DeepDive/API

Documentation

1 Namespace Index **1**

1.1 Namespace List . . . 1

2 Hierarchical Index **3**

2.1 Class Hierarchy . . . 3

3 Class Index **5**

3.1 Class List . . . 5

4 Namespace Documentation **9**

4.1 deepdiveapi Namespace Reference . . . 9

4.2 deepdiveapi.Controllers Namespace Reference . . . 9

4.3 deepdiveapi.Entities Namespace Reference . . . 9

4.4 deepdiveapi.Entities.DataTransferObjects Namespace Reference . . . 10

4.5 deepdiveapi.Entities.Enum Namespace Reference . . . 11

4.5.1 Enumeration Type Documentation . . . 11

4.5.1.1 PasswordResetsStastusEnum . . . 11

4.5.1.2 RegistrationDocumentTypes . . . 12

4.5.1.3 RegistrationStatusEnum . . . 12

4.6 deepdiveapi.Entities.Models Namespace Reference . . . 12

4.7 deepdiveapi.Entities.Validation Namespace Reference . . . 13

4.8 deepdiveapi.Entities.Verification Namespace Reference . . . 13

4.9 deepdiveapi.Entities.Verification.Email Namespace Reference . . . 13

4.10 deepdiveapi.JwtFeatures Namespace Reference . . . 13

4.11 deepdiveapi.Repositories Namespace Reference . . . 13

4.12 deepdiveapi.Repositories.Interfaces Namespace Reference . . . 14

5 Class Documentation **15**

5.1 AdminController Class Reference . . . 15

5.1.1 Detailed Description . . . 15

5.1.2 Constructor & Destructor Documentation . . . 15

5.1.2.1 AdminController() . . . 15

5.1.3 Member Function Documentation . . . 16

5.1.3.1 ViewApplicationsWithRequestedStatus() . . . 16

5.2 deepdiveapi.Repositories.AdminRepository Class Reference . . . 16

5.2.1 Detailed Description . . . 16

5.2.2 Constructor & Destructor Documentation . . . 16

5.2.2.1 AdminRepository() . . . 16

5.2.3 Member Function Documentation . . . 17

5.2.3.1 GetAdminUserEmailsAsync() . . . 17

5.3 deepdiveapi.Entities.ApplicationDbContext Class Reference . . . 17

5.3.1 Detailed Description . . . 18

5.3.2 Constructor & Destructor Documentation . . . 18

5.3.2.1 ApplicationDbContext() . . . 18

5.3.3 Member Function Documentation . . . 18
5.3.3.1 OnModelCreating() . . . 18
5.4 AuthController Class Reference . . . 19
5.4.1 Detailed Description . . . 19
5.4.2 Constructor & Destructor Documentation . . . 19
5.4.2.1 AuthController() . . . 19
5.4.3 Member Function Documentation . . . 20
5.4.3.1 Login() . . . 20
5.4.3.2 RegisterUser() . . . 20
5.4.3.3 ResendValidationEmail() . . . 20
5.4.3.4 ValidateEmailWithId() . . . 21
5.4.3.5 ValidatePassword() . . . 21
5.5 deepdiveapi.Entities.DataTransferObjects.AuthResponseDto Class Reference . . . 21
5.5.1 Detailed Description . . . 22
5.6 deepdiveapi.Entities.DataTransferObjects.CoordinatesDto Class Reference . . . 22
5.6.1 Detailed Description . . . 22
5.7 deepdiveapi.Entities.DataTransferObjects.CoordinatesDto2 Class Reference . . . 22
5.7.1 Detailed Description . . . 23
5.8 deepdiveapi.Entities.DataTransferObjects.DocuIdInput Class Reference . . . 23
5.8.1 Detailed Description . . . 23
5.9 deepdiveapi.Entities.Verification.Email.EmailActivatedHandler Class Reference . . . 23
5.9.1 Detailed Description . . . 24
5.9.2 Constructor & Destructor Documentation . . . 24
5.9.2.1 EmailActivatedHandler() . . . 24
5.9.3 Member Function Documentation . . . 24
5.9.3.1 HandleRequirementAsync() . . . 24
5.10 deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement Class Reference . . . 25
5.10.1 Detailed Description . . . 25
5.11 deepdiveapi.Entities.DataTransferObjects.EmailAttachmentBinaryBypass Class Reference . . . 25
5.11.1 Detailed Description . . . 25
5.12 deepdiveapi.Entities.DataTransferObjects.EmailConfirmationUserId Class Reference . . . 26
5.12.1 Detailed Description . . . 26
5.13 deepdiveapi.Entities.DataTransferObjects.EmailInputDto Class Reference . . . 26
5.13.1 Detailed Description . . . 26
5.14 deepdiveapi.Entities.DataTransferObjects.EmailModelDto Class Reference . . . 26
5.14.1 Detailed Description . . . 27
5.15 deepdiveapi.Entities.DataTransferObjects.ErrorResponseDto Class Reference . . . 27
5.15.1 Detailed Description . . . 27
5.16 deepdiveapi.Entities.Models.Excursion Class Reference . . . 27
5.16.1 Detailed Description . . . 28
5.17 deepdiveapi.Controllers.ExcursionController Class Reference . . . 28
5.17.1 Detailed Description . . . 29

5.17.2 Constructor & Destructor Documentation 29
5.17.2.1 ExcursionController() 29
5.17.3 Member Function Documentation 29
5.17.3.1 AddExcursion() 29
5.17.3.2 DeleteExcursion() 30
5.17.3.3 GetExcursionDetails() 30
5.17.3.4 GetExcursionsList() 30
5.17.3.5 UpdateStatus() 31
5.18 deepdiveapi.Entities.DataTransferObjects.ExcursionDetailsDto Class Reference 31
5.18.1 Detailed Description 31
5.19 deepdiveapi.Entities.DataTransferObjects.ExcursionInfo Class Reference 32
5.19.1 Detailed Description 32
5.20 deepdiveapi.Entities.DataTransferObjects.ExcursionMultipleParticipantsDto Class Reference 32
5.20.1 Detailed Description 32
5.21 deepdiveapi.Entities.Models.ExcursionParticipant Class Reference 32
5.21.1 Detailed Description 33
5.22 deepdiveapi.Entities.DataTransferObjects.ExcursionParticipantDto Class Reference 33
5.22.1 Detailed Description 33
5.23 deepdiveapi.Controllers.ExcursionParticipantsController Class Reference 34
5.23.1 Detailed Description 34
5.23.2 Constructor & Destructor Documentation 34
5.23.2.1 ExcursionParticipantsController() 34
5.23.3 Member Function Documentation 35
5.23.3.1 AddMultipleParticipant() 35
5.23.3.2 AddParticipant() 35
5.23.3.3 RemoveParticipant() 35
5.24 deepdiveapi.Repositories.ExcursionParticipantsRepository Class Reference 36
5.24.1 Detailed Description 36
5.24.2 Constructor & Destructor Documentation 36
5.24.2.1 ExcursionParticipantsRepository() 36
5.24.3 Member Function Documentation 37
5.24.3.1 AddMultipleParticipantsToExcursion() 37
5.24.3.2 AddParticipantToExcursion() 37
5.24.3.3 RemoveParticipantFromExcursion() 37
5.25 deepdiveapi.Repositories.ExcursionRepository Class Reference 38
5.25.1 Detailed Description 38
5.25.2 Constructor & Destructor Documentation 38
5.25.2.1 ExcursionRepository() 38
5.25.3 Member Function Documentation 39
5.25.3.1 AddNewExcursion() 39
5.25.3.2 DeleteExcursion() 39
5.25.3.3 GetExcursionById() 39

5.25.3.4 GetExcursions() 40
5.25.3.5 GetUserExcursions() 40
5.25.3.6 UpdateExcursionAsync() 40
5.26 deepdiveapi.Controllers.HomeController Class Reference 41
5.26.1 Detailed Description 41
5.26.2 Constructor & Destructor Documentation 41
5.26.2.1 HomeController() 41
5.26.3 Member Function Documentation 42
5.26.3.1 GetHomeDashboard() 42
5.27 deepdiveapi.Entities.DataTransferObjects.HomeDashboardDto Class Reference 42
5.27.1 Detailed Description 42
5.28 deepdiveapi.Repositories.Interfaces.IAdminRepository Interface Reference 43
5.28.1 Detailed Description 43
5.28.2 Member Function Documentation 43
5.28.2.1 GetAdminUserEmailsAsync() 43
5.29 deepdiveapi.Entities.DataTransferObjects.IdInputDto Class Reference 43
5.29.1 Detailed Description 44
5.30 deepdiveapi.Entities.DataTransferObjects.IdOutputDto Class Reference 44
5.30.1 Detailed Description 44
5.31 deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository Interface Reference 44
5.31.1 Detailed Description 45
5.31.2 Member Function Documentation 45
5.31.2.1 AddMultipleParticipantsToExcursion() 45
5.31.2.2 AddParticipantToExcursion() 45
5.31.2.3 RemoveParticipantFromExcursion() 45
5.32 deepdiveapi.Repositories.Interfaces.IExcursionRepository Interface Reference 45
5.32.1 Detailed Description 46
5.32.2 Member Function Documentation 46
5.32.2.1 AddNewExcursion() 46
5.32.2.2 DeleteExcursion() 46
5.32.2.3 GetExcursionById() 46
5.32.2.4 GetExcursions() 46
5.32.2.5 GetUserExcursions() 47
5.32.2.6 UpdateExcursionAsync() 47
5.33 deepdiveapi.Repositories.Interfaces.IPasswordResetRepository Interface Reference 47
5.33.1 Detailed Description 47
5.33.2 Member Function Documentation 47
5.33.2.1 AddPasswordReset() 47
5.33.2.2 FindByEmailAndToken() 48
5.34 deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository Interface Reference 48
5.34.1 Detailed Description 48
5.34.2 Member Function Documentation 48

5.34.2.1 AddUserDocumentAsync() 48
5.34.2.2 DeleteDocumentAsync() 49
5.34.2.3 GetDocumentsOfUser() 49
5.35 deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository Interface Reference 49
5.35.1 Detailed Description 49
5.35.2 Member Function Documentation 50
5.35.2.1 AddRegistrationRequest() 50
5.35.2.2 GetRegistrationRequestById() 50
5.35.2.3 GetRegistrationRequestByUserIdAsync() 50
5.35.2.4 UpdateRegistrationRequestAsync() 50
5.36 deepdiveapi.Repositories.Interfaces.ITokenRepository Interface Reference 50
5.36.1 Detailed Description 51
5.36.2 Member Function Documentation 51
5.36.2.1 InvalidateOldRefreshToken() 51
5.36.2.2 IsValidRefreshTokenAsync() 51
5.36.2.3 StoreRefreshTokenAsync() 51
5.37 deepdiveapi.Repositories.Interfaces.IUsersRepository Interface Reference 52
5.37.1 Detailed Description 52
5.37.2 Member Function Documentation 52
5.37.2.1 DeleteAccount() 52
5.37.2.2 DisableAccount() 52
5.37.2.3 EnableAccount() 52
5.37.2.4 GetSafeUserFromID() 53
5.37.2.5 GetSafeUsers() 53
5.37.2.6 GetUsers() 53
5.37.2.7 SearchUsersAsync() 53
5.38 deepdiveapi.JwtFeatures.JwtHandler Class Reference 53
5.38.1 Detailed Description 54
5.38.2 Constructor & Destructor Documentation 54
5.38.2.1 JwtHandler() 54
5.38.3 Member Function Documentation 54
5.38.3.1 GenerateAccessToken() 54
5.38.3.2 GenerateRefreshToken() 54
5.38.3.3 GenerateTokenOptions() 55
5.38.3.4 GetClaims() 55
5.38.3.5 GetPrincipalFromExpiredToken() 55
5.38.3.6 GetSigningCredentials() 56
5.38.3.7 GetUserIdFromToken() 56
5.39 deepdiveapi.Entities.Validation.LaterThanAttribute Class Reference 56
5.39.1 Detailed Description 57
5.39.2 Constructor & Destructor Documentation 57
5.39.2.1 LaterThanAttribute() 57

5.39.3 Member Function Documentation 57
5.39.3.1 IsValid() 57
5.40 deepdiveapi.Entities.DataTransferObjects.NewExcursionDto Class Reference 58
5.40.1 Detailed Description 58
5.41 deepdiveapi.Entities.DataTransferObjects.ParticipantDto Class Reference 58
5.41.1 Detailed Description 58
5.42 deepdiveapi.Entities.DataTransferObjects.ParticipationConfirmationInputDto Class Reference . . . 59
5.42.1 Detailed Description 59
5.43 deepdiveapi.Entities.Models.PasswordReset Class Reference 59
5.43.1 Detailed Description 59
5.44 deepdiveapi.Controllers.PasswordResetController Class Reference 60
5.44.1 Detailed Description 60
5.44.2 Constructor & Destructor Documentation 60
5.44.2.1 PasswordResetController() 60
5.44.3 Member Function Documentation 61
5.44.3.1 AddPasswordRequest() 61
5.44.3.2 UpdatePassword() 61
5.44.3.3 ValidateReset() 61
5.45 deepdiveapi.Repositories.PasswordResetRepository Class Reference 62
5.45.1 Detailed Description 62
5.45.2 Constructor & Destructor Documentation 62
5.45.2.1 PasswordResetRepository() 62
5.45.3 Member Function Documentation 63
5.45.3.1 AddPasswordReset() 63
5.45.3.2 FindByEmailAndToken() 63
5.46 deepdiveapi.Entities.Models.RefreshToken Class Reference 63
5.46.1 Detailed Description 64
5.47 RegisterDocumentsController Class Reference 64
5.47.1 Detailed Description 65
5.47.2 Constructor & Destructor Documentation 65
5.47.2.1 RegisterDocumentsController() 65
5.47.3 Member Function Documentation 65
5.47.3.1 DeleteDocument() 65
5.47.3.2 UploadDocument() 65
5.48 deepdiveapi.Repositories.RegistrationDocumentRepository Class Reference 66
5.48.1 Detailed Description 66
5.48.2 Constructor & Destructor Documentation 66
5.48.2.1 RegistrationDocumentRepository() 66
5.48.3 Member Function Documentation 67
5.48.3.1 AddUserDocumentAsync() 67
5.48.3.2 DeleteDocumentAsync() 67
5.48.3.3 GetDocumentsOfUser() 67

5.49 deepdiveapi.Entities.Models.RegistrationRequest Class Reference 68
5.49.1 Detailed Description 68
5.50 deepdiveapi.Controllers.RegistrationRequestController Class Reference 69
5.50.1 Detailed Description 69
5.50.2 Constructor & Destructor Documentation 69
5.50.2.1 RegistrationRequestController() 69
5.50.3 Member Function Documentation 70
5.50.3.1 UpdateStatus() [1/2] 70
5.50.3.2 UpdateStatus() [2/2] 70
5.51 deepdiveapi.Repositories.RegistrationRequestRepository Class Reference 70
5.51.1 Detailed Description 71
5.51.2 Constructor & Destructor Documentation 71
5.51.2.1 RegistrationRequestRepository() 71
5.51.3 Member Function Documentation 71
5.51.3.1 AddRegistrationRequest() 71
5.51.3.2 GetRegistrationRequestById() 72
5.51.3.3 GetRegistrationRequestByUserIdAsync() 72
5.51.3.4 UpdateRegistrationRequestAsync() 72
5.52 deepdiveapi.Entities.DataTransferObjects.RegistrationResponseDto Class Reference 73
5.52.1 Detailed Description 73
5.53 deepdiveapi.Entities.DataTransferObjects.StringInputDto Class Reference 73
5.53.1 Detailed Description 73
5.54 deepdiveapi.Controllers.TokenController Class Reference 74
5.54.1 Detailed Description 74
5.54.2 Constructor & Destructor Documentation 74
5.54.2.1 TokenController() 74
5.54.3 Member Function Documentation 74
5.54.3.1 RefreshAsync() 74
5.55 deepdiveapi.Entities.DataTransferObjects.TokenRefreshDto Class Reference 75
5.55.1 Detailed Description 75
5.56 deepdiveapi.Repositories.TokenRepository Class Reference 75
5.56.1 Detailed Description 76
5.56.2 Constructor & Destructor Documentation 76
5.56.2.1 TokenRepository() 76
5.56.3 Member Function Documentation 76
5.56.3.1 InvalidateOldRefreshToken() 76
5.56.3.2 IsValidRefreshTokenAsync() 77
5.56.3.3 StoreRefreshTokenAsync() 77
5.57 deepdiveapi.Entities.DataTransferObjects.UpdateExcursionRequestDto Class Reference 77
5.57.1 Detailed Description 78
5.58 deepdiveapi.Entities.DataTransferObjects.UpdateRegistrationRequestDto Class Reference 78
5.58.1 Detailed Description 78

5.59 deepdiveapi.Entities.DataTransferObjects.UpdateStatusRegistrationRequestDto Class Reference . 78
5.59.1 Detailed Description 78
5.60 deepdiveapi.Entities.DataTransferObjects.UpdateUserInputDto Class Reference 79
5.60.1 Detailed Description 79
5.61 deepdiveapi.Entities.DataTransferObjects.UpdateUserPassword Class Reference 79
5.61.1 Detailed Description 79
5.62 deepdiveapi.Entities.DataTransferObjects.UpdateUserResponseDto Class Reference 80
5.62.1 Detailed Description 80
5.63 deepdiveapi.Entities.DataTransferObjects.UploadRegisterDocumentDto Class Reference 80
5.63.1 Detailed Description 80
5.64 deepdiveapi.Entities.Models.User Class Reference 81
5.64.1 Detailed Description 81
5.65 UserController Class Reference 81
5.65.1 Detailed Description 82
5.65.2 Constructor & Destructor Documentation 82
5.65.2.1 UserController() 82
5.65.3 Member Function Documentation 83
5.65.3.1 DeleteAccount() 83
5.65.3.2 DisableAccount() 83
5.65.3.3 EnableAccount() 83
5.65.3.4 GetSafeUsers() [1/2] 84
5.65.3.5 GetSafeUsers() [2/2] 84
5.65.3.6 GetUsers() 84
5.65.3.7 GetUsersSearchTypehead() 84
5.65.3.8 Update() 85
5.65.3.9 ViewApplicationRegistration() 85
5.66 deepdiveapi.Entities.DataTransferObjects.UserDto Class Reference 85
5.66.1 Detailed Description 86
5.67 deepdiveapi.Entities.DataTransferObjects.UserForAuthenticationDto Class Reference 86
5.67.1 Detailed Description 86
5.68 deepdiveapi.Entities.DataTransferObjects.UserForRegistrationDto Class Reference 86
5.68.1 Detailed Description 87
5.69 deepdiveapi.Entities.DataTransferObjects.UserForSafeListDto Class Reference 87
5.69.1 Detailed Description 87
5.70 deepdiveapi.Entities.Models.UserRegisterDocument Class Reference 87
5.70.1 Detailed Description 88
5.71 deepdiveapi.Repositories.UsersRepository Class Reference 88
5.71.1 Detailed Description 88
5.71.2 Constructor & Destructor Documentation 88
5.71.2.1 UsersRepository() 88
5.71.3 Member Function Documentation 89
5.71.3.1 DeleteAccount() 89

5.71.3.2 DisableAccount() 89
5.71.3.3 EnableAccount() 89
5.71.3.4 GetSafeUserFromID() 90
5.71.3.5 GetSafeUsers() 90
5.71.3.6 GetUsers() 90
5.71.3.7 SearchUsersAsync() 90
5.72 deepdiveapi.Entities.DataTransferObjects.UsersTypeahead Class Reference 91
5.72.1 Detailed Description 91
5.73 deepdiveapi.Entities.DataTransferObjects.ValidateEmailAndUserIdResponse Class Reference . . . 91
5.73.1 Detailed Description 92
5.74 deepdiveapi.Entities.DataTransferObjects.ValidatePasswordReset Class Reference 92
5.74.1 Detailed Description 92
5.75 deepdiveapi.Entities.DataTransferObjects.ViewApplicationResponseDto Class Reference 92
5.75.1 Detailed Description 92

Index **93**

Chapter 1

Namespace Index

1.1 Namespace List

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

deepdiveapi 9
deepdiveapi.Controllers 9
deepdiveapi.Entities 9
deepdiveapi.Entities.DataTransferObjects 10
deepdiveapi.Entities.Enum 11
deepdiveapi.Entities.Models 12
deepdiveapi.Entities.Validation 13
deepdiveapi.Entities.Verification 13
deepdiveapi.Entities.Verification.Email 13
deepdiveapi.JwtFeatures 13
deepdiveapi.Repositories 13
deepdiveapi.Repositories.Interfaces 14

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

AuthorizationHandler
- deepdiveapi.Entities.Verification.Email.EmailActivatedHandler 23

deepdiveapi.Entities.DataTransferObjects.AuthResponseDto 21

ControllerBase
- AdminController 15
- AuthController 19
- RegisterDocumentsController 64
- UserController 81
- deepdiveapi.Controllers.ExcursionController 28
- deepdiveapi.Controllers.ExcursionParticipantsController 34
- deepdiveapi.Controllers.HomeController 41
- deepdiveapi.Controllers.PasswordResetController 60
- deepdiveapi.Controllers.RegistrationRequestController 69
- deepdiveapi.Controllers.TokenController 74

deepdiveapi.Entities.DataTransferObjects.CoordinatesDto 22
deepdiveapi.Entities.DataTransferObjects.CoordinatesDto2 22
deepdiveapi.Entities.DataTransferObjects.DocuIdInput 23
deepdiveapi.Entities.DataTransferObjects.EmailAttachmentBinaryBypass 25
deepdiveapi.Entities.DataTransferObjects.EmailConfirmationUserId 26
deepdiveapi.Entities.DataTransferObjects.EmailInputDto 26
deepdiveapi.Entities.DataTransferObjects.EmailModelDto 26
deepdiveapi.Entities.DataTransferObjects.ErrorResponseDto 27
deepdiveapi.Entities.Models.Excursion 27
deepdiveapi.Entities.DataTransferObjects.ExcursionDetailsDto 31
deepdiveapi.Entities.DataTransferObjects.ExcursionInfo 32
deepdiveapi.Entities.DataTransferObjects.ExcursionMultipleParticipantsDto 32
deepdiveapi.Entities.Models.ExcursionParticipant 32
deepdiveapi.Entities.DataTransferObjects.ExcursionParticipantDto 33
deepdiveapi.Entities.DataTransferObjects.HomeDashboardDto 42
deepdiveapi.Repositories.Interfaces.IAdminRepository 43
- deepdiveapi.Repositories.AdminRepository 16

IAuthorizationRequirement
- deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement 25

IdentityDbContext
- deepdiveapi.Entities.ApplicationDbContext 17

IdentityUser
    deepdiveapi.Entities.Models.User 81
deepdiveapi.Entities.DataTransferObjects.IdInputDto 43
deepdiveapi.Entities.DataTransferObjects.IdOutputDto 44
deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository 44
    deepdiveapi.Repositories.ExcursionParticipantsRepository 36
deepdiveapi.Repositories.Interfaces.IExcursionRepository 45
    deepdiveapi.Repositories.ExcursionRepository 38
deepdiveapi.Repositories.Interfaces.IPasswordResetRepository 47
    deepdiveapi.Repositories.PasswordResetRepository 62
deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository 48
    deepdiveapi.Repositories.RegistrationDocumentRepository 66
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository 49
    deepdiveapi.Repositories.RegistrationRequestRepository 70
deepdiveapi.Repositories.Interfaces.ITokenRepository 50
    deepdiveapi.Repositories.TokenRepository 75
deepdiveapi.Repositories.Interfaces.IUsersRepository 52
    deepdiveapi.Repositories.UsersRepository 88
deepdiveapi.JwtFeatures.JwtHandler 53
deepdiveapi.Entities.DataTransferObjects.NewExcursionDto 58
deepdiveapi.Entities.DataTransferObjects.ParticipantDto 58
deepdiveapi.Entities.DataTransferObjects.ParticipationConfirmationInputDto 59
deepdiveapi.Entities.Models.PasswordReset 59
deepdiveapi.Entities.Models.RefreshToken 63
deepdiveapi.Entities.Models.RegistrationRequest 68
deepdiveapi.Entities.DataTransferObjects.RegistrationResponseDto 73
deepdiveapi.Entities.DataTransferObjects.StringInputDto 73
deepdiveapi.Entities.DataTransferObjects.TokenRefreshDto 75
deepdiveapi.Entities.DataTransferObjects.UpdateExcursionRequestDto 77
deepdiveapi.Entities.DataTransferObjects.UpdateRegistrationRequestDto 78
deepdiveapi.Entities.DataTransferObjects.UpdateStatusRegistrationRequestDto 78
deepdiveapi.Entities.DataTransferObjects.UpdateUserInputDto 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserPassword 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserResponseDto 80
deepdiveapi.Entities.DataTransferObjects.UploadRegisterDocumentDto 80
deepdiveapi.Entities.DataTransferObjects.UserDto 85
deepdiveapi.Entities.DataTransferObjects.UserForAuthenticationDto 86
deepdiveapi.Entities.DataTransferObjects.UserForRegistrationDto 86
deepdiveapi.Entities.DataTransferObjects.UserForSafeListDto 87
deepdiveapi.Entities.Models.UserRegisterDocument 87
deepdiveapi.Entities.DataTransferObjects.UsersTypeahead 91
deepdiveapi.Entities.DataTransferObjects.ValidateEmailAndUserIdResponse 91
deepdiveapi.Entities.DataTransferObjects.ValidatePasswordReset 92
ValidationAttribute
    deepdiveapi.Entities.Validation.LaterThanAttribute 56
deepdiveapi.Entities.DataTransferObjects.ViewApplicationResponseDto 92

Chapter 3

Class Index

3.1 Class List

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

AdminController
Controller responsible for handling administrative tasks 15
deepdiveapi.Repositories.AdminRepository
The AdminRepository class manages operations related to administrator users 16
deepdiveapi.Entities.ApplicationDbContext
Represents the database context of the application, integrating Identity framework functionality 17
AuthController
Controller responsible for user authentication and registration 19
deepdiveapi.Entities.DataTransferObjects.AuthResponseDto
DTO representing the response for authentication operations, detailing success, errors, and token data 21
deepdiveapi.Entities.DataTransferObjects.CoordinatesDto
Nested DTO representing coordinates spatial data 22
deepdiveapi.Entities.DataTransferObjects.CoordinatesDto2
Nested DTO representing coordinates spatial data 22
deepdiveapi.Entities.DataTransferObjects.DocuIdInput
DTO for passing a document ID, typically used in operations involving document retrieval or manipulation 23
deepdiveapi.Entities.Verification.Email.EmailActivatedHandler
Handles the authorization requirement checking for a user's email activation status 23
deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement
Represents an authorization requirement that checks if a user's email is activated 25
deepdiveapi.Entities.DataTransferObjects.EmailAttachmentBinaryBypass
DTO for handling binary content of an email attachment, specifying the file name and media type 25
deepdiveapi.Entities.DataTransferObjects.EmailConfirmationUserId
DTO for confirming an email address associated with a specific user ID 26
deepdiveapi.Entities.DataTransferObjects.EmailInputDto
DTO for inputting an email address, typically for sending emails or verifying email-related operations 26
deepdiveapi.Entities.DataTransferObjects.EmailModelDto
DTO for constructing an email with subject, content, recipients, and attachments 26
deepdiveapi.Entities.DataTransferObjects.ErrorResponseDto
DTO for encapsulating error responses from API calls 27
deepdiveapi.Entities.Models.Excursion
Represents an excursion, typically a guided tour or trip 27

deepdiveapi.Controllers.ExcursionController
Controller responsible for managing excursions . . . 28
deepdiveapi.Entities.DataTransferObjects.ExcursionDetailsDto
DTO that provides detailed information about an excursion, including details about the creator and participants . . . 31
deepdiveapi.Entities.DataTransferObjects.ExcursionInfo
Sub-DTO for providing excursion information . . . 32
deepdiveapi.Entities.DataTransferObjects.ExcursionMultipleParticipantsDto
DTO for adding multiple participants to an excursion . . . 32
deepdiveapi.Entities.Models.ExcursionParticipant
Represents a participant in an excursion . . . 32
deepdiveapi.Entities.DataTransferObjects.ExcursionParticipantDto
DTO for managing participants of an excursion . . . 33
deepdiveapi.Controllers.ExcursionParticipantsController
Controller responsible for managing participants of excursions . . . 34
deepdiveapi.Repositories.ExcursionParticipantsRepository
The ExcursionParticipantsRepository class manages operations related to participants of excursions . . . 36
deepdiveapi.Repositories.ExcursionRepository
The ExcursionRepository class manages operations related to excursions . . . 38
deepdiveapi.Controllers.HomeController
Controller responsible for handling home-related requests . . . 41
deepdiveapi.Entities.DataTransferObjects.HomeDashboardDto
DTO representing the data for a home dashboard, including excursion statistics and random excursion information . . . 42
deepdiveapi.Repositories.Interfaces.IAdminRepository
The IExcursionParticipantsRepository interface defines operations related to administrator users 43
deepdiveapi.Entities.DataTransferObjects.IdInputDto
DTO for inputting an ID . . . 43
deepdiveapi.Entities.DataTransferObjects.IdOutputDto
DTO for outputting an ID . . . 44
deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository
The IExcursionParticipantsRepository interface defines methods for managing participants of excursions . . . 44
deepdiveapi.Repositories.Interfaces.IExcursionRepository
The IExcursionRepository interface defines methods for managing excursions . . . 45
deepdiveapi.Repositories.Interfaces.IPasswordResetRepository
The IPasswordResetRepository interface defines methods for managing password reset requests . . . 47
deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository
The IRegistrationDocumentRepository interface defines methods for managing registration documents . . . 48
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository
The IRegistrationRequestRepository interface defines methods for managing registration requests . . . 49
deepdiveapi.Repositories.Interfaces.ITokenRepository
The ITokenRepository interface defines methods for managing authentication tokens . . . 50
deepdiveapi.Repositories.Interfaces.IUsersRepository
The IUsersRepository interface defines methods for user management . . . 52
deepdiveapi.JwtFeatures.JwtHandler
Provides methods for managing JWT tokens including creation and validation . . . 53
deepdiveapi.Entities.Validation.LaterThanAttribute
Specifies that the annotated date or time property must be later than another named property on the same object . . . 56
deepdiveapi.Entities.DataTransferObjects.NewExcursionDto
DTO for creating a new excursion, including required information such as title, description, date/time, and location coordinates . . . 58

deepdiveapi.Entities.DataTransferObjects.ParticipantDto
Nested DTO representing detailed participant information 58
deepdiveapi.Entities.DataTransferObjects.ParticipationConfirmationInputDto
DTO for confirming participation in an event, including the event time, location, and a QR code key for validation 59
deepdiveapi.Entities.Models.PasswordReset
Represents a password reset request 59
deepdiveapi.Controllers.PasswordResetController
Controller responsible for handling password reset requests 60
deepdiveapi.Repositories.PasswordResetRepository
The PasswordResetRepository class manages operations related to user password resets . . 62
deepdiveapi.Entities.Models.RefreshToken
Represents a refresh token for maintaining user authentication sessions 63
RegisterDocumentsController
Controller responsible for managing user registration documents 64
deepdiveapi.Repositories.RegistrationDocumentRepository
The RegistrationDocumentRepository class manages operations related to user registration documents 66
deepdiveapi.Entities.Models.RegistrationRequest
Represents a request for user registration status change 68
deepdiveapi.Controllers.RegistrationRequestController
Controller responsible for managing registration requests 69
deepdiveapi.Repositories.RegistrationRequestRepository
The RegistrationRequestRepository class manages operations related to user registration requests 70
deepdiveapi.Entities.DataTransferObjects.RegistrationResponseDto
DTO representing the response for a registration operation, indicating success or failure and providing details on any errors 73
deepdiveapi.Entities.DataTransferObjects.StringInputDto
DTO for a simple string input, typically used for searches or single field submissions 73
deepdiveapi.Controllers.TokenController
Controller responsible for managing JWT token operations such as refreshing tokens 74
deepdiveapi.Entities.DataTransferObjects.TokenRefreshDto
DTO for refreshing an authentication token 75
deepdiveapi.Repositories.TokenRepository
The TokenRepository class manages operations related to user authentication tokens, including storing and invalidating refresh tokens 75
deepdiveapi.Entities.DataTransferObjects.UpdateExcursionRequestDto
DTO for updating an excursion's details including title, description, and location coordinates . . 77
deepdiveapi.Entities.DataTransferObjects.UpdateRegistrationRequestDto
DTO for updating a registration request with status and optional administrator comments . . . 78
deepdiveapi.Entities.DataTransferObjects.UpdateStatusRegistrationRequestDto
DTO for updating the registration status of a user 78
deepdiveapi.Entities.DataTransferObjects.UpdateUserInputDto
DTO for updating user profile information 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserPassword
DTO for updating a user's password, requiring verification via token 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserResponseDto
DTO representing the response from a user update operation, indicating success or failure and any errors 80
deepdiveapi.Entities.DataTransferObjects.UploadRegisterDocumentDto
DTO for uploading a registration document, specifying the document type and associated user 80
deepdiveapi.Entities.Models.User
Represents user information derived from IdentityUser 81
UserController
Controller responsible for managing user-specific operations such as retrieving and updating user information 81

deepdiveapi.Entities.DataTransferObjects.UserDto
Nested DTO representing basic user information within an excursion context 85
deepdiveapi.Entities.DataTransferObjects.UserForAuthenticationDto
DTO for user login credentials 86
deepdiveapi.Entities.DataTransferObjects.UserForRegistrationDto
DTO for user registration details, including validations for required fields and data consistency . 86
deepdiveapi.Entities.DataTransferObjects.UserForSafeListDto
DTO for displaying user details on a safe list, containing non-sensitive information 87
deepdiveapi.Entities.Models.UserRegisterDocument
Represents a document info registered by a user 87
deepdiveapi.Repositories.UsersRepository
The UsersRepository class is responsible for managing user-related data operations, including CRUD operations and specialized queries 88
deepdiveapi.Entities.DataTransferObjects.UsersTypeahead
DTO for providing a typeahead search functionality for users, displaying minimal user details . 91
deepdiveapi.Entities.DataTransferObjects.ValidateEmailAndUserIdResponse
DTO to communicate the confirmation status of a user's email 91
deepdiveapi.Entities.DataTransferObjects.ValidatePasswordReset
DTO for validating a password reset operation using ID and token 92
deepdiveapi.Entities.DataTransferObjects.ViewApplicationResponseDto
DTO for viewing application responses with user and application status details 92

Chapter 4

Namespace Documentation

4.1 deepdiveapi Namespace Reference

4.2 deepdiveapi.Controllers Namespace Reference

Classes

- class ExcursionController
  Controller responsible for managing excursions.
- class ExcursionParticipantsController
  Controller responsible for managing participants of excursions.
- class HomeController
  Controller responsible for handling home-related requests.
- class PasswordResetController
  Controller responsible for handling password reset requests.
- class RegistrationRequestController
  Controller responsible for managing registration requests.
- class TokenController
  Controller responsible for managing JWT token operations such as refreshing tokens.

4.3 deepdiveapi.Entities Namespace Reference

Classes

- class ApplicationDbContext
  Represents the database context of the application, integrating Identity framework functionality.

4.4 deepdiveapi.Entities.DataTransferObjects Namespace Reference

Classes

- class AuthResponseDto

  DTO representing the response for authentication operations, detailing success, errors, and token data.
- class CoordinatesDto

  Nested DTO representing coordinates spatial data.
- class CoordinatesDto2

  Nested DTO representing coordinates spatial data.
- class DocuIdInput

  DTO for passing a document ID, typically used in operations involving document retrieval or manipulation.
- class EmailAttachmentBinaryBypass

  DTO for handling binary content of an email attachment, specifying the file name and media type.
- class EmailConfirmationUserId

  DTO for confirming an email address associated with a specific user ID.
- class EmailInputDto

  DTO for inputting an email address, typically for sending emails or verifying email-related operations.
- class EmailModelDto

  DTO for constructing an email with subject, content, recipients, and attachments.
- class ErrorResponseDto

  DTO for encapsulating error responses from API calls.
- class ExcursionDetailsDto

  DTO that provides detailed information about an excursion, including details about the creator and participants.
- class ExcursionInfo

  Sub-DTO for providing excursion information.
- class ExcursionMultipleParticipantsDto

  DTO for adding multiple participants to an excursion.
- class ExcursionParticipantDto

  DTO for managing participants of an excursion.
- class HomeDashboardDto

  DTO representing the data for a home dashboard, including excursion statistics and random excursion information.
- class IdInputDto

  DTO for inputting an ID.
- class IdOutputDto

  DTO for outputting an ID.
- class NewExcursionDto

  DTO for creating a new excursion, including required information such as title, description, date/time, and location coordinates.
- class ParticipantDto

  Nested DTO representing detailed participant information.
- class ParticipationConfirmationInputDto

  DTO for confirming participation in an event, including the event time, location, and a QR code key for validation.
- class RegistrationResponseDto

  DTO representing the response for a registration operation, indicating success or failure and providing details on any errors.
- class StringInputDto

  DTO for a simple string input, typically used for searches or single field submissions.
- class TokenRefreshDto

  DTO for refreshing an authentication token.
- class UpdateExcursionRequestDto

DTO for updating an excursion's details including title, description, and location coordinates.

- class UpdateRegistrationRequestDto

  DTO for updating a registration request with status and optional administrator comments.
- class UpdateStatusRegistrationRequestDto

  DTO for updating the registration status of a user.
- class UpdateUserInputDto

  DTO for updating user profile information.
- class UpdateUserPassword

  DTO for updating a user's password, requiring verification via token.
- class UpdateUserResponseDto

  DTO representing the response from a user update operation, indicating success or failure and any errors.
- class UploadRegisterDocumentDto

  DTO for uploading a registration document, specifying the document type and associated user.
- class UserDto

  Nested DTO representing basic user information within an excursion context.
- class UserForAuthenticationDto

  DTO for user login credentials.
- class UserForRegistrationDto

  DTO for user registration details, including validations for required fields and data consistency.
- class UserForSafeListDto

  DTO for displaying user details on a safe list, containing non-sensitive information.
- class UsersTypeahead

  DTO for providing a typeahead search functionality for users, displaying minimal user details.
- class ValidateEmailAndUserIdResponse

  DTO to communicate the confirmation status of a user's email.
- class ValidatePasswordReset

  DTO for validating a password reset operation using ID and token.
- class ViewApplicationResponseDto

  DTO for viewing application responses with user and application status details.

4.5 deepdiveapi.Entities.Enum Namespace Reference

Enumerations

- enum PasswordResetsStastusEnum { Requested = 0 , PwdChanged = 1 }

  Represents the status of a password reset request.
- enum RegistrationDocumentTypes { IdPassport = 0 , Certificate = 1 }

  Defines the types of documents that can be registered.
- enum RegistrationStatusEnum { Requested = 0 , WaitingForUserChanges = 1 , Approved = 2 , Denied = 3 }

  Represents the possible statuses of a registration request.

4.5.1 Enumeration Type Documentation

4.5.1.1 PasswordResetsStastusEnum

```
enum deepdiveapi.Entities.Enum.PasswordResetsStastusEnum
```

Represents the status of a password reset request.

Enumerator

Requested	The password reset has been requested but not completed.
PwdChanged	The password has been changed following the reset request.

4.5.1.2 RegistrationDocumentTypes

`enum deepdiveapi.Entities.Enum.RegistrationDocumentTypes`

Defines the types of documents that can be registered.

Enumerator

IdPassport	Identification document such as a passport.
Certificate	A certificate, such as a birth certificate or marriage certificate.

4.5.1.3 RegistrationStatusEnum

`enum deepdiveapi.Entities.Enum.RegistrationStatusEnum`

Represents the possible statuses of a registration request.

Enumerator

Requested	The request has been made but not yet processed.
WaitingForUserChanges	The request is waiting for the user to make necessary changes.
Approved	The request has been approved.
Denied	The request has been denied.

4.6 deepdiveapi.Entities.Models Namespace Reference

Classes

- class Excursion

  Represents an excursion, typically a guided tour or trip.
- class ExcursionParticipant

  Represents a participant in an excursion.
- class PasswordReset

  Represents a password reset request.
- class RefreshToken

  Represents a refresh token for maintaining user authentication sessions.
- class RegistrationRequest

  Represents a request for user registration status change.
- class User

  Represents user information derived from IdentityUser.
- class UserRegisterDocument

  Represents a document info registered by a user.

4.7 deepdiveapi.Entities.Validation Namespace Reference

Classes

- class LaterThanAttribute

  Specifies that the annotated date or time property must be later than another named property on the same object.

4.8 deepdiveapi.Entities.Verification Namespace Reference

4.9 deepdiveapi.Entities.Verification.Email Namespace Reference

Classes

- class EmailActivatedHandler

  Handles the authorization requirement checking for a user's email activation status.
- class EmailActivatedRequirement

  Represents an authorization requirement that checks if a user's email is activated.

4.10 deepdiveapi.JwtFeatures Namespace Reference

Classes

- class JwtHandler

  Provides methods for managing JWT tokens including creation and validation.

4.11 deepdiveapi.Repositories Namespace Reference

Classes

- class AdminRepository

  The AdminRepository class manages operations related to administrator users.
- class ExcursionParticipantsRepository

  The ExcursionParticipantsRepository class manages operations related to participants of excursions.
- class ExcursionRepository

  The ExcursionRepository class manages operations related to excursions.
- class PasswordResetRepository

  The PasswordResetRepository class manages operations related to user password resets.
- class RegistrationDocumentRepository

  The RegistrationDocumentRepository class manages operations related to user registration documents.
- class RegistrationRequestRepository

  The RegistrationRequestRepository class manages operations related to user registration requests.
- class TokenRepository

  The TokenRepository class manages operations related to user authentication tokens, including storing and invalidating refresh tokens.
- class UsersRepository

  The UsersRepository class is responsible for managing user-related data operations, including CRUD operations and specialized queries.

4.12 deepdiveapi.Repositories.Interfaces Namespace Reference

Classes

- interface IAdminRepository

  The IExcursionParticipantsRepository interface defines operations related to administrator users.
- interface IExcursionParticipantsRepository

  The IExcursionParticipantsRepository interface defines methods for managing participants of excursions.
- interface IExcursionRepository

  The IExcursionRepository interface defines methods for managing excursions.
- interface IPasswordResetRepository

  The IPasswordResetRepository interface defines methods for managing password reset requests.
- interface IRegistrationDocumentRepository

  The IRegistrationDocumentRepository interface defines methods for managing registration documents.
- interface IRegistrationRequestRepository

  The IRegistrationRequestRepository interface defines methods for managing registration requests.
- interface ITokenRepository

  The ITokenRepository interface defines methods for managing authentication tokens.
- interface IUsersRepository

  The IUsersRepository interface defines methods for user management.

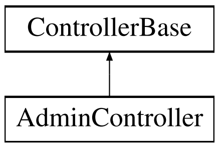
Chapter 5

Class Documentation

5.1 AdminController Class Reference

Controller responsible for handling administrative tasks.

Inheritance diagram for AdminController:



Public Member Functions

- AdminController (ApplicationDbContext dbContext, IRegistrationDocumentRepository registration↩ DocumentRepository)

  Initializes a new instance of the AdminController class.
- async Task< IActionResult > ViewApplicationsWithRequestedStatus ()

  Retrieves applications with requested status.

5.1.1 Detailed Description

Controller responsible for handling administrative tasks.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AdminController()

```
AdminController.AdminController (
            ApplicationDbContext dbContext,
            IRegistrationDocumentRepository registrationDocumentRepository )  [inline]
```

Initializes a new instance of the AdminController class.

Parameters

dbContext	Database context for accessing application data.
registrationDocumentRepository	Repository for managing registration documents.

5.1.3 Member Function Documentation

5.1.3.1 ViewApplicationsWithRequestedStatus()

```
async Task< IActionResult > AdminController.ViewApplicationsWithRequestedStatus ( )  [inline]
```

Retrieves applications with requested status.

Returns

List of application details with requested status.

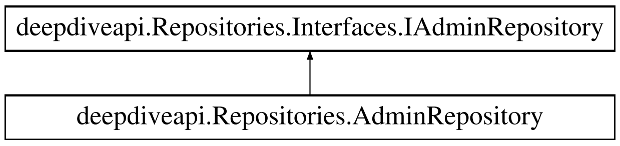
The documentation for this class was generated from the following file:

- Controllers/AdminController.cs

5.2 deepdiveapi.Repositories.AdminRepository Class Reference

The AdminRepository class manages operations related to administrator users.

Inheritance diagram for deepdiveapi.Repositories.AdminRepository:


deepdiveapi.Repositories.Interfaces.IAdminRepository
deepdiveapi.Repositories.AdminRepository

Public Member Functions

- AdminRepository (UserManager< User > userManager)

    Constructor for AdminRepository.
- async Task< List< string > > GetAdminUserEmailsAsync ()

    Retrieves email addresses of admin users asynchronously.

5.2.1 Detailed Description

The AdminRepository class manages operations related to administrator users.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 AdminRepository()

```
deepdiveapi.Repositories.AdminRepository.AdminRepository (
            UserManager< User > userManager )  [inline]
```

Constructor for AdminRepository.

Parameters

userManager	User manager for identity management.

5.2.3 Member Function Documentation

5.2.3.1 GetAdminUserEmailsAsync()

```
async Task< List< string > > deepdiveapi.Repositories.AdminRepository.GetAdminUserEmailsAsync
( ) [inline]
```

Retrieves email addresses of admin users asynchronously.

Returns

A task that represents the asynchronous operation. The task result contains a list of email addresses of admin users.

Implements deepdiveapi.Repositories.Interfaces.IAdminRepository.

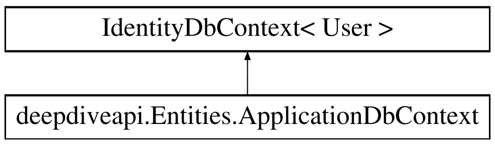
The documentation for this class was generated from the following file:

- Repositories/AdminRepository.cs

5.3 deepdiveapi.Entities.ApplicationDbContext Class Reference

Represents the database context of the application, integrating Identity framework functionality.

Inheritance diagram for deepdiveapi.Entities.ApplicationDbContext:


IdentityDbContext< User >
deepdiveapi.Entities.ApplicationDbContext

Public Member Functions

- ApplicationDbContext (DbContextOptions< ApplicationDbContext > options)
  Initializes a new instance of ApplicationDbContext with the specified options.

Protected Member Functions

- override void OnModelCreating (ModelBuilder modelBuilder)
  Configures the schema needed for the identity framework and sets up entity relationships and properties.

Properties

- DbSet< RefreshToken > **RefreshTokens** `[get, set]`

  Gets or sets the DbSet for Refresh Tokens.
- DbSet< IdentityRole > **IdentityRoles** `[get, set]`

  Gets or sets the DbSet for Identity Roles.
- DbSet< UserRegisterDocument > **UserRegisterDocuments** `[get, set]`

  Gets or sets the DbSet for User Register Documents.
- DbSet< RegistrationRequest > **RegistrationRequests** `[get, set]`

  Gets or sets the DbSet for Registration Requests.
- DbSet< PasswordReset > **PasswordResets** `[get, set]`

  Gets or sets the DbSet for Password Resets.
- DbSet< Excursion > **Excursions** `[get, set]`

  Gets or sets the DbSet for Excursions.
- DbSet< ExcursionParticipant > **ExcursionParticipants** `[get, set]`

  Gets or sets the DbSet for Excursion Participants.

5.3.1 Detailed Description

Represents the database context of the application, integrating Identity framework functionality.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ApplicationDbContext()

```
deepdiveapi.Entities.ApplicationDbContext.ApplicationDbContext (
            DbContextOptions< ApplicationDbContext > options )  [inline]
```

Initializes a new instance of ApplicationDbContext with the specified options.

Parameters

options	The options to be used by a DbContext.

5.3.3 Member Function Documentation

5.3.3.1 OnModelCreating()

```
override void deepdiveapi.Entities.ApplicationDbContext.OnModelCreating (
            ModelBuilder modelBuilder )  [inline], [protected]
```

Configures the schema needed for the identity framework and sets up entity relationships and properties.

Parameters

modelBuilder	The builder being used to construct the model for this context.

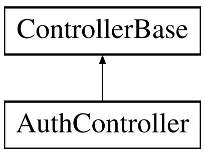
The documentation for this class was generated from the following file:

- Entities/ApplicationDbContext.cs

5.4 AuthController Class Reference

Controller responsible for user authentication and registration.

Inheritance diagram for AuthController:



Public Member Functions

- AuthController (UserManager< User > userManager, JwtHandler jwtHandler, ITokenRepository token←
Repository, IRegistrationRequestRepository registrationRequestRepository, ApplicationDbContext db←
Context, IConfiguration configuration)

    Initializes a new instance of the AuthController class.
- async Task< IActionResult > ValidateEmailWithId ([FromBody] EmailConfirmationUserId input)

    Validates email with user ID for confirmation.
- async Task< IActionResult > Login ([FromBody] UserForAuthenticationDto userForAuthentication)

    Logs in a user.
- async Task< IActionResult > ResendValidationEmail ([FromBody] EmailInputDto userForRegistration)

    Resends email validation confirmation.
- async Task< IActionResult > RegisterUser ([FromBody] UserForRegistrationDto userForRegistration)

    Registers a new user.
- async Task< IActionResult > ValidatePassword ([FromBody] UserForAuthenticationDto userFor←
Authentication)

    Validates user password.

5.4.1 Detailed Description

Controller responsible for user authentication and registration.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 AuthController()

```
AuthController.AuthController (
            UserManager< User > userManager,
            JwtHandler jwtHandler,
            ITokenRepository tokenRepository,
            IRegistrationRequestRepository registrationRequestRepository,
            ApplicationDbContext dbContext,
            IConfiguration configuration ) [inline]
```

Initializes a new instance of the AuthController class.

Parameters

userManager	Manager for handling user-related operations.
jwtHandler	Handler for JWT token generation and validation.
tokenRepository	Repository for managing refresh tokens.
registrationRequestRepository	Repository for managing registration requests.
dbContext	Database context for accessing application data.
configuration	Represents the application configuration properties.

5.4.3 Member Function Documentation

5.4.3.1 Login()

```
async Task< IActionResult > AuthController.Login (
            [FromBody] UserForAuthenticationDto userForAuthentication )  [inline]
```

Logs in a user.

Parameters

userForAuthentication	DTO containing user credentials for authentication.

Returns

Authentication result with access token and refresh token.

5.4.3.2 RegisterUser()

```
async Task< IActionResult > AuthController.RegisterUser (
            [FromBody] UserForRegistrationDto userForRegistration )  [inline]
```

Registers a new user.

Parameters

userForRegistration	DTO containing user details for registration.

Returns

Authentication result with access token and refresh token for the newly registered user.

5.4.3.3 ResendValidationEmail()

```
async Task< IActionResult > AuthController.ResendValidationEmail (
            [FromBody] EmailInputDto userForRegistration )  [inline]
```

Resends email validation confirmation.

Parameters

userForRegistration	DTO containing user email for resending confirmation email.

Returns

Result of resending confirmation email.

5.4.3.4 ValidateEmailWithId()

```
async Task< IActionResult > AuthController.ValidateEmailWithId (
             [FromBody] EmailConfirmationUserId input )  [inline]
```

Validates email with user ID for confirmation.

Parameters

input	DTO containing user ID and email for validation.

Returns

Result of email validation.

5.4.3.5 ValidatePassword()

```
async Task< IActionResult > AuthController.ValidatePassword (
             [FromBody] UserForAuthenticationDto userForAuthentication )  [inline]
```

Validates user password.

Parameters

userForAuthentication	DTO containing user email and password for validation.

Returns

Result of password validation.

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

- Controllers/AuthController.cs

5.5 deepdiveapi.Entities.DataTransferObjects.AuthResponseDto Class Reference

DTO representing the response for authentication operations, detailing success, errors, and token data.

Properties

- bool **IsAuthSuccessful** `[get, set]`
- string? **ErrorMessage** `[get, set]`
- string? **Token** `[get, set]`
- string? **RefreshToken** `[get, set]`

5.5.1 Detailed Description

DTO representing the response for authentication operations, detailing success, errors, and token data.

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

- Entities/DataTransferObjects/AuthResponseDto.cs

5.6 deepdiveapi.Entities.DataTransferObjects.CoordinatesDto Class Reference

Nested DTO representing coordinates spatial data.

Properties

- double **Lat** `[get, set]`
- double **Long** `[get, set]`

5.6.1 Detailed Description

Nested DTO representing coordinates spatial data.

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

- Entities/DataTransferObjects/ExcursionDetailsDto.cs

5.7 deepdiveapi.Entities.DataTransferObjects.CoordinatesDto2 Class Reference

Nested DTO representing coordinates spatial data.

Properties

- double **Lat** `[get, set]`
- double **Long** `[get, set]`

5.7.1 Detailed Description

Nested DTO representing coordinates spatial data.

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

- Entities/DataTransferObjects/UpdateExcursionRequestDto.cs

5.8 deepdiveapi.Entities.DataTransferObjects.DocuIdInput Class Reference

DTO for passing a document ID, typically used in operations involving document retrieval or manipulation.

Properties

- required string **DocumentId** `[get, set]`

5.8.1 Detailed Description

DTO for passing a document ID, typically used in operations involving document retrieval or manipulation.

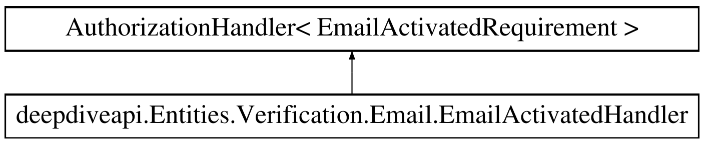
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/DocuIdInput.cs

5.9 deepdiveapi.Entities.Verification.Email.EmailActivatedHandler Class Reference

Handles the authorization requirement checking for a user's email activation status.

Inheritance diagram for deepdiveapi.Entities.Verification.Email.EmailActivatedHandler:


AuthorizationHandler< EmailActivatedRequirement >
deepdiveapi.Entities.Verification.Email.EmailActivatedHandler

Public Member Functions

- EmailActivatedHandler (UserManager< User > userManager)

  Initializes a new instance of EmailActivatedHandler class.

Protected Member Functions

- override async Task HandleRequirementAsync (AuthorizationHandlerContext context, EmailActivatedRequirement requirement)

  Processes the authorization request to determine if the user has an activated email.

5.9.1 Detailed Description

Handles the authorization requirement checking for a user's email activation status.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 EmailActivatedHandler()

```
deepdiveapi.Entities.Verification.Email.EmailActivatedHandler.EmailActivatedHandler (
             UserManager< User > userManager )  [inline]
```

Initializes a new instance of EmailActivatedHandler class.

Parameters

userManager	User manager to access user information.

5.9.3 Member Function Documentation

5.9.3.1 HandleRequirementAsync()

```
override async Task deepdiveapi.Entities.Verification.Email.EmailActivatedHandler.Handle↩
RequirementAsync (
             AuthorizationHandlerContext context,
             EmailActivatedRequirement requirement )  [inline], [protected]
```

Processes the authorization request to determine if the user has an activated email.

Parameters

context	Authorization handling context.
requirement	The authorization requirement to evaluate.

Returns

A task that represents the asynchronous operation.

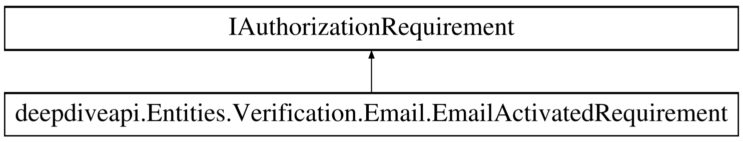
The documentation for this class was generated from the following file:

- Entities/Verification/Email/EmailActivatedHandler.cs

5.10 deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement Class Reference

Represents an authorization requirement that checks if a user's email is activated.

Inheritance diagram for deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement:


IAuthorizationRequirement
deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement

5.10.1 Detailed Description

Represents an authorization requirement that checks if a user's email is activated.

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

- Entities/Verification/Email/EmailActivatedRequirement.cs

5.11 deepdiveapi.Entities.DataTransferObjects.EmailAttachmentBinary↩ Bypass Class Reference

DTO for handling binary content of an email attachment, specifying the file name and media type.

Public Member Functions

- **EmailAttachmentBinaryBypass** (string fileName, string mediaType, byte[ ] content)

Properties

- string **FileName** `[get, set]`
- string **MediaType** `[get, set]`
- byte[ ] **Content** `[get, set]`

5.11.1 Detailed Description

DTO for handling binary content of an email attachment, specifying the file name and media type.

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

- Entities/DataTransferObjects/EmailAttachmentBinaryBypass.cs

5.12 deepdiveapi.Entities.DataTransferObjects.EmailConfirmationUserId Class Reference

DTO for confirming an email address associated with a specific user ID.

Properties

- required string **UserId** `[get, set]`
- required string **Email** `[get, set]`

5.12.1 Detailed Description

DTO for confirming an email address associated with a specific user ID.

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

- Entities/DataTransferObjects/EmailConfirmationUserId.cs

5.13 deepdiveapi.Entities.DataTransferObjects.EmailInputDto Class Reference

DTO for inputting an email address, typically for sending emails or verifying email-related operations.

Properties

- required string **Email** `[get, set]`

5.13.1 Detailed Description

DTO for inputting an email address, typically for sending emails or verifying email-related operations.

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

- Entities/DataTransferObjects/EmailInputDto.cs

5.14 deepdiveapi.Entities.DataTransferObjects.EmailModelDto Class Reference

DTO for constructing an email with subject, content, recipients, and attachments.

Properties

- string **Subject** `[get, set]`
- string **PlainTextContent** `[get, set]`
- string **HtmlContent** `[get, set]`
- List< string > **toRecipients** `[get, set]`
- List< string > **ccRecipients** `[get, set]`
- List< string > **bccRecipients** `[get, set]`
- List< EmailAttachmentBinaryBypass > **Attachments** `[get, set]`

5.14.1 Detailed Description

DTO for constructing an email with subject, content, recipients, and attachments.

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

- Entities/DataTransferObjects/EmailModelDto.cs

5.15 deepdiveapi.Entities.DataTransferObjects.ErrorResponseDto Class Reference

DTO for encapsulating error responses from API calls.

Properties

- IEnumerable< string >? **Errors** `[get, set]`

5.15.1 Detailed Description

DTO for encapsulating error responses from API calls.

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

- Entities/DataTransferObjects/ErrorResponseDto.cs

5.16 deepdiveapi.Entities.Models.Excursion Class Reference

Represents an excursion, typically a guided tour or trip.

Properties

- string **Id** `[get, set]`

  Gets or sets the unique identifier for the excursion.
- string **Title** `[get, set]`

  Gets or sets the title of the excursion.
- string **Description** `[get, set]`

  Gets or sets the description of the excursion.
- string? **CreatedByUserFK** `[get, set]`

  Gets or sets the foreign key for the user who created the excursion.
- User **CreatedByUser** `[get, set]`

  Navigation property for the user who created the excursion.
- DateTime **CreatedOn** `[get, set]`

  Gets or sets the creation date and time for the excursion.
- DateTime **DateTime** `[get, set]`

  Gets or sets the scheduled date and time for the excursion.
- Point **Location** `[get, set]`

  Gets or sets the geographic location of the excursion using spatial data.
- string **ImageName** `[get, set]`

  Gets or sets the image name associated with the excursion.
- ICollection< ExcursionParticipant > **Participants** `[get, set]`

  Navigation property for the participants of the excursion.

5.16.1 Detailed Description

Represents an excursion, typically a guided tour or trip.

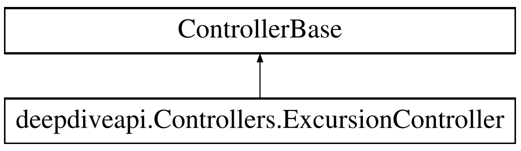
The documentation for this class was generated from the following file:

- Entities/Models/Excursion.cs

5.17 deepdiveapi.Controllers.ExcursionController Class Reference

Controller responsible for managing excursions.

Inheritance diagram for deepdiveapi.Controllers.ExcursionController:



Public Member Functions

- ExcursionController (IExcursionRepository excursionRepository, ApplicationDbContext dbContext, IHttp↩ContextAccessor httpContextAccessor, UserManager< User > userManager)

  Initializes a new instance of the ExcursionController class.
- async Task< IActionResult > GetExcursionsList ()

  Retrieves a list of excursions.
- async Task< IActionResult > GetExcursionDetails (IdInputDto input)

  Retrieves details of a specific excursion.
- async Task< IActionResult > AddExcursion (NewExcursionDto input)

  Adds a new excursion.
- async Task< IActionResult > DeleteExcursion ([FromBody] IdInputDto input)

  Deletes an excursion.
- async Task< IActionResult > UpdateStatus (UpdateExcursionRequestDto input)

  Updates details of an existing excursion.

5.17.1 Detailed Description

Controller responsible for managing excursions.

5.17.2 Constructor & Destructor Documentation

5.17.2.1 ExcursionController()

```
deepdiveapi.Controllers.ExcursionController.ExcursionController (
            IExcursionRepository excursionRepository,
            ApplicationDbContext dbContext,
            IHttpContextAccessor httpContextAccessor,
            UserManager< User > userManager )  [inline]
```

Initializes a new instance of the ExcursionController class.

Parameters

excursionRepository	Repository for managing excursions.
dbContext	Database context for accessing application data.
httpContextAccessor	Accessor for HttpContext.
userManager	Manager for handling user-related operations.

5.17.3 Member Function Documentation

5.17.3.1 AddExcursion()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionController.AddExcursion (
            NewExcursionDto input )  [inline]
```

Adds a new excursion.

Parameters

input	DTO containing details of the new excursion.

Returns

ID of the newly created excursion.

5.17.3.2 DeleteExcursion()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionController.DeleteExcursion (
            [FromBody] IdInputDto input )  [inline]
```

Deletes an excursion.

Parameters

input	DTO containing the ID of the excursion to be deleted.

Returns

Ok if excursion is deleted successfully, BadRequest if excursion not found, or internal server error occurs.

5.17.3.3 GetExcursionDetails()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionController.GetExcursionDetails (
            IdInputDto input )  [inline]
```

Retrieves details of a specific excursion.

Parameters

input	DTO containing the ID of the excursion.

Returns

Details of the excursion.

5.17.3.4 GetExcursionsList()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionController.GetExcursionsList ( )
[inline]
```

Retrieves a list of excursions.

Returns

List of excursions.

5.17.3.5 UpdateStatus()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionController.UpdateStatus (
            UpdateExcursionRequestDto input )  [inline]
```

Updates details of an existing excursion.

Parameters

input	DTO containing details to be updated.

Returns

Ok if excursion details are updated successfully, BadRequest if excursion not found, or internal server error occurs.

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

- Controllers/ExcursionController.cs

5.18 deepdiveapi.Entities.DataTransferObjects.ExcursionDetailsDto Class Reference

DTO that provides detailed information about an excursion, including details about the creator and participants.

Properties

- string **Id** `[get, set]`
- string **Title** `[get, set]`
- string **Description** `[get, set]`
- UserDto **CreatedByUser** `[get, set]`
- DateTime **CreatedOn** `[get, set]`
- DateTime **DateTime** `[get, set]`
- string **ImageName** `[get, set]`
- CoordinatesDto **Coordinates** `[get, set]`
- List< ParticipantDto > **Participants** `[get, set]`

5.18.1 Detailed Description

DTO that provides detailed information about an excursion, including details about the creator and participants.

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

- Entities/DataTransferObjects/ExcursionDetailsDto.cs

5.19 deepdiveapi.Entities.DataTransferObjects.ExcursionInfo Class Reference

Sub-DTO for providing excursion information.

Properties

- string **Id** `[get, set]`
- string **Title** `[get, set]`
- string **Description** `[get, set]`
- string **ImageName** `[get, set]`

5.19.1 Detailed Description

Sub-DTO for providing excursion information.

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

- Entities/DataTransferObjects/HomeDashboardDto.cs

5.20 deepdiveapi.Entities.DataTransferObjects.ExcursionMultiple↩ ParticipantsDto Class Reference

DTO for adding multiple participants to an excursion.

Properties

- List< string > **UserId** `[get, set]`
- string **ExcursionId** `[get, set]`

5.20.1 Detailed Description

DTO for adding multiple participants to an excursion.

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

- Entities/DataTransferObjects/ExcursionMultipleParticipantsDto.cs

5.21 deepdiveapi.Entities.Models.ExcursionParticipant Class Reference

Represents a participant in an excursion.

Properties

- string **Id** `[get, set]`

  Gets or sets the unique identifier for the excursion participant.
- string **ExcursionId** `[get, set]`

  Gets or sets the foreign key for the excursion this participant is registered to.
- Excursion **Excursion** `[get, set]`

  Navigation property for the excursion associated with this participant.
- string **UserId** `[get, set]`

  Gets or sets the foreign key for the user who is participating in the excursion.
- User **User** `[get, set]`

  Navigation property for the user associated with this participant.

5.21.1 Detailed Description

Represents a participant in an excursion.

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

- Entities/Models/ExcursionParticipant.cs

5.22 deepdiveapi.Entities.DataTransferObjects.ExcursionParticipantDto Class Reference

DTO for managing participants of an excursion.

Properties

- string **UserId** `[get, set]`
- string **ExcursionId** `[get, set]`

5.22.1 Detailed Description

DTO for managing participants of an excursion.

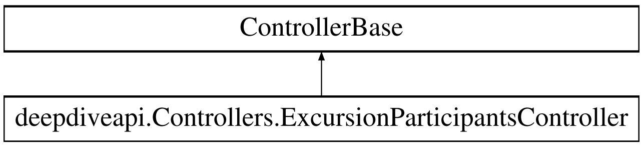
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/ExcursionParticipantDto.cs

5.23 deepdiveapi.Controllers.ExcursionParticipantsController Class Reference

Controller responsible for managing participants of excursions.

Inheritance diagram for deepdiveapi.Controllers.ExcursionParticipantsController:



Public Member Functions

- ExcursionParticipantsController (IExcursionParticipantsRepository excursionParticipantsRepository, ApplicationDbContext dbContext, IHttpContextAccessor httpContextAccessor, UserManager< User > userManager)

  Initializes a new instance of the ExcursionParticipantsController class.
- async Task< IActionResult > AddParticipant (ExcursionParticipantDto input)

  Adds a participant to an excursion.
- async Task< IActionResult > AddMultipleParticipant (ExcursionMultipleParticipantsDto input)

  Adds multiple participants to an excursion.
- async Task< IActionResult > RemoveParticipant (ExcursionParticipantDto input)

  Removes a participant from an excursion.

5.23.1 Detailed Description

Controller responsible for managing participants of excursions.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 ExcursionParticipantsController()

```
deepdiveapi.Controllers.ExcursionParticipantsController.ExcursionParticipantsController (
            IExcursionParticipantsRepository excursionParticipantsRepository,
            ApplicationDbContext dbContext,
            IHttpContextAccessor httpContextAccessor,
            UserManager< User > userManager )  [inline]
```

Initializes a new instance of the ExcursionParticipantsController class.

Parameters

excursionParticipantsRepository	Repository for managing excursion participants.
dbContext	Database context for accessing application data.
httpContextAccessor	Accessor for HttpContext.
userManager	Manager for handling user-related operations.

5.23.3 Member Function Documentation

5.23.3.1 AddMultipleParticipant()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionParticipantsController.Add↩
MultipleParticipant (
            ExcursionMultipleParticipantsDto input )  [inline]
```

Adds multiple participants to an excursion.

Parameters

input	DTO containing the excursion ID and list of participant IDs.

Returns

Ok if participants are added successfully, BadRequest if user not found or internal server error occurs.

5.23.3.2 AddParticipant()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionParticipantsController.Add↩
Participant (
            ExcursionParticipantDto input )  [inline]
```

Adds a participant to an excursion.

Parameters

input	DTO containing the participant and excursion IDs.

Returns

Ok if participant is added successfully, BadRequest if user not found or internal server error occurs.

5.23.3.3 RemoveParticipant()

```
async Task< IActionResult > deepdiveapi.Controllers.ExcursionParticipantsController.Remove↩
Participant (
            ExcursionParticipantDto input )  [inline]
```

Removes a participant from an excursion.

Parameters

input	DTO containing the participant and excursion IDs.

Returns

Ok if participant is removed successfully, BadRequest if user not found or internal server error occurs.

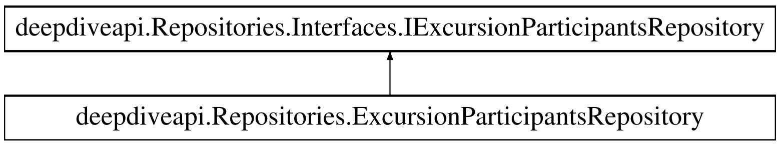
The documentation for this class was generated from the following file:

- Controllers/ExcursionParticipantsController.cs

5.24 deepdiveapi.Repositories.ExcursionParticipantsRepository Class Reference

The ExcursionParticipantsRepository class manages operations related to participants of excursions.

Inheritance diagram for deepdiveapi.Repositories.ExcursionParticipantsRepository:


deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository
deepdiveapi.Repositories.ExcursionParticipantsRepository

Public Member Functions

- ExcursionParticipantsRepository (ApplicationDbContext dbContext, IConfiguration configuration)

    Constructor for ExcursionParticipantsRepository.
- async Task AddParticipantToExcursion (string excursionId, string userId)

    Adds a user as a participant to an excursion.
- async Task RemoveParticipantFromExcursion (string excursionId, string userId)

    Removes a participant from an excursion.
- async Task AddMultipleParticipantsToExcursion (string excursionId, List< string > userIds)

    Adds multiple participants to an excursion.

5.24.1 Detailed Description

The ExcursionParticipantsRepository class manages operations related to participants of excursions.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 ExcursionParticipantsRepository()

```
deepdiveapi.Repositories.ExcursionParticipantsRepository.ExcursionParticipantsRepository (
            ApplicationDbContext dbContext,
            IConfiguration configuration )  [inline]
```

Constructor for ExcursionParticipantsRepository.

Parameters

dbContext	Application database context.
configuration	Configuration interface to access application settings.

5.24.3 Member Function Documentation

5.24.3.1 AddMultipleParticipantsToExcursion()

```
async Task deepdiveapi.Repositories.ExcursionParticipantsRepository.AddMultipleParticipants←
ToExcursion (
            string excursionId,
            List< string > userIds )  [inline]
```

Adds multiple participants to an excursion.

Parameters

excursion←Id	The ID of the excursion to add participants to.
userIds	The IDs of the users to add as participants.

Implements deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.

5.24.3.2 AddParticipantToExcursion()

```
async Task deepdiveapi.Repositories.ExcursionParticipantsRepository.AddParticipantToExcursion
(
            string excursionId,
            string userId )  [inline]
```

Adds a user as a participant to an excursion.

Parameters

excursion←Id	The ID of the excursion to add the participant to.
userId	The ID of the user to add as a participant.

Implements deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.

5.24.3.3 RemoveParticipantFromExcursion()

```
async Task deepdiveapi.Repositories.ExcursionParticipantsRepository.RemoveParticipantFrom←
Excursion (
            string excursionId,
            string userId )  [inline]
```

Removes a participant from an excursion.

Parameters

excursion↩ Id	The ID of the excursion to remove the participant from.
userId	The ID of the user to remove as a participant.

Implements deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.

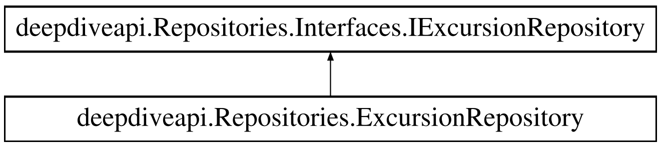
The documentation for this class was generated from the following file:

- Repositories/ExcursionParticipantsRepository.cs

5.25 deepdiveapi.Repositories.ExcursionRepository Class Reference

The ExcursionRepository class manages operations related to excursions.

Inheritance diagram for deepdiveapi.Repositories.ExcursionRepository:


deepdiveapi.Repositories.Interfaces.IExcursionRepository
deepdiveapi.Repositories.ExcursionRepository

Public Member Functions

- ExcursionRepository (ApplicationDbContext context)

  Constructor for ExcursionRepository.
- async Task< List< Excursion > > GetExcursions ()

  Retrieves all excursions.
- async Task< Excursion > GetExcursionById (string input)

  Retrieves an excursion by its ID, including its participants and their user details.
- async Task< string > AddNewExcursion (NewExcursionDto input, string userId)

  Creates a new excursion and automatically adds the creator as a participant.
- async Task DeleteExcursion (string id)

  Deletes an excursion by its ID.
- async Task UpdateExcursionAsync (Excursion excursion)

  Updates an existing excursion.
- async Task<(List< Excursion > PastExcursions, List< Excursion > FutureExcursions)> GetUserExcursions (string userId)

  Retrieves past and future excursions for a specific user.

5.25.1 Detailed Description

The ExcursionRepository class manages operations related to excursions.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 ExcursionRepository()

```
deepdiveapi.Repositories.ExcursionRepository.ExcursionRepository (
            ApplicationDbContext context )   [inline]
```

Constructor for ExcursionRepository.

Parameters

context	Application database context.

5.25.3 Member Function Documentation

5.25.3.1 AddNewExcursion()

```
async Task< string > deepdiveapi.Repositories.ExcursionRepository.AddNewExcursion (
            NewExcursionDto input,
            string userId )  [inline]
```

Creates a new excursion and automatically adds the creator as a participant.

Parameters

input	Data transfer object containing information about the new excursion.
user←Id	The ID of the user creating the excursion.

Returns

A task that represents the asynchronous operation. The task result contains the ID of the newly created excursion.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

5.25.3.2 DeleteExcursion()

```
async Task deepdiveapi.Repositories.ExcursionRepository.DeleteExcursion (
            string id )  [inline]
```

Deletes an excursion by its ID.

Parameters

id	The ID of the excursion to delete.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

5.25.3.3 GetExcursionById()

```
async Task< Excursion > deepdiveapi.Repositories.ExcursionRepository.GetExcursionById (
            string input )  [inline]
```

Retrieves an excursion by its ID, including its participants and their user details.

Parameters

input	The ID of the excursion to retrieve.

Returns

A task that represents the asynchronous operation. The task result contains an Excursion entity, or null if not found.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

5.25.3.4 GetExcursions()

```
async Task< List< Excursion > > deepdiveapi.Repositories.ExcursionRepository.GetExcursions (
)  [inline]
```

Retrieves all excursions.

Returns

A task that represents the asynchronous operation. The task result contains a list of Excursion entities.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

5.25.3.5 GetUserExcursions()

```
async Task<(List< Excursion > PastExcursions, List< Excursion > FutureExcursions)> deepdiveapi.↩
Repositories.ExcursionRepository.GetUserExcursions (
            string userId )  [inline]
```

Retrieves past and future excursions for a specific user.

Parameters

user↩ Id	The ID of the user to retrieve excursions for.

Returns

A task that represents the asynchronous operation. The task result contains two lists of Excursion entities, one for past and one for future excursions.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

5.25.3.6 UpdateExcursionAsync()

```
async Task deepdiveapi.Repositories.ExcursionRepository.UpdateExcursionAsync (
            Excursion excursion )  [inline]
```

Updates an existing excursion.

Parameters

excursion	The excursion entity to update.

Implements deepdiveapi.Repositories.Interfaces.IExcursionRepository.

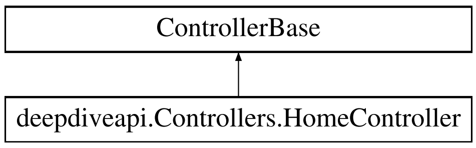
The documentation for this class was generated from the following file:

- Repositories/ExcursionRepository.cs

5.26 deepdiveapi.Controllers.HomeController Class Reference

Controller responsible for handling home-related requests.

Inheritance diagram for deepdiveapi.Controllers.HomeController:



Public Member Functions

- HomeController (IExcursionRepository excursionRepository, ApplicationDbContext dbContext)

    Initializes a new instance of the HomeController class.
- async Task< IActionResult > GetHomeDashboard (IdInputDto input)

    Retrieves the home dashboard data for the specified user.

5.26.1 Detailed Description

Controller responsible for handling home-related requests.

5.26.2 Constructor & Destructor Documentation

5.26.2.1 HomeController()

```
deepdiveapi.Controllers.HomeController.HomeController (
            IExcursionRepository excursionRepository,
            ApplicationDbContext dbContext )  [inline]
```

Initializes a new instance of the HomeController class.

Parameters

excursionRepository	Repository for accessing excursion-related data.
dbContext	Database context for accessing application data.

5.26.3 Member Function Documentation

5.26.3.1 GetHomeDashboard()

```
async Task< IActionResult > deepdiveapi.Controllers.HomeController.GetHomeDashboard (
            IdInputDto input )  [inline]
```

Retrieves the home dashboard data for the specified user.

Parameters

input	DTO containing the user ID for which the dashboard data is requested.

Returns

Ok with the home dashboard data if successful, or InternalServerError on exception.

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

- Controllers/HomeController.cs

5.27 deepdiveapi.Entities.DataTransferObjects.HomeDashboardDto Class Reference

DTO representing the data for a home dashboard, including excursion statistics and random excursion information.

Properties

- List< ExcursionInfo > **RandomExcursionInfos** `[get, set]`
- int **TotalExcursions** `[get, set]`
- int **UpcomingExcursions** `[get, set]`
- int **ParticipatedInExcursions** `[get, set]`
- int **GoingToParticipateInExcursions** `[get, set]`

5.27.1 Detailed Description

DTO representing the data for a home dashboard, including excursion statistics and random excursion information.

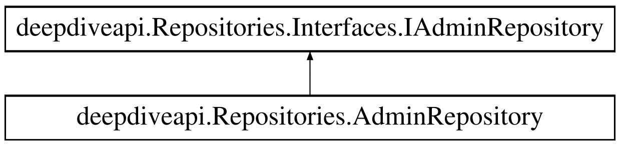
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/HomeDashboardDto.cs

5.28 deepdiveapi.Repositories.Interfaces.IAdminRepository Interface Reference

The IExcursionParticipantsRepository interface defines operations related to administrator users.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IAdminRepository:


deepdiveapi.Repositories.Interfaces.IAdminRepository
deepdiveapi.Repositories.AdminRepository

Public Member Functions

- Task< List< string > > GetAdminUserEmailsAsync ()

5.28.1 Detailed Description

The IExcursionParticipantsRepository interface defines operations related to administrator users.

5.28.2 Member Function Documentation

5.28.2.1 GetAdminUserEmailsAsync()

```
Task< List< string > > deepdiveapi.Repositories.Interfaces.IAdminRepository.GetAdminUser←
EmailsAsync ( )
```

Implemented in deepdiveapi.Repositories.AdminRepository.

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

- Repositories/Interfaces/IAdminRepository.cs

5.29 deepdiveapi.Entities.DataTransferObjects.IdInputDto Class Reference

DTO for inputting an ID.

Properties

- required string **Id** `[get, set]`

5.29.1 Detailed Description

DTO for inputting an ID.

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

- Entities/DataTransferObjects/IdInputDto.cs

5.30 deepdiveapi.Entities.DataTransferObjects.IdOutputDto Class Reference

DTO for outputting an ID.

Properties

- required string **id** `[get, set]`

5.30.1 Detailed Description

DTO for outputting an ID.

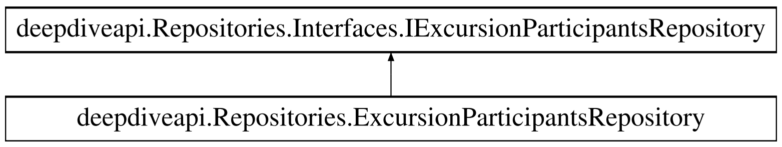
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/IdOutputDto.cs

5.31 deepdiveapi.Repositories.Interfaces.IExcursionParticipants↩ Repository Interface Reference

The IExcursionParticipantsRepository interface defines methods for managing participants of excursions.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository:


deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository
deepdiveapi.Repositories.ExcursionParticipantsRepository

Public Member Functions

- Task AddParticipantToExcursion (string excursionId, string userId)
- Task RemoveParticipantFromExcursion (string excursionId, string userId)
- Task AddMultipleParticipantsToExcursion (string excursionId, List< string > userIds)

5.31.1 Detailed Description

The IExcursionParticipantsRepository interface defines methods for managing participants of excursions.

5.31.2 Member Function Documentation

5.31.2.1 AddMultipleParticipantsToExcursion()

```
Task deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.AddMultipleParticipants↩
ToExcursion (
            string excursionId,
            List< string > userIds )
```

Implemented in deepdiveapi.Repositories.ExcursionParticipantsRepository.

5.31.2.2 AddParticipantToExcursion()

```
Task deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.AddParticipantTo↩
Excursion (
            string excursionId,
            string userId )
```

Implemented in deepdiveapi.Repositories.ExcursionParticipantsRepository.

5.31.2.3 RemoveParticipantFromExcursion()

```
Task deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository.RemoveParticipant↩
FromExcursion (
            string excursionId,
            string userId )
```

Implemented in deepdiveapi.Repositories.ExcursionParticipantsRepository.

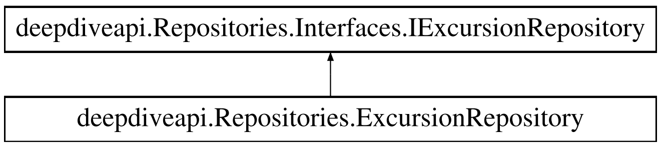
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/IExcursionParticipantsRepository.cs

5.32 deepdiveapi.Repositories.Interfaces.IExcursionRepository Interface Reference

The IExcursionRepository interface defines methods for managing excursions.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IExcursionRepository:


deepdiveapi.Repositories.Interfaces.IExcursionRepository
deepdiveapi.Repositories.ExcursionRepository

Public Member Functions

- Task< List< Excursion > > GetExcursions ()
- Task< Excursion > GetExcursionById (string input)
- Task< string > AddNewExcursion (NewExcursionDto input, string userId)
- Task DeleteExcursion (string id)
- Task UpdateExcursionAsync (Excursion excursion)
- Task<(List< Excursion > PastExcursions, List< Excursion > FutureExcursions)> GetUserExcursions (string userId)

5.32.1 Detailed Description

The IExcursionRepository interface defines methods for managing excursions.

5.32.2 Member Function Documentation

5.32.2.1 AddNewExcursion()

```
Task< string > deepdiveapi.Repositories.Interfaces.IExcursionRepository.AddNewExcursion (
            NewExcursionDto input,
            string userId )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

5.32.2.2 DeleteExcursion()

```
Task deepdiveapi.Repositories.Interfaces.IExcursionRepository.DeleteExcursion (
            string id )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

5.32.2.3 GetExcursionById()

```
Task< Excursion > deepdiveapi.Repositories.Interfaces.IExcursionRepository.GetExcursionById (
            string input )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

5.32.2.4 GetExcursions()

```
Task< List< Excursion > > deepdiveapi.Repositories.Interfaces.IExcursionRepository.Get↵
Excursions ( )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

5.32.2.5 GetUserExcursions()

```
Task<(List< Excursion > PastExcursions, List< Excursion > FutureExcursions)> deepdiveapi.↩
Repositories.Interfaces.IExcursionRepository.GetUserExcursions (
            string userId )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

5.32.2.6 UpdateExcursionAsync()

```
Task deepdiveapi.Repositories.Interfaces.IExcursionRepository.UpdateExcursionAsync (
            Excursion excursion )
```

Implemented in deepdiveapi.Repositories.ExcursionRepository.

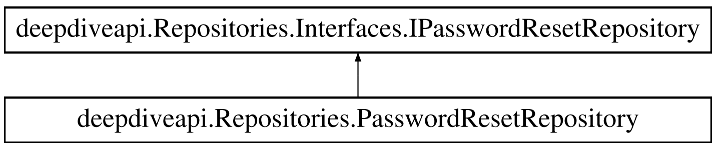
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/IExcursionRepository.cs

5.33 deepdiveapi.Repositories.Interfaces.IPasswordResetRepository Interface Reference

The IPasswordResetRepository interface defines methods for managing password reset requests.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IPasswordResetRepository:


deepdiveapi.Repositories.Interfaces.IPasswordResetRepository
deepdiveapi.Repositories.PasswordResetRepository

Public Member Functions

- Task AddPasswordReset (string email)
- PasswordReset FindByEmailAndToken (ValidatePasswordReset input)

5.33.1 Detailed Description

The IPasswordResetRepository interface defines methods for managing password reset requests.

5.33.2 Member Function Documentation

5.33.2.1 AddPasswordReset()

```
Task deepdiveapi.Repositories.Interfaces.IPasswordResetRepository.AddPasswordReset (
            string email )
```

Implemented in deepdiveapi.Repositories.PasswordResetRepository.

5.33.2.2 FindByEmailAndToken()

```
PasswordReset deepdiveapi.Repositories.Interfaces.IPasswordResetRepository.FindByEmailAndToken
(
            ValidatePasswordReset input )
```

Implemented in deepdiveapi.Repositories.PasswordResetRepository.

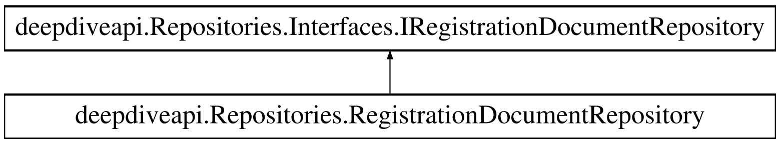
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/IPasswordResetRepository.cs

5.34 deepdiveapi.Repositories.Interfaces.IRegistrationDocument↩ Repository Interface Reference

The IRegistrationDocumentRepository interface defines methods for managing registration documents.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository:


deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository
deepdiveapi.Repositories.RegistrationDocumentRepository

Public Member Functions

- Task AddUserDocumentAsync (UploadRegisterDocumentDto input)
- Task< List< UserRegisterDocument > > GetDocumentsOfUser (string id)
- Task DeleteDocumentAsync (string docuId, string userId)

5.34.1 Detailed Description

The IRegistrationDocumentRepository interface defines methods for managing registration documents.

5.34.2 Member Function Documentation

5.34.2.1 AddUserDocumentAsync()

```
Task deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository.AddUserDocumentAsync
(
            UploadRegisterDocumentDto input )
```

Implemented in deepdiveapi.Repositories.RegistrationDocumentRepository.

5.34.2.2 DeleteDocumentAsync()

```
Task deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository.DeleteDocumentAsync (
            string docuId,
            string userId )
```

Implemented in deepdiveapi.Repositories.RegistrationDocumentRepository.

5.34.2.3 GetDocumentsOfUser()

```
Task< List< UserRegisterDocument > > deepdiveapi.Repositories.Interfaces.IRegistration↩
DocumentRepository.GetDocumentsOfUser (
            string id )
```

Implemented in deepdiveapi.Repositories.RegistrationDocumentRepository.

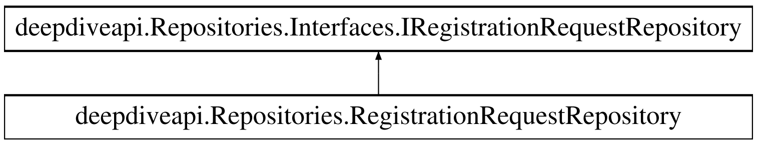
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/IRegistrationDocumentRepository.cs

5.35 deepdiveapi.Repositories.Interfaces.IRegistrationRequest↩ Repository Interface Reference

The IRegistrationRequestRepository interface defines methods for managing registration requests.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository:


deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository
deepdiveapi.Repositories.RegistrationRequestRepository

Public Member Functions

- Task AddRegistrationRequest (string userId)
- Task< RegistrationRequest > GetRegistrationRequestById (int requestId)
- Task< RegistrationRequest > GetRegistrationRequestByUserIdAsync (string userId)
- Task UpdateRegistrationRequestAsync (RegistrationRequest registrationRequest)

5.35.1 Detailed Description

The IRegistrationRequestRepository interface defines methods for managing registration requests.

5.35.2 Member Function Documentation

5.35.2.1 AddRegistrationRequest()

```
Task deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.AddRegistrationRequest
(
             string userId )
```

Implemented in deepdiveapi.Repositories.RegistrationRequestRepository.

5.35.2.2 GetRegistrationRequestById()

```
Task< RegistrationRequest > deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.↩
GetRegistrationRequestById (
             int requestId )
```

Implemented in deepdiveapi.Repositories.RegistrationRequestRepository.

5.35.2.3 GetRegistrationRequestByUserIdAsync()

```
Task< RegistrationRequest > deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.↩
GetRegistrationRequestByUserIdAsync (
             string userId )
```

Implemented in deepdiveapi.Repositories.RegistrationRequestRepository.

5.35.2.4 UpdateRegistrationRequestAsync()

```
Task deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.UpdateRegistration↩
RequestAsync (
             RegistrationRequest registrationRequest )
```

Implemented in deepdiveapi.Repositories.RegistrationRequestRepository.

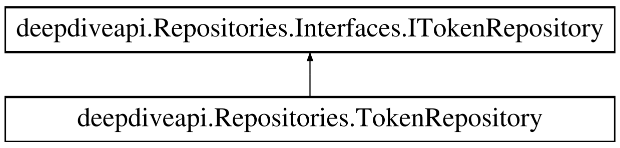
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/IRegistrationRequestRepository.cs

5.36 deepdiveapi.Repositories.Interfaces.ITokenRepository Interface Reference

The ITokenRepository interface defines methods for managing authentication tokens.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.ITokenRepository:


deepdiveapi.Repositories.Interfaces.ITokenRepository
deepdiveapi.Repositories.TokenRepository

Public Member Functions

- Task StoreRefreshTokenAsync (User user, RefreshToken refreshToken)
- Task InvalidateOldRefreshToken (User user, string oldRefreshToken, string newRefreshToken)
- Task< bool > IsValidRefreshTokenAsync (string userId, string refreshToken)

5.36.1 Detailed Description

The ITokenRepository interface defines methods for managing authentication tokens.

5.36.2 Member Function Documentation

5.36.2.1 InvalidateOldRefreshToken()

```
Task deepdiveapi.Repositories.Interfaces.ITokenRepository.InvalidateOldRefreshToken (
            User user,
            string oldRefreshToken,
            string newRefreshToken )
```

Implemented in deepdiveapi.Repositories.TokenRepository.

5.36.2.2 IsValidRefreshTokenAsync()

```
Task< bool > deepdiveapi.Repositories.Interfaces.ITokenRepository.IsValidRefreshTokenAsync (
            string userId,
            string refreshToken )
```

Implemented in deepdiveapi.Repositories.TokenRepository.

5.36.2.3 StoreRefreshTokenAsync()

```
Task deepdiveapi.Repositories.Interfaces.ITokenRepository.StoreRefreshTokenAsync (
            User user,
            RefreshToken refreshToken )
```

Implemented in deepdiveapi.Repositories.TokenRepository.

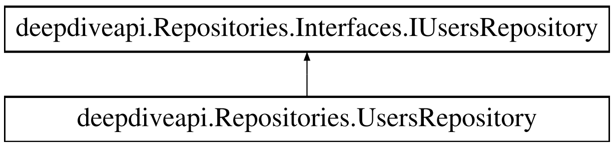
The documentation for this interface was generated from the following file:

- Repositories/Interfaces/ITokenRepository.cs

5.37 deepdiveapi.Repositories.Interfaces.IUsersRepository Interface Reference

The IUsersRepository interface defines methods for user management.

Inheritance diagram for deepdiveapi.Repositories.Interfaces.IUsersRepository:


deepdiveapi.Repositories.Interfaces.IUsersRepository
deepdiveapi.Repositories.UsersRepository

Public Member Functions

- Task< List< User > > GetUsers ()
- Task< List< UserForSafeListDto >?> GetSafeUsers ()
- Task< UserForSafeListDto?> GetSafeUserFromID (string id)
- Task< List< UsersTypeahead > > SearchUsersAsync (string searchString)
- Task EnableAccount (string userId)
- Task DisableAccount (string userId)
- Task DeleteAccount (string userId)

5.37.1 Detailed Description

The IUsersRepository interface defines methods for user management.

5.37.2 Member Function Documentation

5.37.2.1 DeleteAccount()

```
Task deepdiveapi.Repositories.Interfaces.IUsersRepository.DeleteAccount (
            string userId )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.2 DisableAccount()

```
Task deepdiveapi.Repositories.Interfaces.IUsersRepository.DisableAccount (
            string userId )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.3 EnableAccount()

```
Task deepdiveapi.Repositories.Interfaces.IUsersRepository.EnableAccount (
            string userId )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.4 GetSafeUserFromID()

```
Task< UserForSafeListDto?> deepdiveapi.Repositories.Interfaces.IUsersRepository.GetSafeUser←
FromID (
            string id )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.5 GetSafeUsers()

```
Task< List< UserForSafeListDto >?> deepdiveapi.Repositories.Interfaces.IUsersRepository.Get←
SafeUsers ( )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.6 GetUsers()

```
Task< List< User > > deepdiveapi.Repositories.Interfaces.IUsersRepository.GetUsers ( )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

5.37.2.7 SearchUsersAsync()

```
Task< List< UsersTypeahead > > deepdiveapi.Repositories.Interfaces.IUsersRepository.Search←
UsersAsync (
            string searchString )
```

Implemented in deepdiveapi.Repositories.UsersRepository.

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

- Repositories/Interfaces/IUserRepository.cs

5.38 deepdiveapi.JwtFeatures.JwtHandler Class Reference

Provides methods for managing JWT tokens including creation and validation.

Public Member Functions

- JwtHandler (IConfiguration configuration, UserManager< User > userManager)

    Initializes a new instance of the JwtHandler class.
- SigningCredentials GetSigningCredentials ()

    Generates signing credentials using the security key from configuration.
- async Task< List< Claim > > GetClaims (User user)

    Asynchronously generates a list of claims for a specified user.
- JwtSecurityToken GenerateTokenOptions (SigningCredentials signingCredentials, List< Claim > claims)

    Generates a JWT token using the provided signing credentials and claims.
- string GenerateRefreshToken ()

    Generates a refresh token.
- ClaimsPrincipal GetPrincipalFromExpiredToken (string? accessToken)

    Retrieves the principal from an expired JWT token.
- string GenerateAccessToken (IEnumerable< Claim > claims)

    Generates an access token using the specified claims.

Static Public Member Functions

- static string GetUserIdFromToken (HttpRequest request)

  Extracts and decodes the user ID from a JWT token.

5.38.1 Detailed Description

Provides methods for managing JWT tokens including creation and validation.

5.38.2 Constructor & Destructor Documentation

5.38.2.1 JwtHandler()

```
deepdiveapi.JwtFeatures.JwtHandler.JwtHandler (
            IConfiguration configuration,
            UserManager< User > userManager )  [inline]
```

Initializes a new instance of the JwtHandler class.

Parameters

configuration	Application configuration containing JWT settings.
userManager	User manager for accessing user data and roles.

5.38.3 Member Function Documentation

5.38.3.1 GenerateAccessToken()

```
string deepdiveapi.JwtFeatures.JwtHandler.GenerateAccessToken (
            IEnumerable< Claim > claims )  [inline]
```

Generates an access token using the specified claims.

Parameters

claims	Claims to include in the token.

Returns

A string representation of the JWT token.

5.38.3.2 GenerateRefreshToken()

```
string deepdiveapi.JwtFeatures.JwtHandler.GenerateRefreshToken ( )  [inline]
```

Generates a refresh token.

Returns

A randomly generated refresh token.

5.38.3.3 GenerateTokenOptions()

```
JwtSecurityToken deepdiveapi.JwtFeatures.JwtHandler.GenerateTokenOptions (
            SigningCredentials signingCredentials,
            List< Claim > claims )  [inline]
```

Generates a JWT token using the provided signing credentials and claims.

Parameters

signingCredentials	Signing credentials used to sign the token.
claims	Claims to include in the token.

Returns

A JwtSecurityToken object.

5.38.3.4 GetClaims()

```
async Task< List< Claim > > deepdiveapi.JwtFeatures.JwtHandler.GetClaims (
            User user )  [inline]
```

Asynchronously generates a list of claims for a specified user.

Parameters

user	The user for whom to generate claims.

Returns

A task representing the asynchronous operation, containing the list of user claims.

5.38.3.5 GetPrincipalFromExpiredToken()

```
ClaimsPrincipal deepdiveapi.JwtFeatures.JwtHandler.GetPrincipalFromExpiredToken (
            string?  accessToken )  [inline]
```

Retrieves the principal from an expired JWT token.

Parameters

accessToken	The expired JWT token.

Returns

ClaimsPrincipal extracted from the token.

5.38.3.6 GetSigningCredentials()

```
SigningCredentials deepdiveapi.JwtFeatures.JwtHandler.GetSigningCredentials ( )  [inline]
```

Generates signing credentials using the security key from configuration.

Returns

Signing credentials for creating secure JWTs.

5.38.3.7 GetUserIdFromToken()

```
static string deepdiveapi.JwtFeatures.JwtHandler.GetUserIdFromToken (
             HttpRequest request )  [inline], [static]
```

Extracts and decodes the user ID from a JWT token.

Parameters

request	The HttpRequest containing the token.

Returns

The user ID if present, otherwise null.

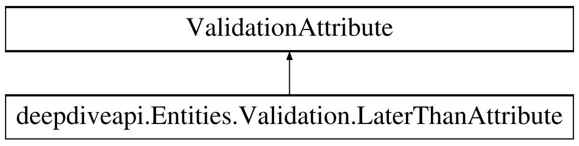
The documentation for this class was generated from the following file:

- JwtFeatures/JwtHandler.cs

5.39 deepdiveapi.Entities.Validation.LaterThanAttribute Class Reference

Specifies that the annotated date or time property must be later than another named property on the same object.

Inheritance diagram for deepdiveapi.Entities.Validation.LaterThanAttribute:



Public Member Functions

- LaterThanAttribute (string earlierPropertyName)

  Initializes a new instance of the LaterThanAttribute class.

Protected Member Functions

- override ValidationResult IsValid (object value, ValidationContext validationContext)

  Validates that the value of the property is later than the value of the referenced property.

5.39.1 Detailed Description

Specifies that the annotated date or time property must be later than another named property on the same object.

5.39.2 Constructor & Destructor Documentation

5.39.2.1 LaterThanAttribute()

```
deepdiveapi.Entities.Validation.LaterThanAttribute.LaterThanAttribute (
            string earlierPropertyName )  [inline]
```

Initializes a new instance of the LaterThanAttribute class.

Parameters

earlierPropertyName	The name of the property to compare with.

5.39.3 Member Function Documentation

5.39.3.1 IsValid()

```
override ValidationResult deepdiveapi.Entities.Validation.LaterThanAttribute.IsValid (
            object value,
            ValidationContext validationContext )  [inline], [protected]
```

Validates that the value of the property is later than the value of the referenced property.

Parameters

value	The value of the property being validated.
validationContext	Describes the context in which a validation check is performed.

Returns

A ValidationResult that represents the success or failure of the validation.

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

- Entities/Validation/LaterThenAttribute.cs

5.40 deepdiveapi.Entities.DataTransferObjects.NewExcursionDto Class Reference

DTO for creating a new excursion, including required information such as title, description, date/time, and location coordinates.

Properties

- string **Title** `[get, set]`
- string **Description** `[get, set]`
- DateTime **DateTime** `[get, set]`
- string **ImageName** `[get, set]`
- CoordinatesDto **Coordinates** `[get, set]`

5.40.1 Detailed Description

DTO for creating a new excursion, including required information such as title, description, date/time, and location coordinates.

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

- Entities/DataTransferObjects/NewExcursionDto.cs

5.41 deepdiveapi.Entities.DataTransferObjects.ParticipantDto Class Reference

Nested DTO representing detailed participant information.

Properties

- string **Id** `[get, set]`
- string **FirstName** `[get, set]`
- string **LastName** `[get, set]`
- string **UserName** `[get, set]`

5.41.1 Detailed Description

Nested DTO representing detailed participant information.

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

- Entities/DataTransferObjects/ExcursionDetailsDto.cs

5.42 deepdiveapi.Entities.DataTransferObjects.Participation↩ ConfirmationInputDto Class Reference

DTO for confirming participation in an event, including the event time, location, and a QR code key for validation.

Properties

- DateTime **DateTime** `[get, set]`
- CoordinatesDto **Coordinates** `[get, set]`
- string **QrCodeKey** `[get, set]`

5.42.1 Detailed Description

DTO for confirming participation in an event, including the event time, location, and a QR code key for validation.

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

- Entities/DataTransferObjects/ParticipationConfirmationInputDto.cs

5.43 deepdiveapi.Entities.Models.PasswordReset Class Reference

Represents a password reset request.

Properties

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

  Gets or sets the unique identifier for the password reset request.
- string **Token** `[get, set]`

  Gets or sets the token associated with the password reset.
- string **UserIdFK** `[get, set]`

  Gets or sets the foreign key of the user associated with this password reset.
- User **User** `[get, set]`

  Navigation property for the user associated with this password reset.
- DateTime **CreatedOn** `[get, set]`

  Gets or sets the date and time when the password reset was created.
- DateTime **ExpireOn** `[get, set]`

  Gets or sets the expiration date and time for the password reset token.
- PasswordResetsStastusEnum **Status** `[get, set]`

  Gets or sets the status of the password reset process.

5.43.1 Detailed Description

Represents a password reset request.

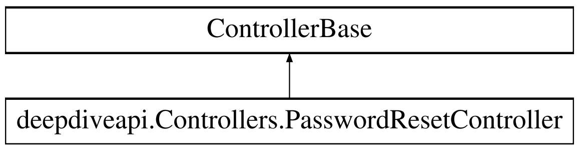
The documentation for this class was generated from the following file:

- Entities/Models/PasswordReset.cs

5.44 deepdiveapi.Controllers.PasswordResetController Class Reference

Controller responsible for handling password reset requests.

Inheritance diagram for deepdiveapi.Controllers.PasswordResetController:



Public Member Functions

- PasswordResetController (IPasswordResetRepository passwordResetrepo, ApplicationDbContext db↩ Context, UserManager< User > userManager)

  Initializes a new instance of the PasswordResetController class.
- async Task< IActionResult > AddPasswordRequest (EmailInputDto email)

  Initiates a password reset request for the given email address.
- async Task< IActionResult > ValidateReset (ValidatePasswordReset input)

  Validates a password reset request based on the provided input.
- async Task< IActionResult > UpdatePassword (UpdateUserPassword input)

  Updates the password for a user based on the provided input.

5.44.1 Detailed Description

Controller responsible for handling password reset requests.

5.44.2 Constructor & Destructor Documentation

5.44.2.1 PasswordResetController()

```
deepdiveapi.Controllers.PasswordResetController.PasswordResetController (
            IPasswordResetRepository passwordResetrepo,
            ApplicationDbContext dbContext,
            UserManager< User > userManager )  [inline]
```

Initializes a new instance of the PasswordResetController class.

Parameters

passwordResetrepo	Repository for handling password reset operations.
dbContext	Database context for accessing application data.
userManager	User manager for user-related operations.

5.44.3 Member Function Documentation

5.44.3.1 AddPasswordRequest()

```
async Task< IActionResult > deepdiveapi.Controllers.PasswordResetController.AddPasswordRequest
(
            EmailInputDto email )  [inline]
```

Initiates a password reset request for the given email address.

Parameters

email	DTO containing the email address for which the password reset is requested.

Returns

Ok if the request is successful, or InternalServerError on exception.

5.44.3.2 UpdatePassword()

```
async Task< IActionResult > deepdiveapi.Controllers.PasswordResetController.UpdatePassword (
            UpdateUserPassword input )  [inline]
```

Updates the password for a user based on the provided input.

Parameters

input	DTO containing information required to update the user's password.

Returns

Ok if the password is updated successfully, BadRequest if not found, expired, or update failed, or Internal←
ServerError on exception.

5.44.3.3 ValidateReset()

```
async Task< IActionResult > deepdiveapi.Controllers.PasswordResetController.ValidateReset (
            ValidatePasswordReset input )  [inline]
```

Validates a password reset request based on the provided input.

Parameters

input	DTO containing information required to validate the password reset request.

Returns

Ok if the request is valid, BadRequest if not found, expired, or already used, or InternalServerError on exception.

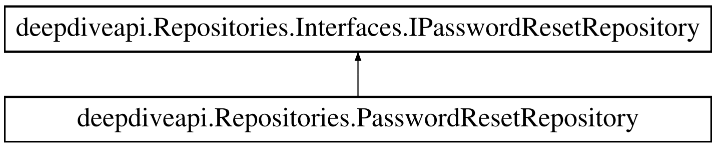
The documentation for this class was generated from the following file:

- Controllers/PasswordResetController.cs

5.45 deepdiveapi.Repositories.PasswordResetRepository Class Reference

The PasswordResetRepository class manages operations related to user password resets.

Inheritance diagram for deepdiveapi.Repositories.PasswordResetRepository:


deepdiveapi.Repositories.Interfaces.IPasswordResetRepository
deepdiveapi.Repositories.PasswordResetRepository

Public Member Functions

- PasswordResetRepository (ApplicationDbContext dbContext, UserManager< User > userManager, IConfiguration configuration)

  Constructor for PasswordResetRepository.
- async Task AddPasswordReset (string email)

  Initiates a password reset process for a user by adding a password reset entry to the database and sending an email with reset instructions.
- PasswordReset FindByEmailAndToken (ValidatePasswordReset input)

  Finds a password reset entry based on a user ID and token.

5.45.1 Detailed Description

The PasswordResetRepository class manages operations related to user password resets.

5.45.2 Constructor & Destructor Documentation

5.45.2.1 PasswordResetRepository()

```
deepdiveapi.Repositories.PasswordResetRepository.PasswordResetRepository (
            ApplicationDbContext dbContext,
            UserManager< User > userManager,
            IConfiguration configuration )  [inline]
```

Constructor for PasswordResetRepository.

Parameters

dbContext	Application database context.
userManager	Manages user data from a database.
configuration	Configuration interface to access application settings.

5.45.3 Member Function Documentation

5.45.3.1 AddPasswordReset()

```
async Task deepdiveapi.Repositories.PasswordResetRepository.AddPasswordReset (
            string email )  [inline]
```

Initiates a password reset process for a user by adding a password reset entry to the database and sending an email with reset instructions.

Parameters

email	The email address of the user requesting the password reset.

Implements deepdiveapi.Repositories.Interfaces.IPasswordResetRepository.

5.45.3.2 FindByEmailAndToken()

```
PasswordReset deepdiveapi.Repositories.PasswordResetRepository.FindByEmailAndToken (
            ValidatePasswordReset input )  [inline]
```

Finds a password reset entry based on a user ID and token.

Parameters

input	Data transfer object containing the user ID and token to validate.

Returns

PasswordReset entity if found; otherwise null.

Implements deepdiveapi.Repositories.Interfaces.IPasswordResetRepository.

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

- Repositories/PasswordResetRepository.cs

5.46 deepdiveapi.Entities.Models.RefreshToken Class Reference

Represents a refresh token for maintaining user authentication sessions.

Properties

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

  Gets or sets the unique identifier for the refresh token.
- string **Token** `[get, set]`

  Gets or sets the token string.
- DateTime **Expires** `[get, set]`

  Gets or sets the expiration date and time for the token.
- bool **IsExpired** `[get]`

  Determines if the token is expired.
- DateTime **Created** `[get, set]`

  Gets or sets the creation date and time for the token.
- DateTime? **Revoked** `[get, set]`

  Optional date and time when the token was revoked.
- string? **ReplacedByToken** `[get, set]`

  Optional token that replaces this one.
- bool **IsActive** `[get]`

  Determines if the token is active and not expired or revoked.
- string **UserIdFK** `[get, set]`

  Gets or sets the foreign key of the user who owns this token.
- User **User** `[get, set]`

  Navigation property for the user associated with this token.

5.46.1 Detailed Description

Represents a refresh token for maintaining user authentication sessions.

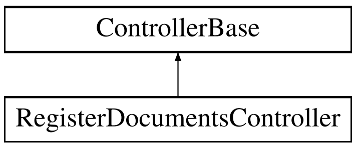
The documentation for this class was generated from the following file:

- Entities/Models/RefreshToken.cs

5.47 RegisterDocumentsController Class Reference

Controller responsible for managing user registration documents.

Inheritance diagram for RegisterDocumentsController:



Public Member Functions

- RegisterDocumentsController (UserManager< User > userManager, IRegistrationDocumentRepository registrationDocumentRepository)

  Initializes a new instance of the RegisterDocumentsController class.
- async Task< IActionResult > UploadDocument ([FromBody] UploadRegisterDocumentDto input)

  Uploads a document for user registration.
- async Task< IActionResult > DeleteDocument ([FromBody] DocuIdInput input)

  Deletes a document associated with a user.

5.47.1 Detailed Description

Controller responsible for managing user registration documents.

5.47.2 Constructor & Destructor Documentation

5.47.2.1 RegisterDocumentsController()

```
RegisterDocumentsController.RegisterDocumentsController (
            UserManager< User > userManager,
            IRegistrationDocumentRepository registrationDocumentRepository )  [inline]
```

Initializes a new instance of the RegisterDocumentsController class.

Parameters

userManager	User manager for user-related operations.
registrationDocumentRepository	Repository for accessing and managing registration documents.

5.47.3 Member Function Documentation

5.47.3.1 DeleteDocument()

```
async Task< IActionResult > RegisterDocumentsController.DeleteDocument (
            [FromBody] DocuIdInput input )  [inline]
```

Deletes a document associated with a user.

Parameters

input	DTO containing the ID of the document to delete.

Returns

Ok if the document is deleted successfully, BadRequest if the user is not found, or InternalServerError on exception.

5.47.3.2 UploadDocument()

```
async Task< IActionResult > RegisterDocumentsController.UploadDocument (
            [FromBody] UploadRegisterDocumentDto input )  [inline]
```

Uploads a document for user registration.

Parameters

input	DTO containing information about the document to upload.

Returns

Ok if the document is uploaded successfully, or InternalServerError on exception.

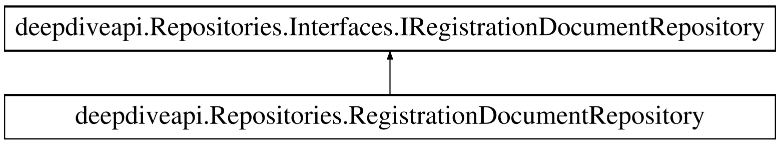
The documentation for this class was generated from the following file:

- Controllers/RegisterDocumentsController.cs

5.48 deepdiveapi.Repositories.RegistrationDocumentRepository Class Reference

The RegistrationDocumentRepository class manages operations related to user registration documents.

Inheritance diagram for deepdiveapi.Repositories.RegistrationDocumentRepository:


deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository
deepdiveapi.Repositories.RegistrationDocumentRepository

Public Member Functions

- RegistrationDocumentRepository (ApplicationDbContext dbContext)

  Constructor for RegistrationDocumentRepository.
- async Task AddUserDocumentAsync (UploadRegisterDocumentDto input)

  Adds a new document to a user's profile.
- async Task< List< UserRegisterDocument > > GetDocumentsOfUser (string id)

  Retrieves all documents associated with a specific user.
- async Task DeleteDocumentAsync (string documentId, string userId)

  Deletes a specific document associated with a user.

5.48.1 Detailed Description

The RegistrationDocumentRepository class manages operations related to user registration documents.

5.48.2 Constructor & Destructor Documentation

5.48.2.1 RegistrationDocumentRepository()

```
deepdiveapi.Repositories.RegistrationDocumentRepository.RegistrationDocumentRepository (
            ApplicationDbContext dbContext )  [inline]
```

Constructor for RegistrationDocumentRepository.

Parameters

dbContext	Application database context.

5.48.3 Member Function Documentation

5.48.3.1 AddUserDocumentAsync()

```
async Task deepdiveapi.Repositories.RegistrationDocumentRepository.AddUserDocumentAsync (
            UploadRegisterDocumentDto input )  [inline]
```

Adds a new document to a user's profile.

Parameters

input	Data transfer object containing information about the document to add.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository.

5.48.3.2 DeleteDocumentAsync()

```
async Task deepdiveapi.Repositories.RegistrationDocumentRepository.DeleteDocumentAsync (
            string documentId,
            string userId )  [inline]
```

Deletes a specific document associated with a user.

Parameters

document↩ Id	The ID of the document to delete.
userId	The ID of the user the document is associated with.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository.

5.48.3.3 GetDocumentsOfUser()

```
async Task< List< UserRegisterDocument > > deepdiveapi.Repositories.RegistrationDocument↩
Repository.GetDocumentsOfUser (
            string id )  [inline]
```

Retrieves all documents associated with a specific user.

Parameters

id	The ID of the user to retrieve documents for.

Returns

A task that represents the asynchronous operation. The task result contains a list of UserRegisterDocument entities.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository.

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

- Repositories/RegisterDocumentRepository.cs

5.49 deepdiveapi.Entities.Models.RegistrationRequest Class Reference

Represents a request for user registration status change.

Properties

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

  Gets or sets the unique identifier for the registration request.
- string **UserIdFK** `[get, set]`

  Gets or sets the foreign key of the user associated with this request.
- User **User** `[get, set]`

  Navigation property for the user associated with this request.
- RegistrationStatusEnum **RegistrationStatus** `[get, set]`

  Gets or sets the current status of the registration request.
- string? **AdminComment** `[get, set]`

  Optional comment by an administrator concerning the request.
- DateTime **CreatedOn** `[get, set]`

  Gets or sets the date and time when the request was created.
- DateTime? **EditedOn** `[get, set]`

  Optional date and time when the request was last edited.
- DateTime? **ApprovedOrDeniedOn** `[get, set]`

  Optional date and time when the request was approved or denied.

5.49.1 Detailed Description

Represents a request for user registration status change.

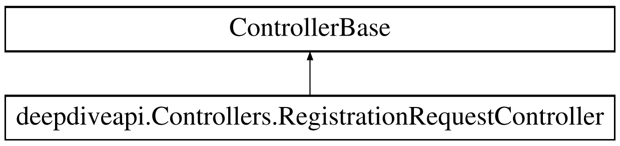
The documentation for this class was generated from the following file:

- Entities/Models/RegistrationRequest.cs

5.50 deepdiveapi.Controllers.RegistrationRequestController Class Reference

Controller responsible for managing registration requests.

Inheritance diagram for deepdiveapi.Controllers.RegistrationRequestController:



Public Member Functions

- RegistrationRequestController (IRegistrationRequestRepository registrationRequestRepository, ApplicationDbContext dbContext, UserManager< User > userManager, IConfiguration configuration)

  Initializes a new instance of the RegistrationRequestController class.
- async Task< IActionResult > UpdateStatus (UpdateStatusRegistrationRequestDto input)

  Updates the status of a registration request for a specific user.
- async Task< IActionResult > UpdateStatus (UpdateRegistrationRequestDto input)

  Updates the status of a registration request for a specific user.

5.50.1 Detailed Description

Controller responsible for managing registration requests.

5.50.2 Constructor & Destructor Documentation

5.50.2.1 RegistrationRequestController()

```
deepdiveapi.Controllers.RegistrationRequestController.RegistrationRequestController (
            IRegistrationRequestRepository registrationRequestRepository,
            ApplicationDbContext dbContext,
            UserManager< User > userManager,
            IConfiguration configuration )  [inline]
```

Initializes a new instance of the RegistrationRequestController class.

Parameters

registrationRequestRepository	Repository for accessing and updating registration requests.
dbContext	Database context for entity operations.
userManager	User manager for user-related operations.
configuration	Configuration for accessing application settings.

5.50.3 Member Function Documentation

5.50.3.1 UpdateStatus() [1/2]

```
async Task< IActionResult > deepdiveapi.Controllers.RegistrationRequestController.UpdateStatus
(
            UpdateRegistrationRequestDto input )  [inline]
```

Updates the status of a registration request for a specific user.

Parameters

input	DTO containing the user ID and new status for the registration request.

Returns

Ok if the update is successful, BadRequest if the request is not found, or InternalServerError on exception.

5.50.3.2 UpdateStatus() [2/2]

```
async Task< IActionResult > deepdiveapi.Controllers.RegistrationRequestController.UpdateStatus
(
            UpdateStatusRegistrationRequestDto input )  [inline]
```

Updates the status of a registration request for a specific user.

Parameters

input	DTO containing the user ID and new status for the registration request.

Returns

Ok if the update is successful, BadRequest if the request is not found, or InternalServerError on exception.

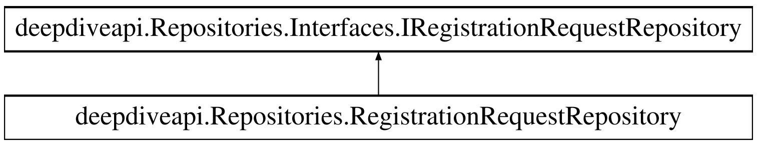
The documentation for this class was generated from the following file:

- Controllers/RegistrationRequestController.cs

5.51 deepdiveapi.Repositories.RegistrationRequestRepository Class Reference

The RegistrationRequestRepository class manages operations related to user registration requests.

Inheritance diagram for deepdiveapi.Repositories.RegistrationRequestRepository:


deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository
deepdiveapi.Repositories.RegistrationRequestRepository

Public Member Functions

- RegistrationRequestRepository (ApplicationDbContext context)

  Constructor for RegistrationRequestRepository.
- async Task AddRegistrationRequest (string userId)

  Adds a new registration request for a user.
- async Task< RegistrationRequest > GetRegistrationRequestByUserIdAsync (string userId)

  Retrieves a registration request by user ID.
- async Task UpdateRegistrationRequestAsync (RegistrationRequest registrationRequest)

  Updates a registration request.
- async Task< RegistrationRequest > GetRegistrationRequestById (int requestId)

  Retrieves a registration request by its ID.

5.51.1 Detailed Description

The RegistrationRequestRepository class manages operations related to user registration requests.

5.51.2 Constructor & Destructor Documentation

5.51.2.1 RegistrationRequestRepository()

```
deepdiveapi.Repositories.RegistrationRequestRepository.RegistrationRequestRepository (
            ApplicationDbContext context )  [inline]
```

Constructor for RegistrationRequestRepository.

Parameters

context	Application database context.

5.51.3 Member Function Documentation

5.51.3.1 AddRegistrationRequest()

```
async Task deepdiveapi.Repositories.RegistrationRequestRepository.AddRegistrationRequest (
            string userId )  [inline]
```

Adds a new registration request for a user.

Parameters

user↩ Id	The ID of the user to add a registration request for.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.

5.51.3.2 GetRegistrationRequestById()

```
async Task< RegistrationRequest > deepdiveapi.Repositories.RegistrationRequestRepository.Get↩
RegistrationRequestById (
             int requestId )  [inline]
```

Retrieves a registration request by its ID.

Parameters

request↩ Id	The ID of the registration request to search for.

Returns

A task that represents the asynchronous operation. The task result contains a RegistrationRequest entity, or null if no request is found.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.

5.51.3.3 GetRegistrationRequestByUserIdAsync()

```
async Task< RegistrationRequest > deepdiveapi.Repositories.RegistrationRequestRepository.Get↩
RegistrationRequestByUserIdAsync (
             string userId )  [inline]
```

Retrieves a registration request by user ID.

Parameters

user↩ Id	The user ID to search for a registration request.

Returns

A task that represents the asynchronous operation. The task result contains a RegistrationRequest entity, or null if no request is found.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.

5.51.3.4 UpdateRegistrationRequestAsync()

```
async Task deepdiveapi.Repositories.RegistrationRequestRepository.UpdateRegistrationRequest↩
Async (
             RegistrationRequest registrationRequest )  [inline]
```

Updates a registration request.

Parameters

registrationRequest	The registration request to update.

Implements deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository.

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

- Repositories/RegistrationRequestRepository.cs

5.52 deepdiveapi.Entities.DataTransferObjects.RegistrationResponse↩ Dto Class Reference

DTO representing the response for a registration operation, indicating success or failure and providing details on any errors.

Properties

- bool **IsSuccessfulRegistration** `[get, set]`
- IEnumerable< string >? **Errors** `[get, set]`

5.52.1 Detailed Description

DTO representing the response for a registration operation, indicating success or failure and providing details on any errors.

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

- Entities/DataTransferObjects/RegistrationResponseDto.cs

5.53 deepdiveapi.Entities.DataTransferObjects.StringInputDto Class Reference

DTO for a simple string input, typically used for searches or single field submissions.

Properties

- string **input** `[get, set]`

5.53.1 Detailed Description

DTO for a simple string input, typically used for searches or single field submissions.

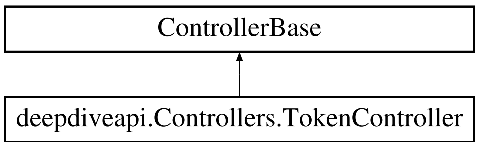
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/StringInputDto.cs

5.54 deepdiveapi.Controllers.TokenController Class Reference

Controller responsible for managing JWT token operations such as refreshing tokens.

Inheritance diagram for deepdiveapi.Controllers.TokenController:



Public Member Functions

- TokenController (UserManager< User > userManager, ITokenRepository tokenRepository, JwtHandler jwt←Handler)

  Initializes a new instance of the TokenController class.
- async Task< IActionResult > RefreshAsync (TokenRefreshDto tokenRefreshDto)

  Refreshes a JWT token using an old token and a refresh token.

5.54.1 Detailed Description

Controller responsible for managing JWT token operations such as refreshing tokens.

5.54.2 Constructor & Destructor Documentation

5.54.2.1 TokenController()

```
deepdiveapi.Controllers.TokenController.TokenController (
            UserManager< User > userManager,
            ITokenRepository tokenRepository,
            JwtHandler jwtHandler ) [inline]
```

Initializes a new instance of the TokenController class.

Parameters

userManager	The user manager for performing user-related operations.
tokenRepository	The repository handling token storage and validation.
jwtHandler	The JWT handler for managing JWT creation and validation.

5.54.3 Member Function Documentation

5.54.3.1 RefreshAsync()

```
async Task< IActionResult > deepdiveapi.Controllers.TokenController.RefreshAsync (
            TokenRefreshDto tokenRefreshDto ) [inline]
```

Refreshes a JWT token using an old token and a refresh token.

Parameters

tokenRefreshDto	The DTO containing the access and refresh tokens.

Returns

A new access token and refresh token if validation is successful; otherwise, unauthorized response.

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

- Controllers/TokenController.cs

5.55 deepdiveapi.Entities.DataTransferObjects.TokenRefreshDto Class Reference

DTO for refreshing an authentication token.

Properties

- required string **AccessToken** `[get, set]`
- required string **RefreshToken** `[get, set]`

5.55.1 Detailed Description

DTO for refreshing an authentication token.

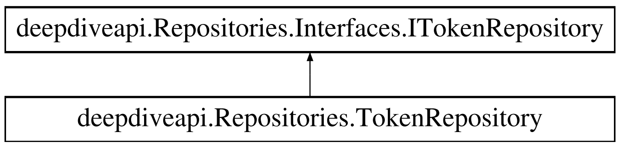
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/TokenRefreshDto.cs

5.56 deepdiveapi.Repositories.TokenRepository Class Reference

The TokenRepository class manages operations related to user authentication tokens, including storing and invalidating refresh tokens.

Inheritance diagram for deepdiveapi.Repositories.TokenRepository:


deepdiveapi.Repositories.Interfaces.ITokenRepository
deepdiveapi.Repositories.TokenRepository

Public Member Functions

- TokenRepository (ApplicationDbContext context)

    Constructor for TokenRepository.
- async Task StoreRefreshTokenAsync (User user, RefreshToken refreshToken)

    Stores a new refresh token for a user.
- async Task InvalidateOldRefreshToken (User user, string oldRefreshToken, string newRefrehToken)

    Invalidates an old refresh token and optionally replaces it with a new one.
- async Task< bool > IsValidRefreshTokenAsync (string userId, string refreshToken)

    Checks if a given refresh token is valid for a specific user.

5.56.1 Detailed Description

The TokenRepository class manages operations related to user authentication tokens, including storing and invalidating refresh tokens.

5.56.2 Constructor & Destructor Documentation

5.56.2.1 TokenRepository()

```
deepdiveapi.Repositories.TokenRepository.TokenRepository (
             ApplicationDbContext context )  [inline]
```

Constructor for TokenRepository.

Parameters

context	Application database context.

5.56.3 Member Function Documentation

5.56.3.1 InvalidateOldRefreshToken()

```
async Task deepdiveapi.Repositories.TokenRepository.InvalidateOldRefreshToken (
             User user,
             string oldRefreshToken,
             string newRefrehToken )  [inline]
```

Invalidates an old refresh token and optionally replaces it with a new one.

Parameters

user	The user associated with the refresh token.
oldRefreshToken	The old refresh token to be invalidated.
newRefreshToken	The new refresh token to replace the old one, if any.

Implements deepdiveapi.Repositories.Interfaces.ITokenRepository.

5.56.3.2 IsValidRefreshTokenAsync()

```
async Task< bool > deepdiveapi.Repositories.TokenRepository.IsValidRefreshTokenAsync (
            string userId,
            string refreshToken )  [inline]
```

Checks if a given refresh token is valid for a specific user.

Parameters

userId	The ID of the user.
refreshToken	The refresh token to validate.

Returns

A task that represents the asynchronous operation. The task result contains a boolean indicating whether the token is valid.

Implements deepdiveapi.Repositories.Interfaces.ITokenRepository.

5.56.3.3 StoreRefreshTokenAsync()

```
async Task deepdiveapi.Repositories.TokenRepository.StoreRefreshTokenAsync (
            User user,
            RefreshToken refreshToken )  [inline]
```

Stores a new refresh token for a user.

Parameters

user	The user to associate the refresh token with.
refreshToken	The refresh token to store.

Implements deepdiveapi.Repositories.Interfaces.ITokenRepository.

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

- Repositories/TokenRepository.cs

5.57 deepdiveapi.Entities.DataTransferObjects.UpdateExcursion↩ RequestDto Class Reference

DTO for updating an excursion's details including title, description, and location coordinates.

Properties

- string **Id** [get, set]
- string **Title** [get, set]
- string **Description** [get, set]
- DateTime **DateTime** [get, set]
- string? **ImageName** [get, set]
- CoordinatesDto2 **Coordinates** [get, set]

5.57.1 Detailed Description

DTO for updating an excursion's details including title, description, and location coordinates.

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

- Entities/DataTransferObjects/UpdateExcursionRequestDto.cs

5.58 deepdiveapi.Entities.DataTransferObjects.UpdateRegistration↩ RequestDto Class Reference

DTO for updating a registration request with status and optional administrator comments.

Properties

- required int **RequestId** `[get, set]`
- required RegistrationStatusEnum **RegistrationStatus** `[get, set]`
- string? **AdminComment** `[get, set]`

5.58.1 Detailed Description

DTO for updating a registration request with status and optional administrator comments.

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

- Entities/DataTransferObjects/UpdateRegistrationRequestDto.cs

5.59 deepdiveapi.Entities.DataTransferObjects.UpdateStatus↩ RegistrationRequestDto Class Reference

DTO for updating the registration status of a user.

Properties

- string **UserId** `[get, set]`
- RegistrationStatusEnum **Status** `[get, set]`

5.59.1 Detailed Description

DTO for updating the registration status of a user.

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

- Entities/DataTransferObjects/UpdateStatusRegistrationRequestDto.cs

5.60 deepdiveapi.Entities.DataTransferObjects.UpdateUserInputDto Class Reference

DTO for updating user profile information.

Properties

- string **Id** `[get, set]`
- string **FirstName** `[get, set]`
- string **LastName** `[get, set]`
- DateOnly **BirthDate** `[get, set]`
- string **UserName** `[get, set]`
- string **Email** `[get, set]`
- string? **Password** `[get, set]`
- string **PhoneNumber** `[get, set]`

5.60.1 Detailed Description

DTO for updating user profile information.

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

- Entities/DataTransferObjects/UpdateUserInputDto.cs

5.61 deepdiveapi.Entities.DataTransferObjects.UpdateUserPassword Class Reference

DTO for updating a user's password, requiring verification via token.

Properties

- required string **Id** `[get, set]`
- required string **Token** `[get, set]`
- required string **Password** `[get, set]`

5.61.1 Detailed Description

DTO for updating a user's password, requiring verification via token.

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

- Entities/DataTransferObjects/UpdateUserPassword.cs

5.62 deepdiveapi.Entities.DataTransferObjects.UpdateUserResponseDto Class Reference

DTO representing the response from a user update operation, indicating success or failure and any errors.

Properties

- bool **IsSuccessfullUpdate** `[get, set]`
- IEnumerable< string >? **Errors** `[get, set]`

5.62.1 Detailed Description

DTO representing the response from a user update operation, indicating success or failure and any errors.

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

- Entities/DataTransferObjects/UpdateUserResponseDto.cs

5.63 deepdiveapi.Entities.DataTransferObjects.UploadRegister↩ DocumentDto Class Reference

DTO for uploading a registration document, specifying the document type and associated user.

Properties

- string **Id** `[get, set]`
- string **DocumentName** `[get, set]`
- string **UserIdFK** `[get, set]`
- RegistrationDocumentTypes **DocumentType** `[get, set]`
- DateTime **CreatedOn** `[get, set]`

5.63.1 Detailed Description

DTO for uploading a registration document, specifying the document type and associated user.

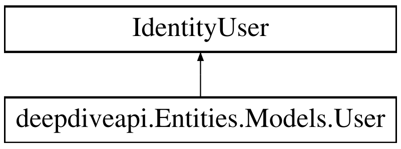
The documentation for this class was generated from the following file:

- Entities/DataTransferObjects/UploadRegisterDocumentDto.cs

5.64 deepdiveapi.Entities.Models.User Class Reference

Represents user information derived from IdentityUser.

Inheritance diagram for deepdiveapi.Entities.Models.User:



Properties

- string **FirstName** `[get, set]`

  Gets or sets the first name of the user.
- string **LastName** `[get, set]`

  Gets or sets the last name of the user.
- DateOnly **BirthDate** `[get, set]`

  Gets or sets the birth date of the user.
- bool **IsDeleted** `[get, set]`

  Indicates whether the user has been logically deleted from the system. (=disabled)
- ICollection< RefreshToken > **RefreshTokens** `[get, set]`
- ICollection< UserRegisterDocument > **UserRegisterDocuments** `[get, set]`
- RegistrationRequest **RegistrationRequest** `[get, set]`
- ICollection< PasswordReset > **PasswordResets** `[get, set]`
- ICollection< Excursion > **Excursions** `[get, set]`
- ICollection< ExcursionParticipant > **ParticipatingInExcursions** `[get, set]`

5.64.1 Detailed Description

Represents user information derived from IdentityUser.

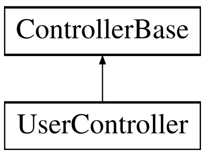
The documentation for this class was generated from the following file:

- Entities/Models/User.cs

5.65 UserController Class Reference

Controller responsible for managing user-specific operations such as retrieving and updating user information.

Inheritance diagram for UserController:



Public Member Functions

- UserController (UserManager< User > userManager, ApplicationDbContext dbContext, IRegistrationDocumentRepository registrationDocumentRepository, IUsersRepository usersRepository, IHttpContextAccessor httpContext←Accessor)

    Initializes a new instance of the UserController class.
- async Task< IActionResult > GetUsers ()

    Retrieves all users from the repository.
- async Task< IActionResult > GetUsersSearchTypehead (StringInputDto input)

    Retrieves users based on a typeahead search.
- async Task< IActionResult > GetSafeUsers ()

    Retrieves a list of safe user information, typically excluding sensitive details.
- async Task< IActionResult > GetSafeUsers (string userId)

    Retrieves safe user information for a specific user ID.
- async Task< IActionResult > ViewApplicationRegistration (IdInputDto input)

    Views the registration application for a specific user by user ID.
- async Task< IActionResult > Update (UpdateUserInputDto input)

    Updates the details of an existing user.
- async Task< IActionResult > DisableAccount (IdInputDto input)

    Disables a user account.
- async Task< IActionResult > EnableAccount (EmailInputDto input)

    Enables a user account.
- async Task< IActionResult > DeleteAccount (IdInputDto input)

    Deletes a user account.

5.65.1 Detailed Description

Controller responsible for managing user-specific operations such as retrieving and updating user information.

5.65.2 Constructor & Destructor Documentation

5.65.2.1 UserController()

```
UserController.UserController (
            UserManager< User > userManager,
            ApplicationDbContext dbContext,
            IRegistrationDocumentRepository registrationDocumentRepository,
            IUsersRepository usersRepository,
            IHttpContextAccessor httpContextAccessor )  [inline]
```

Initializes a new instance of the UserController class.

Parameters

userManager	Provides the APIs for managing user in a persistence store.
dbContext	Database context for entity framework operations.
registrationDocumentRepository	Repository for handling registration documents.
usersRepository	Repository for user-specific operations.
httpContextAccessor	Accessor for retrieving information about the HTTP context.

5.65.3 Member Function Documentation

5.65.3.1 DeleteAccount()

```
async Task< IActionResult > UserController.DeleteAccount (
            IdInputDto input )  [inline]
```

Deletes a user account.

Parameters

input	DTO containing the ID of the user to delete.

Returns

Response indicating success or failure of the delete operation.

5.65.3.2 DisableAccount()

```
async Task< IActionResult > UserController.DisableAccount (
            IdInputDto input )  [inline]
```

Disables a user account.

Parameters

input	DTO containing the ID of the user to disable.

Returns

Response indicating success or failure of the disable operation.

5.65.3.3 EnableAccount()

```
async Task< IActionResult > UserController.EnableAccount (
            EmailInputDto input )  [inline]
```

Enables a user account.

Parameters

input	DTO containing the email of the user to enable.

Returns

Response indicating success or failure of the enable operation.

5.65.3.4 GetSafeUsers() [1/2]

```
async Task< IActionResult > UserController.GetSafeUsers ( )  [inline]
```

Retrieves a list of safe user information, typically excluding sensitive details.

Returns

A list of user information considered safe to share.

5.65.3.5 GetSafeUsers() [2/2]

```
async Task< IActionResult > UserController.GetSafeUsers (
            string userId )  [inline]
```

Retrieves safe user information for a specific user ID.

Parameters

user↩ Id	The ID of the user.

Returns

Safe user information if found; otherwise, a not found result.

5.65.3.6 GetUsers()

```
async Task< IActionResult > UserController.GetUsers ( )  [inline]
```

Retrieves all users from the repository.

Returns

A list of users.

5.65.3.7 GetUsersSearchTypehead()

```
async Task< IActionResult > UserController.GetUsersSearchTypehead (
            StringInputDto input )  [inline]
```

Retrieves users based on a typeahead search.

Parameters

input	Input DTO containing the search string.

Returns

A list of users that match the search criteria.

5.65.3.8 Update()

```
async Task< IActionResult > UserController.Update (
            UpdateUserInputDto input )  [inline]
```

Updates the details of an existing user.

Parameters

input	DTO containing updated user details.

Returns

Response indicating success or failure of the update operation.

5.65.3.9 ViewApplicationRegistration()

```
async Task< IActionResult > UserController.ViewApplicationRegistration (
            IdInputDto input )  [inline]
```

Views the registration application for a specific user by user ID.

Parameters

input	DTO containing the user ID.

Returns

The registration application details if found; otherwise, NotFound.

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

- Controllers/UserController.cs

5.66 deepdiveapi.Entities.DataTransferObjects.UserDto Class Reference

Nested DTO representing basic user information within an excursion context.

Properties

- string **Id** `[get, set]`
- string **FirstName** `[get, set]`
- string **LastName** `[get, set]`

5.66.1 Detailed Description

Nested DTO representing basic user information within an excursion context.

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

- Entities/DataTransferObjects/ExcursionDetailsDto.cs

5.67 deepdiveapi.Entities.DataTransferObjects.UserForAuthentication↩ Dto Class Reference

DTO for user login credentials.

Properties

- string? **Email** `[get, set]`
- string? **Password** `[get, set]`

5.67.1 Detailed Description

DTO for user login credentials.

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

- Entities/DataTransferObjects/UserForAuthenticationDto.cs

5.68 deepdiveapi.Entities.DataTransferObjects.UserForRegistrationDto Class Reference

DTO for user registration details, including validations for required fields and data consistency.

Properties

- string? **FirstName** `[get, set]`
- string? **LastName** `[get, set]`
- string? **UserName** `[get, set]`
- string? **Email** `[get, set]`
- string? **EmailConfirmation** `[get, set]`
- string? **Password** `[get, set]`
- string? **PasswordConfirmation** `[get, set]`
- string? **PhoneNumber** `[get, set]`
- DateOnly **birthDate** `[get, set]`

5.68.1 Detailed Description

DTO for user registration details, including validations for required fields and data consistency.

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

- Entities/DataTransferObjects/UserForRegistrationDto.cs

5.69 deepdiveapi.Entities.DataTransferObjects.UserForSafeListDto Class Reference

DTO for displaying user details on a safe list, containing non-sensitive information.

Properties

- string **Id** `[get, set]`
- string **FirstName** `[get, set]`
- string **LastName** `[get, set]`
- string **Username** `[get, set]`
- DateOnly **Birthdate** `[get, set]`
- string **PhoneNumber** `[get, set]`
- string **Email** `[get, set]`

5.69.1 Detailed Description

DTO for displaying user details on a safe list, containing non-sensitive information.

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

- Entities/DataTransferObjects/UserForSafeListDto.cs

5.70 deepdiveapi.Entities.Models.UserRegisterDocument Class Reference

Represents a document info registered by a user.

Properties

- string **Id** `[get, set]`

  Gets or sets the unique identifier for the user register document.
- string **DocumentName** `[get, set]`

  Gets or sets the name of the document. Must equal the name of the file in Azure Blob Storage.
- string **UserIdFK** `[get, set]`

  Gets or sets the foreign key for the user who owns this document.
- User **User** `[get, set]`

  Navigation property for the user associated with this document.
- RegistrationDocumentTypes **DocumentType** `[get, set]`

  Gets or sets the type of the document.
- DateTime **CreatedOn** `[get, set]`

  Gets or sets the date and time when the document was created.

5.70.1 Detailed Description

Represents a document info registered by a user.

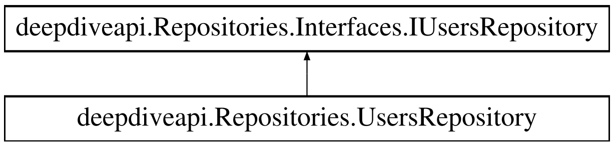
The documentation for this class was generated from the following file:

- Entities/Models/UserRegisterDocument.cs

5.71 deepdiveapi.Repositories.UsersRepository Class Reference

The UsersRepository class is responsible for managing user-related data operations, including CRUD operations and specialized queries.

Inheritance diagram for deepdiveapi.Repositories.UsersRepository:


deepdiveapi.Repositories.Interfaces.IUsersRepository
deepdiveapi.Repositories.UsersRepository

Public Member Functions

- UsersRepository (UserManager< User > userManager, ApplicationDbContext context)

  Constructor for UsersRepository.
- async Task< List< User > > GetUsers ()

  Retrieves a list of all users.
- async Task< List< UserForSafeListDto >?> GetSafeUsers ()

  Retrieves a list of users with safe-to-display data.
- async Task< UserForSafeListDto?> GetSafeUserFromID (string id)

  Retrieves safe-to-display user data for a specific user based on ID.
- async Task< List< UsersTypeahead > > SearchUsersAsync (string searchString)

  Searches for users by a given search string which can match several user attributes.
- async Task DisableAccount (string userId)

  Disables a user account by marking it as deleted.
- async Task EnableAccount (string email)

  Enables a user account by unmarking it as deleted.
- async Task DeleteAccount (string userId)

  Deletes a user account by removing it from the database.

5.71.1 Detailed Description

The UsersRepository class is responsible for managing user-related data operations, including CRUD operations and specialized queries.

5.71.2 Constructor & Destructor Documentation

5.71.2.1 UsersRepository()

```
deepdiveapi.Repositories.UsersRepository.UsersRepository (
            UserManager< User > userManager,
            ApplicationDbContext context )  [inline]
```

Constructor for UsersRepository.

Parameters

userManager	Manages user data from a database.
context	Application database context.

5.71.3 Member Function Documentation

5.71.3.1 DeleteAccount()

```
async Task deepdiveapi.Repositories.UsersRepository.DeleteAccount (
             string userId )  [inline]
```

Deletes a user account by removing it from the database.

Parameters

user↩ Id	The ID of the user whose account is to be deleted.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.2 DisableAccount()

```
async Task deepdiveapi.Repositories.UsersRepository.DisableAccount (
             string userId )  [inline]
```

Disables a user account by marking it as deleted.

Parameters

user↩ Id	The ID of the user whose account is to be disabled.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.3 EnableAccount()

```
async Task deepdiveapi.Repositories.UsersRepository.EnableAccount (
             string email )  [inline]
```

Enables a user account by unmarking it as deleted.

Parameters

email	The email address of the user whose account is to be enabled.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.4 GetSafeUserFromID()

```
async Task< UserForSafeListDto?> deepdiveapi.Repositories.UsersRepository.GetSafeUserFromID (
            string id )  [inline]
```

Retrieves safe-to-display user data for a specific user based on ID.

Parameters

id	The user ID to search for.

Returns

A task that represents the asynchronous operation. The task result contains a UserForSafeListDto entity, or null if the user is not found.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.5 GetSafeUsers()

```
async Task< List< UserForSafeListDto >?> deepdiveapi.Repositories.UsersRepository.GetSafe←
Users ( )  [inline]
```

Retrieves a list of users with safe-to-display data.

Returns

A task that represents the asynchronous operation. The task result contains a list of UserForSafeListDto entities, or null if no users are found.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.6 GetUsers()

```
async Task< List< User > > deepdiveapi.Repositories.UsersRepository.GetUsers ( )  [inline]
```

Retrieves a list of all users.

Returns

A task that represents the asynchronous operation. The task result contains a list of User entities.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

5.71.3.7 SearchUsersAsync()

```
async Task< List< UsersTypeahead > > deepdiveapi.Repositories.UsersRepository.SearchUsers←
Async (
            string searchString )  [inline]
```

Searches for users by a given search string which can match several user attributes.

Parameters

searchString	The string to search for, ignoring case.

Returns

A task that represents the asynchronous operation. The task result contains a list of UsersTypeahead entities.

Implements deepdiveapi.Repositories.Interfaces.IUsersRepository.

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

- Repositories/UserRepository.cs

5.72 deepdiveapi.Entities.DataTransferObjects.UsersTypeahead Class Reference

DTO for providing a typeahead search functionality for users, displaying minimal user details.

Properties

- string **Id** `[get, set]`
- string **FirstName** `[get, set]`
- string **LastName** `[get, set]`
- string **UserName** `[get, set]`

5.72.1 Detailed Description

DTO for providing a typeahead search functionality for users, displaying minimal user details.

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

- Entities/DataTransferObjects/UsersTypeahead.cs

5.73 deepdiveapi.Entities.DataTransferObjects.ValidateEmailAndUserId↩ Response Class Reference

DTO to communicate the confirmation status of a user's email.

Properties

- required bool **EmailHasBeenConfirmed** `[get, set]`

5.73.1 Detailed Description

DTO to communicate the confirmation status of a user's email.

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

- Entities/DataTransferObjects/ValidateEmailAndUserIdResponse.cs

5.74 deepdiveapi.Entities.DataTransferObjects.ValidatePasswordReset Class Reference

DTO for validating a password reset operation using ID and token.

Properties

- required string **Id** `[get, set]`
- required string **Token** `[get, set]`

5.74.1 Detailed Description

DTO for validating a password reset operation using ID and token.

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

- Entities/DataTransferObjects/ValidatePasswordReset.cs

5.75 deepdiveapi.Entities.DataTransferObjects.ViewApplication↩ ResponseDto Class Reference

DTO for viewing application responses with user and application status details.

Properties

- required int **Id** `[get, set]`
- required string **FirstName** `[get, set]`
- required string **LastName** `[get, set]`
- required string **UserName** `[get, set]`
- required string **PhoneNumber** `[get, set]`
- required string **Email** `[get, set]`
- required DateOnly **BirthDate** `[get, set]`
- required RegistrationStatusEnum **Status** `[get, set]`
- string? **AdminComment** `[get, set]`
- required DateTime **CreatedOn** `[get, set]`
- DateTime? **EditedOn** `[get, set]`
- DateTime? **ApprovedOrDeniedOn** `[get, set]`
- List< object > **DocumentsInfo** `[get, set]`

5.75.1 Detailed Description

DTO for viewing application responses with user and application status details.

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

- Entities/DataTransferObjects/ViewApplicationResponseDto.cs

Index

AddExcursion
    deepdiveapi.Controllers.ExcursionController, 29
AddMultipleParticipant
    deepdiveapi.Controllers.ExcursionParticipantsController, 35
AddMultipleParticipantsToExcursion
    deepdiveapi.Repositories.ExcursionParticipantsRepository, 37
    deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository, 45
AddNewExcursion
    deepdiveapi.Repositories.ExcursionRepository, 39
    deepdiveapi.Repositories.Interfaces.IExcursionRepository, 46
AddParticipant
    deepdiveapi.Controllers.ExcursionParticipantsController, 35
AddParticipantToExcursion
    deepdiveapi.Repositories.ExcursionParticipantsRepository, 37
    deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository, 45
AddPasswordRequest
    deepdiveapi.Controllers.PasswordResetController, 61
AddPasswordReset
    deepdiveapi.Repositories.Interfaces.IPasswordResetRepository, 47
    deepdiveapi.Repositories.PasswordResetRepository, 63
AddRegistrationRequest
    deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository, 50
    deepdiveapi.Repositories.RegistrationRequestRepository, 71
AddUserDocumentAsync
    deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository, 48
    deepdiveapi.Repositories.RegistrationDocumentRepository, 67
AdminController, 15
    AdminController, 15
    ViewApplicationsWithRequestedStatus, 16
AdminRepository
    deepdiveapi.Repositories.AdminRepository, 16
ApplicationDbContext
    deepdiveapi.Entities.ApplicationDbContext, 18
Approved
    deepdiveapi.Entities.Enum, 12
AuthController, 19
    AuthController, 19
    Login, 20
    RegisterUser, 20
    ResendValidationEmail, 20
    ValidateEmailWithId, 21
    ValidatePassword, 21
Certificate
    deepdiveapi.Entities.Enum, 12

deepdiveapi, 9
deepdiveapi.Controllers, 9
deepdiveapi.Controllers.ExcursionController, 28
    AddExcursion, 29
    DeleteExcursion, 30
    ExcursionController, 29
    GetExcursionDetails, 30
    GetExcursionsList, 30
    UpdateStatus, 30
deepdiveapi.Controllers.ExcursionParticipantsController, 34
    AddMultipleParticipant, 35
    AddParticipant, 35
    ExcursionParticipantsController, 34
    RemoveParticipant, 35
deepdiveapi.Controllers.HomeController, 41
    GetHomeDashboard, 42
    HomeController, 41
deepdiveapi.Controllers.PasswordResetController, 60
    AddPasswordRequest, 61
    PasswordResetController, 60
    UpdatePassword, 61
    ValidateReset, 61
deepdiveapi.Controllers.RegistrationRequestController, 69
    RegistrationRequestController, 69
    UpdateStatus, 70
deepdiveapi.Controllers.TokenController, 74
    RefreshAsync, 74
    TokenController, 74
deepdiveapi.Entities, 9
deepdiveapi.Entities.ApplicationDbContext, 17
    ApplicationDbContext, 18
    OnModelCreating, 18
deepdiveapi.Entities.DataTransferObjects, 10
deepdiveapi.Entities.DataTransferObjects.AuthResponseDto, 21
deepdiveapi.Entities.DataTransferObjects.CoordinatesDto, 22

deepdiveapi.Entities.DataTransferObjects.CoordinatesDto2, 22
deepdiveapi.Entities.DataTransferObjects.DocuIdInput, 23
deepdiveapi.Entities.DataTransferObjects.EmailAttachmentBinaryBypass, 25
deepdiveapi.Entities.DataTransferObjects.EmailConfirmationUserId, 26
deepdiveapi.Entities.DataTransferObjects.EmailInputDto, 26
deepdiveapi.Entities.DataTransferObjects.EmailModelDto, 26
deepdiveapi.Entities.DataTransferObjects.ErrorResponseDto, 27
deepdiveapi.Entities.DataTransferObjects.ExcursionDetailsDto, 31
deepdiveapi.Entities.DataTransferObjects.ExcursionInfo, 32
deepdiveapi.Entities.DataTransferObjects.ExcursionMultipleParticipantsDto, 32
deepdiveapi.Entities.DataTransferObjects.ExcursionParticipantDto, 33
deepdiveapi.Entities.DataTransferObjects.HomeDashboardDto, 42
deepdiveapi.Entities.DataTransferObjects.IdInputDto, 43
deepdiveapi.Entities.DataTransferObjects.IdOutputDto, 44
deepdiveapi.Entities.DataTransferObjects.NewExcursionDto, 58
deepdiveapi.Entities.DataTransferObjects.ParticipantDto, 58
deepdiveapi.Entities.DataTransferObjects.ParticipationConfirmationInputDto, 59
deepdiveapi.Entities.DataTransferObjects.RegistrationResponseDto, 73
deepdiveapi.Entities.DataTransferObjects.StringInputDto, 73
deepdiveapi.Entities.DataTransferObjects.TokenRefreshDto, 75
deepdiveapi.Entities.DataTransferObjects.UpdateExcursionRequestDto, 77
deepdiveapi.Entities.DataTransferObjects.UpdateRegistrationRequestDto, 78
deepdiveapi.Entities.DataTransferObjects.UpdateStatusRegistrationRequestDto, 78
deepdiveapi.Entities.DataTransferObjects.UpdateUserInputDto, 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserPassword, 79
deepdiveapi.Entities.DataTransferObjects.UpdateUserResponseDto, 80
deepdiveapi.Entities.DataTransferObjects.UploadRegisterDocumentDto, 80
deepdiveapi.Entities.DataTransferObjects.UserDto, 85
deepdiveapi.Entities.DataTransferObjects.UserForAuthenticationDto, 86
deepdiveapi.Entities.DataTransferObjects.UserForRegistrationDto, 86
deepdiveapi.Entities.DataTransferObjects.UserForSafeListDto, 87
deepdiveapi.Entities.DataTransferObjects.UsersTypeahead, 91
deepdiveapi.Entities.DataTransferObjects.ValidateEmailAndUserIdRespon 91
deepdiveapi.Entities.DataTransferObjects.ValidatePasswordReset, 92
deepdiveapi.Entities.DataTransferObjects.ViewApplicationResponseDto, 92
deepdiveapi.Entities.Enum, 11
  Approved, 12
  Certificate, 12
  Denied, 12
  IdPassport, 12
  PasswordResetsStastusEnum, 11
  PwdChanged, 12
  RegistrationDocumentTypes, 12
  RegistrationStatusEnum, 12
  Requested, 12
  WaitingForUserChanges, 12
deepdiveapi.Entities.Models, 12
deepdiveapi.Entities.Models.Excursion, 27
deepdiveapi.Entities.Models.ExcursionParticipant, 32
deepdiveapi.Entities.Models.PasswordReset, 59
deepdiveapi.Entities.Models.RefreshToken, 63
deepdiveapi.Entities.Models.RegistrationRequest, 68
deepdiveapi.Entities.Models.User, 81
deepdiveapi.Entities.Models.UserRegisterDocument, 87
deepdiveapi.Entities.Validation, 13
deepdiveapi.Entities.Validation.LaterThanAttribute, 56
  isValid, 57
  LaterThanAttribute, 57
deepdiveapi.Entities.Verification, 13
deepdiveapi.Entities.Verification.Email, 13
deepdiveapi.Entities.Verification.Email.EmailActivatedHandler, 23
  EmailActivatedHandler, 24
  HandleRequirementAsync, 24
deepdiveapi.Entities.Verification.Email.EmailActivatedRequirement, 25
deepdiveapi.JwtFeatures, 13
deepdiveapi.JwtFeatures.JwtHandler, 53
  GenerateAccessToken, 54
  GenerateRefreshToken, 54
  GenerateTokenOptions, 55
  GetClaims, 55
  GetPrincipalFromExpiredToken, 55
  GetSigningCredentials, 56
  GetUserIdFromToken, 56
  JwtHandler, 54
deepdiveapi.Repositories, 13
deepdiveapi.Repositories.AdminRepository, 16
  AdminRepository, 16
  GetAdminUserEmailsAsync, 17
deepdiveapi.Repositories.ExcursionParticipantsRepository, 36

AddMultipleParticipantsToExcursion, 37
AddParticipantToExcursion, 37
ExcursionParticipantsRepository, 36
RemoveParticipantFromExcursion, 37
deepdiveapi.Repositories.ExcursionRepository, 38
AddNewExcursion, 39
DeleteExcursion, 39
ExcursionRepository, 38
GetExcursionById, 39
GetExcursions, 40
GetUserExcursions, 40
UpdateExcursionAsync, 40
deepdiveapi.Repositories.Interfaces, 14
deepdiveapi.Repositories.Interfaces.IAdminRepository, 43
GetAdminUserEmailsAsync, 43
deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository, 44
AddMultipleParticipantsToExcursion, 45
AddParticipantToExcursion, 45
RemoveParticipantFromExcursion, 45
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 45
AddNewExcursion, 46
DeleteExcursion, 46
GetExcursionById, 46
GetExcursions, 46
GetUserExcursions, 46
UpdateExcursionAsync, 47
deepdiveapi.Repositories.Interfaces.IPasswordResetRepository, 47
AddPasswordReset, 47
FindByEmailAndToken, 47
deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository, 48
AddUserDocumentAsync, 48
DeleteDocumentAsync, 48
GetDocumentsOfUser, 49
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository, 49
AddRegistrationRequest, 50
GetRegistrationRequestById, 50
GetRegistrationRequestByUserIdAsync, 50
UpdateRegistrationRequestAsync, 50
deepdiveapi.Repositories.Interfaces.ITokenRepository, 50
InvalidateOldRefreshToken, 51
IsValidRefreshTokenAsync, 51
StoreRefreshTokenAsync, 51
deepdiveapi.Repositories.Interfaces.IUsersRepository, 52
DeleteAccount, 52
DisableAccount, 52
EnableAccount, 52
GetSafeUserFromID, 52
GetSafeUsers, 53
GetUsers, 53
SearchUsersAsync, 53
deepdiveapi.Repositories.PasswordResetRepository, 62
AddPasswordReset, 63
FindByEmailAndToken, 63
PasswordResetRepository, 62
deepdiveapi.Repositories.RegistrationDocumentRepository, 66
AddUserDocumentAsync, 67
DeleteDocumentAsync, 67
GetDocumentsOfUser, 67
RegistrationDocumentRepository, 66
deepdiveapi.Repositories.RegistrationRequestRepository, 70
AddRegistrationRequest, 71
GetRegistrationRequestById, 71
GetRegistrationRequestByUserIdAsync, 72
RegistrationRequestRepository, 71
UpdateRegistrationRequestAsync, 72
deepdiveapi.Repositories.TokenRepository, 75
InvalidateOldRefreshToken, 76
IsValidRefreshTokenAsync, 76
StoreRefreshTokenAsync, 77
TokenRepository, 76
deepdiveapi.Repositories.UsersRepository, 88
DeleteAccount, 89
DisableAccount, 89
EnableAccount, 89
GetSafeUserFromID, 89
GetSafeUsers, 90
GetUsers, 90
SearchUsersAsync, 90
UsersRepository, 88
DeleteAccount
deepdiveapi.Repositories.Interfaces.IUsersRepository, 52
deepdiveapi.Repositories.UsersRepository, 89
UserController, 83
DeleteDocument
RegisterDocumentsController, 65
DeleteDocumentAsync
deepdiveapi.Repositories.Interfaces.IRegistrationDocumentReposito
48
deepdiveapi.Repositories.RegistrationDocumentRepository, 67
DeleteExcursion
deepdiveapi.Controllers.ExcursionController, 30
deepdiveapi.Repositories.ExcursionRepository, 39
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 46
Denied
deepdiveapi.Entities.Enum, 12
DisableAccount
deepdiveapi.Repositories.Interfaces.IUsersRepository, 52
deepdiveapi.Repositories.UsersRepository, 89
UserController, 83

EmailActivatedHandler

deepdiveapi.Entities.Verification.Email.EmailActivatedHandler, 24
EnableAccount
deepdiveapi.Repositories.Interfaces.IUsersRepository, 52
deepdiveapi.Repositories.UsersRepository, 89
UserController, 83
ExcursionController
deepdiveapi.Controllers.ExcursionController, 29
ExcursionParticipantsController
deepdiveapi.Controllers.ExcursionParticipantsController, 34
ExcursionParticipantsRepository
deepdiveapi.Repositories.ExcursionParticipantsRepository, 36
ExcursionRepository
deepdiveapi.Repositories.ExcursionRepository, 38

FindByEmailAndToken
deepdiveapi.Repositories.Interfaces.IPasswordResetRepository, 47
deepdiveapi.Repositories.PasswordResetRepository, 63

GenerateAccessToken
deepdiveapi.JwtFeatures.JwtHandler, 54
GenerateRefreshToken
deepdiveapi.JwtFeatures.JwtHandler, 54
GenerateTokenOptions
deepdiveapi.JwtFeatures.JwtHandler, 55
GetAdminUserEmailsAsync
deepdiveapi.Repositories.AdminRepository, 17
deepdiveapi.Repositories.Interfaces.IAdminRepository, 43
GetClaims
deepdiveapi.JwtFeatures.JwtHandler, 55
GetDocumentsOfUser
deepdiveapi.Repositories.Interfaces.IRegistrationDocumentRepository, 49
deepdiveapi.Repositories.RegistrationDocumentRepository, 67
GetExcursionById
deepdiveapi.Repositories.ExcursionRepository, 39
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 46
GetExcursionDetails
deepdiveapi.Controllers.ExcursionController, 30
GetExcursions
deepdiveapi.Repositories.ExcursionRepository, 40
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 46
GetExcursionsList
deepdiveapi.Controllers.ExcursionController, 30
GetHomeDashboard
deepdiveapi.Controllers.HomeController, 42
GetPrincipalFromExpiredToken
deepdiveapi.JwtFeatures.JwtHandler, 55
GetRegistrationRequestById
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository, 50
deepdiveapi.Repositories.RegistrationRequestRepository, 71
GetRegistrationRequestByUserIdAsync
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository, 50
deepdiveapi.Repositories.RegistrationRequestRepository, 72
GetSafeUserFromID
deepdiveapi.Repositories.Interfaces.IUsersRepository, 52
deepdiveapi.Repositories.UsersRepository, 89
GetSafeUsers
deepdiveapi.Repositories.Interfaces.IUsersRepository, 53
deepdiveapi.Repositories.UsersRepository, 90
UserController, 83, 84
GetSigningCredentials
deepdiveapi.JwtFeatures.JwtHandler, 56
GetUserExcursions
deepdiveapi.Repositories.ExcursionRepository, 40
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 46
GetUserIdFromToken
deepdiveapi.JwtFeatures.JwtHandler, 56
GetUsers
deepdiveapi.Repositories.Interfaces.IUsersRepository, 53
deepdiveapi.Repositories.UsersRepository, 90
UserController, 84
GetUsersSearchTypehead
UserController, 84

HandleRequirementAsync
deepdiveapi.Entities.Verification.Email.EmailActivatedHandler, 24
HomeController
deepdiveapi.Controllers.HomeController, 41

IdPassport
deepdiveapi.Entities.Enum, 12
InvalidateOldRefreshToken
deepdiveapi.Repositories.Interfaces.ITokenRepository, 51
deepdiveapi.Repositories.TokenRepository, 76
IsValid
deepdiveapi.Entities.Validation.LaterThanAttribute, 57
IsValidRefreshTokenAsync
deepdiveapi.Repositories.Interfaces.ITokenRepository, 51
deepdiveapi.Repositories.TokenRepository, 76

JwtHandler
deepdiveapi.JwtFeatures.JwtHandler, 54

LaterThanAttribute

deepdiveapi.Entities.Validation.LaterThanAttribute, 57
Login
AuthController, 20
OnModelCreating
deepdiveapi.Entities.ApplicationDbContext, 18
PasswordResetController
deepdiveapi.Controllers.PasswordResetController, 60
PasswordResetRepository
deepdiveapi.Repositories.PasswordResetRepository, 62
PasswordResetsStastusEnum
deepdiveapi.Entities.Enum, 11
PwdChanged
deepdiveapi.Entities.Enum, 12
RefreshAsync
deepdiveapi.Controllers.TokenController, 74
RegisterDocumentsController, 64
DeleteDocument, 65
RegisterDocumentsController, 65
UploadDocument, 65
RegisterUser
AuthController, 20
RegistrationDocumentRepository
deepdiveapi.Repositories.RegistrationDocumentRepository, 66
RegistrationDocumentTypes
deepdiveapi.Entities.Enum, 12
RegistrationRequestController
deepdiveapi.Controllers.RegistrationRequestController, 69
RegistrationRequestRepository
deepdiveapi.Repositories.RegistrationRequestRepository, 71
RegistrationStatusEnum
deepdiveapi.Entities.Enum, 12
RemoveParticipant
deepdiveapi.Controllers.ExcursionParticipantsController, 35
RemoveParticipantFromExcursion
deepdiveapi.Repositories.ExcursionParticipantsRepository, 37
deepdiveapi.Repositories.Interfaces.IExcursionParticipantsRepository, 45
Requested
deepdiveapi.Entities.Enum, 12
ResendValidationEmail
AuthController, 20
SearchUsersAsync
deepdiveapi.Repositories.Interfaces.IUsersRepository, 53
deepdiveapi.Repositories.UsersRepository, 90
StoreRefreshTokenAsync
deepdiveapi.Repositories.Interfaces.ITokenRepository, 51
deepdiveapi.Repositories.TokenRepository, 77
TokenController
deepdiveapi.Controllers.TokenController, 74
TokenRepository
deepdiveapi.Repositories.TokenRepository, 76
Update
UserController, 85
UpdateExcursionAsync
deepdiveapi.Repositories.ExcursionRepository, 40
deepdiveapi.Repositories.Interfaces.IExcursionRepository, 47
UpdatePassword
deepdiveapi.Controllers.PasswordResetController, 61
UpdateRegistrationRequestAsync
deepdiveapi.Repositories.Interfaces.IRegistrationRequestRepository, 50
deepdiveapi.Repositories.RegistrationRequestRepository, 72
UpdateStatus
deepdiveapi.Controllers.ExcursionController, 30
deepdiveapi.Controllers.RegistrationRequestController, 70
UploadDocument
RegisterDocumentsController, 65
UserController, 81
DeleteAccount, 83
DisableAccount, 83
EnableAccount, 83
GetSafeUsers, 83, 84
GetUsers, 84
GetUsersSearchTypehead, 84
Update, 85
UserController, 82
ViewApplicationRegistration, 85
UsersRepository
deepdiveapi.Repositories.UsersRepository, 88
ValidateEmailWithId
AuthController, 21
ValidatePassword
AuthController, 21
ValidateReset
deepdiveapi.Controllers.PasswordResetController, 61
ViewApplicationRegistration
UserController, 85
ViewApplicationsWithRequestedStatus
AdminController, 16
WaitingForUserChanges
deepdiveapi.Entities.Enum, 12