](#).

```
00064 { get; set; }
```

6.7.2.7 IsLdapLogin

```
bool DotNetIdentity.Models.Identity.AppUser.IsLdapLogin [get], [set]
```

ldap login property

bool

Definition at line 49 of file [AppUser.cs](#).

```
00049 { get; set; }
```

6.7.2.8 IsMfaForce

```
bool DotNetIdentity.Models.Identity.AppUser.IsMfaForce [get], [set]
```

2fa force property

bool

Definition at line 44 of file [AppUser.cs](#).

```
00044 { get; set; }
```

6.7.2.9 LastName

```
String? DotNetIdentity.Models.Identity.AppUser.LastName = string.Empty [get], [set]
```

lastname property

string

Definition at line 39 of file [AppUser.cs](#).

```
00039 { get; set; } = string.Empty;
```

6.7.2.10 ProfilePicture

`string? DotNetIdentity.Models.Identity.AppUser.ProfilePicture [get], [set]`

profile picture proerty

string

Definition at line 59 of file [AppUser.cs](#).

```
00059 { get; set; }
```

6.7.2.11 RolesCombined

`string? DotNetIdentity.Models.Identity.AppUser.RolesCombined = string.Empty [get], [set]`

roles combined property

Definition at line 69 of file [AppUser.cs](#).

```
00069 { get; set; } = string.Empty;
```

6.7.2.12 TwoFactorType

`TwoFactorType DotNetIdentity.Models.Identity.AppUser.TwoFactorType [get], [set]`

2fa type property

TwoFactorType

Definition at line 14 of file [AppUser.cs](#).

```
00014 { get; set; }
```

6.8 DotNetIdentity.Models.ViewModels.AssignRoleViewModel Class Reference

model for UserEditView

Properties

- string [RoleId](#) = default! [get, set]
the role id
- string [RoleName](#) = default! [get, set]
the role neame
- bool [IsAssigned](#) [get, set]
if role is assigned

6.8.1 Detailed Description

model for UserEditView

Definition at line 6 of file [AssignRoleViewModel.cs](#).

6.8.2 Property Documentation

6.8.2.1 IsAssigned

```
bool DotNetIdentity.Models.ViewModels.AssignRoleViewModel.IsAssigned [get], [set]
```

if role is assigned

bool

Definition at line 22 of file [AssignRoleViewModel.cs](#).

```
00022 { get; set; }
```

6.8.2.2 RoleId

```
string DotNetIdentity.Models.ViewModels.AssignRoleViewModel.RoleId = default! [get], [set]
```

the role id

string

Definition at line 12 of file [AssignRoleViewModel.cs](#).

```
00012 { get; set; } = default!;
```

6.8.2.3 RoleName

```
string DotNetIdentity.Models.ViewModels.AssignRoleViewModel.RoleName = default! [get], [set]
```

the role neame

string

Definition at line 17 of file [AssignRoleViewModel.cs](#).

```
00017 { get; set; } = default!;
```

6.9 DotNetIdentity.Models.ViewModels.ChangePasswordViewModel

Class Reference

model for ChangePassword view

Properties

- string **Password** = default! [get, set]
the password
- string **NewPassword** = default! [get, set]
the new password
- string **NewPasswordConfirm** = default! [get, set]
the new password confirmed

6.9.1 Detailed Description

model for ChangePassword view

Definition at line 8 of file [ChangePasswordViewModel.cs](#).

6.9.2 Property Documentation

6.9.2.1 NewPassword

```
string DotNetIdentity.Models.ViewModels.ChangePasswordViewModel.NewPassword = default! [get],  
[set]
```

the new password

string

Definition at line 19 of file [ChangePasswordViewModel.cs](#).

```
00019 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ChangePassword\(\)](#).

6.9.2.2 NewPasswordConfirm

```
string DotNetIdentity.Models.ViewModels.ChangePasswordViewModel.NewPasswordConfirm = default!  
[get], [set]
```

the new password confirmed

string

Definition at line 25 of file [ChangePasswordViewModel.cs](#).

```
00025 { get; set; } = default!;
```

6.9.2.3 Password

```
string DotNetIdentity.Models.ViewModels.ChangePasswordViewModel.Password = default! [get],  
[set]
```

the password

string

Definition at line 14 of file [ChangePasswordViewModel.cs](#).

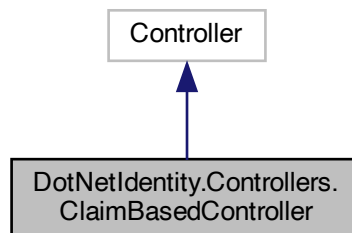
```
00014 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ChangePassword\(\)](#).

6.10 DotNetIdentity.Controllers.ClaimBasedController Class Reference

Controller to test ClaimBased Authorization

Inheritance diagram for DotNetIdentity.Controllers.ClaimBasedController:



Public Member Functions

- IActionResult [HR](#) ()
Controller action for HR view
- IActionResult [Software](#) ()
Controller action for Software view
- IActionResult [Employee](#) ()
Controller action for Employee view
- IActionResult [FreeTrial](#) ()
Controller action for FreeTrial view
- IActionResult [Violence](#) ()
Controller action for AtLeast18Policy view

6.10.1 Detailed Description

Controller to test ClaimBased Authorization

Definition at line 9 of file [ClaimBasedController.cs](#).

6.10.2 Member Function Documentation

6.10.2.1 Employee()

```
ActionResult DotNetIdentity.Controllers.ClaimBasedController.Employee ( )
```

Controller action for Employee view

Returns

view Employee

6.10.2.2 FreeTrial()

```
ActionResult DotNetIdentity.Controllers.ClaimBasedController.FreeTrial ( )
```

Controller action for FreeTrial view

Returns

view FreeTrial

6.10.2.3 HR()

```
ActionResult DotNetIdentity.Controllers.ClaimBasedController.HR ( )
```

Controller action for HR view

Returns

view HR

6.10.2.4 Software()

```
ActionResult DotNetIdentity.Controllers.ClaimBasedController.Software ( )
```

Controller action for Software view

Returns

view Software

6.10.2.5 Violence()

```
ActionResult DotNetIdentity.Controllers.ClaimBasedController.Violence ( )
```

Controller action for AtLeast18Policy view

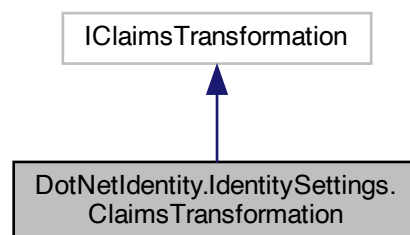
Returns

view AtLeast18Policy

6.11 DotNetIdentity.IdentitySettings.ClaimsTransformation Class Reference

class to define own claims

Inheritance diagram for DotNetIdentity.IdentitySettings.ClaimsTransformation:



Public Member Functions

- [ClaimsTransformation](#) (userManager< [AppUser](#) > userManager, RoleManager< [AppRole](#) > roleManager)
class constructor
- async Task< ClaimsPrincipal > [TransformAsync](#) (ClaimsPrincipal principal)
Task to add claims to identity

6.11.1 Detailed Description

class to define own claims

Definition at line 11 of file [ClaimsTransformation.cs](#).

6.11.2 Constructor & Destructor Documentation

6.11.2.1 ClaimsTransformation()

```
DotNetIdentity.IdentitySettings.ClaimsTransformation.ClaimsTransformation (
    UserManager< AppUser > userManager,
    RoleManager< AppRole > roleManager ) [inline]
```

class constructor

Parameters

<i>userManager</i>	UserManager
<i>roleManager</i>	RoleManager

Definition at line 27 of file [ClaimsTransformation.cs](#).

```
00028     {
00029         _userManager = userManager;
00030         _roleManager = roleManager;
00031     }
```

6.11.3 Member Function Documentation

6.11.3.1 TransformAsync()

```
async Task< ClaimsPrincipal > DotNetIdentity.IdentitySettings.ClaimsTransformation.Transform↔
Async (
    ClaimsPrincipal principal ) [inline]
```

Task to add claims to identoty

Parameters

<i>principal</i>	ClaimsPrincipal
------------------	-----------------

Returns

ClaimsPrincipal

Definition at line 38 of file [ClaimsTransformation.cs](#).

```

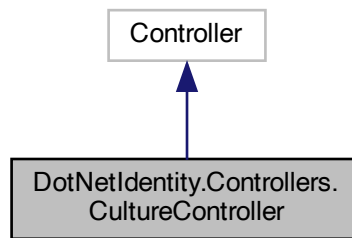
00039     {
00040         var identity = principal.Identity as ClaimsIdentity;
00041         var user = await _userManager.FindByNameAsync(identity?.Name);
00042         if (user != null)
00043         {
00044             if (!principal.HasClaim(c => c.Type == ClaimTypes.Gender))
00045             {
00046                 var genderClaim = new Claim(ClaimTypes.Gender, Enum.GetName(user.Gender!));
00047                 identity?.AddClaim(genderClaim);
00048             }
00049
00050             if (!principal.HasClaim(c => c.Type == ClaimTypes.DateOfBirth))
00051             {
00052                 var birthDayClaim = new Claim(ClaimTypes.DateOfBirth,
00053                     user.BirthDay.ToShortDateString());
00054                 identity?.AddClaim(birthDayClaim);
00055             }
00056
00057             if (!principal.HasClaim(c => c.Type == "FreeTrial"))
00058             {
00059                 var freeTrialClaim = new Claim("FreeTrial", user.CreatedOn.ToShortDateString());
00060                 identity?.AddClaim(freeTrialClaim);
00061             }
00062
00063             if (!principal.HasClaim(c => c.Type == "Department") &&
00064                 !String.IsNullOrEmpty(user.Department))
00065             {
00066                 var departmentClaim = new Claim("Department", user.Department!);
00067                 identity?.AddClaim(departmentClaim);
00068             }
00069
00070             if (!principal.HasClaim(c => c.Type == ClaimTypes.GivenName))
00071             {
00072                 var givennameClaim = new Claim(ClaimTypes.GivenName, user.FirstName!);
00073                 identity?.AddClaim(givennameClaim);
00074             }
00075
00076             if (!principal.HasClaim(c => c.Type == ClaimTypes.Surname))
00077             {
00078                 var surnameClaim = new Claim(ClaimTypes.Surname, user.LastName!);
00079                 identity?.AddClaim(surnameClaim);
00080             }
00081
00082             if(!principal.HasClaim(c=>c.Type=="SelectedCulture")) {
00083                 var culture =
00084                     System.Globalization.CultureInfo.CurrentCulture.ToString().ToLower();
00085                 var cultureClaim = new Claim("SelectedCulture", culture);
00086                 identity?.AddClaim(cultureClaim);
00087             }
00088
00089             if (!principal.HasClaim(c => c.Type == "RolesCombined"))
00090             {
00091                 var rls = await _userManager.GetRolesAsync(user);
00092                 if (rls != null)
00093                 {
00094                     var rlsComb = "";
00095                     if(rls.Count()>0 && rls.Count()<=1) {
00096                         rlsComb = rls[0].ToString();
00097                     } else {
00098                         for(int i=0;i<rls.Count;i++)
00099                         {
00100                             if(i+1!=rls.Count){ rlsComb += rls[i] + ","; }
00101                             else { rlsComb += rls[i]; }
00102                         }
00103                     }
00104                     if(rlsComb.Length>0)
00105                     {
00106                         var rolesCombinedClaim = new Claim("RolesCombined", rlsComb!);
00107                         identity?.AddClaim(rolesCombinedClaim);
00108                     }
00109                 }
00110             }
00111         }
00112     }
00113
00114     return principal;
00115 }

```

6.12 DotNetIdentity.Controllers.CultureController Class Reference

controller for culture settings

Inheritance diagram for DotNetIdentity.Controllers.CultureController:



Public Member Functions

- IActionResult [SetCulture](#) (string culture, string returnUrl)
controller POST method to set current culture

6.12.1 Detailed Description

controller for culture settings

Definition at line 17 of file [CultureController.cs](#).

6.12.2 Member Function Documentation

6.12.2.1 SetCulture()

```
IActionResult DotNetIdentity.Controllers.CultureController.SetCulture (  
    string culture,  
    string returnUrl ) [inline]
```

controller POST method to set current culture

Parameters

<i>culture</i>	the culture as string
<i>returnUrl</i>	the return url as string

Returns

a local redirect to variable returnUrl

Definition at line 26 of file [CultureController.cs](#).

```
00027     {
00028         Response.Cookies.Append(
00029             CookieRequestCultureProvider.DefaultCookieName,
00030             CookieRequestCultureProvider.MakeCookieValue(new RequestCulture(culture)),
00031             new CookieOptions { Expires = DateTimeOffset.UtcNow.AddYears(1) }
00032         );
00033         return LocalRedirect(returnUrl);
00034     }
```

6.13 DotNetIdentity.Models.ViewModels.EditUserViewModel Class Reference

model for EditUser view

Properties

- string **Id** = default! [get, set]
the user id
- string **UserName** = default! [get, set]
the username
- string **Email** = default! [get, set]
the email
- string **PhoneNumber** = default! [get, set]
the phone number
- **Gender** **Gender** = **Gender.Unknown** [get, set]
the gener
- **Department** **Department** = **Department.Software** [get, set]
the department
- List< **AssignRoleViewModel** > **Roles** = new() [get, set]
the roles assigned
- string **FirstName** = string.Empty [get, set]
the firstname
- string **LastName** = string.Empty [get, set]
the lastname

6.13.1 Detailed Description

model for EditUser view

Definition at line 6 of file [EditUserViewModel.cs](#).

6.13.2 Property Documentation

6.13.2.1 Department

`Department` `DotNetIdentity.Models.ViewModels.EditUserViewModel.Department = Department.Software` `[get], [set]`

the department

Department

Definition at line 37 of file [EditUserViewModel.cs](#).

```
00037 { get; set; } = Department.Software;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.2 Email

`string` `DotNetIdentity.Models.ViewModels.EditUserViewModel.Email = default!` `[get], [set]`

the email

string

Definition at line 22 of file [EditUserViewModel.cs](#).

```
00022 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.3 FirstName

`string` `DotNetIdentity.Models.ViewModels.EditUserViewModel.FirstName = string.Empty` `[get], [set]`

the firstname

string

Definition at line 47 of file [EditUserViewModel.cs](#).

```
00047 { get; set; } = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.4 Gender

`Gender` `DotNetIdentity.Models.ViewModels.EditUserViewModel.Gender = Gender.Unknown` `[get], [set]`

the gener

Gender

Definition at line 32 of file [EditUserViewModel.cs](#).

```
00032 { get; set; } = Gender.Unknown;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.5 Id

```
string DotNetIdentity.Models.ViewModels.EditUserViewModel.Id = default! [get], [set]
```

the user id

string

Definition at line 12 of file [EditUserViewModel.cs](#).

```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.6 LastName

```
string DotNetIdentity.Models.ViewModels.EditUserViewModel.LastName = string.Empty [get], [set]
```

the lastname

string

Definition at line 52 of file [EditUserViewModel.cs](#).

```
00052 {get; set;} = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.7 PhoneNumber

```
string DotNetIdentity.Models.ViewModels.EditUserViewModel.PhoneNumber = default! [get], [set]
```

the phone number

string

Definition at line 27 of file [EditUserViewModel.cs](#).

```
00027 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.8 Roles

```
List<AssignRoleViewModel> DotNetIdentity.Models.ViewModels.EditUserViewModel.Roles = new()  
[get], [set]
```

the roles assigned

Returns

List of type AppRole

Definition at line 42 of file [EditUserViewModel.cs](#).

```
00042 { get; set; } = new();
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.13.2.9 Username

```
string DotNetIdentity.Models.ViewModels.EditUserViewModel.Username = default! [get], [set]
```

the username

string

Definition at line 17 of file [EditUserViewModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.EditUser\(\)](#).

6.14 DotNetIdentity.Helpers.EmailHelper Class Reference

class for email methods

Public Member Functions

- [EmailHelper](#) (ISettingsService sett)
class constructor
- async Task [SendAsync](#) ([EmailModel](#) model)
method to send a mail

6.14.1 Detailed Description

class for email methods

Definition at line 12 of file [EmailHelper.cs](#).

6.14.2 Constructor & Destructor Documentation

6.14.2.1 EmailHelper()

```
DotNetIdentity.Helpers.EmailHelper.EmailHelper (  
    ISettingsService sett ) [inline]
```

class constructor

Parameters

<i>sett</i>	type ISettingsService
-------------	-----------------------

Definition at line 23 of file [EmailHelper.cs](#).


```

00024         {
00025             _sett = sett;
00026         }

```

6.14.3 Member Function Documentation

6.14.3.1 SendAsync()

```

async Task DotNetIdentity.Helpers.EmailHelper.SendAsync (
    EmailModel model ) [inline]

```

method to send a mail

Parameters

<i>model</i>	type EmailModel
--------------	-----------------

Returns

Definition at line 33 of file [EmailHelper.cs](#).

```

00034     {
00035         var mail = new MailMessage
00036         {
00037             Subject = model.Subject,
00038             Body = model.Body,
00039             From = new MailAddress(_sett.Mail.SmtptFromAddress!),
00040             IsBodyHtml = true
00041         };
00042
00043         var smtp = new SmtpClient
00044         {
00045             Host = _sett.Mail.SmtptServer!,
00046             Port = Convert.ToInt32(_sett.Mail.SmtptPort),
00047             EnableSsl = Convert.ToBoolean(_sett.Mail.SmtptUseTls),
00048             DeliveryMethod = SmtpDeliveryMethod.Network,
00049             UseDefaultCredentials = false,
00050             Credentials = new NetworkCredential(_sett.Mail.Username, _sett.Mail.Password)
00051         };
00052
00053         mail.To.Add(model.To);
00054
00055         await smtp.SendMailAsync(mail);
00056     }

```

References [DotNetIdentity.Models.BusinessModels.EmailModel.Body](#), [DotNetIdentity.Models.BusinessModels.EmailModel.Subject](#), and [DotNetIdentity.Models.BusinessModels.EmailModel.To](#).

6.15 DotNetIdentity.Models.BusinessModels.EmailModel Class Reference

class to define a email

Properties

- string [Subject](#) = default! [get, set]
property subject
- string [Body](#) = default! [get, set]
property body
- string [To](#) = default! [get, set]
property to

6.15.1 Detailed Description

class to define a email

Definition at line 6 of file [EmailModel.cs](#).

6.15.2 Property Documentation

6.15.2.1 Body

```
string DotNetIdentity.Models.BusinessModels.EmailModel.Body = default! [get], [set]
```

property body

string

Definition at line 17 of file [EmailModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Helpers.EmailHelper.SendAsync\(\)](#).

6.15.2.2 Subject

```
string DotNetIdentity.Models.BusinessModels.EmailModel.Subject = default! [get], [set]
```

property subject

string

Definition at line 12 of file [EmailModel.cs](#).

```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Helpers.EmailHelper.SendAsync\(\)](#).

6.15.2.3 To

```
string DotNetIdentity.Models.BusinessModels.EmailModel.To = default! [get], [set]
```

property to

string

Definition at line 22 of file [EmailModel.cs](#).

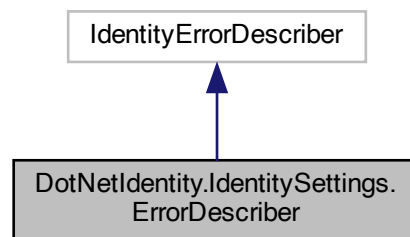
```
00022 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Helpers.EmailHelper.SendAsync\(\)](#).

6.16 DotNetIdentity.IdentitySettings.ErrorDescriber Class Reference

class to override IdentityErrorDescriber IdentityError messages

Inheritance diagram for DotNetIdentity.IdentitySettings.ErrorDescriber:



Static Public Member Functions

- static IdentityError [PasswordContainsUsername](#) ()
custom IdentityError if password contains username
- static IdentityError [UserNameLength](#) ()
custom IdentityError if username length is too short
- static IdentityError [UserNameContainsEmail](#) ()
custom IdentityError if username contains user email

6.16.1 Detailed Description

class to override IdentityErrorDescriber IdentityError messages

Definition at line 8 of file [ErrorDescriber.cs](#).

6.16.2 Member Function Documentation

6.16.2.1 PasswordContainsUsername()

```
static IdentityError DotNetIdentity.IdentitySettings.ErrorDescriber.PasswordContainsUsername (
) [static]
```

custom IdentityError if password contains username

Returns

IdentityError

Referenced by [DotNetIdentity.IdentitySettings.Validators.PasswordValidator.ValidateAsync\(\)](#).

6.16.2.2 UserNameContainsEmail()

```
static IdentityError DotNetIdentity.IdentitySettings.ErrorDescriber.UserNameContainsEmail ( )
[static]
```

custom IdentityError if username contains user email

Returns

IdentityError

Referenced by [DotNetIdentity.IdentitySettings.Validators.UserValidator.ValidateAsync\(\)](#).

6.16.2.3 UserNameLength()

```
static IdentityError DotNetIdentity.IdentitySettings.ErrorDescriber.UserNameLength ( ) [static]
```

custom IdentityError if username length is too short

Returns

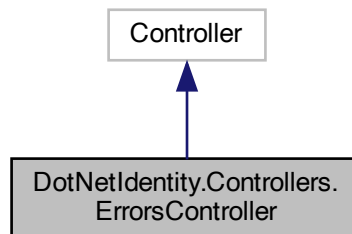
IdentityError

Referenced by [DotNetIdentity.IdentitySettings.Validators.UserValidator.ValidateAsync\(\)](#).

6.17 DotNetIdentity.Controllers.ErrorsController Class Reference

Controller for error views

Inheritance diagram for DotNetIdentity.Controllers.ErrorsController:



Public Member Functions

- IActionResult [ErrorEx](#) ()
controller action for ErrorEx view
- IActionResult [ErrorCd](#) (int statusCode)
Controller action for ErrorCd view

6.17.1 Detailed Description

Controller for error views

Todo add translation

Definition at line 17 of file [ErrorsController.cs](#).

6.17.2 Member Function Documentation

6.17.2.1 ErrorCd()

```
IActionResult DotNetIdentity.Controllers.ErrorsController.ErrorCd (  
    int statusCode ) [inline]
```

Controller action for ErrorCd view

Parameters

<i>statusCode</i>	
-------------------	--

Returns

view ErrorCd

Definition at line 45 of file [ErrorsController.cs](#).

```

00046     {
00047         var statusCodeData = HttpContext.Features.Get<IStatusCodeReExecuteFeature>();
00048         StatusCodesModel md = new StatusCodesModel();
00049         md.StatusCode = statusCode;
00050         md.RouteOfException = statusCodeData!.OriginalPath;
00051
00052         switch (statusCode)
00053         {
00054             // request timeout
00055             case 408:
00056                 md.ErrorMessage = "Timeout. The request timed out..";
00057                 break;
00058             // badrequest
00059             case 400:
00060                 md.ErrorMessage = "Badrequest. The request meets not the controllers expetation.";
00061                 break;
00062             // forbidden
00063             case 403:
00064                 md.ErrorMessage = "Access denied!";
00065                 break;
00066             // unauthorized
00067             case 401:
00068                 md.ErrorMessage = "You're not authorized to access this resource!";
00069                 break;
00070             //page not found
00071             case 404:
00072                 md.ErrorMessage = "Sorry the page you requested could not be found";
00073
00074                 break;
00075             // server error
00076             case 500:
00077                 md.ErrorMessage = "Sorry something went wrong on the server";
00078                 break;
00079             // service unavailable
00080             case 503:
00081                 md.ErrorMessage = "Sorry, the service is currently unavailable.";
00082                 break;
00083             default:
00084                 md.ErrorMessage = "Sorry, there was an error.";
00085                 break;
00086         }
00087         return View("ErrorCd", md);
00088     }

```

6.17.2.2 ErrorEx()

`IActionResult DotNetIdentity.Controllers.ErrorsController.ErrorEx () [inline]`

controller action for ErrorEx view

Returns

view ErrorEx

Definition at line 23 of file [ErrorsController.cs](#).

```
00024     {
00025         var exceptionFeature = HttpContext.Features.Get<ExceptionHandlerPathFeature>();
00026
00027         if (exceptionFeature != null)
00028         {
00029             if(exceptionFeature.Error.StackTrace!=null) {
00030                 ViewBag.StackTrace = exceptionFeature.Error.StackTrace;
00031             }
00032             ViewBag.ErrorMessage = exceptionFeature.Error.Message;
00033             ViewBag.RouteOfException = exceptionFeature.Path;
00034         }
00035
00036         return View();
00037     }
```

6.18 DotNetIdentity.Models.ViewModels.ErrorViewModel Class Reference

model for displaying errors

Properties

- string? [RequestId](#) [get, set]
the request id
- bool [ShowRequestId](#) [get]
show the request id

6.18.1 Detailed Description

model for displaying errors

Definition at line 5 of file [ErrorViewModel.cs](#).

6.18.2 Property Documentation

6.18.2.1 RequestId

```
string? DotNetIdentity.Models.ViewModels.ErrorViewModel.RequestId [get], [set]
```

the request id

string

Definition at line 11 of file [ErrorViewModel.cs](#).

```
00011 { get; set; }
```

6.18.2.2 ShowRequestId

```
bool DotNetIdentity.Models.ViewModels.ErrorViewModel.ShowRequestId [get]
```

show the request id

Returns

bool

Definition at line 16 of file [ErrorViewModel.cs](#).

6.19 DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel Class Reference

model for ForgotPassword view

Properties

- string [Email](#) = default! [get, set]
the user email

6.19.1 Detailed Description

model for ForgotPassword view

Definition at line 6 of file [ForgotPasswordViewModel.cs](#).

6.19.2 Property Documentation

6.19.2.1 Email

```
string DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel.Email = default! [get], [set]
```

the user email

string

Definition at line 12 of file [ForgotPasswordViewModel.cs](#).

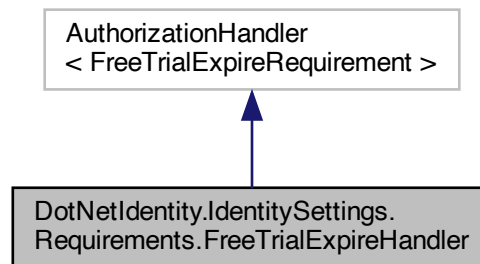
```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ForgotPassword\(\)](#).

6.20 DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler Class Reference

class to define own Identity requirements type of [FreeTrialExpireRequirement](#)

Inheritance diagram for DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler:



Protected Member Functions

- override Task [HandleRequirementAsync](#) (AuthorizationHandlerContext context, [FreeTrialExpireRequirement](#) requirement)
overriding HandleRequirementAsync

6.20.1 Detailed Description

class to define own Identity requirements type of [FreeTrialExpireRequirement](#)

Definition at line 13 of file [FreeTrialExpireRequirement.cs](#).

6.20.2 Member Function Documentation

6.20.2.1 HandleRequirementAsync()

```

override Task DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler.Handle<
RequirementAsync (
    AuthorizationHandlerContext context,
    FreeTrialExpireRequirement requirement ) [inline], [protected]
  
```

overriding HandleRequirementAsync

Parameters

<i>context</i>	AuthorizationHandlerContext
<i>requirement</i>	FreeTrialExpireRequirement

Returns

Task

Definition at line 21 of file [FreeTrialExpireRequirement.cs](#).

```

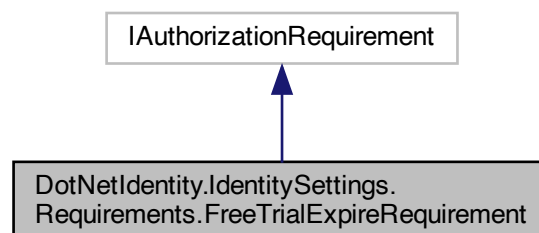
00022     {
00023         var freeTrialClaim = context.User.FindFirst(f => f.Type == "FreeTrial");
00024         if (freeTrialClaim != null)
00025         {
00026             var freeTrialExpires = Convert.ToDateTime(freeTrialClaim.Value).AddDays(30);
00027             if (DateTime.Now < freeTrialExpires)
00028             {
00029                 context.Succeed(requirement);
00030             }
00031             else
00032             {
00033                 context.Fail();
00034             }
00035         }
00036         return Task.CompletedTask;
00037     }
00038 }

```

6.21 DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement Class Reference

class to define own Identity requirements type of IAuthorizationRequirement

Inheritance diagram for DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement:



6.21.1 Detailed Description

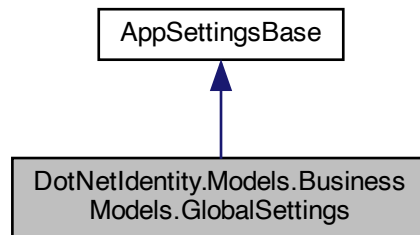
class to define own Identity requirements type of IAuthorizationRequirement

Definition at line 8 of file [FreeTrialExpireRequirement.cs](#).

6.22 DotNetIdentity.Models.BusinessModels.GlobalSettings Class Reference

class to define a global settings object

Inheritance diagram for DotNetIdentity.Models.BusinessModels.GlobalSettings:



Properties

- string? [ApplicationName](#) [get, set]
property ApplicationName
- string? [SessionTimeoutWarnAfter](#) [get, set]
property SessionTimeoutWarnAfter
- string? [SessionTimeoutRedirAfter](#) [get, set]
property SessionTimeoutRedirAfter
- string? [SessionCookieExpiration](#) [get, set]
property SessionCookieExpiration

Additional Inherited Members

6.22.1 Detailed Description

class to define a global settings object

Definition at line 7 of file [GlobalSettings.cs](#).

6.22.2 Property Documentation

6.22.2.1 ApplicationName

```
string? DotNetIdentity.Models.BusinessModels.GlobalSettings.ApplicationName [get], [set]
```

property ApplicationName

string

Definition at line 12 of file [GlobalSettings.cs](#).

```
00012 {get; set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.22.2.2 SessionCookieExpiration

```
string? DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionCookieExpiration [get],  
[set]
```

property SessionCookieExpiration

string

Definition at line 27 of file [GlobalSettings.cs](#).

```
00027 {get; set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.22.2.3 SessionTimeoutRedirAfter

```
string? DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionTimeoutRedirAfter [get],  
[set]
```

property SessionTimeoutRedirAfter

string

Definition at line 22 of file [GlobalSettings.cs](#).

```
00022 {get; set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.22.2.4 SessionTimeoutWarnAfter

```
string? DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionTimeoutWarnAfter [get],  
[set]
```

property SessionTimeoutWarnAfter

string

Definition at line 17 of file [GlobalSettings.cs](#).

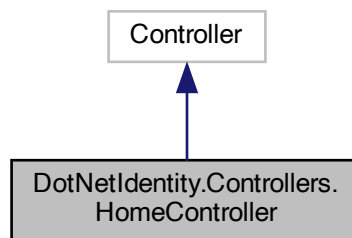
```
00017 {get; set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.23 DotNetIdentity.Controllers.HomeController Class Reference

Controller for Home views

Inheritance diagram for DotNetIdentity.Controllers.HomeController:



Public Member Functions

- [HomeController](#) (UserManager< [AppUser](#) > userManager)
Controller constructor
- IActionResult [Index](#) ()
Controller action for view Index
- IActionResult [AccessDenied](#) ()
controller action for view AccessDenied

6.23.1 Detailed Description

Controller for Home views

Definition at line 11 of file [HomeController.cs](#).

6.23.2 Constructor & Destructor Documentation

6.23.2.1 HomeController()

```
DotNetIdentity.Controllers.HomeController.HomeController (
    UserManager< AppUser > userManager ) [inline]
```

Controller constructor

Parameters

<i>userManager</i>	Property UserManager
--------------------	----------------------

Definition at line 22 of file [HomeController.cs](#).

```
00023     {  
00024         _userManager = userManager;  
00025     }
```

6.23.3 Member Function Documentation

6.23.3.1 AccessDenied()

```
IActionResult DotNetIdentity.Controllers.HomeController.AccessDenied ( )
```

controller action for view AccessDenied

Returns

view AccessDenied

6.23.3.2 Index()

```
IActionResult DotNetIdentity.Controllers.HomeController.Index ( )
```

Controller action for view Index

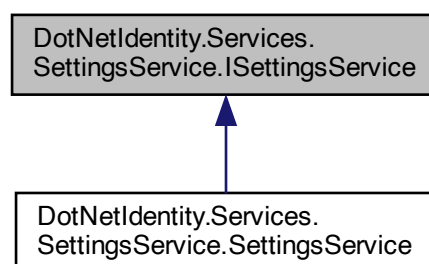
Returns

view Index

6.24 DotNetIdentity.Services.SettingService.ISettingService Interface Reference

Interface class for [SettingService](#)

Inheritance diagram for DotNetIdentity.Services.SettingService.ISettingService:



Public Member Functions

- Task [Save](#) ()
save method

Properties

- [GlobalSettings Global](#) [get]
property global settings
- [MailSettings Mail](#) [get]
property mail settings
- [LdapSettings Ldap](#) [get]
property mail settings

6.24.1 Detailed Description

Interface class for [SettingsService](#)

Definition at line 8 of file [ISettingsService.cs](#).

6.24.2 Member Function Documentation

6.24.2.1 Save()

```
Task DotNetIdentity.Services.SettingsService.ISettingsService.Save ( )
```

save method

Implemented in [DotNetIdentity.Services.SettingsService.SettingsService](#).

6.24.3 Property Documentation

6.24.3.1 Global

```
GlobalSettings DotNetIdentity.Services.SettingsService.ISettingsService.Global [get]
```

property global settings

GlobalSettings

Implemented in [DotNetIdentity.Services.SettingsService.SettingsService](#).

Definition at line 13 of file [ISettingsService.cs](#).

```
00013 { get; }
```

6.24.3.2 Ldap

`LdapSettings` `DotNetIdentity.Services.SettingsService.ISettingsService.Ldap` [get]

property mail settings

MailSettings

Implemented in `DotNetIdentity.Services.SettingsService.SettingsService`.

Definition at line 23 of file `ISettingsService.cs`.

```
00023 { get; }
```

6.24.3.3 Mail

`MailSettings` `DotNetIdentity.Services.SettingsService.ISettingsService.Mail` [get]

property mail settings

MailSettings

Implemented in `DotNetIdentity.Services.SettingsService.SettingsService`.

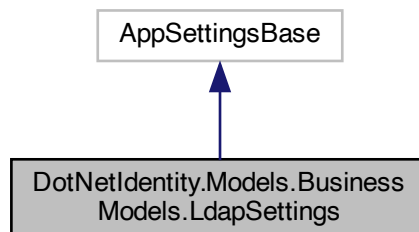
Definition at line 18 of file `ISettingsService.cs`.

```
00018 { get; }
```

6.25 DotNetIdentity.Models.BusinessModels.LdapSettings Class Reference

class to define ldap settings

Inheritance diagram for `DotNetIdentity.Models.BusinessModels.LdapSettings`:



Properties

- string? [LdapDomainController](#) [get, set]
domain controller property
- string? [LdapDomainName](#) [get, set]
domain name property
- string? [LdapBaseDn](#) [get, set]
domain base dn property
- string? [LdapGroup](#) [get, set]
domain group name property
- bool [LdapEnabled](#) [get, set]
ldap enabled property

6.25.1 Detailed Description

class to define ldap settings

Definition at line 5 of file [LdapSettings.cs](#).

6.25.2 Property Documentation

6.25.2.1 LdapBaseDn

```
string? DotNetIdentity.Models.BusinessModels.LdapSettings.LdapBaseDn [get], [set]
```

domain base dn property

string

Definition at line 20 of file [LdapSettings.cs](#).

```
00020 {get;set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsLdap\(\)](#).

6.25.2.2 LdapDomainController

```
string? DotNetIdentity.Models.BusinessModels.LdapSettings.LdapDomainController [get], [set]
```

domain controller property

string

Definition at line 10 of file [LdapSettings.cs](#).

```
00010 {get;set;}
```

6.25.2.3 LdapDomainName

```
string? DotNetIdentity.Models.BusinessModels.LdapSettings.LdapDomainName [get], [set]
```

domain name property

string

Definition at line 15 of file [LdapSettings.cs](#).

```
00015 {get;set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsLdap\(\)](#).

6.25.2.4 LdapEnabled

```
bool DotNetIdentity.Models.BusinessModels.LdapSettings.LdapEnabled [get], [set]
```

Ldap enabled property

bool

Definition at line 30 of file [LdapSettings.cs](#).

```
00030 {get;set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsLdap\(\)](#).

6.25.2.5 LdapGroup

```
string? DotNetIdentity.Models.BusinessModels.LdapSettings.LdapGroup [get], [set]
```

domain group name property

string

Definition at line 25 of file [LdapSettings.cs](#).

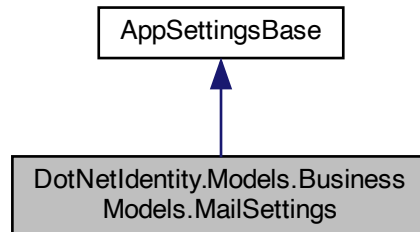
```
00025 {get;set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsLdap\(\)](#).

6.26 DotNetIdentity.Models.BusinessModels.MailSettings Class Reference

class to define a mail settings object

Inheritance diagram for DotNetIdentity.Models.BusinessModels.MailSettings:



Properties

- string? [Username](#) [get, set]
property Username
- string? [Password](#) [get, set]
property Password
- string? [SmtpServer](#) [get, set]
property SmtpServer
- string? [SmtpPort](#) [get, set]
property SmtpPort
- string? [SmtpFromAddress](#) [get, set]
property SmtpFromAddress
- bool [SmtpUseTls](#) [get, set]
property SmtpUseTls

Additional Inherited Members

6.26.1 Detailed Description

class to define a mail settings object

Definition at line 7 of file [MailSettings.cs](#).

6.26.2 Property Documentation

6.26.2.1 Password

```
string? DotNetIdentity.Models.BusinessModels.MailSettings.Password [get], [set]
```

property Password

string

Definition at line 17 of file [MailSettings.cs](#).

```
00017 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.26.2.2 SmtptFromAddress

```
string? DotNetIdentity.Models.BusinessModels.MailSettings.SmtptFromAddress [get], [set]
```

property SmtptFromAddress

string

Definition at line 32 of file [MailSettings.cs](#).

```
00032 {get; set;}
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.26.2.3 SmtptPort

```
string? DotNetIdentity.Models.BusinessModels.MailSettings.SmtptPort [get], [set]
```

property SmtptPort

string

Definition at line 27 of file [MailSettings.cs](#).

```
00027 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.26.2.4 SmtptServer

```
string? DotNetIdentity.Models.BusinessModels.MailSettings.SmtptServer [get], [set]
```

property SmtptServer

string

Definition at line 22 of file [MailSettings.cs](#).

```
00022 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.26.2.5 SmtptUseTls

```
bool DotNetIdentity.Models.BusinessModels.MailSettings.SmtptUseTls [get], [set]
```

property SmtptUseTls

bool

Definition at line 37 of file [MailSettings.cs](#).

```
00037 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.26.2.6 Username

```
string? DotNetIdentity.Models.BusinessModels.MailSettings.Username [get], [set]
```

property Username

string

Definition at line 12 of file [MailSettings.cs](#).

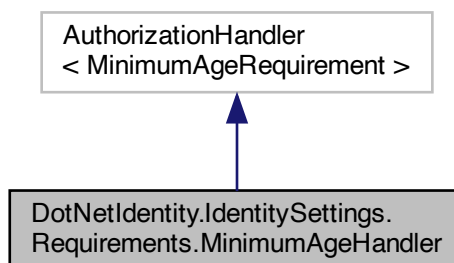
```
00012 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.SettingsController.SettingsMail\(\)](#).

6.27 DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler Class Reference

class implementing AuthorizationHandler of type [MinimumAgeRequirement](#)

Inheritance diagram for DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler:



Protected Member Functions

- override Task [HandleRequirementAsync](#) (AuthorizationHandlerContext context, [MinimumAgeRequirement](#) requirement)
overriding HandleRequirementAsync

6.27.1 Detailed Description

class implementing AuthorizationHandler of type [MinimumAgeRequirement](#)

Definition at line 30 of file [MinimumAgeRequirement.cs](#).

6.27.2 Member Function Documentation

6.27.2.1 HandleRequirementAsync()

```
override Task DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler.HandleRequirement↵
Async (
```

```
    AuthorizationHandlerContext context,
    MinimumAgeRequirement requirement ) [inline], [protected]
```

overriding HandleRequirementAsync

Parameters

<i>context</i>	AuthorizationHandlerContext
<i>requirement</i>	MinimumAgeRequirement

Returns

Task

Definition at line 38 of file [MinimumAgeRequirement.cs](#).

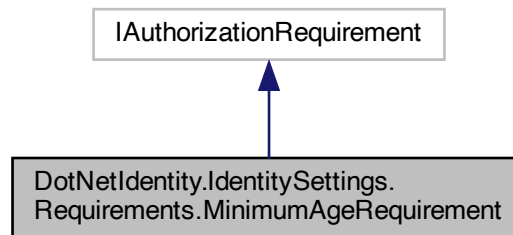
```
00039     {
00040         var birthDayClaim = context.User.FindFirst(f => f.Type == ClaimTypes.DateOfBirth);
00041         if (birthDayClaim != null)
00042         {
00043             var birthDay = Convert.ToDateTime(birthDayClaim.Value);
00044             var calculatedAge = DateTime.Now.Year - birthDay.Year;
00045             if (calculatedAge >= requirement.MinimumAge)
00046             {
00047                 context.Succeed(requirement);
00048             }
00049             else
00050             {
00051                 context.Fail();
00052             }
00053         }
00054         return Task.CompletedTask;
00055     }
```

References [DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement.MinimumAge](#).

6.28 DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement Class Reference

class to define own Identity requirements type of IAuthorizationRequirement

Inheritance diagram for DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement:



Public Member Functions

- [MinimumAgeRequirement](#) (int minimumAge)
class constructor

Properties

- int [MinimumAge](#) [get, set]
property MinimumAge

6.28.1 Detailed Description

class to define own Identity requirements type of IAuthorizationRequirement

Definition at line 9 of file [MinimumAgeRequirement.cs](#).

6.28.2 Constructor & Destructor Documentation

6.28.2.1 MinimumAgeRequirement()

```
DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement.MinimumAgeRequirement (
    int minimumAge ) [inline]
```

class constructor

Parameters

<i>minimumAge</i>	
-------------------	--

Definition at line 15 of file [MinimumAgeRequirement.cs](#).

```
00016     {
00017         MinimumAge = minimumAge;
00018     }
```

References [DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement.MinimumAge](#).

6.28.3 Property Documentation**6.28.3.1 MinimumAge**

```
int DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement.MinimumAge [get], [set]
```

property MinimumAge

Definition at line 24 of file [MinimumAgeRequirement.cs](#).

```
00024 { get; set; }
```

Referenced by [DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler.HandleRequirementAsync\(\)](#), and [DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement.MinimumAgeRequirement\(\)](#).

6.29 DotNetIdentity.Models.ViewModels.NewUserModel Class Reference

model for NewUser view

Properties

- string [UserName](#) = default! [get, set]
the username
- string [Email](#) = default! [get, set]
the users email address
- [Gender](#) [Gender](#) = [Gender.Unknown](#) [get, set]
the gender
- [Department](#) [Department](#) = [Department.Software](#) [get, set]
the department
- List< [AssignRoleViewModel](#) > [Roles](#) = new() [get, set]
the roles assigned
- string [FirstName](#) = string.Empty [get, set]
the firstname
- string [LastName](#) = string.Empty [get, set]
the lastname
- bool [IsLdapLogin](#) [get, set]
if user is ldap login
- bool [IsMfaForce](#) [get, set]
is 2fa enforced
- string [Password](#) = string.Empty [get, set]
the password

6.29.1 Detailed Description

model for NewUser view

Definition at line 9 of file [NewUserModel.cs](#).

6.29.2 Property Documentation

6.29.2.1 Department

```
Department DotNetIdentity.Models.ViewModels.NewUserModel.Department = Department.Software  
[get], [set]
```

the department

Department

Definition at line 30 of file [NewUserModel.cs](#).

```
00030 { get; set; } = Department.Software;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.2 Email

```
string DotNetIdentity.Models.ViewModels.NewUserModel.Email = default! [get], [set]
```

the users email address

string

Definition at line 20 of file [NewUserModel.cs](#).

```
00020 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.3 FirstName

```
string DotNetIdentity.Models.ViewModels.NewUserModel.FirstName = string.Empty [get], [set]
```

the firstname

string

Definition at line 40 of file [NewUserModel.cs](#).

```
00040 { get; set; } = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.4 Gender

`Gender` `DotNetIdentity.Models.ViewModels.NewUserModel.Gender` = `Gender.Unknown` [get], [set]

the gender

Gender

Definition at line 25 of file [NewUserModel.cs](#).

```
00025 { get; set; } = Gender.Unknown;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.5 IsLdapLogin

`bool` `DotNetIdentity.Models.ViewModels.NewUserModel.IsLdapLogin` [get], [set]

if user is ldap login

bool

Definition at line 50 of file [NewUserModel.cs](#).

```
00050 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.6 IsMfaForce

`bool` `DotNetIdentity.Models.ViewModels.NewUserModel.IsMfaForce` [get], [set]

is 2fa enforced

bool

Definition at line 55 of file [NewUserModel.cs](#).

```
00055 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.7 LastName

`string` `DotNetIdentity.Models.ViewModels.NewUserModel.LastName` = `string.Empty` [get], [set]

the lastname

string

Definition at line 45 of file [NewUserModel.cs](#).

```
00045 { get; set; } = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.8 Password

```
string DotNetIdentity.Models.ViewModels.NewUserModel.Password = string.Empty [get], [set]
```

the password

string

Definition at line 60 of file [NewUserModel.cs](#).

```
00060 { get; set; } = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.9 Roles

```
List<AssignRoleViewModel> DotNetIdentity.Models.ViewModels.NewUserModel.Roles = new() [get],  
[set]
```

the roles assigned

Returns

List of type AppRole

Definition at line 35 of file [NewUserModel.cs](#).

```
00035 { get; set; } = new();
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.29.2.10 UserName

```
string DotNetIdentity.Models.ViewModels.NewUserModel.UserName = default! [get], [set]
```

the username

string

Definition at line 15 of file [NewUserModel.cs](#).

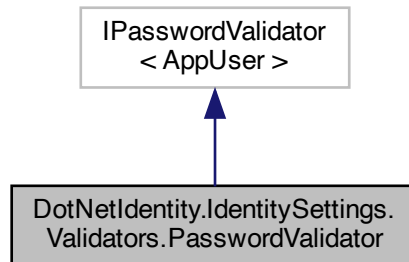
```
00015 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.NewUser\(\)](#).

6.30 DotNetIdentity.IdentitySettings.Validators.PasswordValidator Class Reference

class to validate a password

Inheritance diagram for DotNetIdentity.IdentitySettings.Validators.PasswordValidator:



Public Member Functions

- Task< IdentityResult > [ValidateAsync](#) (UserManager< [AppUser](#) > manager, [AppUser](#) user, string password)
Task to validate a password

6.30.1 Detailed Description

class to validate a password

Definition at line 9 of file [PasswordValidator.cs](#).

6.30.2 Member Function Documentation

6.30.2.1 ValidateAsync()

```
Task< IdentityResult > DotNetIdentity.IdentitySettings.Validators.PasswordValidator.ValidateAsync (
    UserManager< AppUser > manager,
    AppUser user,
    string password ) [inline]
```

Task to validate a password

Parameters

<i>manager</i>	UserManager
<i>user</i>	AppUser
<i>password</i>	string

Returns

Task of type IdentityResult

Definition at line 18 of file [PasswordValidator.cs](#).

```

00019     {
00020         var errors = new List<IdentityError>();
00021
00022         if (user.UserName == password)
00023         {
00024             errors.Add(ErrorDescriber.PasswordContainsUsername());
00025         }
00026
00027         if (errors.Any())
00028         {
00029             return Task.FromResult(IdentityResult.Failed(errors.ToArray()));
00030         }
00031         return Task.FromResult(IdentityResult.Success);
00032     }

```

References [DotNetIdentity.IdentitySettings.ErrorDescriber.PasswordContainsUsername\(\)](#).

Here is the call graph for this function:



6.31 DotNetIdentity.Models.ViewModels.ResetPasswordViewModel Class Reference

model for ResetPassword view

Properties

- string [UserId](#) = default! [get, set]
the user id
- string [Token](#) = default! [get, set]
the reset token
- string [Password](#) = default! [get, set]
the password

6.31.1 Detailed Description

model for ResetPassword view

Definition at line 6 of file [ResetPasswordViewModel.cs](#).

6.31.2 Property Documentation

6.31.2.1 Password

```
string DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.Password = default! [get],  
[set]
```

the password

string

Definition at line 22 of file [ResetPasswordViewModel.cs](#).

```
00022 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ResetPassword\(\)](#).

6.31.2.2 Token

```
string DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.Token = default! [get], [set]
```

the reset token

string

Definition at line 17 of file [ResetPasswordViewModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ResetPassword\(\)](#).

6.31.2.3 UserId

```
string DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.UserId = default! [get], [set]
```

the user id

string

Definition at line 12 of file [ResetPasswordViewModel.cs](#).

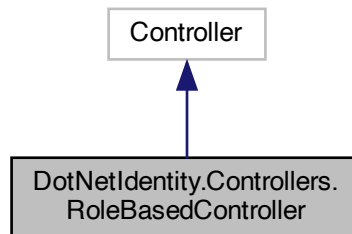
```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.ResetPassword\(\)](#).

6.32 DotNetIdentity.Controllers.RoleBasedController Class Reference

Controller to test RoleBased Authorization

Inheritance diagram for DotNetIdentity.Controllers.RoleBasedController:



Public Member Functions

- IActionResult [Dashboard](#) ()
Controller Action for Dashboard view
- IActionResult [Users](#) ()
Controller Action for Users view
- IActionResult [Articles](#) ()
Controller Action for Articles view
- IActionResult [Profile](#) ()
Controller Action for Profile view

6.32.1 Detailed Description

Controller to test RoleBased Authorization

Definition at line 9 of file [RoleBasedController.cs](#).

6.32.2 Member Function Documentation

6.32.2.1 Articles()

```
IActionResult DotNetIdentity.Controllers.RoleBasedController.Articles ( )
```

Controller Action for Articles view

Returns

6.32.2.2 Dashboard()

```
ActionResult DotNetIdentity.Controllers.RoleBasedController.Dashboard ( )
```

Controller Action for Dashboard view

Returns

6.32.2.3 Profile()

```
ActionResult DotNetIdentity.Controllers.RoleBasedController.Profile ( )
```

Controller Action for Profile view

Returns

6.32.2.4 Users()

```
ActionResult DotNetIdentity.Controllers.RoleBasedController.Users ( )
```

Controller Action for Users view

Returns

6.33 DotNetIdentity.Models.DataModels.SessionCache Class Reference

class to define a [SessionCache](#) object

Properties

- string? [id](#) [get, set]
property id
- byte?[] [Value](#) [get, set]
property value
- DateTime [ExpiresAtTime](#) [get, set]
property ExpiresAtTime
- int [SlidingExpirationInSeconds](#) [get, set]
property SlidingExpirationInSeconds
- DateTime? [AbsoluteExpiration](#) [get, set]
property AbsoluteExpiration

6.33.1 Detailed Description

class to define a [SessionCache](#) object

Definition at line 11 of file [SessionCache.cs](#).

6.33.2 Property Documentation

6.33.2.1 AbsoluteExpiration

DateTime? DotNetIdentity.Models.DataModels.SessionCache.AbsoluteExpiration [get], [set]

property AbsoluteExpiration

DateTime

Definition at line 37 of file [SessionCache.cs](#).

```
00037 {get; set;}
```

6.33.2.2 ExpiresAtTime

DateTime DotNetIdentity.Models.DataModels.SessionCache.ExpiresAtTime [get], [set]

property ExpiresAtTime

DateTime

Definition at line 27 of file [SessionCache.cs](#).

```
00027 {get; set;}
```

6.33.2.3 id

string? DotNetIdentity.Models.DataModels.SessionCache.id [get], [set]

property id

int

Definition at line 17 of file [SessionCache.cs](#).

```
00017 {get; set;}
```

6.33.2.4 SlidingExpirationInSeconds

```
int DotNetIdentity.Models.DataModels.SessionCache.SlidingExpirationInSeconds [get], [set]
```

property SlidingExpirationInSeconds

int

Definition at line 32 of file [SessionCache.cs](#).

```
00032 {get; set;}
```

6.33.2.5 Value

```
byte? [ ] DotNetIdentity.Models.DataModels.SessionCache.Value [get], [set]
```

property value

byte[]

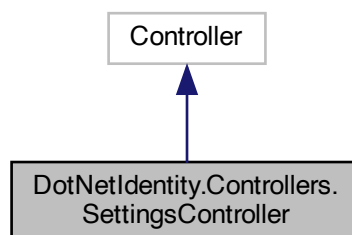
Definition at line 22 of file [SessionCache.cs](#).

```
00022 {get; set;}
```

6.34 DotNetIdentity.Controllers.SettingsController Class Reference

controller class for settings views

Inheritance diagram for DotNetIdentity.Controllers.SettingsController:



Public Member Functions

- [SettingsController](#) (ILogger< [SettingsController](#) > log, ISettingsService sett, IStringLocalizer< [SettingsController](#) > loc, UserManager< [AppUser](#) > usr, RoleManager< [AppRole](#) > rol, IWebHostEnvironment env)
class constructor
- ActionResult [SettingsLdap](#) ()
Controller action for SettingsLdap view
- ActionResult [SettingsMail](#) ()
Controller action for SettingsMail view
- ActionResult [SettingsApp](#) ()
Controller action for SettingsApp view
- async Task< IActionResult > [SettingsMail](#) ([MailSettings](#) viewModel)
controller POST method to save Mail settings
- async Task< IActionResult > [SettingsMail](#) ([GlobalSettings](#) viewModel)
controller POST method to save Application settings
- async Task< IActionResult > [SettingsLdap](#) ([LdapSettings](#) viewModel)
controller POST method to save ldap settings

6.34.1 Detailed Description

controller class for settings views

Todo add translation

Definition at line 21 of file [SettingsController.cs](#).

6.34.2 Constructor & Destructor Documentation

6.34.2.1 SettingsController()

```
DotNetIdentity.Controllers.SettingsController.SettingsController (
    ILogger< SettingsController > log,
    ISettingsService sett,
    IStringLocalizer< SettingsController > loc,
    UserManager< AppUser > usr,
    RoleManager< AppRole > rol,
    IWebHostEnvironment env ) [inline]
```

class constructor

Parameters

<i>sett</i>	ISettingsService
<i>loc</i>	IStringLocalizer
<i>usr</i>	UserManager
<i>rol</i>	RoleManager
<i>log</i>	ILogger

```
///
```

Parameters

<i>env</i>	IWebHostEnvironment
------------	---------------------

Definition at line 57 of file [SettingsController.cs](#).

```
00057
    {
00058         _hostEnvironment = env;
00059         _userManager = usr;
00060         _roleManager = rol;
00061         _localizer = loc;
00062         _settings = sett;
00063         _logger = log;
00064     }
```

6.34.3 Member Function Documentation

6.34.3.1 SettingsApp()

ActionResult DotNetIdentity.Controllers.SettingsController.SettingsApp () [inline]

Controller action for SettingsApp view

Returns

view SettingsApp

Definition at line 90 of file [SettingsController.cs](#).

```
00090     {
00091         var viewModel = new GlobalSettings();
00092         viewModel = _settings.Global;
00093         return View(viewModel);
00094     }
```

6.34.3.2 SettingsLdap() [1/2]

ActionResult DotNetIdentity.Controllers.SettingsController.SettingsLdap () [inline]

Controller action for SettingsLdap view

Returns

view SettingsLdap

Definition at line 70 of file [SettingsController.cs](#).

```
00070     {
00071         var viewModel = new LdapSettings();
00072         viewModel = _settings.Ldap;
00073         return View(viewModel);
00074     }
```

6.34.3.3 SettingsLdap() [2/2]

async Task< IActionResult > DotNetIdentity.Controllers.SettingsController.SettingsLdap (
LdapSettings viewModel) [inline]

controller POST method to save ldap settings

Parameters

<i>viewModel</i>	LdapSettings
------------------	--------------

Returns

view

Definition at line 143 of file [SettingsController.cs](#).

```

00143                                     {
00144
00145         _settings.Ldap.LdapDomainName = viewModel.LdapDomainName;
00146         _settings.Ldap.LdapDomainController = viewModel.LdapDomainName;
00147         _settings.Ldap.LdapBaseDn = viewModel.LdapBaseDn;
00148         _settings.Ldap.LdapEnabled = viewModel.LdapEnabled;
00149         _settings.Ldap.LdapGroup = viewModel.LdapGroup;
00150         await _settings.Save();
00151         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " modified Ldap settings! ");
00152         ViewData["message"] = _localizer["1"];
00153         return View();
00154     }

```

References [DotNetIdentity.Models.BusinessModels.LdapSettings.LdapBaseDn](#), [DotNetIdentity.Models.BusinessModels.LdapSettings.LdapDomainController](#), [DotNetIdentity.Models.BusinessModels.LdapSettings.LdapEnabled](#), and [DotNetIdentity.Models.BusinessModels.LdapSettings.LdapGroup](#)

6.34.3.4 SettingsMail() [1/3]

```
ActionResult DotNetIdentity.Controllers.SettingsController.SettingsMail ( ) [inline]
```

Controller action for SettingsMail view

Returns

view SettingsMail

Definition at line 80 of file [SettingsController.cs](#).

```

00080                                     {
00081         var viewModel = new MailSettings();
00082         viewModel = _settings.Mail;
00083         return View(viewModel);
00084     }

```

6.34.3.5 SettingsMail() [2/3]

```
async Task< IActionResult > DotNetIdentity.Controllers.SettingsController.SettingsMail (
    GlobalSettings viewModel ) [inline]
```

controller POST method to save Application settings

Parameters

<i>viewModel</i>	GlobalSettings
------------------	----------------

Returns

view

Definition at line 124 of file [SettingsController.cs](#).

```

00124                                     {
00125
00126         _settings.Global.ApplicationName = viewModel.ApplicationName;
00127         _settings.Global.SessionCookieExpiration = viewModel.SessionCookieExpiration;
00128         _settings.Global.SessionTimeoutWarnAfter = viewModel.SessionTimeoutWarnAfter;
00129         _settings.Global.SessionTimeoutRedirAfter = viewModel.SessionTimeoutRedirAfter;
00130         await _settings.Save();
00131         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " modified Application
settings! ");
00132         ViewData["message"] = _localizer["1"];
00133         return View();
00134     }

```

References [DotNetIdentity.Models.BusinessModels.GlobalSettings.ApplicationName](#), [DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionCookieExpiration](#), [DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionTimeoutWarnAfter](#), [DotNetIdentity.Models.BusinessModels.GlobalSettings.SessionTimeoutRedirAfter](#), and [DotNetIdentity.Models.BusinessModels.GlobalSettings.ApplicationName](#).

6.34.3.6 SettingsMail() [3/3]

```

async Task< IActionResult > DotNetIdentity.Controllers.SettingsController.SettingsMail (
    MailSettings viewModel ) [inline]

```

controller POST method to save Mail settings

Parameters*viewModel*

MailSettings

Returns

view

Definition at line 103 of file [SettingsController.cs](#).

```

00103                                     {
00104
00105         _settings.Mail.SmtpServer = viewModel.SmtpServer;
00106         _settings.Mail.SmtpFromAddress = viewModel.SmtpFromAddress;
00107         _settings.Mail.SmtpPort = viewModel.SmtpPort;
00108         _settings.Mail.SmtpUseTls = viewModel.SmtpUseTls;
00109         _settings.Mail.Username = viewModel.Username;
00110         _settings.Mail.Password = viewModel.Password;
00111         await _settings.Save();
00112         _logger.LogInformation("AUDIT: " + User.Identity!.Name + " modified Mail settings! ");
00113         ViewData["message"] = _localizer["1"];
00114         return View();
00115     }

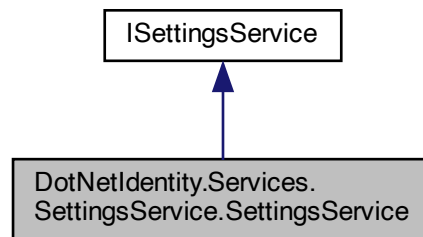
```

References [DotNetIdentity.Models.BusinessModels.MailSettings.Password](#), [DotNetIdentity.Models.BusinessModels.MailSettings.SmtpServer](#), [DotNetIdentity.Models.BusinessModels.MailSettings.SmtpPort](#), [DotNetIdentity.Models.BusinessModels.MailSettings.SmtpUseTls](#), and [DotNetIdentity.Models.BusinessModels.MailSettings.Username](#).

6.35 DotNetIdentity.Services.SettingsService.SettingsService Class Reference

class for settings service

Inheritance diagram for DotNetIdentity.Services.SettingsService.SettingsService:



Public Member Functions

- [SettingsService](#) ([AppDbContext](#) unitOfWork)
class constructor
- async Task [Save](#) ()
save method

Properties

- [LdapSettings](#) [Ldap](#) [get]
global settings property
- [GlobalSettings](#) [Global](#) [get]
global settings property
- [MailSettings](#) [Mail](#) [get]
mail settings property

6.35.1 Detailed Description

class for settings service

Definition at line 8 of file [SettingsService.cs](#).

6.35.2 Constructor & Destructor Documentation

6.35.2.1 SettingsService()

```
DotNetIdentity.Services.SettingsService.SettingsService.SettingsService (
    AppDbContext unitOfWork ) [inline]
```

class constructor

Parameters

<i>unitOfWork</i>	
-------------------	--

Definition at line 45 of file [SettingsService.cs](#).

```
00046 {
00047     // ARGUMENT CHECKING SKIPPED FOR BREVITY
00048     _unitOfWork = unitOfWork;
00049     // 3
00050     _globalSettings = new Lazy<GlobalSettings>(CreateSettings<GlobalSettings>);
00051     _mailSettings = new Lazy<MailSettings>(CreateSettings<MailSettings>);
00052     _ldapSettings = new Lazy<LdapSettings>(CreateSettings<LdapSettings>);
00053 }
```

6.35.3 Member Function Documentation

6.35.3.1 Save()

`async Task DotNetIdentity.Services.SettingsService.SettingsService.Save () [inline]`

save method

Returns

Implements [DotNetIdentity.Services.SettingsService.ISettingsService](#).

Definition at line 59 of file [SettingsService.cs](#).

```
00060 {
00061     // only save changes to settings that have been loaded
00062     if (_globalSettings.IsValueCreated)
00063         await _globalSettings.Value.Save(_unitOfWork);
00064
00065     if (_mailSettings.IsValueCreated)
00066         await _mailSettings.Value.Save(_unitOfWork);
00067
00068     if (_ldapSettings.IsValueCreated)
00069         await _ldapSettings.Value.Save(_unitOfWork);
00070
00071     await _unitOfWork.SaveChangesAsync();
00072 }
```

6.35.4 Property Documentation

6.35.4.1 Global

[GlobalSettings](#) `DotNetIdentity.Services.SettingsService.SettingsService.Global [get]`

global settings property

Implements [DotNetIdentity.Services.SettingsService.ISettingsService](#).

Definition at line 26 of file [SettingsService.cs](#).

```
00026 { get { return _globalSettings.Value; } }
```


6.35.4.2 Ldap

`LdapSettings` `DotNetIdentity.Services.SettingsService.SettingsService.Ldap` [get]

global settings property

Implements `DotNetIdentity.Services.SettingsService.ISettingsService`.

Definition at line 17 of file `SettingsService.cs`.

```
00017 { get { return _ldapSettings.Value; } }
```

6.35.4.3 Mail

`MailSettings` `DotNetIdentity.Services.SettingsService.SettingsService.Mail` [get]

mail settings property

Implements `DotNetIdentity.Services.SettingsService.ISettingsService`.

Definition at line 35 of file `SettingsService.cs`.

```
00035 { get { return _mailSettings.Value; } }
```

6.36 DotNetIdentity.Models.ViewModels.SignInViewModel Class Reference

model for Login view

Properties

- string `UserName` = string.Empty [get, set]
the username
- string `Password` = default! [get, set]
the password
- bool `RememberMe` = true [get, set]
the set cookie persisten option

6.36.1 Detailed Description

model for Login view

Definition at line 6 of file `SignInViewModel.cs`.

6.36.2 Property Documentation

6.36.2.1 Password

```
string DotNetIdentity.Models.ViewModels.SignInViewModel.Password = default! [get], [set]
```

the password

string

Definition at line 17 of file [SignInViewModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Login\(\)](#).

6.36.2.2 RememberMe

```
bool DotNetIdentity.Models.ViewModels.SignInViewModel.RememberMe = true [get], [set]
```

the set cookie persisten option

bool

Definition at line 22 of file [SignInViewModel.cs](#).

```
00022 { get; set; } = true;
```

Referenced by [DotNetIdentity.Controllers.UserController.Login\(\)](#).

6.36.2.3 UserName

```
string DotNetIdentity.Models.ViewModels.SignInViewModel.UserName = string.Empty [get], [set]
```

the username

string

Definition at line 12 of file [SignInViewModel.cs](#).

```
00012 {get; set;} = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.UserController.Login\(\)](#).

6.37 DotNetIdentity.Models.ViewModels.SignUpViewModel Class Reference

model for Register view

Properties

- string [UserName](#) = default! [get, set]
the username
- string [Email](#) = default! [get, set]
the email
- [Gender](#) [Gender](#) = [Gender.Unknown](#) [get, set]
the gender
- DateTime [BirthDay](#) [get, set]
the birthday
- [Department](#) [Department](#) = [Department.Software](#) [get, set]
the department
- string [Password](#) = default! [get, set]
the password
- string [FirstName](#) = string.Empty [get, set]
the firstname
- string [LastName](#) = string.Empty [get, set]
the lastname

6.37.1 Detailed Description

model for Register view

Definition at line 6 of file [SignUpViewModel.cs](#).

6.37.2 Property Documentation

6.37.2.1 BirthDay

```
DateTime DotNetIdentity.Models.ViewModels.SignUpViewModel.BirthDay [get], [set]
```

the birthday

DateTime

Definition at line 27 of file [SignUpViewModel.cs](#).

```
00027 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.2 Department

`Department` `DotNetIdentity.Models.ViewModels.SignUpViewModel.Department = Department.Software` [get], [set]

the department

Department

Definition at line 32 of file [SignUpViewModel.cs](#).

```
00032 { get; set; } = Department.Software;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.3 Email

`string` `DotNetIdentity.Models.ViewModels.SignUpViewModel.Email = default!` [get], [set]

the email

string

Definition at line 17 of file [SignUpViewModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.4 FirstName

`string` `DotNetIdentity.Models.ViewModels.SignUpViewModel.FirstName = string.Empty` [get], [set]

the firstname

string

Definition at line 42 of file [SignUpViewModel.cs](#).

```
00042 {get; set;} = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.5 Gender

`Gender` `DotNetIdentity.Models.ViewModels.SignUpViewModel.Gender = Gender.Unknown` [get], [set]

the gender

Gender

Definition at line 22 of file [SignUpViewModel.cs](#).

```
00022 { get; set; } = Gender.Unknown;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.6 LastName

```
string DotNetIdentity.Models.ViewModels.SignUpViewModel.LastName = string.Empty [get], [set]
```

the lastname

string

Definition at line 47 of file [SignUpViewModel.cs](#).

```
00047 {get; set;} = string.Empty;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.7 Password

```
string DotNetIdentity.Models.ViewModels.SignUpViewModel.Password = default! [get], [set]
```

the password

string

Definition at line 37 of file [SignUpViewModel.cs](#).

```
00037 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.37.2.8 UserName

```
string DotNetIdentity.Models.ViewModels.SignUpViewModel.UserName = default! [get], [set]
```

the username

string

Definition at line 12 of file [SignUpViewModel.cs](#).

```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Register\(\)](#).

6.38 DotNetIdentity.Models.ViewModels.StatusCodesModel Class Reference

model for Error Statuscodes

Properties

- int [StatusCode](#) [get, set]
the status code
- string? [ErrorMessage](#) [get, set]
the error message
- string? [RouteOfException](#) [get, set]
the route that thowed the exception

6.38.1 Detailed Description

model for Error Statuscodes

Definition at line 5 of file [StatusCodesModel.cs](#).

6.38.2 Property Documentation

6.38.2.1 ErrorMessage

```
string? DotNetIdentity.Models.ViewModels.StatusCodesModel.ErrorMessage [get], [set]
```

the error message

string

Definition at line 15 of file [StatusCodesModel.cs](#).

```
00015 {get; set;}
```

6.38.2.2 RouteOfException

```
string? DotNetIdentity.Models.ViewModels.StatusCodesModel.RouteOfException [get], [set]
```

the route that thowed the exception

string

Definition at line 20 of file [StatusCodesModel.cs](#).

```
00020 {get; set;}
```

6.38.2.3 StatusCode

```
int DotNetIdentity.Models.ViewModels.StatusCodesModel.StatusCode [get], [set]
```

the status code

int

Definition at line 10 of file [StatusCodesModel.cs](#).

```
00010 {get; set;}
```

6.39 DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService Class Reference

class providing 2fa auth services

Public Member Functions

- [TwoFactorAuthenticationService](#) (IConfiguration cnf, UrlEncoder urlEncoder, ISettingsService sett)
class constructor
- string [GenerateQrCodeUri](#) (string email, string authenticatorKey)
method to generate a barcode for 2fa configuration

6.39.1 Detailed Description

class providing 2fa auth services

Definition at line 8 of file [TwoFactorAuthenticationService.cs](#).

6.39.2 Constructor & Destructor Documentation

6.39.2.1 TwoFactorAuthenticationService()

```
DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService.TwoFactorAuthenticationService  
(  
    IConfiguration cnf,  
    UrlEncoder urlEncoder,  
    ISettingsService sett ) [inline]
```

class constructor

Parameters

<i>cnf</i>	IConfiguration
<i>urlEncoder</i>	UrlEncoder
<i>sett</i>	ISettingsService

Generated by Doxygen

Definition at line 29 of file [TwoFactorAuthenticationService.cs](#).

```
00030     {
00031         _urlEncoder = urlEncoder;
00032         _configuration = cnf;
00033         _sett = sett;
00034     }
```

6.39.3 Member Function Documentation

6.39.3.1 GenerateQrCodeUri()

```
string DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService.GenerateQrCodeUri (
    string email,
    string authenticatorKey ) [inline]
```

method to generate a barcode for 2fa configuration

Parameters

<i>email</i>	string
<i>authenticatorKey</i>	string

Returns

string

Definition at line 42 of file [TwoFactorAuthenticationService.cs](#).

```
00043     {
00044         var encodedUrl = _urlEncoder.Encode(_sett.Global.ApplicationName!);
00045         var encodedEmail = _urlEncoder.Encode(email);
00046
00047         return
00048         $"otpauth://totp/{encodedUrl}:{encodedEmail}?secret={authenticatorKey}&issuer={encodedUrl}&digits=6";
00048     }
```

6.40 DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorView

Model Class Reference

model for TwoFactorAuthenticator view

Properties

- string [SharedKey](#) = default! [get, set]
the shared key
- string [AuthenticationUri](#) = default! [get, set]
the auth uri
- string [VerificationCode](#) = default! [get, set]
the otp code

6.40.1 Detailed Description

model for TwoFactorAuthenticator view

Definition at line 6 of file [TwoFactorAuthenticatorViewModel.cs](#).

6.40.2 Property Documentation

6.40.2.1 AuthenticationUri

```
string DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel.AuthenticationUri =  
default! [get], [set]
```

the auth uri

string

Definition at line 17 of file [TwoFactorAuthenticatorViewModel.cs](#).

```
00017 { get; set; } = default!;
```

6.40.2.2 SharedKey

```
string DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel.SharedKey = default!  
[get], [set]
```

the shared key

string

Definition at line 12 of file [TwoFactorAuthenticatorViewModel.cs](#).

```
00012 { get; set; } = default!;
```

6.40.2.3 VerificationCode

```
string DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel.VerificationCode =  
default! [get], [set]
```

the otp code

string

Definition at line 22 of file [TwoFactorAuthenticatorViewModel.cs](#).

```
00022 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.EnforceTwoFactorAuthenticator\(\)](#), and [DotNetIdentity.Controllers.UserContro](#)

6.41 DotNetIdentity.Models.ViewModels.TwoFactorLoginVieWModel Class Reference

the model for TwoFactorLogin view

Properties

- string [VerificationCode](#) = default! [get, set]
the otp code
- bool [IsRecoveryCode](#) [get, set]
bool if otp code is a recovery code
- [TwoFactorType](#) [TwoFactorType](#) [get, set]
the type of 2fa

6.41.1 Detailed Description

the model for TwoFactorLogin view

Definition at line 6 of file [TwoFactorLoginVieWModel.cs](#).

6.41.2 Property Documentation

6.41.2.1 IsRecoveryCode

```
bool DotNetIdentity.Models.ViewModels.TwoFactorLoginVieWModel.IsRecoveryCode [get], [set]
```

bool if otp code is a recovery code

bool

Definition at line 17 of file [TwoFactorLoginVieWModel.cs](#).

```
00017 { get; set; }
```

6.41.2.2 TwoFactorType

```
TwoFactorType DotNetIdentity.Models.ViewModels.TwoFactorLoginVieWModel.TwoFactorType [get],  
[set]
```

the type of 2fa

TwoFactorType

Definition at line 22 of file [TwoFactorLoginVieWModel.cs](#).

```
00022 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.TwoFactorLogin\(\)](#).

6.41.2.3 VerificationCode

```
string DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWModel.VerificationCode = default!  
[get], [set]
```

the otp code

string

Definition at line 12 of file [TwoFactorLoginViewWModel.cs](#).

```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.TwoFactorLogin\(\)](#).

6.42 DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel Class Reference

model for TwoFactorType view

Properties

- [TwoFactorType](#) [TwoFactorType](#) [get, set]
the 2fa type

6.42.1 Detailed Description

model for TwoFactorType view

Definition at line 6 of file [TwoFactorTypeViewModel.cs](#).

6.42.2 Property Documentation

6.42.2.1 TwoFactorType

```
TwoFactorType DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel.TwoFactorType [get],  
[set]
```

the 2fa type

TwoFactorType

Definition at line 12 of file [TwoFactorTypeViewModel.cs](#).

```
00012 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.TwoFactorType\(\)](#).

6.43 DotNetIdentity.Models.ViewModels.UpdateProfileViewModel Class Reference

model for UpdateProfile view

Properties

- string [UserName](#) = default! [get, set]
the username
- string [Email](#) = default! [get, set]
the email
- string [PhoneNumber](#) = default! [get, set]
the phone number
- [Gender](#) [Gender](#) [get, set]
the gender
- DateTime [BirthDay](#) [get, set]
the birthday
- IFormFile? [ProfilePicture](#) [get, set]
the profile picture

6.43.1 Detailed Description

model for UpdateProfile view

Definition at line 6 of file [UpdateProfileViewModel.cs](#).

6.43.2 Property Documentation

6.43.2.1 BirthDay

```
DateTime DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.BirthDay [get], [set]
```

the birthday

DateTime

Definition at line 32 of file [UpdateProfileViewModel.cs](#).

```
00032 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.43.2.2 Email

```
string DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.Email = default! [get], [set]
```

the email

string

Definition at line 17 of file [UpdateProfileViewModel.cs](#).

```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.43.2.3 Gender

```
Gender DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.Gender [get], [set]
```

the gender

Gender

Definition at line 27 of file [UpdateProfileViewModel.cs](#).

```
00027 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.43.2.4 PhoneNumber

```
string DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.PhoneNumber = default! [get],  
[set]
```

the phone number

string

Definition at line 22 of file [UpdateProfileViewModel.cs](#).

```
00022 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.43.2.5 ProfilePicture

```
IFormFile? DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.ProfilePicture [get],  
[set]
```

the profile picture

IFormFile

Definition at line 37 of file [UpdateProfileViewModel.cs](#).

```
00037 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.43.2.6 UserName

```
string DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.UserName = default! [get],  
[set]
```

the username

string

Definition at line 12 of file [UpdateProfileViewModel.cs](#).

```
00012 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.UserController.Profile\(\)](#).

6.44 DotNetIdentity.Models.ViewModels.UpsertRoleViewModel Class Reference

model for the UpsertRole view

Properties

- string? [Id](#) [get, set]
the role id
- string [Name](#) = default! [get, set]
the role name

6.44.1 Detailed Description

model for the UpsertRole view

Definition at line 6 of file [UpsertRoleViewModel.cs](#).

6.44.2 Property Documentation

6.44.2.1 Id

```
string? DotNetIdentity.Models.ViewModels.UpsertRoleViewModel.Id [get], [set]
```

the role id

string

Definition at line 12 of file [UpsertRoleViewModel.cs](#).

```
00012 { get; set; }
```

Referenced by [DotNetIdentity.Controllers.AdminController.UpsertRole\(\)](#).

6.44.2.2 Name

```
string DotNetIdentity.Models.ViewModels.UpsertRoleViewModel.Name = default! [get], [set]
```

the role name

string

Definition at line 17 of file [UpsertRoleViewModel.cs](#).

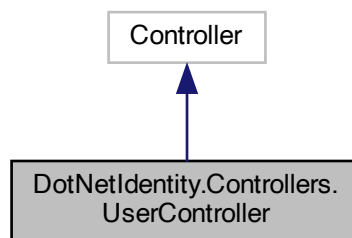
```
00017 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Controllers.AdminController.UpsertRole\(\)](#).

6.45 DotNetIdentity.Controllers.UserController Class Reference

Controller Class for User views

Inheritance diagram for DotNetIdentity.Controllers.UserController:



Public Member Functions

- [UserController](#) (IStringLocalizer< [UserController](#) > localizer, ILogger< [UserController](#) > log, IWebHostEnvironment webhostEnviroment, UserManager< [AppUser](#) > userManager, SignInManager< [AppUser](#) > signInManager, [TwoFactorAuthenticationService](#) twoFactorAuthService, [EmailHelper](#) emailHelper)
Controller constructor
- IActionResult [Register](#) ()
Controller action for Register view
- IActionResult [debugUserInfo](#) ()
Controller action for DebugUserInfo view
- IActionResult [ping](#) ()
Controller action for Ping view
- async Task< IActionResult > [Register](#) ([SignUpViewModel](#) viewModel)
Controller POST method for Register view
- async Task< IActionResult > [Login](#) (string? returnUrl)
Controller action for Login view
- async Task< IActionResult > [Login](#) ([SignInViewModel](#) viewModel)

- Controller POST method for Login view*
- async Task< IActionResult > [TwoFactorLogin](#) (string? returnUrl)
- Controller action for view TwoFactorLogin*
- async Task< IActionResult > [TwoFactorLogin](#) ([TwoFactorLoginViewWModel](#) viewWModel)
- controller POST method for TwoFactorLogin view*
- async Task [Logout](#) ()
- controller action for Logout*
- async Task< IActionResult > [Profile](#) ()
- controller action for Profile view*
- async Task< IActionResult > [Profile](#) ([UpdateProfileViewModel](#) viewModel)
- controller POST method for Profile view*
- IActionResult [ChangePassword](#) ()
- controller action for ChangePassword view*
- async Task< IActionResult > [ChangePassword](#) ([ChangePasswordViewModel](#) viewModel)
- controller POST method for ChangePassword view*
- IActionResult [ForgotPassword](#) ()
- controller action for ForgotPassword view*
- async Task< IActionResult > [ForgotPassword](#) ([ForgotPasswordViewModel](#) viewModel)
- controller POST method for ForgotPassword view*
- IActionResult [ResetPassword](#) (string userId, string token)
- controller action for ResetPassword view*
- async Task< IActionResult > [ResetPassword](#) ([ResetPasswordViewModel](#) viewModel)
- controller POST method for ResetPassword view*
- async Task< IActionResult > [ConfirmEmail](#) (string userId, string token)
- controller action for ConfirmEmail view*
- async Task< IActionResult > [TwoFactorType](#) ()
- controller action for view TwiFactorType*
- async Task< IActionResult > [TwoFactorType](#) ([TwoFactorTypeViewModel](#) viewModel)
- controller POST method for view TwoFactorType*
- async Task< IActionResult > [EnforceTwoFactorAuthenticator](#) ()
- controller action for view EnforceTwoFactorAuthenticator*
- async Task< IActionResult > [EnforceTwoFactorAuthenticator](#) ([TwoFactorAuthenticatorViewModel](#) view←
Model)
- controller POST method for view EnforceTwoFactorAuthenticator*
- async Task< IActionResult > [EnforceTwoFactorResult](#) ()
- controller action for view EnforceTwoFactorResult*
- async Task< IActionResult > [TwoFactorAuthenticator](#) ()
- controller action for view TwoFactorAuthenticator*
- async Task< IActionResult > [TwoFactorAuthenticator](#) ([TwoFactorAuthenticatorViewModel](#) viewModel)
- controller POST method for view TwoFactorAuthenticator*

6.45.1 Detailed Description

Controller Class for User views

Definition at line 19 of file [UserController.cs](#).

6.45.2 Constructor & Destructor Documentation

6.45.2.1 UserController()

```
DotNetIdentity.Controllers.UserController.UserController (
    IStringLocalizer< UserController > localizer,
    ILogger< UserController > log,
    IWebHostEnvironment webhostEnviroment,
    UserManager< AppUser > userManager,
    SignInManager< AppUser > signInManager,
    TwoFactorAuthenticationService twoFactorAuthService,
    EmailHelper emailHelper ) [inline]
```

Controller constructor

Parameters

<i>localizer</i>	type IStringLocalizer
<i>log</i>	type ILogger
<i>webhostEnviroment</i>	type IWebHostEnviroment
<i>userManager</i>	type UserManager
<i>signInManager</i>	type SignInManager
<i>twoFactorAuthService</i>	type TwoFactorAuthService
<i>emailHelper</i>	työe EmailHelper

Definition at line 60 of file [UserController.cs](#).

```
00061     {
00062         _userManager = userManager;
00063         _signInManager = signInManager;
00064         _emailHelper = emailHelper;
00065         _twoFactorAuthService = twoFactorAuthService;
00066         _hostEnvironment = webhostEnviroment;
00067         _logger = log;
00068         _localizer = localizer;
00069     }
```

6.45.3 Member Function Documentation

6.45.3.1 ChangePassword() [1/2]

```
IActionResult DotNetIdentity.Controllers.UserController.ChangePassword ( )
```

controller action for ChangePassword view

Returns

view ChangePassword

6.45.3.2 ChangePassword() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.ChangePassword (
    ChangePasswordViewModel viewModel ) [inline]
```

controller POST method for ChangePassword view

Parameters

<i>viewModel</i>	ChangePasswordViewModel
------------------	-------------------------

Returns

redirect to action Index

Definition at line 409 of file [UserController.cs](#).

```

00410     {
00411         if (ModelState.IsValid)
00412         {
00413             var me = await _userManager.FindByNameAsync(User.Identity?.Name);
00414
00415             var passwordValid = await _userManager.CheckPasswordAsync(me, viewModel.Password);
00416             if (passwordValid)
00417             {
00418                 var result = await _userManager.ChangePasswordAsync(me, viewModel.Password,
viewModel.NewPassword);
00419                 if (result.Succeeded)
00420                 {
00421                     await _userManager.UpdateSecurityStampAsync(me);
00422
00423                     await _signInManager.SignOutAsync();
00424                     await _signInManager.SignInAsync(me, true);
00425                     _logger.LogInformation("AUDIT: " + me.UserName + " successfully changed own
password!");
00426
00427                     return RedirectToAction("Index", "Admin");
00428                 }
00429                 _logger.LogWarning("AUDIT: " + me.UserName + " tried to change password. Error: " +
result.Errors.FirstOrDefault()!.ToString());
00430                 result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
f.Description));
00431             }
00432             else
00433             {
00434                 _logger.LogWarning("AUDIT: " + me.UserName + " tried to change password. But
provided invalid current password!");
00435                 ModelState.AddModelError(string.Empty, _localizer["8"]);
00436             }
00437         }
00438
00439         return View();
00440     }

```

References [DotNetIdentity.Models.ViewModels.ChangePasswordViewModel.NewPassword](#), and [DotNetIdentity.Models.ViewModels.C](#)

6.45.3.3 ConfirmEmail()

```

async Task< IActionResult > DotNetIdentity.Controllers.UserController.ConfirmEmail (
    string userId,
    string token ) [inline]

```

controller action for ConfirmEmail view

Parameters

<i>userId</i>	the user id
<i>token</i>	the confirmation token

Returns

redirect to action Index

Definition at line 545 of file [UserController.cs](#).

```

00546     {
00547         var user = await _userManager.FindByIdAsync(userId);
00548         if (user != null)
00549         {
00550             var result = await _userManager.ConfirmEmailAsync(user, token);
00551             if (result.Succeeded)
00552             {
00553                 _logger.LogInformation("AUDIT: " + user.UserName + " successfully confirmed email
address!");
00554                 return RedirectToAction("Login");
00555             }
00556         }
00557         return RedirectToAction("Index", "Home");
00558     }

```

6.45.3.4 debugUserInfo()

`ActionResult DotNetIdentity.Controllers.UserController.debugUserInfo ()`

Controller action for DebugUserInfo view

Returns

view DebugUserInfo

6.45.3.5 EnforceTwoFactorAuthenticator() [1/2]

`async Task< ActionResult > DotNetIdentity.Controllers.UserController.EnforceTwoFactorAuthenticator () [inline]`

controller action for view EnforceTwoFactorAuthenticator

Returns

view EnforceTwoFactorAuthenticator

Definition at line 602 of file [UserController.cs](#).

```

00603     {
00604         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00605         user.TwoFactorType = Models.TwoFactorType.Authenticator;
00606         await _userManager.UpdateAsync(user);
00607
00608         var authenticatorKey = await _userManager.GetAuthenticatorKeyAsync(user);
00609         if (authenticatorKey == null)
00610         {
00611             await _userManager.ResetAuthenticatorKeyAsync(user);
00612             authenticatorKey = await _userManager.GetAuthenticatorKeyAsync(user);
00613         }
00614
00615         return View(new TwoFactorAuthenticatorViewModel
00616         {
00617             SharedKey = authenticatorKey,
00618             AuthenticationUri = _twoFactorAuthService.GenerateQrCodeUri(user.Email,
authenticatorKey)
00619         });
00620     }

```

6.45.3.6 EnforceTwoFactorAuthenticator() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.EnforceTwoFactorAuthenticator
(
    TwoFactorAuthenticatorViewModel viewModel ) [inline]
```

controller POST method for view EnforceTwoFactorAuthenticator

Parameters

<i>viewModel</i>	type TwoFactorAuthenticatorViewModel
------------------	--------------------------------------

Returns

redirect to action EnforceTwoFactorResult

Definition at line 628 of file [UserController.cs](#).

```
00629     {
00630         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00631
00632         var verificationCode = viewModel.VerificationCode.Replace(" ", string.Empty).Replace("-",
string.Empty);
00633
00634         var isValidToken = await _userManager.VerifyTwoFactorTokenAsync(user,
_userManager.Options.Tokens.AuthenticatorTokenProvider, verificationCode);
00635         if (isValidToken)
00636         {
00637             if (user.IsMfaForce && user.TwoFactorEnabled==false) {
00638                 await _userManager.SetTwoFactorEnabledAsync(user, true);
00639                 user.IsMfaForce = false;
00640                 await _userManager.UpdateAsync(user);
00641             }
00642
00643             TempData["RecoveryCodes"] = await
_userManager.GenerateNewTwoFactorRecoveryCodesAsync(user, 5);
00644         }
00645         _logger.LogInformation("AUDIT: " + user.UserName + " successfully configured 2fa after
enforcement. Redirecting to action: EnforceTwoFactorResult");
00646         return RedirectToAction("EnforceTwoFactorResult", "User");
00647     }
```

References [DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel.VerificationCode](#).

6.45.3.7 EnforceTwoFactorResult()

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.EnforceTwoFactorResult (
) [inline]
```

controller action for view EnforceTwoFactorResult

Returns

view EnforceTwoFactorResult

Definition at line 653 of file [UserController.cs](#).

```
00653     {
00654         await _signInManager.SignOutAsync();
00655         return View();
00656     }
```

6.45.3.8 ForgotPassword() [1/2]

`ActionResult DotNetIdentity.Controllers.UserController.ForgotPassword ()`

controller action for ForgotPassword view

Returns

view ForgotPassword

6.45.3.9 ForgotPassword() [2/2]

`async Task< ActionResult > DotNetIdentity.Controllers.UserController.ForgotPassword (ForgotPasswordViewModel viewModel) [inline]`

controller POST method for ForgotPassword view

Parameters

<i>viewModel</i>	ForgotPasswordViewModel
------------------	-------------------------

Returns

redirect to action Index

Definition at line 454 of file [UserController.cs](#).

```

00455     {
00456         if (ModelState.IsValid)
00457         {
00458             var user = await _userManager.FindByEmailAsync(viewModel.Email);
00459             if (user != null)
00460             {
00461                 var passwordResetToken = await _userManager.GeneratePasswordResetTokenAsync(user);
00462
00463                 var passwordLink = Url.Action("ResetPassword", "User", new
00464                 {
00465                     userId = user.Id,
00466                     token = passwordResetToken
00467                 }, HttpContext.Request.Scheme);
00468
00469                 await _emailHelper.SendAsync(new()
00470                 {
00471                     Subject = "Reset password",
00472                     Body = $"Please <a href='{passwordLink}'>click</a> to reset your password.",
00473                     To = user.Email
00474                 });
00475                 _logger.LogWarning("AUDIT: " + user.UserName + " requested a password reset!");
00476                 return RedirectToAction("Index", "Home");
00477             }
00478             else
00479             {
00480                 ModelState.AddModelError(string.Empty, _localizer["9"]);
00481             }
00482         }
00483         return View(viewModel);
00484     }

```

References [DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel.Email](#).

6.45.3.10 Login() [1/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.Login (
    SignInViewModel viewModel ) [inline]
```

Controller POST method for Login view

Todo Implement Ldap Login

Returns

Redirect to action

Definition at line 161 of file [UserController.cs](#).

```
00162     {
00163         if (ModelState.IsValid)
00164         {
00165             var user = await _userManager.FindByNameAsync(viewModel.UserName);
00166             if (user != null)
00167             {
00168                 await _signInManager.SignOutAsync();
00169
00170                 var result = await _signInManager.PasswordSignInAsync(user, viewModel.Password,
viewModel.RememberMe, true);
00171                 if (result.Succeeded && user.IsEnabled==false) {
00172                     await _signInManager.SignOutAsync();
00173                     _logger.LogWarning("AUDIT: " + user.UserName + " logged in successfully. But
account is disabled!");
00174                     ModelState.AddModelError(string.Empty, _localizer["1"]);
00175                 }
00176                 else if (result.Succeeded && user.IsMfaForce==true) {
00177                     await _userManager.ResetAccessFailedCountAsync(user);
00178                     await _userManager.SetLockoutEndDateAsync(user, null);
00179                     _logger.LogInformation("AUDIT: " + user.UserName + " logged in successfully.
But MFA is enforced. Redirecting to action: EnforceTwoFactorAuthenticator");
00180                     return RedirectToAction("EnforceTwoFactorAuthenticator");
00181                 }
00182                 else if (result.Succeeded && user.IsMfaForce==false)
00183                 {
00184                     await _userManager.ResetAccessFailedCountAsync(user);
00185                     await _userManager.SetLockoutEndDateAsync(user, null);
00186                     _logger.LogInformation("AUDIT: " + user.UserName + " logged in
successfully.");
00187                     var returnUrl = TempData["ReturnUrl"];
00188                     if (returnUrl != null)
00189                     {
00190                         return Redirect(returnUrl.ToString() ?? "/");
00191                     }
00192                     return RedirectToAction("Index", "Admin");
00193                 }
00194                 else if (result.RequiresTwoFactor)
00195                 {
00196                     _logger.LogInformation("AUDIT: " + user.UserName + " logged in successfully.
But MFA is enabled. Redirecting to action: TwoFactorLogin");
00197                     return RedirectToAction("TwoFactorLogin", new { ReturnUrl =
TempData["ReturnUrl"] });
00198                 }
00199                 else if (result.IsLockedOut)
00200                 {
00201                     var lockoutEndUtc = await _userManager.GetLockoutEndDateAsync(user);
00202                     var timeLeft = lockoutEndUtc.Value - DateTime.UtcNow;
00203                     _logger.LogWarning("AUDIT: " + user.UserName + " logged in successfully. But
Account is locked out for at least another " + timeLeft.Minutes + "minutes!");
00204                     ModelState.AddModelError(string.Empty, _localizer["2",
timeLeft.Minutes.ToString()]);
00205                 }
00206                 else if (result.IsNotAllowed)
00207                 {
00208                     _logger.LogWarning("AUDIT: " + user.UserName + " logged in successfully. But
email is not confirmed!");
00209                     ModelState.AddModelError(string.Empty, _localizer["3"]);
00210                 }
00211                 else
00212                 {
00213                     _logger.LogWarning("AUDIT: " + user.UserName + " login in failed. Invalid
username or password!");
00214                     ModelState.AddModelError(string.Empty, _localizer["4"]);
```

```
00215         }
00216     }
00217     else
00218     {
00219         _logger.LogWarning("AUDIT: " + "Login in failed. Invalid username or password!");
00220         ModelState.AddModelError(string.Empty, _localizer["4"]);
00221     }
00222 }
00223 return View(viewModel);
00224 }
```

References [DotNetIdentity.Models.ViewModels.SignInViewModel.Password](#), [DotNetIdentity.Models.ViewModels.SignInViewModel.Password](#) and [DotNetIdentity.Models.ViewModels.SignInViewModel.UserName](#).

6.45.3.11 Login() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.Login (
    string? returnUrl ) [inline]
```

Controller action for Login view

Parameters

<i>returnUrl</i>	
------------------	--

Returns

view Login

Definition at line 145 of file [UserController.cs](#).

```
00146     {
00147         await _signInManager.SignOutAsync();
00148         if (returnUrl != null)
00149         {
00150             TempData["ReturnUrl"] = returnUrl;
00151         }
00152         return View();
00153     }
```

6.45.3.12 Logout()

```
async Task DotNetIdentity.Controllers.UserController.Logout ( )
```

controller action for Logout

Returns

6.45.3.13 ping()

```
IActionResult DotNetIdentity.Controllers.UserController.ping ( )
```

Controller action for Ping view

Returns

view Ping

6.45.3.14 Profile() [1/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.Profile ( ) [inline]
```

controller action for Profile view

Returns

view Profile

Definition at line 304 of file [UserController.cs](#).

```
00305     {
00306         var me = await _userManager.FindByNameAsync(User.Identity?.Name);
00307         if (me == null)
00308         {
00309             await _signInManager.SignOutAsync();
00310             return RedirectToAction("Index", "Home");
00311         }
00312         var mod = new UpdateProfileViewModel();
00313         mod.Email = me.Email;
00314         mod.Birthday = me.Birthday;
00315         mod.UserName = me.UserName;
00316         mod.PhoneNumber = me.PhoneNumber;
00317         mod.Gender = me.Gender;
00318         //return View(me.Adapt<UpdateProfileViewModel>());
00319         return View(mod);
00320     }
```

6.45.3.15 Profile() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.Profile (
    UpdateProfileViewModel viewModel ) [inline]
```

controller POST method for Profile view

Parameters

<i>viewModel</i>	type UpdateProfileViewModel
------------------	-----------------------------

Returns

redirect to action Index

Definition at line 329 of file [UserController.cs](#).

```

00330     {
00331         if (ModelState.IsValid)
00332         {
00333             var me = await _userManager.FindByNameAsync(User.Identity?.Name);
00334             if (me != null)
00335             {
00336                 if (me.PhoneNumber != viewModel.PhoneNumber && _userManager.Users.Any(a =>
00337                     a.PhoneNumber == viewModel.PhoneNumber))
00338                 {
00339                     ModelState.AddModelError(string.Empty, _localizer["6"]);
00340                 }
00341                 else
00342                 {
00343                     if (viewModel.ProfilePicture != null) {
00344                         using (var memoryStream = new MemoryStream())
00345                         {
00346                             await viewModel.ProfilePicture.CopyToAsync(memoryStream);
00347
00348                             // Upload the file if less than 2 MB
00349                             if (memoryStream.Length < 2097152)
00350                             {
00351                                 var format = "image/png";
00352                                 var Content = memoryStream.ToArray();
00353                                 var imageData =
00354                                     $"data:{format};base64,{Convert.ToBase64String(Content)}";
00355                                 me.ProfilePicture = imageData;
00356                             }
00357                             else
00358                             {
00359                                 ModelState.AddModelError("File", _localizer["7"]);
00360                             }
00361                         } else {
00362                             me.ProfilePicture = null;
00363                         }
00364
00365                         me.UserName = viewModel.UserName;
00366                         me.Email = viewModel.Email;
00367                         me.PhoneNumber = viewModel.PhoneNumber;
00368                         me.Gender = viewModel.Gender;
00369                         me.BirthDay = viewModel.BirthDay;
00370                         //me.Department = viewModel.Department.ToString();
00371
00372                         var result = await _userManager.UpdateAsync(me);
00373                         if (result.Succeeded)
00374                         {
00375                             await _userManager.UpdateSecurityStampAsync(me);
00376                             await _signInManager.SignOutAsync();
00377                             await _signInManager.SignInAsync(me, true);
00378
00379                             _logger.LogInformation("AUDIT: " + me.UserName + "successfully updated own
00380 user profile");
00381
00382                             return RedirectToAction("Index", "Home");
00383                         }
00384                         result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
00385 f.Description));
00386                     }
00387                     else
00388                     {
00389                         await _signInManager.SignOutAsync();
00390                         return RedirectToAction("Index", "Home");
00391                     }
00392                 }
00393                 return View(viewModel);
00394             }
00395         }

```

References [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.BirthDay](#), [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.Email](#), [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.Gender](#), [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.PhoneNumber](#), [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.ProfilePicture](#), and [DotNetIdentity.Models.ViewModels.UpdateProfileViewModel.UserName](#).

6.45.3.16 Register() [1/2]

`IActionResult DotNetIdentity.Controllers.UserController.Register ()`

Controller action for Register view

Returns

view Register

6.45.3.17 Register() [2/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.Register (
    SignUpViewModel viewModel ) [inline]
```

Controller POST method for Register view

Parameters

viewModel	
-----------	--

Returns

Redirect to action Login

Definition at line 97 of file [UserController.cs](#).

```
00098     {
00099         if (ModelState.IsValid)
00100         {
00101             var user = new AppUser
00102             {
00103                 FirstName = viewModel.FirstName,
00104                 LastName = viewModel.LastName,
00105                 UserName = viewModel.UserName,
00106                 Email = viewModel.Email,
00107                 Gender = viewModel.Gender,
00108                 BirthDay = viewModel.BirthDay,
00109                 Department = viewModel.Department.ToString(),
00110                 TwoFactorType = Models.TwoFactorType.None,
00111                 CreatedOn = DateTime.UtcNow,
00112                 IsEnabled = true
00113             };
00114
00115             var result = await _userManager.CreateAsync(user, viewModel.Password);
00116             if (result.Succeeded)
00117             {
00118                 await _userManager.AddToRoleAsync(user, "User");
00119                 var confirmationToken = await
00120                 _userManager.GenerateEmailConfirmationTokenAsync(user);
00121                 var confirmationLink = Url.Action("ConfirmEmail", "User", new
00122                 {
00123                     userId = user.Id,
00124                     token = confirmationToken
00125                 }, HttpContext.Request.Scheme);
00126                 await _emailHelper.SendAsync(new()
00127                 {
00128                     Subject = "Confirm e-mail",
00129                     Body = $"Please <a href='{confirmationLink}'>click</a> to confirm your e-mail
00130 address.",
00131                     To = user.Email
00132                 });
00133                 _logger.LogInformation("AUDIT: " + "registered successfully new user: " +
00134                 user.UserName);
00135                 return RedirectToAction("Login");
00136             }
00137             result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
00138             f.Description));
00139             return View(viewModel);
00140         }
```

References [DotNetIdentity.Models.ViewModels.SignUpViewModel.BirthDay](#), [DotNetIdentity.Models.ViewModels.SignUpViewModel.D](#), [DotNetIdentity.Models.ViewModels.SignUpViewModel.Email](#), [DotNetIdentity.Models.ViewModels.SignUpViewModel.FirstName](#),

[DotNetIdentity.Models.ViewModels.SignUpViewModel.Gender](#), [DotNetIdentity.Models.ViewModels.SignUpViewModel.LastName](#), [DotNetIdentity.Models.ViewModels.SignUpViewModel.Password](#), and [DotNetIdentity.Models.ViewModels.SignUpViewModel.UserName](#)

6.45.3.18 ResetPassword() [1/2]

```
async Task< IActionResult > DotNetIdentity.Controllers.UserController.ResetPassword (
    ResetPasswordViewModel viewModel ) [inline]
```

controller POST method for ResetPassword view

Parameters

<i>viewModel</i>	type ResetPasswordViewModel
------------------	-----------------------------

Returns

redirect to action Login

Definition at line 512 of file [UserController.cs](#).

```
00513     {
00514         if (ModelState.IsValid)
00515         {
00516             var user = await _userManager.FindByIdAsync(viewModel.UserId);
00517             if (user != null)
00518             {
00519                 var result = await _userManager.ResetPasswordAsync(user, viewModel.Token,
00520                     viewModel.Password);
00521                 if (result.Succeeded)
00522                 {
00523                     await _userManager.UpdateSecurityStampAsync(user);
00524                     _logger.LogInformation("AUDIT: " + user.UserName + "successfully resetted own
00525                     password.");
00526                     return RedirectToAction("Login", "User");
00527                 }
00528                 else
00529                 {
00530                     result.Errors.ToList().ForEach(f => ModelState.AddModelError(string.Empty,
00531                         f.Description));
00532                 }
00533             }
00534             else
00535             {
00536                 ModelState.AddModelError(string.Empty, "User not found.");
00537             }
00538         }
00539         return View(viewModel);
00540     }
```

References [DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.Password](#), [DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.Token](#), and [DotNetIdentity.Models.ViewModels.ResetPasswordViewModel.UserId](#).

6.45.3.19 ResetPassword() [2/2]

```
IActionResult DotNetIdentity.Controllers.UserController.ResetPassword (
    string userId,
    string token ) [inline]
```

controller action for ResetPassword view

Parameters

<i>userId</i> ↔	the user id
<i>token</i>	the reset token

Returns

view ResetPassword

Definition at line 492 of file [UserController.cs](#).

```

00493     {
00494         if (userId == null || token == null)
00495         {
00496             return RedirectToAction("Login", "User");
00497         }
00498
00499         return View(new ResetPasswordViewModel
00500         {
00501             UserId = userId,
00502             Token = token
00503         });
00504     }

```

6.45.3.20 TwoFactorAuthenticator() [1/2]

```

async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorAuthenticator (
) [inline]

```

controller action for view TwoFactorAuthenticator

Returns

view TwoFactorAuthenticator

Definition at line 662 of file [UserController.cs](#).

```

00663     {
00664         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00665         if (user.TwoFactorEnabled && user.TwoFactorType == Models.TwoFactorType.Authenticator)
00666         {
00667             var authenticatorKey = await _userManager.GetAuthenticatorKeyAsync(user);
00668             if (authenticatorKey == null)
00669             {
00670                 await _userManager.ResetAuthenticatorKeyAsync(user);
00671                 authenticatorKey = await _userManager.GetAuthenticatorKeyAsync(user);
00672             }
00673
00674             return View(new TwoFactorAuthenticatorViewModel
00675             {
00676                 SharedKey = authenticatorKey,
00677                 AuthenticationUri = _twoFactorAuthService.GenerateQrCodeUri(user.Email,
authenticatorKey)
00678             });
00679         }
00680         return RedirectToAction("Index", "Home");
00681     }

```

6.45.3.21 TwoFactorAuthenticator() [2/2]

```

async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorAuthenticator (
    TwoFactorAuthenticatorViewModel viewModel ) [inline]

```

controller POST method for view TwoFactorAuthenticator

Parameters

<i>viewModel</i>	type TwoFactorAuthenticatorViewModel
------------------	--------------------------------------

Returns

redirect to action TwoFactorType

Definition at line 689 of file [UserController.cs](#).

```

00690     {
00691         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00692
00693         var verificationCode = viewModel.VerificationCode.Replace(" ", string.Empty).Replace("-",
string.Empty);
00694
00695         var isValidToken = await _userManager.VerifyTwoFactorTokenAsync(user,
_userManager.Options.Tokens.AuthenticatorTokenProvider, verificationCode);
00696         if (isValidToken)
00697         {
00698             if (user.IsMfaForce && user.TwoFactorEnabled==false) {
00699                 await _userManager.SetTwoFactorEnabledAsync(user, true);
00700                 user.IsMfaForce = false;
00701                 await _userManager.UpdateAsync(user);
00702             }
00703
00704             TempData["RecoveryCodes"] = await
_userManager.GenerateNewTwoFactorRecoveryCodesAsync(user, 5);
00705         }
00706         _logger.LogInformation("AUDIT: " + user.UserName + " successfully configured 2fa!");
00707         return RedirectToAction("TwoFactorType", "User");
00708     }

```

References [DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel.VerificationCode](#).

6.45.3.22 TwoFactorLogin() [1/2]

```

async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorLogin (
    string? returnUrl ) [inline]

```

Controller action for view TwoFactorLogin

Parameters

<i>returnUrl</i>	optional return url as string
------------------	-------------------------------

Returns

view TwoFactorLogin

Definition at line 231 of file [UserController.cs](#).

```

00232     {
00233         if (returnUrl != null)
00234         {
00235             TempData["ReturnUrl"] = returnUrl;
00236         }
00237
00238         var user = await _signInManager.GetTwoFactorAuthenticationUserAsync();
00239
00240         if (user.TwoFactorType == DotNetIdentity.Models.TwoFactorType.Email) {
00241             var token = await _userManager.GenerateTwoFactorTokenAsync(user, "Email");
00242             await _emailHelper.SendAsync(new()

```

```

00243         {
00244             Subject = "Your 2fa code",
00245             Body = $"Please use this " + token + " 2fa code to verify your login.",
00246             To = user.Email
00247         });
00248         _logger.LogInformation("AUDIT: send otp code to " + user.Email);
00249     }
00250
00251     return View(new TwoFactorLoginVieWModel
00252     {
00253         TwoFactorType = user.TwoFactorType,
00254     });
00255 }

```

References [DotNetIdentity.Models.ViewModels.TwoFactorLoginVieWModel.TwoFactorType](#).

6.45.3.23 TwoFactorLogin() [2/2]

```

async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorLogin (
    TwoFactorLoginVieWModel viewModel ) [inline]

```

controller POST method for TwoFactorLogin view

Parameters

<i>viewModel</i>	type TwoFactorLoginVieWModel
------------------	------------------------------

Returns

rediret to return url

Definition at line 263 of file [UserController.cs](#).

```

00264     {
00265         var user = await _signInManager.GetTwoFactorAuthenticationUserAsync();
00266
00267         if (user.TwoFactorType == Models.TwoFactorType.Authenticator)
00268         {
00269             var result = viewModel.IsRecoveryCode ? await
                _signInManager.TwoFactorRecoveryCodeSignInAsync(viewModel.VerificationCode) : await
                _signInManager.TwoFactorAuthenticatorSignInAsync(viewModel.VerificationCode, true, false);
00270             if (result.Succeeded)
00271             {
00272                 _logger.LogInformation("AUDIT: " + user.UserName + "Authenticator 2fa verified
successfully");
00273                 return Redirect(TempData["ReturnUrl"]?.ToString() ?? "/");
00274             }
00275             _logger.LogWarning("AUDIT: " + user.UserName + "Authenticator 2fa verification failed.
Invalid otp code");
00276             ModelState.AddModelError(string.Empty, _localizer["5"]);
00277         }
00278         else if (user.TwoFactorType == Models.TwoFactorType.Email)
00279         {
00280             var result = viewModel.IsRecoveryCode ? await
                _signInManager.TwoFactorRecoveryCodeSignInAsync(viewModel.VerificationCode) : await
                _signInManager.TwoFactorSignInAsync("Email", viewModel.VerificationCode, true, false);
00281             if (result.Succeeded)
00282             {
00283                 _logger.LogInformation("AUDIT: " + user.UserName + "Email 2fa verified
successfully");
00284                 return Redirect(TempData["ReturnUrl"]?.ToString() ?? "/");
00285             }
00286             _logger.LogWarning("AUDIT: " + user.UserName + "Email 2fa verification failed. Invalid
otp code");
00287             ModelState.AddModelError(string.Empty, _localizer["5"]);
00288         }
00289
00290         return View(viewModel);
00291     }

```

References [DotNetIdentity.Models.ViewModels.TwoFactorLoginVieWModel.VerificationCode](#).

6.45.3.24 TwoFactorType() [1/2]

`async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorType () [inline]`

controller action for view TwiFactorType

Returns

view TwoFactorTaype

Definition at line 565 of file [UserController.cs](#).

```
00566     {
00567         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00568
00569         return View(new TwoFactorTypeViewModel
00570         {
00571             TwoFactorType = user.TwoFactorType
00572         });
00573     }
```

6.45.3.25 TwoFactorType() [2/2]

`async Task< IActionResult > DotNetIdentity.Controllers.UserController.TwoFactorType (TwoFactorTypeViewModel viewModel) [inline]`

controller POST method for view TwoFactorType

Parameters

<i>viewModel</i>	TwoFactorTypeViewModel
------------------	------------------------

Returns

redireoct to action TwoFactorType

Definition at line 582 of file [UserController.cs](#).

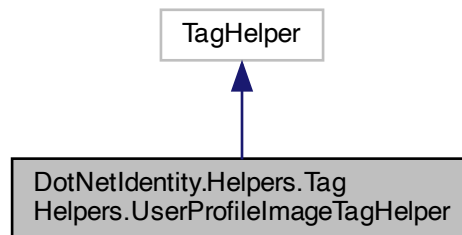
```
00583     {
00584         var user = await _userManager.FindByNameAsync(User.Identity?.Name);
00585         user.TwoFactorType = viewModel.TwoFactorType;
00586         await _userManager.UpdateAsync(user);
00587         await _userManager.SetTwoFactorEnabledAsync(user, user.TwoFactorType !=
Models.TwoFactorType.None);
00588
00589         if (viewModel.TwoFactorType == Models.TwoFactorType.Authenticator)
00590         {
00591             _logger.LogInformation("AUDIT: " + user.UserName + " set authenticator as 2fa method.
Redirecting to action: TwoFactorAuthenticator");
00592             return RedirectToAction("TwoFactorAuthenticator", "User");
00593         }
00594         _logger.LogInformation("AUDIT: " + user.UserName + " set " + user.TwoFactorType.ToString()
+ " as 2fa method.");
00595         return RedirectToAction("TwoFactorType", "User");
00596     }
```

References [DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel.TwoFactorType](#).

6.46 DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper Class Reference

html taghelper class

Inheritance diagram for DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper:



Public Member Functions

- [UserProfileImageTagHelper](#) (UserManager< [AppUser](#) > userManager)
class constructor
- override async Task [ProcessAsync](#) (TagHelperContext context, TagHelperOutput output)
override ProcessAsync

Properties

- string [UserId](#) = default! [get, set]
html attribute definition

6.46.1 Detailed Description

html taghelper class

Definition at line 12 of file [UserProfileImageTagHelper.cs](#).

6.46.2 Constructor & Destructor Documentation

6.46.2.1 UserProfileImageTagHelper()

```
DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper.UserProfileImageTagHelper (
    UserManager< AppUser > userManager ) [inline]
```

class constructor

Parameters

<i>userManager</i>	type UserManager
--------------------	------------------

Definition at line 23 of file [UserProfileImageTagHelper.cs](#).

```
00024     {  
00025         _userManager = userManager;  
00026     }
```

6.46.3 Member Function Documentation

6.46.3.1 ProcessAsync()

```
override async Task DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper.ProcessAsync (  
    TagHelperContext context,  
    TagHelperOutput output ) [inline]
```

override ProcessAsync

Parameters

<i>context</i>	type TagHelperContext
<i>output</i>	type TagHelperOutput

Returns

TagHelperOutput

Definition at line 41 of file [UserProfileImageTagHelper.cs](#).

```
00042     {  
00043         var user = await _userManager.FindByIdAsync(UserId);  
00044         var sb = new StringBuilder();  
00045         if(user.ProfilePicture==null) {  
00046             sb.Append($"<img src='/assets/images/faces/user.png' height='32' />");  
00047         } else {  
00048             sb.Append($"<img src='" + user.ProfilePicture + "' height='32' />");  
00049         }  
00050     }  
00051     output.Content.SetHtmlContent(sb.ToString());  
00052 }  
00053 }
```

References [DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper.UserId](#).

6.46.4 Property Documentation

6.46.4.1 UserId

```
string DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper.UserId = default! [get],
[set]
```

html attribute definition

Definition at line 33 of file [UserProfileImageTagHelper.cs](#).

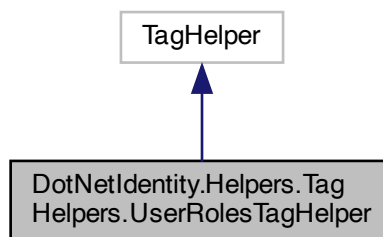
```
00033 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper.ProcessAsync\(\)](#).

6.47 DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper Class Reference

html tag helper class

Inheritance diagram for DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper:



Public Member Functions

- [UserRolesTagHelper](#) (userManager< [AppUser](#) > userManager)
class constructor
- override async Task [ProcessAsync](#) (TagHelperContext context, TagHelperOutput output)
method to override ProcessAsync

Properties

- string [UserId](#) = default! [get, set]
attribute definition

6.47.1 Detailed Description

html tag helper class

Definition at line 12 of file [UserRolesTagHelper.cs](#).

6.47.2 Constructor & Destructor Documentation

6.47.2.1 UserRolesTagHelper()

```
DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper.UserRolesTagHelper (
    UserManager< AppUser > userManager ) [inline]
```

class constructor

Parameters

<i>userManager</i>	
--------------------	--

Definition at line 23 of file [UserRolesTagHelper.cs](#).

```
00024     {
00025         _userManager = userManager;
00026     }
```

6.47.3 Member Function Documentation

6.47.3.1 ProcessAsync()

```
override async Task DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper.ProcessAsync (
    TagHelperContext context,
    TagHelperOutput output ) [inline]
```

method to override ProcessAsync

Parameters

<i>context</i>	type TagHelperContext
<i>output</i>	type TagHelperOutput

Returns

TagHelperOutput

Definition at line 41 of file [UserRolesTagHelper.cs](#).

```
00042     {
00043         var user = await _userManager.FindByIdAsync(UserId);
00044         var roles = await _userManager.GetRolesAsync(user);
00045
00046         var sb = new StringBuilder();
00047
00048         if (roles.Count() > 0)
00049         {
00050             foreach (var role in roles)
00051             {
```

```

00052         sb.Append($"<span class='badge rounded-pill bg-danger'>{role}</span>");
00053     }
00054 }
00055 else
00056 {
00057     sb.Append("No roles found");
00058 }
00059
00060 output.Content.SetHtmlContent(sb.ToString());
00061 }

```

References [DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper.UserId](#).

6.47.4 Property Documentation

6.47.4.1 UserId

```
string DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper.UserId = default! [get], [set]
```

attribute definition

Definition at line 33 of file [UserRolesTagHelper.cs](#).

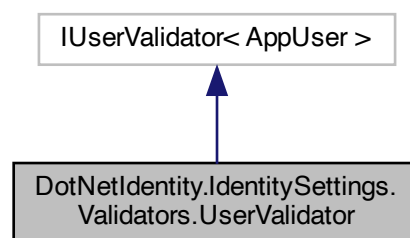
```
00033 { get; set; } = default!;
```

Referenced by [DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper.ProcessAsync\(\)](#).

6.48 DotNetIdentity.IdentitySettings.Validators.UserValidator Class Reference

class to validate IdentityUser

Inheritance diagram for DotNetIdentity.IdentitySettings.Validators.UserValidator:



Public Member Functions

- Task< IdentityResult > [ValidateAsync](#) (userManager< [AppUser](#) > manager, [AppUser](#) user)
Task to validate IdentityUser

6.48.1 Detailed Description

class to validate IdentityUser

Definition at line 9 of file [UserValidator.cs](#).

6.48.2 Member Function Documentation

6.48.2.1 ValidateAsync()

```
Task< IdentityResult > DotNetIdentity.IdentitySettings.Validators.UserValidator.ValidateAsync
(
    UserManager< AppUser > manager,
    AppUser user ) [inline]
```

Task to validate IdentityUser

Parameters

<i>manager</i>	UserManager
<i>user</i>	AppUser

Returns

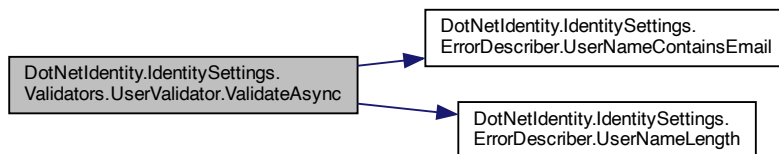
Task of type IdentityResult

Definition at line 17 of file [UserValidator.cs](#).

```
00018     {
00019         var errors = new List<IdentityError>();
00020
00021         if (user.UserName.Length < 6)
00022         {
00023             errors.Add(ErrorDescriber.UserNameLength());
00024         }
00025         if (user.Email.Substring(0, user.Email.IndexOf("@")) == user.UserName)
00026         {
00027             errors.Add(ErrorDescriber.UserNameContainsEmail());
00028         }
00029
00030         if (errors.Any())
00031         {
00032             return Task.FromResult(IdentityResult.Failed(errors.ToArray()));
00033         }
00034         return Task.FromResult(IdentityResult.Success);
00035     }
```

References [DotNetIdentity.IdentitySettings.ErrorDescriber.UserNameContainsEmail\(\)](#), and [DotNetIdentity.IdentitySettings.ErrorDescriber.UserNameLength\(\)](#).

Here is the call graph for this function:



Index

AbsoluteExpiration
 DotNetIdentity.Models.DataModels.SessionCache, [93](#)

AccessDenied
 DotNetIdentity.Controllers.HomeController, [74](#)

Accounting
 DotNetIdentity.Models, [13](#)

AdminController
 DotNetIdentity.Controllers.AdminController, [20](#)

AppDbContext
 DotNetIdentity.Data.AppDbContext, [35](#)

ApplicationName
 DotNetIdentity.Models.BusinessModels.GlobalSettings, [71](#)

AppLogs
 DotNetIdentity.Data.AppDbContext, [36](#)

AppSessionCache
 DotNetIdentity.Data.AppDbContext, [37](#)

AppSettings
 DotNetIdentity.Data.AppDbContext, [37](#)

AppSettingsBase
 DotNetIdentity.Services.SettingsService.AppSettingsBase, [42](#)

Articles
 DotNetIdentity.Controllers.RoleBasedController, [91](#)

AuditLogs
 DotNetIdentity.Controllers.AdminController, [21](#)

AuthenticationUri
 DotNetIdentity.Models.ViewModels.TwoFactorAuthenticationViewModel, [109](#)

BirthDay
 DotNetIdentity.Models.Identity.AppUser, [45](#)
 DotNetIdentity.Models.ViewModels.SignUpViewModel, [103](#)
 DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, [112](#)

Body
 DotNetIdentity.Models.BusinessModels.EmailModel, [62](#)

ChangePassword
 DotNetIdentity.Controllers.UserController, [117](#)

ClaimsTransformation
 DotNetIdentity.IdentitySettings.ClaimsTransformation, [54](#)

ConfirmEmail
 DotNetIdentity.Controllers.UserController, [118](#)

CreatedOn
 DotNetIdentity.Models.Identity.AppRole, [41](#)
 DotNetIdentity.Models.Identity.AppUser, [45](#)

Dashboard
 DotNetIdentity.Controllers.RoleBasedController, [91](#)

debugUserInfo
 DotNetIdentity.Controllers.UserController, [119](#)

DeleteRole
 DotNetIdentity.Controllers.AdminController, [21](#)

DeleteUser
 DotNetIdentity.Controllers.AdminController, [22](#)

Department
 DotNetIdentity.Models, [13](#)
 DotNetIdentity.Models.Identity.AppUser, [46](#)
 DotNetIdentity.Models.ViewModels.EditUserViewModel, [57](#)
 DotNetIdentity.Models.ViewModels.NewUserModel, [85](#)
 DotNetIdentity.Models.ViewModels.SignUpViewModel, [103](#)

DisableLdapLogin
 DotNetIdentity.Controllers.AdminController, [22](#)

DisableMfaForce
 DotNetIdentity.Controllers.AdminController, [23](#)

DisableUser
 DotNetIdentity.Controllers.AdminController, [23](#)

DotNetIdentity, [11](#)

DotNetIdentity.Controllers, [11](#)

DotNetIdentity.Controllers.AdminController, [19](#)
 AdminController, [20](#)
 AuditLogs, [21](#)
 DeleteRole, [21](#)
 DeleteUser, [22](#)
 DisableLdapLogin, [22](#)
 DisableMfaForce, [23](#)
 DisableUser, [23](#)
 EditUser, [24](#), [26](#)
 EnableLdapLogin, [26](#)
 EnableMfaForce, [27](#)
 EnableUser, [27](#)
 ErrorLogs, [28](#)
 GetAppLogs, [28](#)
 GetAuditLogs, [29](#)
 GetErrorLogs, [29](#)
 GetUsers, [29](#)
 Index, [30](#)
 NewUser, [30](#)
 Roles, [31](#)
 SystemLogs, [32](#)
 UpsertRole, [32](#)
 Users, [33](#)

- DotNetIdentity.Controllers.ClaimBasedController, 51
 - Employee, 52
 - FreeTrial, 52
 - HR, 52
 - Software, 52
 - Violence, 53
- DotNetIdentity.Controllers.CultureController, 55
 - SetCulture, 56
- DotNetIdentity.Controllers.ErrorsController, 65
 - ErrorCd, 65
 - ErrorEx, 66
- DotNetIdentity.Controllers.HomeController, 73
 - AccessDenied, 74
 - HomeController, 73
 - Index, 74
- DotNetIdentity.Controllers.RoleBasedController, 91
 - Articles, 91
 - Dashboard, 91
 - Profile, 92
 - Users, 92
- DotNetIdentity.Controllers.SettingsController, 94
 - SettingsApp, 96
 - SettingsController, 95
 - SettingsLdap, 96
 - SettingsMail, 97, 98
- DotNetIdentity.Controllers.UserController, 115
 - ChangePassword, 117
 - ConfirmEmail, 118
 - debugUserInfo, 119
 - EnforceTwoFactorAuthenticator, 119
 - EnforceTwoFactorResult, 120
 - ForgotPassword, 120, 121
 - Login, 121, 123
 - Logout, 123
 - ping, 123
 - Profile, 124
 - Register, 125, 126
 - ResetPassword, 127
 - TwoFactorAuthenticator, 128
 - TwoFactorLogin, 129, 130
 - TwoFactorType, 130, 131
 - UserController, 116
- DotNetIdentity.Data, 11
- DotNetIdentity.Data.AppDbContext, 34
 - AppDbContext, 35
 - AppLogs, 36
 - AppSessionCache, 37
 - AppSettings, 37
 - OnModelCreating, 35
- DotNetIdentity.Helpers, 12
- DotNetIdentity.Helpers.EmailHelper, 60
 - EmailHelper, 60
 - SendAsync, 61
- DotNetIdentity.Helpers.TagHelpers, 12
- DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper, To, 62
 - 132
 - ProcessAsync, 133
 - UserId, 133
 - UserProfileImageTagHelper, 132
 - DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper, 134
 - ProcessAsync, 135
 - UserId, 136
 - UserRolesTagHelper, 135
- DotNetIdentity.IdentitySettings, 12
- DotNetIdentity.IdentitySettings.ClaimsTransformation, 53
 - ClaimsTransformation, 54
 - TransformAsync, 54
- DotNetIdentity.IdentitySettings.ErrorDescriber, 63
 - PasswordContainsUsername, 64
 - UserNameContainsEmail, 64
 - UserNameLength, 64
- DotNetIdentity.IdentitySettings.Requirements, 12
- DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireHandler, 69
 - HandleRequirementAsync, 69
- DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpireRequirement, 70
- DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler, 81
 - HandleRequirementAsync, 82
- DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement, 83
 - MinimumAge, 84
 - MinimumAgeRequirement, 83
- DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService, 107
 - GenerateQrCodeUri, 108
 - TwoFactorAuthenticationService, 107
- DotNetIdentity.IdentitySettings.Validators, 13
- DotNetIdentity.IdentitySettings.Validators.PasswordValidator, 88
 - ValidateAsync, 88
- DotNetIdentity.IdentitySettings.Validators.UserValidator, 136
 - ValidateAsync, 137
- DotNetIdentity.Models, 13
 - Accounting, 13
 - Department, 13
 - Email, 15
 - Gender, 13
 - HR, 13
 - Male, 15
 - None, 15
 - Software, 13
 - TwoFactorType, 15
 - Unknown, 15
- DotNetIdentity.Models.BusinessModels, 15
- DotNetIdentity.Models.BusinessModels.EmailModel, 61
 - Body, 62
 - Subject, 62
- DotNetIdentity.Models.BusinessModels.GlobalSettings, 71
 - ApplicationName, 71

- SessionCookieExpiration, [72](#)
- SessionTimeoutRedirAfter, [72](#)
- SessionTimeoutWarnAfter, [72](#)
- DotNetIdentity.Models.BusinessModels.LdapSettings, [76](#)
 - LdapBaseDn, [77](#)
 - LdapDomainController, [77](#)
 - LdapDomainName, [77](#)
 - LdapEnabled, [78](#)
 - LdapGroup, [78](#)
- DotNetIdentity.Models.BusinessModels.MailSettings, [79](#)
 - Password, [79](#)
 - SmtpFromAddress, [80](#)
 - SmtpPort, [80](#)
 - SmtpServer, [80](#)
 - SmtpUseTls, [80](#)
 - Username, [81](#)
- DotNetIdentity.Models.DataModels, [15](#)
- DotNetIdentity.Models.DataModels.ApplicationSettings, [37](#)
 - Name, [38](#)
 - Type, [38](#)
 - Value, [38](#)
- DotNetIdentity.Models.DataModels.AppLogs, [38](#)
 - Exception, [39](#)
 - id, [39](#)
 - Level, [39](#)
 - Message, [40](#)
 - MessageTemplate, [40](#)
 - Properties, [40](#)
 - Timestamp, [40](#)
- DotNetIdentity.Models.DataModels.SessionCache, [92](#)
 - AbsoluteExpiration, [93](#)
 - ExpiresAtTime, [93](#)
 - id, [93](#)
 - SlidingExpirationInSeconds, [93](#)
 - Value, [94](#)
- DotNetIdentity.Models.Identity, [16](#)
- DotNetIdentity.Models.Identity.AppRole, [41](#)
 - CreatedOn, [41](#)
- DotNetIdentity.Models.Identity.AppUser, [44](#)
 - BirthDay, [45](#)
 - CreatedOn, [45](#)
 - Department, [46](#)
 - FirstName, [46](#)
 - Gender, [46](#)
 - IsEnabled, [46](#)
 - IsLdapLogin, [47](#)
 - IsMfaForce, [47](#)
 - LastName, [47](#)
 - ProfilePicture, [47](#)
 - RolesCombined, [48](#)
 - TwoFactorType, [48](#)
- DotNetIdentity.Models.ViewModels, [16](#)
- DotNetIdentity.Models.ViewModels.AssignRoleViewModel, [48](#)
 - IsAssigned, [49](#)
 - RoleId, [49](#)
- RoleName, [49](#)
- DotNetIdentity.Models.ViewModels.ChangePasswordViewModel, [50](#)
 - NewPassword, [50](#)
 - NewPasswordConfirm, [50](#)
 - Password, [50](#)
- DotNetIdentity.Models.ViewModels.EditUserViewModel, [57](#)
 - Department, [57](#)
 - Email, [58](#)
 - FirstName, [58](#)
 - Gender, [58](#)
 - Id, [58](#)
 - LastName, [59](#)
 - PhoneNumber, [59](#)
 - Roles, [59](#)
 - UserName, [59](#)
- DotNetIdentity.Models.ViewModels.ErrorViewModel, [67](#)
 - RequestId, [67](#)
 - ShowRequestId, [67](#)
- DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel, [68](#)
 - Email, [68](#)
- DotNetIdentity.Models.ViewModels.NewUserModel, [84](#)
 - Department, [85](#)
 - Email, [85](#)
 - FirstName, [85](#)
 - Gender, [85](#)
 - IsLdapLogin, [86](#)
 - IsMfaForce, [86](#)
 - LastName, [86](#)
 - Password, [86](#)
 - Roles, [87](#)
 - UserName, [87](#)
- DotNetIdentity.Models.ViewModels.ResetPasswordViewModel, [89](#)
 - Password, [90](#)
 - Token, [90](#)
 - UserId, [90](#)
- DotNetIdentity.Models.ViewModels.SignInViewModel, [101](#)
 - Password, [101](#)
 - RememberMe, [102](#)
 - UserName, [102](#)
- DotNetIdentity.Models.ViewModels.SignUpViewModel, [102](#)
 - BirthDay, [103](#)
 - Department, [103](#)
 - Email, [104](#)
 - FirstName, [104](#)
 - Gender, [104](#)
 - LastName, [104](#)
 - Password, [105](#)
 - UserName, [105](#)
- DotNetIdentity.Models.ViewModels.StatusCodesModel, [105](#)
 - ErrorMessage, [106](#)
 - RouteOfException, [106](#)

- StatusCode, 106
- DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel, 108
 - AuthenticationUri, 109
 - SharedKey, 109
 - VerificationCode, 109
- DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWModel, 110
 - IsRecoveryCode, 110
 - TwoFactorType, 110
 - VerificationCode, 110
- DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel, 111
 - TwoFactorType, 111
- DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 112
 - BirthDay, 112
 - Email, 112
 - Gender, 113
 - PhoneNumber, 113
 - ProfilePicture, 113
 - UserName, 113
- DotNetIdentity.Models.ViewModels.UpsertRoleViewModel, 114
 - Id, 114
 - Name, 114
- DotNetIdentity.Services, 17
- DotNetIdentity.Services.SettingsService, 17
- DotNetIdentity.Services.SettingsService.AppSettingsBase, 42
 - AppSettingsBase, 42
 - Load, 43
 - Save, 43
- DotNetIdentity.Services.SettingsService.ISettingsService, 74
 - Global, 75
 - Ldap, 75
 - Mail, 76
 - Save, 75
- DotNetIdentity.Services.SettingsService.SettingsService, 98
 - Global, 100
 - Ldap, 100
 - Mail, 101
 - Save, 100
 - SettingsService, 99
- EditUser
 - DotNetIdentity.Controllers.AdminController, 24, 26
- Email
 - DotNetIdentity.Models, 15
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 58
 - DotNetIdentity.Models.ViewModels.ForgotPasswordViewModel, 68
 - DotNetIdentity.Models.ViewModels.NewUserModel, 85
 - DotNetIdentity.Models.ViewModels.SignUpViewModel, 104
- DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 112
 - EmailHelper
 - DotNetIdentity.Helpers.EmailHelper, 60
 - Employee
 - DotNetIdentity.Controllers.ClaimBasedController, 52
 - EnableLdapLogin
 - DotNetIdentity.Controllers.AdminController, 26
 - EnableMfaForce
 - DotNetIdentity.Controllers.AdminController, 27
 - EnableUser
 - DotNetIdentity.Controllers.AdminController, 27
 - EnforceTwoFactorAuthenticator
 - DotNetIdentity.Controllers.UserController, 119
 - EnforceTwoFactorResult
 - DotNetIdentity.Controllers.UserController, 120
 - ErrorCd
 - DotNetIdentity.Controllers.ErrorsController, 65
 - ErrorEx
 - DotNetIdentity.Controllers.ErrorsController, 66
 - ErrorLogs
 - DotNetIdentity.Controllers.AdminController, 28
 - ErrorMessage
 - DotNetIdentity.Models.ViewModels.StatusCodesModel, 106
 - Exception
 - DotNetIdentity.Models.DataModels.AppLogs, 39
 - ExpiresAtTime
 - DotNetIdentity.Models.DataModels.SessionCache, 93
 - FirstName
 - DotNetIdentity.Models.Identity.AppUser, 46
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 58
 - DotNetIdentity.Models.ViewModels.NewUserModel, 85
 - DotNetIdentity.Models.ViewModels.SignUpViewModel, 104
 - ForgotPassword
 - DotNetIdentity.Controllers.UserController, 120, 121
 - FreeTrial
 - DotNetIdentity.Controllers.ClaimBasedController, 52
 - Gender
 - DotNetIdentity.Models, 13
 - DotNetIdentity.Models.Identity.AppUser, 46
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 58
 - DotNetIdentity.Models.ViewModels.NewUserModel, 85
 - DotNetIdentity.Models.ViewModels.SignUpViewModel, 104
 - DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 113
 - GenerateQrCodeUri

- DotNetIdentity.IdentitySettings.TwoFactorAuthenticationSettings, 108
- GetAppLogs
 - DotNetIdentity.Controllers.AdminController, 28
- GetAuditLogs
 - DotNetIdentity.Controllers.AdminController, 29
- GetErrorLogs
 - DotNetIdentity.Controllers.AdminController, 29
- GetUsers
 - DotNetIdentity.Controllers.AdminController, 29
- Global
 - DotNetIdentity.Services.SettingsService.ISettingsService, 75
 - DotNetIdentity.Services.SettingsService.SettingsService, 100
- HandleRequirementAsync
 - DotNetIdentity.IdentitySettings.Requirements.FreeTrialExpirationHandler, 69
 - DotNetIdentity.IdentitySettings.Requirements.MinimumAgeHandler, 82
- HomeController
 - DotNetIdentity.Controllers.HomeController, 73
- HR
 - DotNetIdentity.Controllers.ClaimBasedController, 52
 - DotNetIdentity.Models, 13
- Id
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 58
 - DotNetIdentity.Models.ViewModels.UpsertRoleViewModel, 114
- id
 - DotNetIdentity.Models.DataModels.AppLogs, 39
 - DotNetIdentity.Models.DataModels.SessionCache, 93
- Index
 - DotNetIdentity.Controllers.AdminController, 30
 - DotNetIdentity.Controllers.HomeController, 74
- IsAssigned
 - DotNetIdentity.Models.ViewModels.AssignRoleViewModel, 49
- IsEnabled
 - DotNetIdentity.Models.Identity.AppUser, 46
- IsLdapLogin
 - DotNetIdentity.Models.Identity.AppUser, 47
 - DotNetIdentity.Models.ViewModels.NewUserModel, 86
- IsMfaForce
 - DotNetIdentity.Models.Identity.AppUser, 47
 - DotNetIdentity.Models.ViewModels.NewUserModel, 86
- IsRecoveryCode
 - DotNetIdentity.Models.ViewModels.TwoFactorLoginViewModel, 110
- LastName
 - DotNetIdentity.Models.Identity.AppUser, 47
- DotNetIdentity.Models.ViewModels.EditUserViewModel, 59
- DotNetIdentity.Models.ViewModels.NewUserModel, 86
- DotNetIdentity.Models.ViewModels.SignUpViewModel, 104
- Ldap
 - DotNetIdentity.Services.SettingsService.ISettingsService, 75
 - DotNetIdentity.Services.SettingsService.SettingsService, 100
- LdapBaseDn
 - DotNetIdentity.Models.BusinessModels.LdapSettings, 77
- LdapDomainController
 - DotNetIdentity.Models.BusinessModels.LdapSettings, 77
- LdapDomainName
 - DotNetIdentity.Models.BusinessModels.LdapSettings, 77
- LdapEnabled
 - DotNetIdentity.Models.BusinessModels.LdapSettings, 78
- LdapGroup
 - DotNetIdentity.Models.BusinessModels.LdapSettings, 78
- Level
 - DotNetIdentity.Models.DataModels.AppLogs, 39
- Load
 - DotNetIdentity.Services.SettingsService.AppSettingsBase, 43
- Login
 - DotNetIdentity.Controllers.UserController, 121, 123
- Logout
 - DotNetIdentity.Controllers.UserController, 123
- Mail
 - DotNetIdentity.Services.SettingsService.ISettingsService, 76
 - DotNetIdentity.Services.SettingsService.SettingsService, 101
- Model
 - DotNetIdentity.Models, 15
- Message
 - DotNetIdentity.Models.DataModels.AppLogs, 40
- MessageTemplate
 - DotNetIdentity.Models.DataModels.AppLogs, 40
- MinimumAge
 - DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement, 84
- MinimumAgeRequirement
 - DotNetIdentity.IdentitySettings.Requirements.MinimumAgeRequirement, 83
- Name
 - DotNetIdentity.Models.DataModels.ApplicationSettings, 38
- DotNetIdentity.Models.ViewModels.UpsertRoleViewModel, 114

- NewPassword
 - DotNetIdentity.Models.ViewModels.ChangePasswordViewModel, 50
- NewPasswordConfirm
 - DotNetIdentity.Models.ViewModels.ChangePasswordViewModel, 50
- NewUser
 - DotNetIdentity.Controllers.AdminController, 30
- None
 - DotNetIdentity.Models, 15
- OnModelCreating
 - DotNetIdentity.Data.AppDbContext, 35
- Password
 - DotNetIdentity.Models.BusinessModels.MailSettings, 79
 - DotNetIdentity.Models.ViewModels.ChangePasswordViewModel, 50
 - DotNetIdentity.Models.ViewModels.NewUserModel, 86
 - DotNetIdentity.Models.ViewModels.ResetPasswordViewModel, 90
 - DotNetIdentity.Models.ViewModels.SignInViewModel, 101
 - DotNetIdentity.Models.ViewModels.SignUpViewModel, 105
- PasswordContainsUsername
 - DotNetIdentity.IdentitySettings.ErrorDescriber, 64
- PhoneNumber
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 59
 - DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 113
- ping
 - DotNetIdentity.Controllers.UserController, 123
- ProcessAsync
 - DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper, 133
 - DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper, 135
- Profile
 - DotNetIdentity.Controllers.RoleBasedController, 92
 - DotNetIdentity.Controllers.UserController, 124
- ProfilePicture
 - DotNetIdentity.Models.Identity.AppUser, 47
 - DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 113
- Properties
 - DotNetIdentity.Models.DataModels.AppLogs, 40
- Register
 - DotNetIdentity.Controllers.UserController, 125, 126
- RememberMe
 - DotNetIdentity.Models.ViewModels.SignInViewModel, 102
- RequestId
 - DotNetIdentity.Models.ViewModels.ErrorViewModel, 67
- ResetPassword
 - DotNetIdentity.Controllers.UserController, 127
- Role
 - DotNetIdentity.Models.ViewModels.AssignRoleViewModel, 49
- RoleName
 - DotNetIdentity.Models.ViewModels.AssignRoleViewModel, 49
- Roles
 - DotNetIdentity.Controllers.AdminController, 31
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 59
 - DotNetIdentity.Models.ViewModels.NewUserModel, 87
- RolesCombined
 - DotNetIdentity.Models.Identity.AppUser, 48
- RouteOfException
 - DotNetIdentity.Models.ViewModels.StatusCodesModel, 106
- Save
 - DotNetIdentity.Services.SettingsService.AppSettingsBase, 43
 - DotNetIdentity.Services.SettingsService.ISettingsService, 75
 - DotNetIdentity.Services.SettingsService.SettingsService, 100
- SendAsync
 - DotNetIdentity.Helpers.EmailHelper, 61
- SessionCookieExpiration
 - DotNetIdentity.Models.BusinessModels.GlobalSettings, 72
- SessionTimeoutRedirAfter
 - DotNetIdentity.Models.BusinessModels.GlobalSettings, 72
- SessionTimeoutWarnAfter
 - DotNetIdentity.Models.BusinessModels.GlobalSettings, 72
- TagHelper
 - DotNetIdentity.Controllers.CultureController, 56
- SettingsApp
 - DotNetIdentity.Controllers.SettingsController, 96
- SettingsController
 - DotNetIdentity.Controllers.SettingsController, 95
- SettingsLdap
 - DotNetIdentity.Controllers.SettingsController, 96
- SettingsMail
 - DotNetIdentity.Controllers.SettingsController, 97, 98
- SettingsService
 - DotNetIdentity.Services.SettingsService.SettingsService, 99
- SharedKey
 - DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel, 109
- ShowRequestId
 - DotNetIdentity.Models.ViewModels.ErrorViewModel, 67
- SlidingExpirationInSeconds

- DotNetIdentity.Models.DataModels.SessionCache, 93
- SmtpFromAddress
 - DotNetIdentity.Models.BusinessModels.MailSettings, 80
- SmtpPort
 - DotNetIdentity.Models.BusinessModels.MailSettings, 80
- SmtpServer
 - DotNetIdentity.Models.BusinessModels.MailSettings, 80
- SmtpUseTls
 - DotNetIdentity.Models.BusinessModels.MailSettings, 80
- Software
 - DotNetIdentity.Controllers.ClaimBasedController, 52
 - DotNetIdentity.Models, 13
- StatusCode
 - DotNetIdentity.Models.ViewModels.StatusCodesModel, 106
- Subject
 - DotNetIdentity.Models.BusinessModels.EmailModel, 62
- SystemLogs
 - DotNetIdentity.Controllers.AdminController, 32
- Timestamp
 - DotNetIdentity.Models.DataModels.AppLogs, 40
- To
 - DotNetIdentity.Models.BusinessModels.EmailModel, 62
- Token
 - DotNetIdentity.Models.ViewModels.ResetPasswordViewModel, 90
- TransformAsync
 - DotNetIdentity.IdentitySettings.ClaimsTransformation, 54
- TwoFactorAuthenticationService
 - DotNetIdentity.IdentitySettings.TwoFactorAuthenticationService, 107
- TwoFactorAuthenticator
 - DotNetIdentity.Controllers.UserController, 128
- TwoFactorLogin
 - DotNetIdentity.Controllers.UserController, 129, 130
- TwoFactorType
 - DotNetIdentity.Controllers.UserController, 130, 131
 - DotNetIdentity.Models, 15
 - DotNetIdentity.Models.Identity.AppUser, 48
 - DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWMModel, 110
 - DotNetIdentity.Models.ViewModels.TwoFactorTypeViewModel, 111
- Type
 - DotNetIdentity.Models.DataModels.ApplicationSettings, 38
- Unknown
 - DotNetIdentity.Models, 15
- UpsertRole
 - DotNetIdentity.Controllers.AdminController, 32
- UserController
 - DotNetIdentity.Controllers.UserController, 116
- UserId
 - DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper, 133
 - DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper, 136
- DotNetIdentity.Models.ViewModels.ResetPasswordViewModel, 90
- UserName
 - DotNetIdentity.Models.ViewModels.EditUserViewModel, 59
 - DotNetIdentity.Models.ViewModels.NewUserModel, 87
 - DotNetIdentity.Models.ViewModels.SignInViewModel, 102
 - DotNetIdentity.Models.ViewModels.SignUpViewModel, 105
 - DotNetIdentity.Models.ViewModels.UpdateProfileViewModel, 113
- Username
 - DotNetIdentity.Models.BusinessModels.MailSettings, 81
- UserNameContainsEmail
 - DotNetIdentity.IdentitySettings.ErrorDescriber, 64
- UserNameLength
 - DotNetIdentity.IdentitySettings.ErrorDescriber, 64
- UserProfileImageTagHelper
 - DotNetIdentity.Helpers.TagHelpers.UserProfileImageTagHelper, 132
- UserRolesTagHelper
 - DotNetIdentity.Helpers.TagHelpers.UserRolesTagHelper, 135
- Users
 - DotNetIdentity.Controllers.AdminController, 33
 - DotNetIdentity.Controllers.RoleBasedController, 92
- ValidateAsync
 - DotNetIdentity.IdentitySettings.Validators.PasswordValidator, 88
 - DotNetIdentity.IdentitySettings.Validators.UserValidator, 137
- Value
 - DotNetIdentity.Models.DataModels.ApplicationSettings, 38
 - DotNetIdentity.Models.DataModels.SessionCache, 94
- ViewModelCode
 - DotNetIdentity.Models.ViewModels.TwoFactorAuthenticatorViewModel, 109
 - DotNetIdentity.Models.ViewModels.TwoFactorLoginViewWMModel, 110
- Violence
 - DotNetIdentity.Controllers.ClaimBasedController, 53