

Augmented Education Website Code

Contents

1	Namespace Index	1
1.1	Packages	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	ARFE Namespace Reference	9
4.1.1	Detailed Description	10
4.2	ARFE.Controllers Namespace Reference	10
4.2.1	Detailed Description	11
4.3	ARFE.Models Namespace Reference	11
4.3.1	Detailed Description	12
4.4	ARFE.Tests Namespace Reference	12
4.4.1	Detailed Description	12
4.5	ARFE.Tests.Controllers Namespace Reference	12
4.5.1	Detailed Description	12
4.6	AuthenticationTokenCache Namespace Reference	13
4.6.1	Detailed Description	13
4.7	AuthenticationTokenCache.Tests Namespace Reference	13
4.7.1	Detailed Description	13
4.8	Common Namespace Reference	13
4.8.1	Detailed Description	13

5	Class Documentation	15
5.1	ARFE.Controllers.AccountController Class Reference	15
5.1.1	Detailed Description	16
5.1.2	Constructor & Destructor Documentation	16
5.1.2.1	AccountController() [1/2]	17
5.1.2.2	AccountController() [2/2]	17
5.1.3	Member Function Documentation	17
5.1.3.1	ConfirmEmail()	17
5.1.3.2	Dispose()	17
5.1.3.3	ForgotPassword() [1/2]	18
5.1.3.4	ForgotPassword() [2/2]	18
5.1.3.5	ForgotPasswordConfirmation()	18
5.1.3.6	Login() [1/2]	19
5.1.3.7	Login() [2/2]	19
5.1.3.8	LogOff()	19
5.1.3.9	Register() [1/2]	20
5.1.3.10	Register() [2/2]	20
5.1.3.11	ResetPassword() [1/2]	20
5.1.3.12	ResetPassword() [2/2]	21
5.1.3.13	ResetPasswordConfirmation()	21
5.1.3.14	SendCode() [1/2]	21
5.1.3.15	SendCode() [2/2]	22
5.1.3.16	VerifyCode() [1/2]	22
5.1.3.17	VerifyCode() [2/2]	22
5.1.4	Member Data Documentation	23
5.1.4.1	SignInManager	23
5.1.4.2	UserManager	23
5.2	ARFE.Models.AddPhoneNumberViewModel Class Reference	23
5.2.1	Detailed Description	23
5.2.2	Property Documentation	24

5.2.2.1	Number	24
5.3	ARFE.Models.ApplicationDbContext Class Reference	24
5.3.1	Detailed Description	24
5.3.2	Constructor & Destructor Documentation	25
5.3.2.1	ApplicationDbContext()	25
5.3.3	Member Function Documentation	25
5.3.3.1	Create()	25
5.4	ARFE.ApplicationSignInManager Class Reference	25
5.4.1	Detailed Description	26
5.4.2	Constructor & Destructor Documentation	26
5.4.2.1	ApplicationSignInManager()	26
5.4.3	Member Function Documentation	26
5.4.3.1	Create()	26
5.4.3.2	CreateUserIdentityAsync()	27
5.5	ARFE.Models.ApplicationUser Class Reference	27
5.5.1	Detailed Description	27
5.5.2	Member Function Documentation	28
5.5.2.1	GenerateUserIdentityAsync()	28
5.6	ARFE.ApplicationUserManager Class Reference	28
5.6.1	Detailed Description	29
5.6.2	Constructor & Destructor Documentation	29
5.6.2.1	ApplicationUserManager()	29
5.6.3	Member Function Documentation	29
5.6.3.1	Create()	29
5.7	ARFE.BlobManager Class Reference	30
5.7.1	Detailed Description	31
5.7.2	Constructor & Destructor Documentation	31
5.7.2.1	BlobManager()	31
5.7.3	Member Function Documentation	31
5.7.3.1	ConvertAndDownloadBlobFromUserContainer()	31

5.7.3.2	DeleteBlobByNameInUserContainer()	31
5.7.3.3	FormatBlobContainerName()	32
5.7.3.4	GetOrCreateBlobContainer()	32
5.7.3.5	ListBlobNamesToUrisInPublicContainer()	33
5.7.3.6	ListBlobNamesToUrisInPublicContainerOwnedBy()	33
5.7.3.7	ListBlobNamesToUrisInUserContainer()	33
5.7.3.8	ListBlobUrisInPublicContainerOwnedBy()	34
5.7.3.9	ListBlobUrisInUserContainer()	34
5.7.3.10	ListPrivateBlobInfoForUI()	34
5.7.3.11	ListPublicBlobInfoForUI()	35
5.7.3.12	RemoveExpiredTemporaryBlobs()	35
5.7.3.13	UploadBlobToPublicContainer()	35
5.7.3.14	UploadBlobToUserContainer()	36
5.8	ARFE.Tests.BlobTests Class Reference	36
5.8.1	Detailed Description	37
5.8.2	Member Function Documentation	37
5.8.2.1	Test_Constructor()	37
5.8.2.2	Test_FormatBlobContainerName()	37
5.9	ARFE.BundleConfig Class Reference	37
5.9.1	Detailed Description	37
5.9.2	Member Function Documentation	38
5.9.2.1	RegisterBundles()	38
5.10	ARFE.Models.ChangePasswordViewModel Class Reference	38
5.10.1	Detailed Description	38
5.10.2	Property Documentation	38
5.10.2.1	ConfirmPassword	39
5.10.2.2	NewPassword	39
5.10.2.3	OldPassword	39
5.11	ARFE.Models.ConfigureTwoFactorViewModel Class Reference	39
5.11.1	Detailed Description	39

5.11.2	Property Documentation	40
5.11.2.1	Providers	40
5.11.2.2	SelectedProvider	40
5.12	ARFE.EmailService Class Reference	40
5.12.1	Detailed Description	41
5.12.2	Member Function Documentation	41
5.12.2.1	SendAsync()	41
5.13	ARFE.FileConverter Class Reference	41
5.13.1	Detailed Description	42
5.13.2	Constructor & Destructor Documentation	42
5.13.2.1	FileConverter()	42
5.13.3	Member Function Documentation	43
5.13.3.1	ConvertToDAE()	43
5.13.3.2	ConvertToFBX()	43
5.13.3.3	ConvertToOBJ()	43
5.13.3.4	ConvertToPLY()	44
5.13.3.5	ConvertToSTL()	44
5.13.4	Member Data Documentation	45
5.13.4.1	BasePath	45
5.13.4.2	InputFolder	45
5.13.4.3	OutputFolder	45
5.14	FileTypeModel Class Reference	45
5.15	ARFE.Controllers.FileTypeModel Class Reference	45
5.15.1	Detailed Description	46
5.15.2	Property Documentation	46
5.15.2.1	FileType	46
5.15.2.2	FileTypes	46
5.16	Common.FileUIInfo Class Reference	46
5.16.1	Detailed Description	47
5.16.2	Constructor & Destructor Documentation	47

5.16.2.1	FileUIInfo()	47
5.16.3	Property Documentation	47
5.16.3.1	Author	47
5.16.3.2	Description	48
5.16.3.3	FileName	48
5.16.3.4	UploadDate	48
5.17	ARFE.FilterConfig Class Reference	48
5.17.1	Detailed Description	48
5.17.2	Member Function Documentation	48
5.17.2.1	RegisterGlobalFilters()	48
5.18	ARFE.Models.ForgotPasswordViewModel Class Reference	49
5.18.1	Detailed Description	49
5.18.2	Property Documentation	49
5.18.2.1	Email	49
5.19	ARFE.Controllers.HelpController Class Reference	49
5.19.1	Detailed Description	50
5.19.2	Member Function Documentation	50
5.19.2.1	Help()	50
5.20	ARFE.Tests.Controllers.HomeControllerTest Class Reference	50
5.21	ARFE.Models.IndexViewModel Class Reference	50
5.21.1	Detailed Description	51
5.21.2	Property Documentation	51
5.21.2.1	BrowserRemembered	51
5.21.2.2	HasPassword	51
5.21.2.3	Logins	51
5.21.2.4	PhoneNumber	52
5.21.2.5	TwoFactor	52
5.22	ARFE.Models.LoginViewModel Class Reference	52
5.22.1	Detailed Description	52
5.22.2	Property Documentation	52

5.22.2.1	Email	52
5.22.2.2	Password	53
5.22.2.3	RememberMe	53
5.23	ARFE.Controllers.ManageController Class Reference	53
5.23.1	Detailed Description	54
5.23.2	Constructor & Destructor Documentation	54
5.23.2.1	ManageController() [1/2]	55
5.23.2.2	ManageController() [2/2]	55
5.23.3	Member Data Documentation	55
5.23.3.1	SignInManager	55
5.23.3.2	UserManager	55
5.24	ARFE.Models.ManageLoginsViewModel Class Reference	56
5.24.1	Detailed Description	56
5.24.2	Property Documentation	56
5.24.2.1	CurrentLogins	56
5.24.2.2	OtherLogins	56
5.25	ARFE.Controllers.MobileAuthController Class Reference	56
5.25.1	Detailed Description	57
5.25.2	Constructor & Destructor Documentation	57
5.25.2.1	MobileAuthController()	57
5.25.3	Member Function Documentation	57
5.25.3.1	DownloadFile()	58
5.25.3.2	ListFiles()	58
5.25.3.3	RequestAuthToken()	58
5.25.4	Member Data Documentation	59
5.25.4.1	UserManager	59
5.26	ARFE.MvcApplication Class Reference	59
5.27	ARFE.Controllers.PublicContentController Class Reference	59
5.27.1	Detailed Description	60
5.27.2	Member Function Documentation	60

5.27.2.1	ContentSelect()	60
5.27.2.2	Index()	60
5.27.2.3	Search()	61
5.28	ARFE.Models.RegisterViewModel Class Reference	61
5.28.1	Detailed Description	62
5.28.2	Property Documentation	62
5.28.2.1	ConfirmPassword	62
5.28.2.2	Email	62
5.28.2.3	Password	62
5.29	ARFE.Models.ResetPasswordViewModel Class Reference	62
5.29.1	Detailed Description	63
5.29.2	Property Documentation	63
5.29.2.1	Code	63
5.29.2.2	ConfirmPassword	63
5.29.2.3	Email	63
5.29.2.4	Password	64
5.30	ARFE.RouteConfig Class Reference	64
5.30.1	Detailed Description	64
5.30.2	Member Function Documentation	64
5.30.2.1	RegisterRoutes()	64
5.31	ARFE.SearchQueries Class Reference	65
5.31.1	Detailed Description	65
5.31.2	Member Function Documentation	65
5.31.2.1	FilterByFileNameOrderByDateAscending()	65
5.31.2.2	FilterByFileNameOrderByDateDescending()	66
5.31.2.3	FilterByFileNameOrderByNameAscending()	66
5.31.2.4	FilterByFileNameOrderByNameDescending()	67
5.31.2.5	FilterDateOrderByDateAscending()	67
5.31.2.6	FilterDateOrderByDateDescending()	67
5.31.2.7	FilterDateOrderByNameAscending()	68

5.31.2.8	FilterDateOrderByNameDescending()	68
5.32	ARFE.Models.SendCodeViewModel Class Reference	69
5.32.1	Detailed Description	69
5.32.2	Property Documentation	69
5.32.2.1	Providers	69
5.32.2.2	RememberMe	70
5.32.2.3	ReturnUrl	70
5.32.2.4	SelectedProvider	70
5.33	ARFE.Models.SetPasswordViewModel Class Reference	70
5.33.1	Detailed Description	70
5.33.2	Property Documentation	70
5.33.2.1	ConfirmPassword	71
5.33.2.2	NewPassword	71
5.34	ARFE.SmsService Class Reference	71
5.34.1	Detailed Description	71
5.34.2	Member Function Documentation	71
5.34.2.1	SendAsync()	71
5.35	ARFE.Startup Class Reference	72
5.35.1	Detailed Description	72
5.35.2	Member Function Documentation	72
5.35.2.1	Configuration()	72
5.35.2.2	ConfigureAuth()	73
5.35.2.3	TimerInit()	73
5.36	Common.SupportedFileTypes Class Reference	73
5.36.1	Detailed Description	73
5.36.2	Member Data Documentation	73
5.36.2.1	FileList	74
5.37	AuthenticationTokenCache.TokenCache Class Reference	74
5.37.1	Detailed Description	74
5.37.2	Member Function Documentation	74

5.37.2.1	GenerateToken()	74
5.37.2.2	Init()	75
5.37.2.3	ValidateToken()	75
5.38	AuthenticationTokenCache.Tests.TokenCacheTests Class Reference	76
5.38.1	Detailed Description	76
5.38.2	Member Function Documentation	76
5.38.2.1	Test_BadTokenReturnsNull()	76
5.38.2.2	Test_Init()	76
5.38.2.3	Test_Insert()	77
5.38.2.4	Test_MultipleTokensToSameUser()	77
5.38.2.5	Test_PrivateConstructor()	77
5.38.2.6	Test_Singleton()	77
5.38.2.7	Test_ValidateMultiple()	77
5.38.2.8	Test_ValidateSingle()	77
5.39	ARFE.Controllers.UploadController Class Reference	78
5.39.1	Detailed Description	78
5.39.2	Member Function Documentation	78
5.39.2.1	Index()	79
5.39.2.2	OverWriteUpload()	79
5.39.2.3	UploadFile() [1/2]	79
5.39.2.4	UploadFile() [2/2]	80
5.40	ARFE.UploadedFileCache Class Reference	80
5.40.1	Detailed Description	81
5.40.2	Member Function Documentation	81
5.40.2.1	DeleteAndRemoveFile()	82
5.40.2.2	DeleteOldFiles()	82
5.40.2.3	FindContainingFolderGUIDForFile()	82
5.40.2.4	GetFileDescription()	83
5.40.2.5	GetInstance()	83
5.40.2.6	IsSavedFilePublic()	83

5.40.2.7	MarkForDelete()	84
5.40.2.8	RemoveFileTracking()	84
5.40.2.9	SaveFile() [1/2]	84
5.40.2.10	SaveFile() [2/2]	85
5.40.3	Property Documentation	86
5.40.3.1	BasePath	86
5.41	ARFE.Tests.UploadedFileCacheTests Class Reference	86
5.41.1	Detailed Description	86
5.41.2	Member Function Documentation	86
5.41.2.1	Test_Init()	87
5.41.2.2	Test_PrivateConstructor()	87
5.42	ARFE.Controllers.UserContentController Class Reference	87
5.42.1	Detailed Description	88
5.42.2	Member Function Documentation	88
5.42.2.1	Index()	88
5.42.2.2	PrivateContentSelect()	88
5.42.2.3	PrivateSearch()	89
5.42.2.4	PublicContentSelect()	89
5.42.2.5	PublicSearch()	90
5.43	ARFE.Models.VerifyCodeViewModel Class Reference	90
5.43.1	Detailed Description	91
5.43.2	Property Documentation	91
5.43.2.1	Code	91
5.43.2.2	Provider	91
5.43.2.3	RememberBrowser	91
5.43.2.4	RememberMe	91
5.43.2.5	ReturnUrl	91
5.44	ARFE.Models.VerifyPhoneNumberViewModel Class Reference	92
5.44.1	Detailed Description	92
5.44.2	Property Documentation	92
5.44.2.1	Code	92
5.44.2.2	PhoneNumber	92

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

ARFE	This is the over-arching namespace for all website related code.	9
ARFE.Controllers	This namespaces is a sub-namespace of the ARFE project namespace specifically for the AS↔P.NET Controllers	10
ARFE.Models	The model classes that contain data representation objects for passing information between the web .cshtml code and the backend server code.	11
ARFE.Tests	The Main namespace containing unit testing code for the ARFE Web project	12
ARFE.Tests.Controllers	The sub-namespace of the main namespace containing unit testing code for the ARFE Web project. This namespace is for testing the ARFE.Controllers code specifically.	12
AuthenticationTokenCache	The namespace containing all code relating to the in memory cache used for mobile authentication.	13
AuthenticationTokenCache.Tests	The namespace containing all tests that pertain to the AuthenticationTokenCache.TokenCache class and it's methods.	13
Common	This namespace is for containing all custom objects and methods that should be made publicly available and easily accessible for all other projects within the appliaction.	13

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

ARFE.Models.AddPhoneNumberViewModel	23
ARFE.BlobManager	30
ARFE.Tests.BlobTests	36
ARFE.BundleConfig	37
ARFE.Models.ChangePasswordViewModel	38
ARFE.Models.ConfigureTwoFactorViewModel	39
Controller	
ARFE.Controllers.AccountController	15
ARFE.Controllers.HelpController	49
ARFE.Controllers.ManageController	53
ARFE.Controllers.MobileAuthController	56
ARFE.Controllers.PublicContentController	59
ARFE.Controllers.UploadController	78
ARFE.Controllers.UserContentController	87
ARFE.FileConverter	41
FileTypeModel	45
ARFE.Controllers.FileTypeModel	45
Common.FileUIInfo	46
ARFE.FilterConfig	48
ARFE.Models.ForgotPasswordViewModel	49
ARFE.Tests.Controllers.HomeControllerTest	50
HttpApplication	
ARFE.MvcApplication	59
IdentityDbContext	
ARFE.Models.ApplicationDbContext	24
IdentityUser	
ARFE.Models.ApplicationUser	27
IMessageService	
ARFE.EmailService	40
ARFE.SmsService	71
ARFE.Models.IndexViewModel	50
ARFE.Models.LoginViewModel	52
ARFE.Models.ManageLoginsViewModel	56
ARFE.Models.RegisterViewModel	61
ARFE.Models.ResetPasswordViewModel	62

ARFE.RouteConfig	64
ARFE.SearchQueries	65
ARFE.Models.SendCodeViewModel	69
ARFE.Models.SetPasswordViewModel	70
SignInManager	
ARFE.ApplicationSignInManager	25
ARFE.Startup	72
Common.SupportedFileTypes	73
AuthenticationTokenCache.TokenCache	74
AuthenticationTokenCache.Tests.TokenCacheTests	76
ARFE.UploadedFileCache	80
ARFE.Tests.UploadedFileCacheTests	86
UserManager	
ARFE.ApplicationUserManager	28
ARFE.Models.VerifyCodeViewModel	90
ARFE.Models.VerifyPhoneNumberViewModel	92

Chapter 3

Class Index

3.1 Class List

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

ARFE.Controllers.AccountController	A class derived from the Controller class that has all of the controller actions to manage user accounts.	15
ARFE.Models.AddPhoneNumberViewModel	A model for registering a phone number to a user account.	23
ARFE.Models.ApplicationDbContext	This class inherits from and can override the Identity default Db that is automatically generated from the ApplicationUser class. If additionaly database fields are needed for the application database tables, they can be added here.	24
ARFE.ApplicationSignInManager	Creates a useful interface to the Microsoft.AspNet.Identity.UserManager<TUser> class for easily verifying user accounts with account userName/password verification handling.	25
ARFE.Models.ApplicationUser	This class holds the profile information for a user within the appliation. Profile data for the user can be added by adding more properties to the ApplicationUser class, please visit https://go.microsoft.com/fwlink/?LinkId=317594 to learn more.	27
ARFE.ApplicationUserManager	Creates a useful interface to the Microsoft.AspNet.Identity.UserManager<TUser> class for easily registering new user accounts with account parameter verification handling.	28
ARFE.BlobManager	This class is for managing all transactions between the AE website and Azure blob storage. All user containers, user blob creation/deletion, blob metadata, blob permissions, and blob listings are handled here.	30
ARFE.Tests.BlobTests	Unit tests covering the non-Azure-facing functionality of the BlobManager.cs file in ARFE.csproj	36
ARFE.BundleConfig	Configure all of the javascript and css include bundles. Register these with the entire application for simple inclusion in the .cshtml files.	37
ARFE.Models.ChangePasswordViewModel	A model to change a user's registered account password.	38
ARFE.Models.ConfigureTwoFactorViewModel	A model for registering third party two-factor authentication providers. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.	39

ARFE.EmailService	
A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via email. This implementation is currently empty as no third party email verification is currently supported.	40
ARFE.FileConverter	
The FileConverter class serves as an interface between the ASP.NET controllers and other C# classes to FileConversion.exe. This is used throughout the application for managing file compatibility and efficiency in both file storage and transfer.	41
FileTypeModel	45
ARFE.Controllers.FileTypeModel	
A POCO class for the file type combo box	45
Common.FileUIInfo	
A class that contains file representation information for displaying information about a user's files to the web interface.	46
ARFE.FilterConfig	
A class for defining filtered attributes to the ASP.NET web request and response communication. For example, the [RequireHttps] filter could be applied here to demand all traffic is only allowed if it is transferred over SSL.	48
ARFE.Models.ForgotPasswordViewModel	
A model for the user requesting a password reset via their registered email address.	49
ARFE.Controllers.HelpController	
A class derived from the Controller class that has all of the controller actions to display the website's help pages.	49
ARFE.Tests.Controllers.HomeControllerTest	50
ARFE.Models.IndexViewModel	
A model for returning user login and verification.	50
ARFE.Models.LoginViewModel	
A model for the user to log in to the website.	52
ARFE.Controllers.ManageController	
A class derived from the Controller class that has all of the controller actions to manage user registration and authentication.	
53	
ARFE.Models.ManageLoginsViewModel	
A model for viewing and reporting current and past user login information.	56
ARFE.Controllers.MobileAuthController	
The MobileAuthController serves as an API to the mobile application. The [Route] attribute allows this Controller to specify that to call any individual controller action within this class, you have to explicitly list the action method in the request URL.	56
ARFE.MvcApplication	59
ARFE.Controllers.PublicContentController	
A class derived from the Controller class that has all of the controller actions to display and interact with the public file content on the website.	59
ARFE.Models.RegisterViewModel	
A model for registering a new user.	61
ARFE.Models.ResetPasswordViewModel	
A model to reset a user's password.	62
ARFE.RouteConfig	
This class is for defining the valid url rout patterns for the application.	64
ARFE.SearchQueries	
This class contains all of the linq queries for the search and order functionality.	65
ARFE.Models.SendCodeViewModel	
A model for sending a two-factor authentication message through a registered third party two-factor authentication provider.	69
ARFE.Models.SetPasswordViewModel	
A model for registering a user's password upon account creation.	70

ARFE.SmsService	
A class derived from the <code>Microsoft.AspNet.Identity.IIdentityMessageService</code> class to implement third party account verification via SMS text messaging. This implementation is currently empty as no third party SMS text messaging verification is currently supported.	71
ARFE.Startup	
The application startup class is called upon launching the application.	72
Common.SupportedFileTypes	
The list of file type extensions that are supported by the file conversion software that this application uses.	73
AuthenticationTokenCache.TokenCache	
A class to manage tokens that are associated to user login information. This is used for mobile user authentication by verifying tokens, that the user explicitly requests, to that user's log in credentials. This system of token passing makes for simpler and more secure communication between the mobile application and the web service.	74
AuthenticationTokenCache.Tests.TokenCacheTests	
The class containing all tests that pertain to the AuthenticationTokenCache.TokenCache class and it's methods.	76
ARFE.Controllers.UploadController	
A class derived from the Controller class that has all of the controller actions to manage user file uploads.	78
ARFE.UploadedFileCache	
A singleton class, constructed via the GetInstance method call. This class serves as a cache for tracking the location, details about, and automatic removal of files that are saved for intermediate storage between being uploaded to Azure blob storage, or converted to some other file type pending a download.	80
ARFE.Tests.UploadedFileCacheTests	
Summary description for UploadedFileCacheTests	86
ARFE.Controllers.UserContentController	
A class derived from the Controller class that has all of the controller actions to display and interact with the user's file content on the website.	87
ARFE.Models.VerifyCodeViewModel	
A model for verifying a two-factor authentication code that has been sent to a user.	90
ARFE.Models.VerifyPhoneNumberViewModel	
A model for verifying a user's phone number via two-factor authentication. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.	92

Chapter 4

Namespace Documentation

4.1 ARFE Namespace Reference

This is the over-arching namespace for all website related code.

Namespaces

- namespace [Controllers](#)
This namespace is a sub-namespace of the [ARFE](#) project namespace specifically for the ASP.NET [Controllers](#).
- namespace [Models](#)
The model classes that contain data representation objects for passing information between the web .cshtml code and the backend server code.
- namespace [Tests](#)
The Main namespace containing unit testing code for the [ARFE](#) Web project

Classes

- class [ApplicationSignInManager](#)
Creates a useful interface to the `Microsoft.AspNet.Identity.UserManager<TUser>` class for easily verifying user accounts with account userName/password verification handling.
- class [ApplicationUserManager](#)
Creates a useful interface to the `Microsoft.AspNet.Identity.UserManager<TUser>` class for easily registering new user accounts with account parameter verification handling.
- class [BlobManager](#)
This class is for managing all transactions between the AE website and Azure blob storage. All user containers, user blob creation/deletion, blob metadata, blob permissions, and blob listings are handled here.
- class [BundleConfig](#)
Configure all of the javascript and css include bundles. Register these with the entire application for simple inclusion in the .cshtml files.
- class [EmailService](#)
A class derived from the `Microsoft.AspNet.Identity.IdentityMessageService` class to implement third party account verification via email. This implementation is currently empty as no third party email verification is currently supported.
- class [FileConverter](#)
The [FileConverter](#) class serves as an interface between the ASP.NET controllers and other C# classes to `File↔Conversion.exe`. This is used throughout the application for managing file compatability and efficiency in both file storage and transfer.

- class [FilterConfig](#)
A class for defining filtered attributes to the ASP.NET web request and response communication. For example, the [RequireHttps] filter could be applied here to demand all traffic is only allowed if is transferred over SSL.
- class [MvcApplication](#)
- class [RouteConfig](#)
This class is for defining the valid url rout patterns for the application.
- class [SearchQueries](#)
This class contains all of the linq queries for the search and order functionality.
- class [SmsService](#)
A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via SMS text messaging. This implementation is currently empty as no third party SMS text messaging verification is currently supported.
- class [Startup](#)
The application startup class is called upon launching the application.
- class [UploadedFileCache](#)
A singleton class, constructed via the [GetInstance](#) method call. This class serves as a cache for tracking the location, details about, and automatic removal of files that are saved for intermediate storage between being uploaded to Azure blob storage, or converted to some other file type pending a download.

4.1.1 Detailed Description

This is the over-arching namespace for all website related code.

This namespaces is the [ARFE](#) project.

4.2 ARFE.Controllers Namespace Reference

This namespaces is a sub-namespace of the [ARFE](#) project namespace specifically for the ASP.NET [Controllers](#).

Classes

- class [AccountController](#)
A class derived from the Controller class that has all of the controller actions to manage user accounts.
- class [FileTypeModel](#)
A POCO class for the file type combo box
- class [HelpController](#)
A class derived from the Controller class that has all of the controller actions to dispaly the website's help pages.
- class [ManageController](#)
A class derived from the Controller class that has all of the controller actions to manage user registration and authentication.
- class [MobileAuthController](#)
The [MobileAuthController](#) serves as an API to the mobile application. The [Route] attribute allows this Controller to specify that to call any individual controller action within this class, you have to explicitly list the action method in the request URL.
- class [PublicContentController](#)
A class derived from the Controller class that has all of the controller actions to dispaly and interact with the public file content on the website.
- class [UploadController](#)
A class derived from the Controller class that has all of the controller actions to manage user file uploads.
- class [UserContentController](#)
A class derived from the Controller class that has all of the controller actions to dispaly and interact with the user's file content on the website.

Enumerations

- enum **OrderingOption** { **NewestFirst** = 1, **OldestFirst** = 2, **Alphabetical** = 3, **ReverseAlphabetical** = 4 }

4.2.1 Detailed Description

This namespace is a sub-namespace of the [ARFE](#) project namespace specifically for the ASP.NET [Controllers](#).

4.3 ARFE.Models Namespace Reference

The model classes that contain data representation objects for passing information between the web .cshtml code and the backend server code.

Classes

- class [AddPhoneNumberViewModel](#)
A model for registering a phone number to a user account.
- class [ApplicationDbContext](#)
This class inherits from and can override the Identity default Db that is automatically generated from the [ApplicationUser](#) class. If additional database fields are needed for the application database tables, they can be added here.
- class [ApplicationUser](#)
This class holds the profile information for a user within the application. Profile data for the user can be added by adding more properties to the [ApplicationUser](#) class, please visit <https://go.microsoft.com/fwlink/?LinkID=317594> to learn more.
- class [ChangePasswordViewModel](#)
A model to change a user's registered account password.
- class [ConfigureTwoFactorViewModel](#)
A model for registering third party two-factor authentication providers. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.
- class [ForgotPasswordViewModel](#)
A model for the user requesting a password reset via their registered email address.
- class [IndexViewModel](#)
A model for returning user login and verification.
- class [LoginViewModel](#)
A model for the user to log in to the website.
- class [ManageLoginsViewModel](#)
A model for viewing and reporting current and past user login information.
- class [RegisterViewModel](#)
A model for registering a new user.
- class [ResetPasswordViewModel](#)
A model to reset a user's password.
- class [SendCodeViewModel](#)
A model for sending a two-factor authentication message through a registered third party two-factor authentication provider.
- class [SetPasswordViewModel](#)
A model for registering a user's password upon account creation.
- class [VerifyCodeViewModel](#)
A model for verifying a two-factor authentication code that has been sent to a user.
- class [VerifyPhoneNumberViewModel](#)
A model for verifying a user's phone number via two-factor authentication. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.

4.3.1 Detailed Description

The model classes that contain data representation objects for passing information between the web .cshtml code and the backend server code.

4.4 ARFE.Tests Namespace Reference

The Main namespace containing unit testing code for the [ARFE](#) Web project

Namespaces

- namespace [Controllers](#)

The sub-namespace of the main namespace containing unit testing code for the [ARFE](#) Web project. This namespace is for testing the [ARFE.Controllers](#) code specifically.

Classes

- class [BlobTests](#)

Unit tests covering the non-Azure-facing functionality of the BlobManager.cs file in ARFE.csproj

- class [UploadedFileCacheTests](#)

Summary description for [UploadedFileCacheTests](#)

4.4.1 Detailed Description

The Main namespace containing unit testing code for the [ARFE](#) Web project

4.5 ARFE.Tests.Controllers Namespace Reference

The sub-namespace of the main namespace containing unit testing code for the [ARFE](#) Web project. This namespace is for testing the [ARFE.Controllers](#) code specifically.

Classes

- class [HomeControllerTest](#)

4.5.1 Detailed Description

The sub-namespace of the main namespace containing unit testing code for the [ARFE](#) Web project. This namespace is for testing the [ARFE.Controllers](#) code specifically.

4.6 AuthenticationTokenCache Namespace Reference

The namespace containing all code relating to the in memory cache used for mobile authentication.

Namespaces

- namespace [Tests](#)

The namespace containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

Classes

- class [TokenCache](#)

A class to manage tokens that are associated to user login information. This is used for mobile user authentication by verifying tokens, that the user explicitly requests, to that user's log in credentials. This system of token passing makes for simpler and more secure communication between the mobile application and the web service.

4.6.1 Detailed Description

The namespace containing all code relating to the in memory cache used for mobile authentication.

4.7 AuthenticationTokenCache.Tests Namespace Reference

The namespace containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

Classes

- class [TokenCacheTests](#)

The class containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

4.7.1 Detailed Description

The namespace containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

4.8 Common Namespace Reference

This namespace is for containing all custom objects and methods that should be made publicly available and easily accessible for all other projects within the appliaction.

Classes

- class [FileUIInfo](#)

A class that contains file representation information for displaying information about a user's files to the web interface.

- class [SupportedFileTypes](#)

The list of file type extensions that are supported by the file conversion software that this application uses.

4.8.1 Detailed Description

This namespace is for containing all custom objects and methods that should be made publicly available and easily accessible for all other projects within the appliaction.

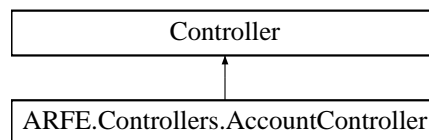
Chapter 5

Class Documentation

5.1 ARFE.Controllers.AccountController Class Reference

A class derived from the Controller class that has all of the controller actions to manage user accounts.

Inheritance diagram for ARFE.Controllers.AccountController:



Classes

- class **ChallengeResult**

Public Member Functions

- [AccountController](#) ()
The default constructor.
- [AccountController](#) ([ApplicationUserManager](#) userManager, [ApplicationSignInManager](#) signInManager)
An overloaded constructor that takes instances of the class properties as parameters.
- ActionResult [Login](#) (string returnUrl)
The default Login method.
- async Task< ActionResult > [Login](#) ([LoginViewModel](#) model, string returnUrl)
The login method that takes the completed login model object as login credentials and parameters.
- async Task< ActionResult > [VerifyCode](#) (string provider, string returnUrl, bool rememberMe)
Third party code verification
- async Task< ActionResult > [VerifyCode](#) ([VerifyCodeViewModel](#) model)
Third party code verification.
- ActionResult [Register](#) ()
Register a new user.
- async Task< ActionResult > [Register](#) ([RegisterViewModel](#) model)

Register a new user.

- async Task< ActionResult > [ConfirmEmail](#) (string userId, string code)
This method currently does nothing without third party email support.
- ActionResult [ForgotPassword](#) ()
This method currently does nothing without third party verification.
- async Task< ActionResult > [ForgotPassword](#) ([ForgotPasswordViewModel](#) model)
This method currently does nothing without third party verification.
- ActionResult [ForgotPasswordConfirmation](#) ()
This method currently does nothing without third party verification.
- ActionResult [ResetPassword](#) (string code)
Default password reset view from two-factor auth code.
- async Task< ActionResult > [ResetPassword](#) ([ResetPasswordViewModel](#) model)
Reset user password.
- ActionResult [ResetPasswordConfirmation](#) ()
Redirect to a page after password reset
- async Task< ActionResult > [SendCode](#) (string returnUrl, bool rememberMe)
This method currently does nothing without third party verification.
- async Task< ActionResult > [SendCode](#) ([SendCodeViewModel](#) model)
This method currently does nothing without third party verification.
- ActionResult [LogOff](#) ()
Log out.

Public Attributes

- [ApplicationSignInManager](#) [SignInManager](#) => [_SignInManager](#) ?? HttpContext.GetOwinContext().Get<[ApplicationSignInManager](#)>
A property reference to the [ApplicationSignInManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.
- [ApplicationUserManager](#) [UserManager](#) => [_UserManager](#) ?? HttpContext.GetOwinContext().GetUser<[ApplicationUserManager](#)>()
A property reference to the [ApplicationUserManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.

Protected Member Functions

- override void [Dispose](#) (bool disposing)
Clean up any application resources used by this controller.

5.1.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to manage user accounts.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AccountController() [1/2]

```
ARFE.Controllers.AccountController.AccountController ( )
```

The default constructor.

5.1.2.2 AccountController() [2/2]

```
ARFE.Controllers.AccountController.AccountController (
    ApplicationUserManager userManager,
    ApplicationSignInManager signInManager )
```

An overloaded constructor that takes instances of the class properties as parameters.

Parameters

<i>userManager</i>	An instance of the ApplicationUserManager class.
<i>signInManager</i>	An instance of the ApplicationSignInManager class.

5.1.3 Member Function Documentation

5.1.3.1 ConfirmEmail()

```
async Task<ActionResult> ARFE.Controllers.AccountController.ConfirmEmail (
    string userId,
    string code )
```

This method currently does nothing without third party email support.

Parameters

<i>userId</i>	
<i>code</i>	

Returns

5.1.3.2 Dispose()

```
override void ARFE.Controllers.AccountController.Dispose (
    bool disposing ) [protected]
```

Clean up any application resources used by this controller.

Parameters

<i>disposing</i>	should dispose
------------------	----------------

5.1.3.3 ForgotPassword() [1/2]

```
ActionResult ARFE.Controllers.AccountController.ForgotPassword ( )
```

This mehtod currently does nothing without third party verification.

Returns

5.1.3.4 ForgotPassword() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.ForgotPassword (
    ForgotPasswordViewModel model )
```

This method currently does nothing without third party verification.

Parameters

<i>model</i>	
--------------	--

Returns

5.1.3.5 ForgotPasswordConfirmation()

```
ActionResult ARFE.Controllers.AccountController.ForgotPasswordConfirmation ( )
```

This method currently does nothing without third party verification.

Returns

5.1.3.6 Login() [1/2]

```
ActionResult ARFE.Controllers.AccountController.Login (
    string returnUrl )
```

The default Login method.

Parameters

<i>returnUrl</i>	The page url to redirect to after login in.
------------------	---

Returns

A web page redirect to the *returnUrl* address.

5.1.3.7 Login() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.Login (
    LoginViewModel model,
    string returnUrl )
```

The login method that takes the completed login model object as login credentials and parameters.

Parameters

<i>model</i>	The login object with all of the user credentials and parameters.
<i>returnUrl</i>	The page url to redirect to after login in.

Returns

A web page redirect to the appropriate page following the login attempt.

5.1.3.8 LogOff()

```
ActionResult ARFE.Controllers.AccountController.LogOff ( )
```

Log out.

Returns

Web page redirect to the Public Content page.

5.1.3.9 Register() [1/2]

```
ActionResult ARFE.Controllers.AccountController.Register ( )
```

Register a new user.

Returns

A web page redirect to the Index page.

5.1.3.10 Register() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.Register (
    RegisterViewModel model )
```

Register a new user.

Parameters

<i>model</i>	The user-provided registration information.
--------------	---

Returns

A web page redirect to the appropriate url after user registration.

5.1.3.11 ResetPassword() [1/2]

```
ActionResult ARFE.Controllers.AccountController.ResetPassword (
    string code )
```

Default password reset view from two-factor auth code.

Parameters

<i>code</i>	Code provided from a two-factor auth provider.
-------------	--

Returns

A web page redirect to the appropriate page.

5.1.3.12 ResetPassword() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.ResetPassword (
    ResetPasswordViewModel model )
```

Reset user password.

Parameters

<i>model</i>	User-provided credentials and password reset.
--------------	---

Returns

A web page redirect to the appropriate confirmation page.

5.1.3.13 ResetPasswordConfirmation()

```
ActionResult ARFE.Controllers.AccountController.ResetPasswordConfirmation ( )
```

Redirect to a page after password reset

Returns

A web page redirect.

5.1.3.14 SendCode() [1/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.SendCode (
    string returnUrl,
    bool rememberMe )
```

This method currently does nothing without third party verification.

Parameters

<i>returnUrl</i>	
<i>rememberMe</i>	

Returns

5.1.3.15 SendCode() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.SendCode (
    SendCodeViewModel model )
```

This method currently does nothing without third party verification.

Parameters

<i>model</i>	
--------------	--

Returns

5.1.3.16 VerifyCode() [1/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.VerifyCode (
    string provider,
    string returnUrl,
    bool rememberMe )
```

Third party code verification

Parameters

<i>provider</i>	The provider name.
<i>returnUrl</i>	The redirect Url.
<i>rememberMe</i>	Remember the logged in user.

Returns

A web page redirect to the *returnUrl* address.

5.1.3.17 VerifyCode() [2/2]

```
async Task<ActionResult> ARFE.Controllers.AccountController.VerifyCode (
    VerifyCodeViewModel model )
```

Third party code verification.

Parameters

<i>model</i>	The user-provided code verification information.
--------------	--

Returns

A web page redirect to the appropriate page after code verification.

5.1.4 Member Data Documentation

5.1.4.1 SignInManager

```
ApplicationSignInManager ARFE.Controllers.AccountController.SignInManager => _SignInManager ??  
HttpContext.GetOwinContext().Get<ApplicationSignInManager>()
```

A property reference to the [ApplicationSignInManager](#) class. This reference is either to the instance provided by the `ManageController(ApplicationUserManager, ApplicationSignInManager)` constructor, or to the default instance that exists within the context of Microsoft.Owin.

5.1.4.2 UserManager

```
ApplicationUserManager ARFE.Controllers.AccountController.UserManager => _UserManager ?? Http←  
Context.GetOwinContext().GetUserManager<ApplicationUserManager>()
```

A property reference to the [ApplicationUserManager](#) class. This reference is either to the instance provided by the `ManageController(ApplicationUserManager, ApplicationSignInManager)` constructor, or to the default instance that exists within the context of Microsoft.Owin.

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

- ARFE/Controllers/AccountController.cs

5.2 ARFE.Models.AddPhoneNumberViewModel Class Reference

A model for registering a phone number to a user account.

Properties

- string [Number](#) [get, set]
The user's newly registered phone number.

5.2.1 Detailed Description

A model for registering a phone number to a user account.

5.2.2 Property Documentation

5.2.2.1 Number

```
string ARFE.Models.AddPhoneNumberViewModel.Number [get], [set]
```

The user's newly registered phone number.

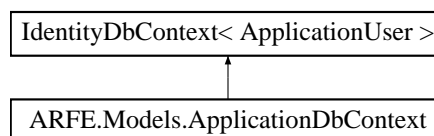
The documentation for this class was generated from the following file:

- ARFE/Models/ManageViewModels.cs

5.3 ARFE.Models.ApplicationDbContext Class Reference

This class inherits from and can override the Identity default Db that is automatically generated from the [ApplicationUser](#) class. If additionally database fields are needed for the application database tables, they can be added here.

Inheritance diagram for ARFE.Models.ApplicationDbContext:



Public Member Functions

- [ApplicationDbContext](#) ()
Default constructor that creates an instance from the DefaultConnection of the Identity registered Db.

Static Public Member Functions

- static [ApplicationDbContext Create](#) ()
A statically available method to generate a new default instance of this class.

5.3.1 Detailed Description

This class inherits from and can override the Identity default Db that is automatically generated from the [ApplicationUser](#) class. If additionally database fields are needed for the application database tables, they can be added here.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ApplicationDbContext()

```
ARFE.Models.ApplicationDbContext.ApplicationDbContext ( )
```

Default constructor that creates an instance from the DefaultConnection of the Identity registered Db.

5.3.3 Member Function Documentation

5.3.3.1 Create()

```
static ApplicationDbContext ARFE.Models.ApplicationDbContext.Create ( ) [static]
```

A statically available method to generate a new default instance of this class.

Returns

A new default instance of this class.

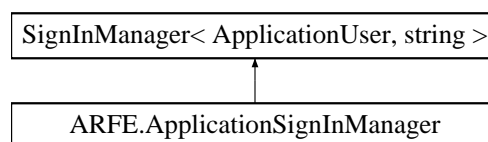
The documentation for this class was generated from the following file:

- ARFE/Models/IdentityModels.cs

5.4 ARFE.ApplicationSignInManager Class Reference

Creates a useful interface to the Microsoft.AspNet.Identity.UserManager<TUser> class for easily verifying user accounts with account userName/password verification handling.

Inheritance diagram for ARFE.ApplicationSignInManager:



Public Member Functions

- [ApplicationSignInManager](#) ([ApplicationUserManager](#) userManager, [IAuthenticationManager](#) authenticationManager)
Constructor, serves as a pass-through entity to the parent class constructor.
- override Task< ClaimsIdentity > [CreateUserIdentityAsync](#) ([ApplicationUser](#) user)
Creates a user identity object that represents a validated user and their credentials.

Static Public Member Functions

- static [ApplicationSignInManager Create](#) (IdentityFactoryOptions< [ApplicationSignInManager](#) > options, I← OwinContext context)

A method to register a new user sign in validator.

5.4.1 Detailed Description

Creates a useful interface to the Microsoft.AspNet.Identity.UserManager<TUser> class for easily verifying user accounts with account userName/password verification handling.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 ApplicationSignInManager()

```
ARFE.ApplicationSignInManager.ApplicationSignInManager (
    ApplicationUserManager userManager,
    IAuthenticationManager authenticationManager )
```

Constructor, serves as a pass-through entity to the parent class constructor.

Parameters

<i>userManager</i>	A reference to the ApplicationUserManager class.
<i>authenticationManager</i>	An instance of the Microsoft.Owin.Security.IAuthenticationManager class.

5.4.3 Member Function Documentation

5.4.3.1 Create()

```
static ApplicationSignInManager ARFE.ApplicationSignInManager.Create (
    IdentityFactoryOptions< ApplicationSignInManager > options,
    IOwinContext context ) [static]
```

A method to register a new user sign in validator.

Parameters

<i>options</i>	The user configuration option.
<i>context</i>	Serves as the configuration and user verification context.

Returns

A reference to the [ApplicationSignInManager](#) class.

5.4.3.2 CreateUserIdentityAsync()

```
override Task<ClaimsIdentity> ARFE.ApplicationSignInManager.CreateUserIdentityAsync (
    ApplicationUser user )
```

Creates a user identity object that represents a validated user and their credentials.

Parameters

<i>user</i>	The registered user to create the identity object for.
-------------	--

Returns

The authenticated user's identity object.

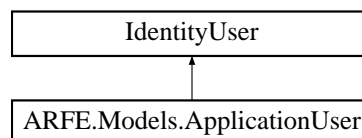
The documentation for this class was generated from the following file:

- ARFE/App_Start/IdentityConfig.cs

5.5 ARFE.Models.ApplicationUser Class Reference

This class holds the profile information for a user within the application. Profile data for the user can be added by adding more properties to the [ApplicationUser](#) class, please visit <https://go.microsoft.com/fwlink/?LinkID=317594> to learn more.

Inheritance diagram for ARFE.Models.ApplicationUser:



Public Member Functions

- async Task< ClaimsIdentity > [GenerateUserIdentityAsync](#) (userManager< [ApplicationUser](#) > manager)
Create and register new users within the application and store their credentials.

5.5.1 Detailed Description

This class holds the profile information for a user within the application. Profile data for the user can be added by adding more properties to the [ApplicationUser](#) class, please visit <https://go.microsoft.com/fwlink/?LinkID=317594> to learn more.

5.5.2 Member Function Documentation

5.5.2.1 GenerateUserIdentityAsync()

```
async Task<ClaimsIdentity> ARFE.Models.ApplicationUser.GenerateUserIdentityAsync (
    UserManager< ApplicationUser > manager )
```

Create and register new users within the application and store their credentials.

Parameters

<i>manager</i>	The class object used for user state interaction and representation.
----------------	--

Returns

A new claim identity for the newly created user.

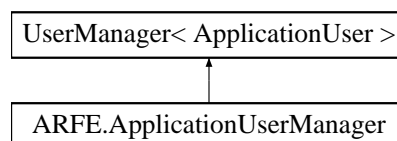
The documentation for this class was generated from the following file:

- ARFE/Models/IdentityModels.cs

5.6 ARFE.ApplicationUserManager Class Reference

Creates a useful interface to the Microsoft.AspNet.Identity.UserManager<TUser> class for easily registering new user accounts with account parameter verification handling.

Inheritance diagram for ARFE.ApplicationUserManager:



Public Member Functions

- **ApplicationUserManager** (IUserStore< ApplicationUser > store)
Constructor, serves as a pass-through entity to the parent class constructor.

Static Public Member Functions

- static **ApplicationUserManager Create** (IdentityFactoryOptions< ApplicationUserManager > options, I←OwinContext context)
A method to register a new user with validated credentials.

5.6.1 Detailed Description

Creates a useful interface to the `Microsoft.AspNet.Identity.UserManager<TUser>` class for easily registering new user accounts with account parameter verification handling.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 ApplicationUserManager()

```
ARFE.ApplicationUserManager.ApplicationUserManager (
    IUserStore< ApplicationUser > store )
```

Constructor, serves as a pass-through entity to the parent class constructor.

Parameters

<i>store</i>	An instance of <code>Microsoft.AspNet.Identity.IUserStore<TUser></code> , which serves as the collection of application registered users.
--------------	---

5.6.3 Member Function Documentation

5.6.3.1 Create()

```
static ApplicationUserManager ARFE.ApplicationUserManager.Create (
    IdentityFactoryOptions< ApplicationUserManager > options,
    IOwinContext context ) [static]
```

A method to register a new user with validated credentials.

Parameters

<i>options</i>	The user configuration option.
<i>context</i>	Serves as the configuration and user verification context.

Returns

A reference to the `ApplicationUserManager` class.

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

- ARFE/App_Start/IdentityConfig.cs
-

5.7 ARFE.BlobManager Class Reference

This class is for managing all transactions between the AE website and Azure blob storage. All user containers, user blob creation/deletion, blob metadata, blob permissions, and blob listings are handled here.

Public Member Functions

- [BlobManager](#) ()
Default constructor. No special parameters or attributes are needed to use this class.
- [CloudBlobContainer](#) [GetOrCreateBlobContainer](#) (string userName)
Given the user name, get a reference to the user's associated blob container, or create a container if none exists.
- string [FormatBlobContainerName](#) (string userName)
Azure Blob Containers have a rigid rule set on Container name format. <https://blogs.msdn.microsoft.com/jmstall/2014/06/12/azure-storage-naming-rules/> Only lowercase letters, numbers, and dashes are allowed. Only one consecutive dash allowed. Dash must be surrounded by numers or letters. Name must be between 3 and 63 characters in length
- bool [UploadBlobToUserContainer](#) (string userName, string fileName, string filePath, string description="", bool overwrite=false)
Upload a file as a blob to cloud storage in the provided user's blob container.
- bool [DeleteBlobByNameInUserContainer](#) (string userName, string blobName)
Delete a blob by name stored in the user's container. Blob's stored in the public container can only be deleted by the owner (handled by the UI).
- string [ConvertAndDownloadBlobFromUserContainer](#) (string userName, string fileName, string request←
Extension, string path)
Attempt to download the requested file from blob storage. If the file does not exist, perform the appropriate file conversion and give a downloadable link to the converted file that matches the user's request.
- List< Uri > [ListBlobUrisInUserContainer](#) (string userName)
Get a list of blob Uris within the blob container for a given user name.
- List< Tuple< string, Uri > > [ListBlobNamesToUrisInUserContainer](#) (string userName)
Get a list of associations of blob names to Uris for a given user name/container.
- bool [UploadBlobToPublicContainer](#) (string userName, string fileName, string filePath, string description="", bool overwrite=false)
Given the current logged in user name and the name of the file to upload, and the local path to the file, upload the file to the public blob container as a new blob owned by the user.
- List< [FileUIInfo](#) > [ListPrivateBlobInfoForUI](#) (string userName)
Iterate through all blobs in a given user's container. For each blob get the meta information (blob Name, author name, description, last modify time) about that blob as a POCO (Plain Old C# Object) and create a list of all of the blob's information.
- List< [FileUIInfo](#) > [ListPublicBlobInfoForUI](#) ()
Make a call to [ListPrivateBlobInfoForUI\(string\)](#) with the userName "public" to query the public container.
- List< Tuple< string, Uri > > [ListBlobNamesToUrisInPublicContainer](#) ()
Get a list of associations of blob names to Uris within the public container
- List< Uri > [ListBlobUrisInPublicContainerOwnedBy](#) (string userName)
Gets a list of all blob Uris in the public container and filters by the ones that are owned by the provided user name.
- List< Tuple< string, Uri > > [ListBlobNamesToUrisInPublicContainerOwnedBy](#) (string userName)
Get all Uris for blobs owned by the given user name by calling [ListBlobUrisInPublicContainerOwnedBy\(string\)](#). For each Uri. The blob name is the last portion of the string following the last '/' character.

Static Public Member Functions

- static void [RemoveExpiredTemporaryBlobs](#) ()
Lazily delete blobs that are stored as the result of conversion requests. Maintain the small .fbx files but delete the extras as they become old.

5.7.1 Detailed Description

This class is for managing all transactions between the AE website and Azure blob storage. All user containers, user blob creation/deletion, blob metadata, blob permissions, and blob listings are handled here.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 BlobManager()

```
ARFE.BlobManager.BlobManager ( )
```

Default constructor. No special parameters or attributes are needed to use this class.

5.7.3 Member Function Documentation

5.7.3.1 ConvertAndDownloadBlobFromUserContainer()

```
string ARFE.BlobManager.ConvertAndDownloadBlobFromUserContainer (
    string userName,
    string fileName,
    string requestExtension,
    string path )
```

Attempt to download the requested file from blob storage. If the file does not exist, perform the appropriate file conversion and give a downloadable link to the converted file that matches the user's request.

Parameters

<i>userName</i>	The name of the user/blob container for the download request.
<i>fileName</i>	The name of the blob for the download request.
<i>requestExtension</i>	The requested file type.
<i>path</i>	The file path for download/conversion if necessary.

Returns

5.7.3.2 DeleteBlobByNameInUserContainer()

```
bool ARFE.BlobManager.DeleteBlobByNameInUserContainer (
    string userName,
    string blobName )
```

Delete a blob by name stored in the user's container. Blob's stored in the public container can only be deleted by the owner (handled by the UI).

Parameters

<i>userName</i>	The name of the user/container to delete the blob from.
<i>blobName</i>	The name of the blob to delete.

Returns

5.7.3.3 FormatBlobContainerName()

```
string ARFE.BlobManager.FormatBlobContainerName (  
    string userName )
```

Azure Blob Containers have a rigid rule set on Container name format. <https://blogs.msdn.microsoft.com/jmstall/2014/06/12/azure-storage-naming-rules/> Only lowercase letters, numbers, and dashes are allowed. Only one consecutive dash allowed. Dash must be surrounded by numbers or letters. Name must be between 3 and 63 characters in length

Parameters

<i>userName</i>	Container names are created from the user's user name.
-----------------	--

Returns

- string.Empty - The provided name was unsalvageable.
- The appropriately formatted container name.

5.7.3.4 GetOrCreateBlobContainer()

```
CloudBlobContainer ARFE.BlobManager.GetOrCreateBlobContainer (  
    string userName )
```

Given the user name, get a reference to the user's associated blob container, or create a container if none exists.

Parameters

<i>userName</i>	The name of the user and associated blob container.
-----------------	---

Returns

- The blob container if able to reference or create it.
- Null if unable to reference or create the container.

5.7.3.5 ListBlobNamesToUrisInPublicContainer()

```
List<Tuple<string, Uri> > ARFE.BlobManager.ListBlobNamesToUrisInPublicContainer ( )
```

Get a list of associations of blob names to Uris within the public container

Returns

A list of tuples of blob names and their associated Uris from the public container

5.7.3.6 ListBlobNamesToUrisInPublicContainerOwnedBy()

```
List<Tuple<string, Uri> > ARFE.BlobManager.ListBlobNamesToUrisInPublicContainerOwnedBy (
    string userName )
```

Get all Uris for blobs owned by the given user name by calling [ListBlobUrisInPublicContainerOwnedBy\(string\)](#). For each Uri. The blob name is the last portion of the string following the last '/' character.

Parameters

<i>userName</i>	The name of the user to filter by.
-----------------	------------------------------------

Returns

A list of object pairs of (blob Name, blob Uri) for each blob in the public container owned by the user.

5.7.3.7 ListBlobNamesToUrisInUserContainer()

```
List<Tuple<string, Uri> > ARFE.BlobManager.ListBlobNamesToUrisInUserContainer (
    string userName )
```

Get a list of associations of blob names to Uris for a given user name/container.

Parameters

<i>userName</i>	The user/container name.
-----------------	--------------------------

Returns

A list of associations of blob names to Uris.

5.7.3.8 ListBlobUrisInPublicContainerOwnedBy()

```
List<Uri> ARFE.BlobManager.ListBlobUrisInPublicContainerOwnedBy (
    string userName )
```

Gets a list of all blob Uris in the public container and filters by the ones that are owned by the provided user name.

Parameters

<i>userName</i>	The name of the owner to filter the list by.
-----------------	--

Returns

A list of all Uris to blobs that are owned by the user with the provided user name.

5.7.3.9 ListBlobUrisInUserContainer()

```
List<Uri> ARFE.BlobManager.ListBlobUrisInUserContainer (
    string userName )
```

Get a list of blob Uris within the blob container for a given user name.

Parameters

<i>userName</i>	The user name to look up associated blob Uris.
-----------------	--

Returns

The list of blob Uris for the give user.

5.7.3.10 ListPrivateBlobInfoForUI()

```
List<FileUIInfo> ARFE.BlobManager.ListPrivateBlobInfoForUI (
    string userName )
```

Iterate through all blobs in a given user's container. For each blob get the meta information (blob Name, author name, description, last modify time) about that blob as a POCO (Plain Old C# Object) and create a list of all of the blob's information.

Parameters

<i>userName</i>	The name of the user/blob container.
-----------------	--------------------------------------

Returns

A list of POCOs containing blob metaData information.

5.7.3.11 ListPublicBlobInfoForUI()

```
List<FileUIInfo> ARFE.BlobManager.ListPublicBlobInfoForUI ( )
```

Make a call to [ListPrivateBlobInfoForUI\(string\)](#) with the userName "public" to query the public container.

Returns

A list of POCOs containing blob metaData information.

5.7.3.12 RemoveExpiredTemporaryBlobs()

```
static void ARFE.BlobManager.RemoveExpiredTemporaryBlobs ( ) [static]
```

Lazily delete blobs that are stored as the result of conversion requests. Maintain the small .fbx files but delete the extras as they become old.

5.7.3.13 UploadBlobToPublicContainer()

```
bool ARFE.BlobManager.UploadBlobToPublicContainer (
    string userName,
    string fileName,
    string filePath,
    string description = "",
    bool overwrite = false )
```

Given the current logged in user name and the name of the file to upload, and the local path to the file, upload the file to the public blob container as a new blob owned by the user.

Parameters

<i>userName</i>	Name of the owner
<i>fileName</i>	Name of the file
<i>filePath</i>	Path to the file

Returns

- True: The file was successfully uploaded.
- False: The file failed to upload.

5.7.3.14 UploadBlobToUserContainer()

```
bool ARFE.BlobManager.UploadBlobToUserContainer (
    string userName,
    string fileName,
    string filePath,
    string description = "",
    bool overwrite = false )
```

Upload a file as a blob to cloud storage in the provided user's blob container.

Parameters

<i>userName</i>	The name of the user and associated blob container.
<i>fileName</i>	The name of the file to upload to blob storage.
<i>filePath</i>	The path to the file to upload to blob storage.
<i>description</i>	The user's description of the file.
<i>overwrite</i>	Whether or not to overwrite any previously existing blobs with the same name.

Returns

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

- ARFE/BlobManager.cs

5.8 ARFE.Tests.BlobTests Class Reference

Unit tests covering the non-Azure-facing functionality of the BlobManager.cs file in ARFE.csproj

Public Member Functions

- void [Test_Constructor](#) ()
Test the validity of the using alias at the top of this file (using BlobMan = [ARFE.BlobManager](#);) and the correctness of the constructor of the [BlobManager](#) class.
- void [Test_FormatBlobContainerName](#) ()
Test that the method for formatting blob container names follows Microsoft naming procedures available at <https://blogs.msdn.microsoft.com/jmstall/2014/06/12/azure-storage-naming-rules/>

5.8.1 Detailed Description

Unit tests covering the non-Azure-facing functionality of the BlobManager.cs file in ARFE.csproj

Calls to Azure storage are unable to be made from this test project. The [BlobManager](#) class is unable to appropriately access the configuration string required from the Web.config file, located at ARFE/Web.config, without the application startup sequence. Azure-facing functionality of the BlobManager.cs file must be manually tested.

5.8.2 Member Function Documentation

5.8.2.1 Test_Constructor()

```
void ARFE.Tests.BlobTests.Test_Constructor ( )
```

Test the validity of the using alias at the top of this file (using BlobMan = [ARFE.BlobManager](#);) and the correctness of the constructor of the [BlobManager](#) class.

5.8.2.2 Test_FormatBlobContainerName()

```
void ARFE.Tests.BlobTests.Test_FormatBlobContainerName ( )
```

Test that the method for formatting blob container names follows Microsoft naming procedures available at <https://blogs.msdn.microsoft.com/jmstall/2014/06/12/azure-storage-naming-rules/>

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

- ARFE.Tests/BlobTests.cs

5.9 ARFE.BundleConfig Class Reference

Configure all of the javascript and css include bundles. Register these with the entire application for simple inclusion in the .cshtml files.

Static Public Member Functions

- static void [RegisterBundles](#) (BundleCollection bundles)

The collection of script bundles to register with the application. For more information on bundling, visit <https://go.microsoft.com/fwlink/?LinkId=301862>

5.9.1 Detailed Description

Configure all of the javascript and css include bundles. Register these with the entire application for simple inclusion in the .cshtml files.

5.9.2 Member Function Documentation

5.9.2.1 RegisterBundles()

```
static void ARFE.BundleConfig.RegisterBundles (
    BundleCollection bundles ) [static]
```

The collection of script bundles to register with the application. For more information on bundling, visit <https://go.microsoft.com/fwlink/?LinkId=301862>

Parameters

<i>bundles</i>	
----------------	--

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

- ARFE/App_Start/BundleConfig.cs

5.10 ARFE.Models.ChangePasswordViewModel Class Reference

A model to change a user's registered account password.

Properties

- string **OldPassword** [get, set]
The property field for the user's current password that they are changing.
- string **NewPassword** [get, set]
The property field for the user's new password that they wish to set.
- string **ConfirmPassword** [get, set]
The property field for the confirmed re-entry of the user's new password that they wish to set.

5.10.1 Detailed Description

A model to change a user's registered account password.

5.10.2 Property Documentation

5.10.2.1 ConfirmPassword

```
string ARFE.Models.ChangePasswordViewModel.ConfirmPassword [get], [set]
```

The property field for the confirmed re-entry of the user's new password that they wish to set.

5.10.2.2 NewPassword

```
string ARFE.Models.ChangePasswordViewModel.NewPassword [get], [set]
```

The property field for the user's new password that they wish to set.

5.10.2.3 OldPassword

```
string ARFE.Models.ChangePasswordViewModel.OldPassword [get], [set]
```

The property field for the user's current password that they are changing.

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

- ARFE/Models/ManageViewModels.cs

5.11 ARFE.Models.ConfigureTwoFactorViewModel Class Reference

A model for registering third party two-factor authentication providers. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.

Properties

- string [SelectedProvider](#) [get, set]
The name of the third party two-factor authentication provider.
- ICollection< System.Web.Mvc.SelectListItem > [Providers](#) [get, set]
A List of third party two-factor authentication provider that can be used as a combo-box for user selection on a web page UI.

5.11.1 Detailed Description

A model for registering third party two-factor authentication providers. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.

5.11.2 Property Documentation

5.11.2.1 Providers

```
ICollection<System.Web.Mvc.SelectListItem> ARFE.Models.ConfigureTwoFactorViewModel.Providers  
[get], [set]
```

A List of third party two-factor authentication provider that can be used as a combo-box for user selection on a web page UI.

5.11.2.2 SelectedProvider

```
string ARFE.Models.ConfigureTwoFactorViewModel.SelectedProvider [get], [set]
```

The name of the third party two-factor authentication provider.

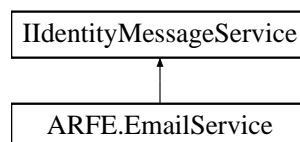
The documentation for this class was generated from the following file:

- ARFE/Models/ManageViewModels.cs

5.12 ARFE.EmailService Class Reference

A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via email. This implementation is currently empty as no third party email verification is currently supported.

Inheritance diagram for ARFE.EmailService:



Public Member Functions

- Task [SendAsync](#) (IdentityMessage message)

This method is required to be overwritten by the inheritance pattern of the Microsoft.AspNet.Identity.IIdentityMessageService class. As no third party email verification is currently supported, the method just returns 0 to the caller.

5.12.1 Detailed Description

A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via email. This implementation is currently empty as no third party email verification is currently supported.

5.12.2 Member Function Documentation

5.12.2.1 SendAsync()

```
Task ARFE.EmailService.SendAsync (
    IdentityMessage message )
```

This method is required to be overwritten by the inheritance pattern of the Microsoft.AspNet.Identity.IIdentityMessageService class. As no third party email verification is currently supported, the method just returns 0 to the caller.

Parameters

<i>message</i>	The message to send via the registered email provider.
----------------	--

Returns

0

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

- ARFE/App_Start/IdentityConfig.cs

5.13 ARFE.FileConverter Class Reference

The [FileConverter](#) class serves as an interface between the ASP.NET controllers and other C# classes to FileConversion.exe. This is used throughout the application for managing file compatability and efficiency in both file storage and transfer.

Public Member Functions

- [FileConverter](#) (string inputFolder, string outputFolder)
Public constructor that requires parameters for both the file path location of the file to convert and the file path location for the files produced by the conversion process.
 - bool [ConvertToFBX](#) (string fileName)
First checks if the file is of a supported type, by calling ValidateExtension(string). If the file type is supported, a call to FileConversion.exe is made to convert the file to a .fbx file.
 - bool [ConvertToDAE](#) (string fileName)
-

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .dae file.

- bool `ConvertToOBJ` (string fileName)

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .obj file.

- bool `ConvertToSTL` (string fileName)

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .stl file.

- bool `ConvertToPLY` (string fileName)

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .ply file.

Public Attributes

- string `InputFolder` => `_InputFolder`

Public read-only property defined with an expression-bodied member. The property is the file path location of the file to convert.

- string `OutputFolder` => `_OutputFolder`

Public read-only property defined with an expression-bodied member. The property is the file path location of the files produced by the conversion process.

- string `BasePath` => `_AppDomainPath`

Public read-only property defined with an expression-bodied member. The property is the base directory of the `AppDomainPath`.

5.13.1 Detailed Description

The `FileConverter` class serves as an interface between the ASP.NET controllers and other C# classes to `FileConversion.exe`. This is used throughout the application for managing file compatability and efficiency in both file storage and transfer.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 `FileConverter()`

```
ARFE.FileConverter.FileConverter (
    string inputFolder,
    string outputFolder )
```

Public constructor that requires parameters for both the file path location of the file to convert and the file path location for the files produced by the conversion process.

Parameters

<code>inputFolder</code>	The file path location of the file to convert.
<code>outputFolder</code>	The file path location for the files produced by the conversion process.

5.13.3 Member Function Documentation

5.13.3.1 ConvertToDAE()

```
bool ARFE.FileConverter.ConvertToDAE (
    string fileName )
```

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .dae file.

Parameters

<i>fileName</i>	The name of the file to convert in the InputFolder directory.
-----------------	---

Returns

- True: The file was converted successfully.
- False: The file was not converted successfully.

5.13.3.2 ConvertToFBX()

```
bool ARFE.FileConverter.ConvertToFBX (
    string fileName )
```

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .fbx file.

Parameters

<i>fileName</i>	The name of the file to convert in the InputFolder directory.
-----------------	---

Returns

- True: The file was converted successfully.
- False: The file was not converted successfully.

5.13.3.3 ConvertToOBJ()

```
bool ARFE.FileConverter.ConvertToOBJ (
    string fileName )
```

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .obj file.

Parameters

<i>fileName</i>	The name of the file to convert in the InputFolder directory.
-----------------	---

Returns

- True: The file was converted successfully.
- False: The file was not converted successfully.

5.13.3.4 ConvertToPLY()

```
bool ARFE.FileConverter.ConvertToPLY (
    string fileName )
```

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .ply file.

Parameters

<i>fileName</i>	The name of the file to convert in the InputFolder directory.
-----------------	---

Returns

- True: The file was converted successfully.
- False: The file was not converted successfully.

5.13.3.5 ConvertToSTL()

```
bool ARFE.FileConverter.ConvertToSTL (
    string fileName )
```

First checks if the file is of a supported type, by calling `ValidateExtension(string)`. If the file type is supported, a call to `FileConversion.exe` is made to convert the file to a .stl file.

Parameters

<i>fileName</i>	The name of the file to convert in the InputFolder directory.
-----------------	---

Returns

- True: The file was converted successfully.
 - False: The file was not converted successfully.
-

5.13.4 Member Data Documentation

5.13.4.1 BasePath

```
string ARFE.FileConverter.BasePath => _AppDomainPath
```

Public read-only property defined with an expression-bodied member. The property is the base directory of the AppDomainPath.

5.13.4.2 InputFolder

```
string ARFE.FileConverter.InputFolder => _InputFolder
```

Public read-only property defined with an expression-bodied member. The property is the file path location of the file to convert.

5.13.4.3 OutputFolder

```
string ARFE.FileConverter.OutputFolder => _OutputFolder
```

Public read-only property defined with an expression-bodied member. The property is the file path location of the files produced by the conversion process.

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

- ARFE/FileConverter.cs

5.14 FileTypeModel Class Reference

Properties

- string **FileType** [get, set]
- IEnumerable< SelectListItem > **FileTypes** [get, set]

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

- ARFE/Controllers/PublicContentController.cs

5.15 ARFE.Controllers.FileTypeModel Class Reference

A POCO class for the file type combo box

Properties

- string `FileType` [get, set]
The selected file type
- List< SelectListItem > `FileTypes` [get, set]
The list of file type options.

5.15.1 Detailed Description

A POCO class for the file type combo box

5.15.2 Property Documentation

5.15.2.1 FileType

```
string ARFE.Controllers.FileTypeModel.FileType [get], [set]
```

The selected file type

5.15.2.2 FileTypes

```
List<SelectListItem> ARFE.Controllers.FileTypeModel.FileTypes [get], [set]
```

The list of file type options.

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

- ARFE/Controllers/UserContentController.cs

5.16 Common.FileUIInfo Class Reference

A class that contains file representation information for displaying information about a user's files to the web interface.

Public Member Functions

- `FileUIInfo` (string fileName="", string author="", string description="", DateTime uploaded=new DateTime())
A constructor that takes any or all parameters to fill the defining object properties.

Properties

- string `FileName` [get, set]
The name of the file.
- string `Author` [get, set]
The name of the owner of the file.
- string `Description` [get, set]
The user-provided description of the file.
- DateTime `UploadDate` [get, set]
The UTC time when the file was uploaded to Azure Blob storage.

5.16.1 Detailed Description

A class that contains file representation information for displaying information about a user's files to the web interface.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 FileUIInfo()

```
Common.FileUIInfo.FileUIInfo (
    string fileName = "",
    string author = "",
    string description = "",
    DateTime uploaded = new DateTime() )
```

A constructor that takes any or all parameters to fill the defining object properties.

Parameters

<i>fileName</i>	The name of the file. The default value is empty.
<i>author</i>	The name of the owner of the file. the default value is empty.
<i>description</i>	The user-provided description of the file. The default value is empty.
<i>uploaded</i>	The UTC time when the file was uploaded to Azure Blob storage.

5.16.3 Property Documentation

5.16.3.1 Author

```
string Common.FileUIInfo.Author [get], [set]
```

The name of the owner of the file.

5.16.3.2 Description

```
string Common.FileUIInfo.Description [get], [set]
```

The user-provided description of the file.

5.16.3.3 FileName

```
string Common.FileUIInfo.FileName [get], [set]
```

The name of the file.

5.16.3.4 UploadDate

```
DateTime Common.FileUIInfo.UploadDate [get], [set]
```

The UTC time when the file was uploaded to Azure Blob storage.

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

- Common/CommonClasses.cs

5.17 ARFE.FilterConfig Class Reference

A class for defining filtered attributes to the ASP.NET web request and response communication. For example, the [RequireHttps] filter could be applied here to demand all traffic is only allowed if is transferred over SSL.

Static Public Member Functions

- static void [RegisterGlobalFilters](#) (GlobalFilterCollection filters)
Register the filters to the ASP.NET application

5.17.1 Detailed Description

A class for defining filtered attributes to the ASP.NET web request and response communication. For example, the [RequireHttps] filter could be applied here to demand all traffic is only allowed if is transferred over SSL.

5.17.2 Member Function Documentation

5.17.2.1 RegisterGlobalFilters()

```
static void ARFE.FilterConfig.RegisterGlobalFilters (  
    GlobalFilterCollection filters ) [static]
```

Register the filters to the ASP.NET application

Parameters

<i>filters</i>	The collection of filters to add to the application.
----------------	--

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

- ARFE/App_Start/FilterConfig.cs

5.18 ARFE.Models.ForgotPasswordViewModel Class Reference

A model for the user requesting a password reset via their registered email address.

Properties

- string [Email](#) [get, set]
The user's registered email address.

5.18.1 Detailed Description

A model for the user requesting a password reset via their registered email address.

5.18.2 Property Documentation

5.18.2.1 Email

```
string ARFE.Models.ForgotPasswordViewModel.Email [get], [set]
```

The user's registered email address.

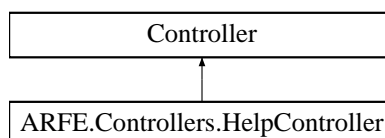
The documentation for this class was generated from the following file:

- ARFE/Models/AccountViewModels.cs

5.19 ARFE.Controllers.HelpController Class Reference

A class derived from the Controller class that has all of the controller actions to display the website's help pages.

Inheritance diagram for ARFE.Controllers.HelpController:



Public Member Functions

- ActionResult [Help](#) ()

The Index controller action is the default action called when the UserContent page is browsed to.

5.19.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to display the website's help pages.

5.19.2 Member Function Documentation

5.19.2.1 Help()

```
ActionResult ARFE.Controllers.HelpController.Help ( )
```

The Index controller action is the default action called when the UserContent page is browsed to.

Returns

A view to the "Index.cshtml" page in the Views/Help/ folder.

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

- ARFE/Controllers/HelpController.cs

5.20 ARFE.Tests.Controllers.HomeControllerTest Class Reference

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

- ARFE.Tests/Controllers/HomeControllerTest.cs

5.21 ARFE.Models.IndexViewModel Class Reference

A model for returning user login and verification.

Properties

- bool [HasPassword](#) [get, set]
A property for whether the returning site visitor has a registered password.
- IList< UserLoginInfo > [Logins](#) [get, set]
A property for the list of returning site visitor log in records.
- string [PhoneNumber](#) [get, set]
A property for the returning site visitor's registered phone number.
- bool [TwoFactor](#) [get, set]
A property for if the returning site visitor turned on two-factor authentication.
- bool [BrowserRemembered](#) [get, set]
A property for if the returning site visitor wants their browser usage remembered.

5.21.1 Detailed Description

A model for returning user login and verification.

5.21.2 Property Documentation

5.21.2.1 BrowserRemembered

```
bool ARFE.Models.IndexViewModel.BrowserRemembered [get], [set]
```

A property for if the returning site visitor wants their browser usage remembered.

5.21.2.2 HasPassword

```
bool ARFE.Models.IndexViewModel.HasPassword [get], [set]
```

A property for whether the returning site visitor has a registered password.

5.21.2.3 Logins

```
IList<UserLoginInfo> ARFE.Models.IndexViewModel.Logins [get], [set]
```

A property for the list of returning site visitor log in records.

5.21.2.4 PhoneNumber

```
string ARFE.Models.IndexViewModel.PhoneNumber [get], [set]
```

A property for the returning site visitor's registered phone number.

5.21.2.5 TwoFactor

```
bool ARFE.Models.IndexViewModel.TwoFactor [get], [set]
```

A property for if the returning site visitor turned on two-factor authentication.

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

- ARFE/Models/ManageViewModels.cs

5.22 ARFE.Models.LoginViewModel Class Reference

A model for the user to log in to the website.

Properties

- string [Email](#) [get, set]
The user's email account that has been registered with their account.
- string [Password](#) [get, set]
The user's account password.
- bool [RememberMe](#) [get, set]
If the user wants their sign-in information to be automatically verified.

5.22.1 Detailed Description

A model for the user to log in to the website.

5.22.2 Property Documentation

5.22.2.1 Email

```
string ARFE.Models.LoginViewModel.Email [get], [set]
```

The user's email account that has been registered with their account.

5.22.2.2 Password

```
string ARFE.Models.LoginViewModel.Password [get], [set]
```

The user's account password.

5.22.2.3 RememberMe

```
bool ARFE.Models.LoginViewModel.RememberMe [get], [set]
```

If the user wants their sign-in information to be automatically verified.

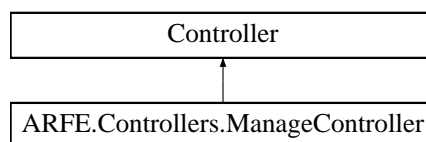
The documentation for this class was generated from the following file:

- ARFE/Models/AccountViewModels.cs

5.23 ARFE.Controllers.ManageController Class Reference

A class derived from the Controller class that has all of the controller actions to manage user registration and authentication.

Inheritance diagram for ARFE.Controllers.ManageController:



Public Types

- enum **ManageMessageId** {
 AddPhoneSuccess, ChangePasswordSuccess, SetTwoFactorSuccess, SetPasswordSuccess,
 RemoveLoginSuccess, RemovePhoneSuccess, Error }

Public Member Functions

- [ManageController](#) ()
The default constructor.
- [ManageController](#) ([ApplicationUserManager](#) userManager, [ApplicationSignInManager](#) signInManager)
An overloaded constructor that takes instances of the class properties as parameters.
- async Task< ActionResult > **Index** (ManageMessageId? message)
- async Task< ActionResult > **RemoveLogin** (string loginProvider, string providerKey)
- ActionResult **AddPhoneNumber** ()
- async Task< ActionResult > **AddPhoneNumber** ([AddPhoneNumberViewModel](#) model)
- async Task< ActionResult > **EnableTwoFactorAuthentication** ()
- async Task< ActionResult > **DisableTwoFactorAuthentication** ()
- async Task< ActionResult > **VerifyPhoneNumber** (string phoneNumber)
- async Task< ActionResult > **VerifyPhoneNumber** ([VerifyPhoneNumberViewModel](#) model)
- async Task< ActionResult > **RemovePhoneNumber** ()
- ActionResult **ChangePassword** ()
- async Task< ActionResult > **ChangePassword** ([ChangePasswordViewModel](#) model)
- ActionResult **SetPassword** ()
- async Task< ActionResult > **SetPassword** ([SetPasswordViewModel](#) model)
- async Task< ActionResult > **ManageLogins** (ManageMessageId? message)
- ActionResult **LinkLogin** (string provider)
- async Task< ActionResult > **LinkLoginCallback** ()

Public Attributes

- [ApplicationSignInManager](#) [SignInManager](#) => _SignInManager ?? HttpContext.GetOwinContext().Get<[ApplicationSignInManager](#)>()
A property reference to the [ApplicationSignInManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.
- [ApplicationUserManager](#) [UserManager](#) => _userManager ?? HttpContext.GetOwinContext().GetUserManager<[ApplicationUserManager](#)>()
A property reference to the [ApplicationUserManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.

Protected Member Functions

- override void **Dispose** (bool disposing)

5.23.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to manage user registration and authentication.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 ManageController() [1/2]

```
ARFE.Controllers.ManageController.ManageController ( )
```

The default constructor.

5.23.2.2 ManageController() [2/2]

```
ARFE.Controllers.ManageController.ManageController (
    ApplicationUserManager userManager,
    ApplicationSignInManager signInManager )
```

An overloaded constructor that takes instances of the class properties as parameters.

Parameters

<i>userManager</i>	An instance of the ApplicationUserManager class.
<i>signInManager</i>	An instance of the ApplicationSignInManager class.

5.23.3 Member Data Documentation

5.23.3.1 SignInManager

```
ApplicationSignInManager ARFE.Controllers.ManageController.SignInManager => _SignInManager ??
HttpContext.GetOwinContext().Get<ApplicationSignInManager>()
```

A property reference to the [ApplicationSignInManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.

5.23.3.2 UserManager

```
ApplicationUserManager ARFE.Controllers.ManageController.UserManager => _UserManager ??
HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>()
```

A property reference to the [ApplicationUserManager](#) class. This reference is either to the instance provided by the [ManageController\(ApplicationUserManager, ApplicationSignInManager\)](#) constructor, or to the default instance that exists within the context of Microsoft.Owin.

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

- ARFE/Controllers/ManageController.cs

5.24 ARFE.Models.ManageLoginsViewModel Class Reference

A model for viewing and reporting current and past user login information.

Properties

- `IList< UserLoginInfo > CurrentLogins [get, set]`
A property for the list of currently logged in users.
- `IList< AuthenticationDescription > OtherLogins [get, set]`
a property for the list of previously logged in users.

5.24.1 Detailed Description

A model for viewing and reporting current and past user login information.

5.24.2 Property Documentation

5.24.2.1 CurrentLogins

```
IList<UserLoginInfo> ARFE.Models.ManageLoginsViewModel.CurrentLogins [get], [set]
```

A property for the list of currently logged in users.

5.24.2.2 OtherLogins

```
IList<AuthenticationDescription> ARFE.Models.ManageLoginsViewModel.OtherLogins [get], [set]
```

a property for the list of previously logged in users.

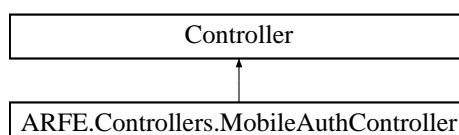
The documentation for this class was generated from the following file:

- ARFE/Models/ManageViewModels.cs

5.25 ARFE.Controllers.MobileAuthController Class Reference

The [MobileAuthController](#) serves as an API to the mobile application. The [Route] attribute allows this Controller to specify that to call any individual controller action within this class, you have to explicitly list the action method in the request URL.

Inheritance diagram for ARFE.Controllers.MobileAuthController:



Public Member Functions

- [MobileAuthController](#) ()

The default constructor either instantiates the TokenCache or gets a reference to it. The reference is stored in the _TokenCache private member.

- async Task< string > [RequestAuthToken](#) ()

Read the userName and password values from the JSON and create a unique Authorization Token string associated to the user credentials and stored in a cache. This method is only accessible via an Http POST request.

- async Task< string > [ListFiles](#) (int descriptor, int? pageNumber)

Return a list of files that a mobile user has access to denoted by an ownership descriptor value. The discriptor values are

- async Task< string > [DownloadFile](#) ()

Using the Authentication Token and the FileUri, both provided in the HTTP header as fileUri: [fileUri], token: [token] Authenticate the user from the provided token, and request to Azure Blob storage to get a downloadable file URI for the file. This method is only accessible via an Http GET request.

Public Attributes

- [ApplicationUserManager UserManager](#) => HttpContext.GetOwinContext().Get<[ApplicationUserManager](#)>()

A reference to the applications UserManager for password validation.

5.25.1 Detailed Description

The [MobileAuthController](#) serves as an API to the mobile application. The [Route] attribute allows this Controller to specify that to call any individual controller action within this class, you have to explicitly list the action method in the request URL.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 MobileAuthController()

```
ARFE.Controllers.MobileAuthController.MobileAuthController ( )
```

The default constructor either instantiates the TokenCache or gets a reference to it. The reference is stored in the _TokenCache private member.

5.25.3 Member Function Documentation

5.25.3.1 DownloadFile()

```
async Task<string> ARFE.Controllers.MobileAuthController.DownloadFile ( )
```

Using the Authentication Token and the FileUri, both provided in the HTTP header as fileUri: [fileUri], token: [token] Authenticate the user from the provided token, and request to Azure Blob storage to get a downloadable file URI for the file. This method is only accessible via an Http GET request.

Returns

The JSON response containing the downloadable file URI.

5.25.3.2 ListFiles()

```
async Task<string> ARFE.Controllers.MobileAuthController.ListFiles (
    int descriptor,
    int? pageNumber )
```

Return a list of files that a mobile user has access to denoted by an ownership descriptor value. The discriptor values are

- 0: ALL (All public and private files that the mobile user has access to.)
- 1: OWNED_ALL (All public and private files that the mobile user owns)
- 2: OWNED_PRIVATE (All of the private files owned by the mobile user)
- 3: OWNED_PUBLIC (All of the public files owned by the mobile user)
- 4: NOT_OWNED_PUBLIC (All of the public files not owned by the mobile user)

If a *pageNumber* value is supplied, the response list will be paginated with at most 10 results per page, and the correct page of values will be returned. This method is only accessible via an Http GET request.

Parameters

<i>descriptor</i>	The file descriptor value
<i>pageNumber</i>	The page index to return. If null, return all pages.

Returns

The JSON response of the requested file names to associated non-downloadable URIs.

5.25.3.3 RequestAuthToken()

```
async Task<string> ARFE.Controllers.MobileAuthController.RequestAuthToken ( )
```

Read the userName and password values from the JSON and create a unique Authorization Token string associated to the user credentials and stored in a cache. This method is only accessible via an Http POST request.

Returns

JSON response of the Authentication Token

5.25.4 Member Data Documentation

5.25.4.1 UserManager

```
ApplicationUserManager ARFE.Controllers.MobileAuthController.UserManager => HttpContext.Get<OwinContext>().Get<ApplicationUserManager>()
```

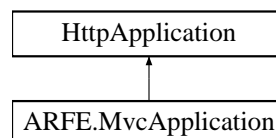
A reference to the applications UserManager for password validation.

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

- ARFE/Controllers/MobileAuthController.cs

5.26 ARFE.MvcApplication Class Reference

Inheritance diagram for ARFE.MvcApplication:



Protected Member Functions

- void **Application_Start** ()

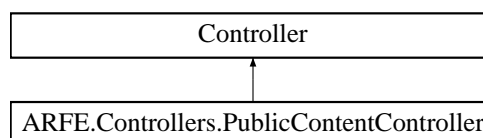
The documentation for this class was generated from the following file:

- ARFE/Global.asax.cs

5.27 ARFE.Controllers.PublicContentController Class Reference

A class derived from the Controller class that has all of the controller actions to display and interact with the public file content on the website.

Inheritance diagram for ARFE.Controllers.PublicContentController:



Public Member Functions

- ActionResult [Index](#) ()

The Index controller action is the default action called when the UserContent page is browsed to.

- ActionResult [Search](#) (int OrderType, string SearchType, string TextCriteria, string StartDateCrit, string EndDateCrit)

The controller action responsible for the filtering and ordering the files, based on the user input.

- ActionResult [ContentSelect](#) ([FileTypeModel](#) model, string downloadType)

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the content that the user owns but is publicly available. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only.

5.27.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to display and interact with the public file content on the website.

5.27.2 Member Function Documentation

5.27.2.1 ContentSelect()

```
ActionResult ARFE.Controllers.PublicContentController.ContentSelect (
    FileTypeModel model,
    string downloadType )
```

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the content that the user owns but is publicly available. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only.

Parameters

<i>model</i>	A FileTypeModel object representing the file selected.
<i>downloadType</i>	The selected file type for potential conversion

Returns

A view redirection to refresh the content of the Public Content page.

5.27.2.2 Index()

```
ActionResult ARFE.Controllers.PublicContentController.Index ( )
```

The Index controller action is the default action called when the UserContent page is browsed to.

Returns

A view to the "Index.cshtml" page in the Views/PublicContent/ folder.

5.27.2.3 Search()

```
ActionResult ARFE.Controllers.PublicContentController.Search (
    int OrderType,
    string SearchType,
    string TextCriteria,
    string StartDateCrit,
    string EndDateCrit )
```

The controller action responsible for the filtering and ordering the files, based on the user input.

Parameters

<i>OrderType</i>	An integer that references the Ordering Option enum
<i>SearchType</i>	A string containing "name" or "date". The denotes which type of search is begin submitted.

Parameters

<i>TextCriteria</i>	User inputted search string for the linq query
---------------------	--

Parameters

<i>StartDateCrit</i>	A string containing the lower bound date for the date search
----------------------	--

Parameters

<i>EndDateCrit</i>	A string containing the upper bound date for the date search
--------------------	--

Returns

A view to the "Index.cshtml" page in the Views/PublicContent/ folder.

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

- ARFE/Controllers/PublicContentController.cs

5.28 ARFE.Models.RegisterViewModel Class Reference

A model for registering a new user.

Properties

- string [Email](#) [get, set]
The user's email that they wish to register with their account.
- string [Password](#) [get, set]
The user's account password.
- string [ConfirmPassword](#) [get, set]
The user's account password, entered a second time to verify that the [Password](#) field was entered corretly.

5.28.1 Detailed Description

A model for registering a new user.

5.28.2 Property Documentation

5.28.2.1 ConfirmPassword

```
string ARFE.Models.RegisterViewModel.ConfirmPassword [get], [set]
```

The user's account password, entered a second time to verify that the [Password](#) field was entered corretly.

5.28.2.2 Email

```
string ARFE.Models.RegisterViewModel.Email [get], [set]
```

The user's email that they wish to register with their account.

5.28.2.3 Password

```
string ARFE.Models.RegisterViewModel.Password [get], [set]
```

The user's account password.

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

- ARFE/Models/AccountViewModels.cs

5.29 ARFE.Models.ResetPasswordViewModel Class Reference

A model to reset a user's password.

Properties

- string [Email](#) [get, set]
The user's email address that has been registered with their account.
- string [Password](#) [get, set]
The user's new password that they wish to set.
- string [ConfirmPassword](#) [get, set]
The user's new account password, entered a second time to verify that the [Password](#) field was entered corretly.
- string [Code](#) [get, set]
The two-factor authentication code that had been sent to the user that will allow them to reset their password without knowing their old password.

5.29.1 Detailed Description

A model to reset a user's password.

5.29.2 Property Documentation

5.29.2.1 Code

```
string ARFE.Models.ResetPasswordViewModel.Code [get], [set]
```

The two-factor authentication code that had been sent to the user that will allow them to reset their password without knowing their old password.

5.29.2.2 ConfirmPassword

```
string ARFE.Models.ResetPasswordViewModel.ConfirmPassword [get], [set]
```

The user's new account password, entered a second time to verify that the [Password](#) field was entered corretly.

5.29.2.3 Email

```
string ARFE.Models.ResetPasswordViewModel.Email [get], [set]
```

The user's email address that has been registered with their account.

5.29.2.4 Password

```
string ARFE.Models.ResetPasswordViewModel.Password [get], [set]
```

The user's new password that they wish to set.

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

- ARFE/Models/AccountViewModels.cs

5.30 ARFE.RouteConfig Class Reference

This class is for defining the valid url rout patterns for the application.

Static Public Member Functions

- static void [RegisterRoutes](#) (RouteCollection routes)
The configuration for url route patterns used within the application.

5.30.1 Detailed Description

This class is for defining the valid url rout patterns for the application.

5.30.2 Member Function Documentation

5.30.2.1 RegisterRoutes()

```
static void ARFE.RouteConfig.RegisterRoutes (  
    RouteCollection routes ) [static]
```

The configuration for url route patterns used within the application.

Parameters

<i>routes</i>	The collection of acceptable url route patterns.
---------------	--

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

- ARFE/App_Start/RouteConfig.cs
-

5.31 ARFE.SearchQueries Class Reference

This class contains all of the linq queries for the search and order functionality.

Static Public Member Functions

- static List< [Common.FileUIInfo](#) > [FilterByFileNameOrderByNameAscending](#) (List< [Common.FileUIInfo](#) > fileList, string SearchCriteria)
This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them in alphabetical order and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterByFileNameOrderByNameDescending](#) (List< [Common.FileUIInfo](#) > fileList, string SearchCriteria)
This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them in reverse alphabetical order and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterByFileNameOrderByNameAscending](#) (List< [Common.FileUIInfo](#) > fileList, string SearchCriteria)
This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them by date ascending and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterByFileNameOrderByNameDescending](#) (List< [Common.FileUIInfo](#) > fileList, string SearchCriteria)
This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them by date in descending order and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterDateOrderByNameAscending](#) (List< [Common.FileUIInfo](#) > fileList, DateTime startDate, DateTime endDate)
This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them in alphabetical order and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterDateOrderByNameDescending](#) (List< [Common.FileUIInfo](#) > fileList, DateTime startDate, DateTime endDate)
This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them in reverse alphabetical order and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterDateOrderByNameAscending](#) (List< [Common.FileUIInfo](#) > fileList, DateTime startDate, DateTime endDate)
This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them oldest first and return it in a list.
- static List< [Common.FileUIInfo](#) > [FilterDateOrderByNameDescending](#) (List< [Common.FileUIInfo](#) > fileList, DateTime startDate, DateTime endDate)
This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them by newest first and return it in a list.

5.31.1 Detailed Description

This class contains all of the linq queries for the search and order functionality.

5.31.2 Member Function Documentation

5.31.2.1 FilterByFileNameOrderByNameAscending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterByFileNameOrderByNameAscending (
    List< Common.FileUIInfo > fileList,
    string SearchCriteria ) [static]
```

This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them by date ascending and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered
<i>SearchCriteria</i>	A string holding the search criteria

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.2 FilterByFileNameOrderByDateDescending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterByFileNameOrderByDateDescending (  
    List< Common.FileUIInfo > fileList,  
    string SearchCriteria ) [static]
```

This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them by date in descending order and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered
<i>SearchCriteria</i>	A string holding the search criteria

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.3 FilterByFileNameOrderByNameAscending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterByFileNameOrderByNameAscending (  
    List< Common.FileUIInfo > fileList,  
    string SearchCriteria ) [static]
```

This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them in alphabetical order and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered
<i>SearchCriteria</i>	A string holding the search criteria

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.4 FilterByFileNameOrderByNameDescending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterByFileNameOrderByNameDescending (
    List< Common.FileUIInfo > fileList,
    string SearchCriteria ) [static]
```

This function takes in a list of fileUIInfo and search string. It uses linq to find the files with the name containing that string, order them in reverse alphabetical order and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered
<i>SearchCriteria</i>	A string holding the search criteria

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.5 FilterDateOrderByDateAscending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterDateOrderByDateAscending (
    List< Common.FileUIInfo > fileList,
    DateTime startDate,
    DateTime endDate ) [static]
```

This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them oldest first and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered.
<i>startDate</i>	A datetime object holding the lower bound date.
<i>endDate</i>	A datetime object holding the upper bound date.

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.6 FilterDateOrderByDateDescending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterDateOrderByDateDescending (
    List< Common.FileUIInfo > fileList,
```

```
DateTime startDate,  
DateTime endDate ) [static]
```

This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them by newest first and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered.
<i>startDate</i>	A datetime object holding the lower bound date.
<i>endDate</i>	A datetime object holding the upper bound date.

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.7 FilterDateOrderByNameAscending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterDateOrderByNameAscending (  
    List< Common.FileUIInfo > fileList,  
    DateTime startDate,  
    DateTime endDate ) [static]
```

This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them in alphebetical order and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered.
<i>startDate</i>	A datetime object holding the lower bound date.
<i>endDate</i>	A datetime object holding the upper bound date.

Returns

A filtered and ordered list of FileUIInfo objects

5.31.2.8 FilterDateOrderByNameDescending()

```
static List<Common.FileUIInfo> ARFE.SearchQueries.FilterDateOrderByNameDescending (  
    List< Common.FileUIInfo > fileList,  
    DateTime startDate,  
    DateTime endDate ) [static]
```

This function takes in a list of fileUIInfo, a start and end date. It uses linq to find the files between the start and end date, then order them in reverse alphebetical order and return it in a list.

Parameters

<i>fileList</i>	A list of FileUIInfo objects that need to be filtered and ordered.
<i>startDate</i>	A datetime object holding the lower bound date.
<i>endDate</i>	A datetime object holding the upper bound date.

Returns

A filtered and ordered list of FileUIInfo objects

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

- ARFE/SearchQueries.cs

5.32 ARFE.Models.SendCodeViewModel Class Reference

A model for sending a two-factor authentication message through a registered third party two-factor authentication provider.

Properties

- string [SelectedProvider](#) [get, set]
The name of the user-seleted third party two-factor authentication provider.
- ICollection< System.Web.Mvc.SelectListItem > [Providers](#) [get, set]
The list of available third party two-factor authentication providers.
- string [ReturnUrl](#) [get, set]
The redirection Url to send the user to after having verified their account information via the two-factor authentication code.
- bool [RememberMe](#) [get, set]
The user's preference to have their account automatically verified, when possible.

5.32.1 Detailed Description

A model for sending a two-factor authentication message through a registered third party two-factor authentication provider.

5.32.2 Property Documentation

5.32.2.1 Providers

```
ICollection<System.Web.Mvc.SelectListItem> ARFE.Models.SendCodeViewModel.Providers [get],
[set]
```

The list of available third party two-factor authentication providers.

5.32.2.2 RememberMe

```
bool ARFE.Models.SendCodeViewModel.RememberMe [get], [set]
```

The user's preference to have their account automatically verified, when possible.

5.32.2.3 returnUrl

```
string ARFE.Models.SendCodeViewModel.ReturnUrl [get], [set]
```

The redirection Url to send the user to after having verified their account information via the two-factor authentication code.

5.32.2.4 SelectedProvider

```
string ARFE.Models.SendCodeViewModel.SelectedProvider [get], [set]
```

The name of the user-seleted third party two-factor authentication provider.

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

- ARFE/Models/AccountViewModels.cs

5.33 ARFE.Models.SetPasswordViewModel Class Reference

A model for registering a user's password upon account creation.

Properties

- string [NewPassword](#) [get, set]
The property field for the user's new password that they wish to set.
- string [ConfirmPassword](#) [get, set]
The property field for the confirmed re-entry of the user's new password that they wish to set.

5.33.1 Detailed Description

A model for registering a user's password upon account creation.

5.33.2 Property Documentation

5.33.2.1 ConfirmPassword

```
string ARFE.Models.SetPasswordViewModel.ConfirmPassword [get], [set]
```

The property field for the confirmed re-entry of the user's new password that they wish to set.

5.33.2.2 NewPassword

```
string ARFE.Models.SetPasswordViewModel.NewPassword [get], [set]
```

The property field for the user's new password that they wish to set.

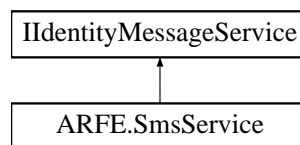
The documentation for this class was generated from the following file:

- ARFE/Models/ManageViewModels.cs

5.34 ARFE.SmsService Class Reference

A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via SMS text messaging. This implementation is currently empty as no third party SMS text messaging verification is currently supported.

Inheritance diagram for ARFE.SmsService:



Public Member Functions

- Task [SendAsync](#) (IdentityMessage message)

This method is required to be overwritten by the inheritance pattern of the Microsoft.AspNet.Identity.IIdentityMessageService class. As no third party SMS text messaging verification is currently supported, the method just returns 0 to the caller.

5.34.1 Detailed Description

A class derived from the Microsoft.AspNet.Identity.IIdentityMessageService class to implement third party account verification via SMS text messaging. This implementation is currently empty as no third party SMS text messaging verification is currently supported.

5.34.2 Member Function Documentation

5.34.2.1 SendAsync()

```
Task ARFE.SmsService.SendAsync (
    IdentityMessage message )
```

This method is required to be overwritten by the inheritance pattern of the Microsoft.AspNet.Identity.IIdentityMessageService class. As no third party SMS text messaging verification is currently supported, the method just returns 0 to the caller.

Parameters

<i>message</i>	The message to send via the registered SMS provider.
----------------	--

Returns

0

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

- ARFE/App_Start/IdentityConfig.cs

5.35 ARFE.Startup Class Reference

The application startup class is called upon launching the application.

Public Member Functions

- void [ConfigureAuth](#) (IAppBuilder app)
Application startup configuration via Microsoft.AspNet.Identity For more information on configuring authentication, please visit <https://go.microsoft.com/fwlink/?LinkId=301864>
- void [Configuration](#) (IAppBuilder app)
Method called automatically upon startup. Initial app setup logic and configuration goes here.
- void [TimerInit](#) ()
Create a timer that will fire a clean-up event every 15 minutes.

5.35.1 Detailed Description

The application startup class is called upon launching the application.

5.35.2 Member Function Documentation

5.35.2.1 Configuration()

```
void ARFE.Startup.Configuration (
    IAppBuilder app )
```

Method called automatically upon startup. Initial app setup logic and configuration goes here.

Parameters

<i>app</i>	The AE web app
------------	----------------

5.35.2.2 ConfigureAuth()

```
void ARFE.Startup.ConfigureAuth (
    IApplicationBuilder app )
```

Application startup configuration via Microsoft.AspNetCore.Identity For more information on configuring authentication, please visit <https://go.microsoft.com/fwlink/?LinkId=301864>

Parameters

<i>app</i>	The application being configured.
------------	-----------------------------------

5.35.2.3 TimerInit()

```
void ARFE.Startup.TimerInit ( )
```

Create a timer that will fire a clean-up event every 15 minutes.

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

- ARFE/App_Start/Startup.Auth.cs
- ARFE/Startup.cs

5.36 Common.SupportedFileTypes Class Reference

The list of file type extensions that are supported by the file conversion software that this application uses.

Static Public Attributes

- static List< string > [FileList](#) => new List<string> { ".dae", ".fbx", ".obj", ".ply", ".stl" }
The list of supported file types.

5.36.1 Detailed Description

The list of file type extensions that are supported by the file conversion software that this application uses.

5.36.2 Member Data Documentation

5.36.2.1 FileList

```
List<string> Common.SupportedFileTypes.FileList => new List<string> { ".dae", ".fbx", ".obj", ".ply", ".stl" } [static]
```

The list of supported file types.

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

- Common/CommonClasses.cs

5.37 AuthenticationTokenCache.TokenCache Class Reference

A class to manage tokens that are associated to user login information. This is used for mobile user authentication by verifying tokens, that the user explicitly requests, to that user's log in credentials. This system of token passing makes for simpler and more secure communication between the mobile application and the web service.

Public Member Functions

- string [GenerateToken](#) (string userName, string password)
Given a user's user name and password credentials, generate a new authentication token associated with those credentials. The token will be valid for two hours. The users credentials are not validated at this point, they are only stored. If incorrect credentials are supplied then the application will catch that internally via ASP.NET Identity.
- Tuple< string, string > [ValidateToken](#) (string token)
Verify that a token exists within the current state of the in-memory storage. If it does and the token has yet to expire, return a tuple of the associated user name and password that the token was created with. If it does not exist or the token has expired, the response is null.

Static Public Member Functions

- static [TokenCache Init](#) ()
A static initialization method that calls a private constructor internally to ensure a static single instance of this object is maintained throughout the run-time life of the application.

5.37.1 Detailed Description

A class to manage tokens that are associated to user login information. This is used for mobile user authentication by verifying tokens, that the user explicitly requests, to that user's log in credentials. This system of token passing makes for simpler and more secure communication between the mobile application and the web service.

5.37.2 Member Function Documentation

5.37.2.1 GenerateToken()

```
string AuthenticationTokenCache.TokenCache.GenerateToken (
    string userName,
    string password )
```

Given a user's user name and password credentials, generate a new authentication token associated with those credentials. The token will be valid for two hours. The users credentials are not validated at this point, they are only stored. If incorrect credentials are supplied then the application will catch that internally via ASP.NET Identity.

Parameters

<i>userName</i>	The user provided user name.
<i>password</i>	The user provided password.

Returns

A new base64 encoded string of unique bytes.

5.37.2.2 Init()

```
static TokenCache AuthenticationTokenCache.TokenCache.Init ( ) [static]
```

A static initialization method that calls a private constructor internally to ensure a static single instance of this object is maintained throughout the run-time life of the application.

Returns

The once-constructed instance of the [TokenCache](#).

5.37.2.3 ValidateToken()

```
Tuple<string, string> AuthenticationTokenCache.TokenCache.ValidateToken (
    string token )
```

Verify that a token exists within the current state of the in-memory storage. If it does and the token has yet to expire, return a tuple of the associated user name and password that the token was created with. If it does not exist or the token has expired, the response is null.

Parameters

<i>token</i>	The randomized string provided by the GenerateToken(string, string) method.
--------------	---

Returns

- A tuple of the associated user name and password - the token is valid.
- Null - the token is invalid or expired.

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

- [AuthenticationTokenCache/TokenCache.cs](#)
-

5.38 AuthenticationTokenCache.Tests.TokenCacheTests Class Reference

The class containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

Public Member Functions

- void [Test_PrivateConstructor](#) ()
Test that the constructor of the [TokenCache](#) is in fact private. Calling `Activator.CreateInstance` will try to call the object constructor but will throw a `MissingMethodException` since one hasn't been made publicly accessible due to the fact that it's a singleton cache.
- void [Test_Init](#) ()
Test that the publicly available static `Init()` method creates an instance of the [TokenCache](#) object.
- void [Test_Insert](#) ()
Test that inserting user info to the cache creates a non-empty string token.
- void [Test_ValidateSingle](#) ()
Test that the token produced for a users information is able to successfully retrieve the same user information.
- void [Test_Singleton](#) ()
Test that creating two separate variables of type [TokenCache](#) with the static `Init()` method sets each variable to the same static instance of the [TokenCache](#) object.
- void [Test_BadTokenReturnsNull](#) ()
Test that users information can be supplied but providing an invalid token will not return any user's information.
- void [Test_ValidateMultiple](#) ()
Test that the cache successfully handles more than one user's information.
- void [Test_MultipleTokensToSameUser](#) ()
Test that a single user can get multiple tokens that will all remain valid.

5.38.1 Detailed Description

The class containing all tests that pertain to the [AuthenticationTokenCache.TokenCache](#) class and it's methods.

5.38.2 Member Function Documentation

5.38.2.1 [Test_BadTokenReturnsNull\(\)](#)

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_BadTokenReturnsNull ( )
```

Test that users information can be supplied but providing an invalid token will not return any user's information.

5.38.2.2 [Test_Init\(\)](#)

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_Init ( )
```

Test that the publicly available static `Init()` method creates an instance of the [TokenCache](#) object.

5.38.2.3 Test_Insert()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_Insert ( )
```

Test that inserting user info to the cache creates a non-empty string token.

5.38.2.4 Test_MultipleTokensToSameUser()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_MultipleTokensToSameUser ( )
```

Test that a single user can get multiple tokens that will all remain valid.

5.38.2.5 Test_PrivateConstructor()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_PrivateConstructor ( )
```

Test that the constructor of the [TokenCache](#) is in fact private. Calling `Activator.CreateInstance` will try to call the object constructor but will throw a `MissingMethodException` since one hasn't been made publicly accessible due to the fact that it's a singleton cache.

5.38.2.6 Test_Singleton()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_Singleton ( )
```

Test that creating two separate variables of type [TokenCache](#) with the static `Init()` method sets each variable to the same static instance of the [TokenCache](#) object.

5.38.2.7 Test_ValidateMultiple()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_ValidateMultiple ( )
```

Test that the cache successfully handles more than one user's information.

5.38.2.8 Test_ValidateSingle()

```
void AuthenticationTokenCache.Tests.TokenCacheTests.Test_ValidateSingle ( )
```

Test that the token produced for a users information is able to successfully retrieve the same user information.

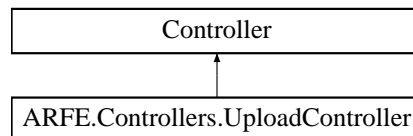
The documentation for this class was generated from the following file:

- `AuthTokenCache.Tests/TokenCacheTests.cs`
-

5.39 ARFE.Controllers.UploadController Class Reference

A class derived from the Controller class that has all of the controller actions to manage user file uploads.

Inheritance diagram for ARFE.Controllers.UploadController:



Public Member Functions

- ActionResult [Index](#) ()
The call to load the index page of the file upload section of the website.
- ActionResult [UploadFile](#) ()
This method is functionally the same as calling the [Index](#) method. If Upload() is called as a get action - do nothing. The [HttpGet] assembly tag is used to register this method as eligible for GET requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.
- ActionResult [UploadFile](#) (HttpPostedFileBase baseFile, HttpPostedFileBase matFile, bool publicFile, string altFileName, string fileDescription)
Allows a user to upload a file from their computer via the web browser. The file is checked for validity and if found valid, is saved to temporary local storage and makes a request to upload the file to Azure Blob storage. If the file is not valid or already exists in blob storage, an error message is displayed to the user. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.
- ActionResult [OverWriteUpload](#) (string overWrite, string uploadFileName)
This controller action is only to be called after the [UploadFile\(HttpPostedFileBase, HttpPostedFileBase, bool, string, string\)](#) controller action has been called and has returned an error that the file already exists in Azure Blob storage. This controller action gives the user the option to overwrite the previously existing file with the current one. If the value of overWrite is true, this method performs the same action as [UploadFile\(HttpPostedFileBase, HttpPostedFileBase, bool, string, string\)](#). The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

5.39.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to manage user file uploads.

5.39.2 Member Function Documentation

5.39.2.1 Index()

```
ActionResult ARFE.Controllers.UploadController.Index ( )
```

The call to load the index page of the file upload section of the website.

Returns

A view that refreshes the web page to the current url location.

5.39.2.2 OverWriteUpload()

```
ActionResult ARFE.Controllers.UploadController.OverWriteUpload (
    string overWrite,
    string uploadFileName )
```

This controller action is only to be called after the [UploadFile\(HttpPostedFileBase, HttpPostedFileBase, bool, string, string\)](#) controller action has been called and has returned an error that the file already exists in Azure Blob storage. This controller action gives the user the option to overwrite the previously existing file with the current one. If the value of *overWrite* is true, this method performs the same action as [UploadFile\(HttpPostedFileBase, HttpPostedFileBase, bool, string, string\)](#). The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>overWrite</i>	A boolean status indicating whether a file that has the same name as the new one should be overwritten in order to save the new one.
<i>uploadFileName</i>	The name of the file saved in temporary storage from the UploadFile(HttpPostedFileBase, HttpPostedFileBase, bool, string, string) call prior. This will either be the <i>baseFile.FileName</i> or the <i>altFileName</i> if one was provided.

Returns

The current web page view with a message as to the status of the file upload.

5.39.2.3 UploadFile() [1/2]

```
ActionResult ARFE.Controllers.UploadController.UploadFile ( )
```

This method is functionally the same as calling the [Index](#) method. If Upload() is called as a get action - do nothing. The [HttpGet] assembly tag is used to register this method as eligible for GET requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Returns

A view that refreshes the web page to the current url location.

5.39.2.4 UploadFile() [2/2]

```
ActionResult ARFE.Controllers.UploadController.UploadFile (
    HttpPostedFileBase baseFile,
    HttpPostedFileBase matFile,
    bool publicFile,
    string altFileName,
    string fileDescription )
```

Allows a user to upload a file from their computer via the web browser. The file is checked for validity and if found valid, is saved to temporary local storage and makes a request to upload the file to Azure Blob storage. If the file is not valid or already exists in blob storage, an error message is displayed to the user. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>baseFile</i>	The file to be uploaded to Azure Blob storage.
<i>matFile</i>	If <i>baseFile</i> is a '.obj' file, this may be an accompanying '.mat' materials file.
<i>publicFile</i>	A boolean indicator for if the user wants this file to be publicly available via the website interface.
<i>altFileName</i>	If supplied, will be the alternate file name that the uploaded file will be saved as.
<i>fileDescription</i>	A user-provided description of the file.

Returns

The current web page view with a message as to the status of the file upload.

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

- ARFE/Controllers/UploadController.cs

5.40 ARFE.UploadedFileCache Class Reference

A singleton class, constructed via the [GetInstance](#) method call. This class serves as a cache for tracking the location, details about, and automatic removal of files that are saved for intermediate storage between being uploaded to Azure blob storage, or converted to some other file type pending a download.

Public Member Functions

- bool [SaveFile](#) (HttpPostedFileBase file, string userName, string fileName, string altFileName="", string description="", bool isPublic=false)
Create a unique sub-directory for the file to save the file in. Information about the file associated with the unique sub-directory will be added to the thread-safe s_UsedGuids and s_FileDataByGuid members for tracking lifespan and information regarding the temporary file(s) stored.
- Guid [SaveFile](#) (CloudBlockBlob file, string userName, string fileName)
Create a unique sub-directory for the file to save the file in. Information about the file associated with the unique sub-directory will be added to the thread-safe s_UsedGuids and s_FileDataByGuid members for tracking lifespan and information regarding the temporary file(s) stored.

- void [RemoveFileTracking](#) (Guid fileGuid)
For a provided Guid representing a sub-directory, remove the Guid from s_UsedGuids and remove the Guid:FileData record from s_FileDataByGuid. An immediate removal is not guaranteed from either members due to their thread-safe design. Removal is retried at most 50 times each.
- Guid [FindContainingFolderGUIDForFile](#) (string userName, string uploadFileName)
Check each FileData(Guid, string, string, string, string, bool) in s_FileDataByGuid values. Attempt to match user name and file name. If multiple matches are found, which should almost never be the case, the most recent is assumed.
- string [GetFileDescription](#) (string userName, string uploadFileName)
Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. From the Guid, get the FileData from s_FileDataByGuid. Use the FileData to access the user provided file description.
- bool [IsSavedFilePublic](#) (string userName, string uploadFileName)
Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. From the Guid, get the FileData from s_FileDataByGuid. Use the FileData to access the user provided public/private flag.
- bool [DeleteAndRemoveFile](#) (string userName, string uploadFileName)
Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. Use the Guid to delete the sub-directory and all of its contents. Call [RemoveFileTracking\(Guid\)](#) to free the Guid from tracking.
- void [MarkForDelete](#) (string userName, string uploadFileName)
*Mark a FileData(Guid, string, string, string, string, bool) record for deletion by calling FileData.MarkForDelete to set its expiration time to some time in the past.
The next time that entries are examined for expiration, it should be deleted.*

Static Public Member Functions

- static [UploadedFileCache GetInstance](#) ()
A publicly available method for creating a single instance of this cache if one doesn't already exist, or getting a reference to the single instance.
- static void [DeleteOldFiles](#) ()
A publicly available static method for cleaning the root upload directory. This method is called at timed intervals from Startup.Timer_Elapsed(object, System.Timers.ElapsedEventArgs).

Properties

- string [BasePath](#) [get, set]
The publicly accessible property for the root directory for file upload/download intermediate storage. This property can only be set once and upon being set, purges the supplied directory path of all unnecessary files and folders.

5.40.1 Detailed Description

A singleton class, constructed via the [GetInstance](#) method call. This class serves as a cache for tracking the location, details about, and automatic removal of files that are saved for intermediate storage between being uploaded to Azure blob storage, or converted to some other file type pending a download.

5.40.2 Member Function Documentation

5.40.2.1 DeleteAndRemoveFile()

```
bool ARFE.UploadedFileCache.DeleteAndRemoveFile (
    string userName,
    string uploadFileName )
```

Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. Use the Guid to delete the sub-directory and all of its contents. Call [RemoveFileTracking\(Guid\)](#) to free the Guid from tracking.

Parameters

<i>userName</i>	The name of the user who saved the file.
<i>uploadFileName</i>	The name of the file that was saved.

Returns

- True : The sub-directory that the file is in was deleted.
- False : The sub-directory was not deleted, but was marked for deletion.

5.40.2.2 DeleteOldFiles()

```
static void ARFE.UploadedFileCache.DeleteOldFiles ( ) [static]
```

A publicly available static method for cleaning the root upload directory. This method is called at timed intervals from `Startup.Timer_Elapsed(object, System.Timers.ElapsedEventArgs)`.

5.40.2.3 FindContainingFolderGUIDForFile()

```
Guid ARFE.UploadedFileCache.FindContainingFolderGUIDForFile (
    string userName,
    string uploadFileName )
```

Check each `FileData(Guid, string, string, string, string, bool)` in `s_FileDataByGuid` values. Attempt to match user name and file name. If multiple matches are found, which should almost never be the case, the most recent is assumed.

Parameters

<i>userName</i>	The name of the user who owns the file.
<i>uploadFileName</i>	The file name (or alternate file name) that the file was saved as.

Returns

- Guid : The Guid that is the name of the sub-directory that the file is in.
-

- Guid.Empty : File not found.

5.40.2.4 GetFileDescription()

```
string ARFE.UploadedFileCache.GetFileDescription (
    string userName,
    string uploadFileName )
```

Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. From the Guid, get the FileData from `s_FileDataByGuid`. Use the FileData to access the user provided file description.

Parameters

<i>userName</i>	The name of the user who saved the file.
<i>uploadFileName</i>	The name of the file that was saved.

Returns

- Description : The user-provided description of the file.
- String.Empty : File Not Found

5.40.2.5 GetInstance()

```
static UploadedFileCache ARFE.UploadedFileCache.GetInstance ( ) [static]
```

A publicly available method for creating a single instance of this cache if one doesn't already exist, or getting a reference to the single instance.

Returns

A reference to the single instance of this cache class.

5.40.2.6 IsSavedFilePublic()

```
bool ARFE.UploadedFileCache.IsSavedFilePublic (
    string userName,
    string uploadFileName )
```

Get the Guid that is the sub-directory name of the file from the [FindContainingFolderGUIDForFile\(string, string\)](#) method. From the Guid, get the FileData from `s_FileDataByGuid`. Use the FileData to access the user provided public/private flag.

Parameters

<i>userName</i>	The name of the user who saved the file.
<i>uploadFileName</i>	The name of the file that was saved.

Returns

- **IsPublic** : The user-provided public/private flag value.
- **Default** : False, meaning private.

5.40.2.7 MarkForDelete()

```
void ARFE.UploadedFileCache.MarkForDelete (
    string userName,
    string uploadFileName )
```

Mark a `FileData(Guid, string, string, string, string, bool)` record for deletion by calling `FileData.MarkForDelete` to set its expiration time to some time in the past.

The next time that entries are examined for expiration, it should be deleted.

Parameters

<i>userName</i>	The name of the user who saved the file.
<i>uploadFileName</i>	The name of the file that was saved.

5.40.2.8 RemoveFileTracking()

```
void ARFE.UploadedFileCache.RemoveFileTracking (
    Guid fileGuid )
```

For a provided `Guid` representing a sub-directory, remove the `Guid` from `s_UsedGuids` and remove the `Guid:FileData` record from `s_FileDataByGuid`. An immediate removal is not guaranteed from either members due to their thread-safe design. Removal is retried at most 50 times each.

Parameters

<i>fileGuid</i>	The <code>Guid</code> to stop tracking.
-----------------	---

5.40.2.9 SaveFile() [1/2]

```
bool ARFE.UploadedFileCache.SaveFile (
    HttpPostedFileBase file,
```

```

    string userName,
    string fileName,
    string altFileName = "",
    string description = "",
    bool isPublic = false )

```

Create a unique sub-directory for the file to save the file in. Information about the file associated with the unique sub-directory will be added to the thread-safe `s_UsedGuids` and `s_FileDataByGuid` members for tracking lifespan and information regarding the temporary file(s) stored.

Parameters

<i>file</i>	The file uploaded to the website via HTTP POST in the Controllers.UploadController.UploadFile(HttpPostedFileBase, HttpPostedFileBase, bool, string, string) method or the Controllers.UploadController.UploadFile method.
<i>userName</i>	The name of the user uploading the file.
<i>fileName</i>	The name of the file being uploaded.
<i>altFileName</i>	An alternative file name that the file should be saved as instead, provided by the user.
<i>description</i>	A description of the file provided by user.
<i>isPublic</i>	A flag indicating if the file should be uploaded to the public blob container of the user's private blob container.

Returns

- True: The file was uploaded successfully.
- False: The file failed to upload.

5.40.2.10 SaveFile() [2/2]

```

Guid ARFE.UploadedFileCache.SaveFile (
    CloudBlockBlob file,
    string userName,
    string fileName )

```

Create a unique sub-directory for the file to save the file in. Information about the file associated with the unique sub-directory will be added to the thread-safe `s_UsedGuids` and `s_FileDataByGuid` members for tracking lifespan and information regarding the temporary file(s) stored.

Parameters

<i>file</i>	The file being downloaded from the BlobManager.ConvertBlobToBlob(CloudBlockBlob, string, string, string, string) method.
<i>userName</i>	The name of the user uploading the file.
<i>fileName</i>	The name of the file being uploaded.

Returns

- Guid : The Guid that is the name of the sub-directory that the file was saved to.
- Guid.Empty : The file could not be saved.

5.40.3 Property Documentation

5.40.3.1 BasePath

```
string ARFE.UploadedFileCache.BasePath [get], [set]
```

The publicly accessible property for the root directory for file upload/download intermediate storage. This property can only be set once and upon being set, purges the supplied directory path of all unnecessary files and folders.

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

- ARFE/UploadedFileCache.cs

5.41 ARFE.Tests.UploadedFileCacheTests Class Reference

Summary description for [UploadedFileCacheTests](#)

Public Member Functions

- void [Test_PrivateConstructor](#) ()
Test that the constructor of the [UploadedFileCache](#) is in fact private. Calling `Activator.CreateInstance` will try to call the object constructor but will throw a `MissingMethodException` since one hasn't been made publicly accessible due to the fact that it's a singleton cache.
- void [Test_Init](#) ()
Test that the publicly available static init method, [UploadedFileCache.GetInstance](#), creates an instance of the [UploadedFileCache](#) object.

Properties

- TestContext [TestContext](#) [get, set]
Gets or sets the test context which provides information about and functionality for the current test run. /summary>

5.41.1 Detailed Description

Summary description for [UploadedFileCacheTests](#)

5.41.2 Member Function Documentation

5.41.2.1 Test_Init()

```
void ARFE.Tests.UploadedFileCacheTests.Test_Init ( )
```

Test that the publicly available static init method, [UploadedFileCache.GetInstance](#), creates an instance of the [UploadedFileCache](#) object.

5.41.2.2 Test_PrivateConstructor()

```
void ARFE.Tests.UploadedFileCacheTests.Test_PrivateConstructor ( )
```

Test that the constructor of the [UploadedFileCache](#) is in fact private. Calling `Activator.CreateInstance` will try to call the object constructor but will throw a `MissingMethodException` since one hasn't been made publicly accessible due to the fact that it's a singleton cache.

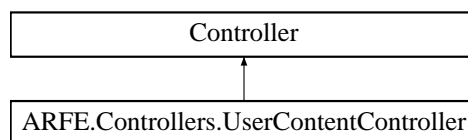
The documentation for this class was generated from the following file:

- ARFE.Tests/UploadedFileCacheTests.cs

5.42 ARFE.Controllers.UserContentController Class Reference

A class derived from the `Controller` class that has all of the controller actions to display and interact with the user's file content on the website.

Inheritance diagram for `ARFE.Controllers.UserContentController`:



Public Member Functions

- ActionResult [Index](#) ()

The `Index` controller action is the default action called when the `UserContent` page is browsed to. The `[Authorize]` assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

- ActionResult [PrivateContentSelect](#) ([FileTypeModel](#) model, string downloadType)

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the user's privately owned content. The `[HttpPost]` assembly tag is used to register this method as eligible for POST requests only. The `[Authorize]` assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

- ActionResult [PublicContentSelect](#) ([FileTypeModel](#) model, string downloadType)

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the content that the user owns but is publicly available. The `[HttpPost]` assembly tag is used to register this method as eligible for POST requests only. The `[Authorize]` assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

- ActionResult [PrivateSearch](#) (int prOrderType, string prSearchType, string prTextCriteria, string prStartDate↵ Crit, string prEndDateCrit)

The Index controller action is called when the user in doing a unique search of the private files The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

- ActionResult [PublicSearch](#) (int puOrderType, string puSearchType, string puTextCriteria, string puStartDate↵ Crit, string puEndDateCrit)

The Index controller action is called when the user in doing a unique search of the public files The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

5.42.1 Detailed Description

A class derived from the Controller class that has all of the controller actions to dispaly and interact with the user's file content on the website.

5.42.2 Member Function Documentation

5.42.2.1 Index()

```
ActionResult ARFE.Controllers.UserContentController.Index ( )
```

The Index controller action is the default action called when the UserContent page is browsed to. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Returns

A view to the "Index.cshtml" page in the Views/UserContent/ folder.

5.42.2.2 PrivateContentSelect()

```
ActionResult ARFE.Controllers.UserContentController.PrivateContentSelect (
    FileTypeModel model,
    string downloadType )
```

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the user's privately owned content. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>model</i>	A FileTypeModel object representing the file selected.
<i>downloadType</i>	The selected file type for potential conversion

Returns

A view redirection to refresh the content of the User Content page.

5.42.2.3 PrivateSearch()

```
ActionResult ARFE.Controllers.UserContentController.PrivateSearch (
    int prOrderType,
    string prSearchType,
    string prTextCriteria,
    string prStartDateCrit,
    string prEndDateCrit )
```

The Index controller action is called when the user is doing a unique search of the private files. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>prOrderType</i>	An integer that designates the ordering style. Value corresponds to Ordering Option Enum
<i>prSearchType</i>	A string containing the type of search the user is trying to do. Value is either "name" or "date"
<i>prTextCriteria</i>	The user inputted string to filter files that only hold that name
<i>prStartDateCrit</i>	String containing the lower bound date for the date search. Format: "YYYY/MM/DD"
<i>prEndDateCrit</i>	String containing the upper bound date for the date search. Format: "YYYY/MM/DD"

Returns

A view to the "Index.cshtml" page in the Views/UserContent/ folder.

5.42.2.4 PublicContentSelect()

```
ActionResult ARFE.Controllers.UserContentController.PublicContentSelect (
    FileTypeModel model,
    string downloadType )
```

The controller action responsible for the sub-menu button clicks on the menu that corresponds to the content that the user owns but is publicly available. The [HttpPost] assembly tag is used to register this method as eligible for POST requests only. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>model</i>	A FileTypeModel object representing the file selected.
<i>downloadType</i>	The selected file type for potential conversion

Returns

A view redirection to refresh the content of the User Content page.

5.42.2.5 PublicSearch()

```
ActionResult ARFE.Controllers.UserContentController.PublicSearch (
    int puOrderType,
    string puSearchType,
    string puTextCriteria,
    string puStartDateCrit,
    string puEndDateCrit )
```

The Index controller action is called when the user is doing a unique search of the public files. The [Authorize] assembly tag is used in collaboration with ASP.NET Identity. If the action is attempted to be browsed to without the browser being correctly signed in with Identity, the request is denied.

Parameters

<i>puOrderType</i>	An integer that designates the ordering style. Value corresponds to Ordering Option Enum
<i>puSearchType</i>	A string containing the type of search the user is trying to do. Value is either "name" or "date"
<i>puTextCriteria</i>	The user inputted string to filter files that only hold that name
<i>puStartDateCrit</i>	String containing the lower bound date for the date search. Format: "YYYY/MM/DD"
<i>puEndDateCrit</i>	String containing the upper bound date for the date search. Format: "YYYY/MM/DD"

Returns

A view to the "Index.cshtml" page in the Views/UserContent/ folder.

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

- ARFE/Controllers/UserContentController.cs

5.43 ARFE.Models.VerifyCodeViewModel Class Reference

A model for verifying a two-factor authentication code that has been sent to a user.

Properties

- string **Provider** [get, set]
The name of the user-selected third party two-factor authentication provider.
- string **Code** [get, set]
The two-factor authentication code that had been sent to the user.
- string **ReturnUrl** [get, set]
The redirection Url to send the user to after having verified their account information via the two-factor authentication code.
- bool **RememberBrowser** [get, set]
The user's preference to have their account automatically verified, when possible, by the browser that the user is currently using to browse the website.
- bool **RememberMe** [get, set]
The user's preference to have their account automatically verified, when possible.

5.43.1 Detailed Description

A model for verifying a two-factor authentication code that has been sent to a user.

5.43.2 Property Documentation

5.43.2.1 Code

```
string ARFE.Models.VerifyCodeViewModel.Code [get], [set]
```

The two-factor authentication code that had been sent to the user.

5.43.2.2 Provider

```
string ARFE.Models.VerifyCodeViewModel.Provider [get], [set]
```

The name of the user-seleted third party two-factor authentication provider.

5.43.2.3 RememberBrowser

```
bool ARFE.Models.VerifyCodeViewModel.RememberBrowser [get], [set]
```

The user's preference to have their account automatically verified, when possible, by the browser that the user is currently using to browse the website.

5.43.2.4 RememberMe

```
bool ARFE.Models.VerifyCodeViewModel.RememberMe [get], [set]
```

The user's preference to have their account automatically verified, when possible.

5.43.2.5 returnUrl

```
string ARFE.Models.VerifyCodeViewModel.ReturnUrl [get], [set]
```

The redirection Url to send the user to after having verified their account information via the two-factor authentication code.

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

- ARFE/Models/AccountViewModels.cs
-

5.44 ARFE.Models.VerifyPhoneNumberViewModel Class Reference

A model for verifying a user's phone number via two-factor authentication. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.

Properties

- string `Code` [get, set]
The code provided by two-factor authentication.
- string `PhoneNumber` [get, set]
The phone number being validated.

5.44.1 Detailed Description

A model for verifying a user's phone number via two-factor authentication. This model is currently not being used as two-factor authentication is not supported. However this serves as an extension point for supporting two-factor authentication at a later time.

5.44.2 Property Documentation

5.44.2.1 Code

```
string ARFE.Models.VerifyPhoneNumberViewModel.Code [get], [set]
```

The code provided by two-factor authentication.

5.44.2.2 PhoneNumber

```
string ARFE.Models.VerifyPhoneNumberViewModel.PhoneNumber [get], [set]
```

The phone number being validated.

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

- ARFE/Models/ManageViewModels.cs
-

Index

- ARFE.ApplicationSignInManager, [25](#)
- ARFE.ApplicationUserManager, [28](#)
- ARFE.BlobManager, [30](#)
- ARFE.BundleConfig, [37](#)
- ARFE.Controllers, [10](#)
- ARFE.Controllers.AccountController, [15](#)
- ARFE.Controllers.FileTypeModel, [45](#)
- ARFE.Controllers.HelpController, [49](#)
- ARFE.Controllers.ManageController, [53](#)
- ARFE.Controllers.MobileAuthController, [56](#)
- ARFE.Controllers.PublicContentController, [59](#)
- ARFE.Controllers.UploadController, [78](#)
- ARFE.Controllers.UserContentController, [87](#)
- ARFE.EmailService, [40](#)
- ARFE.FileConverter, [41](#)
- ARFE.FilterConfig, [48](#)
- ARFE.Models, [11](#)
- ARFE.Models.AddPhoneNumberViewModel, [23](#)
- ARFE.Models.ApplicationDbContext, [24](#)
- ARFE.Models.ApplicationUser, [27](#)
- ARFE.Models.ChangePasswordViewModel, [38](#)
- ARFE.Models.ConfigureTwoFactorViewModel, [39](#)
- ARFE.Models.ForgotPasswordViewModel, [49](#)
- ARFE.Models.IndexViewModel, [50](#)
- ARFE.Models.LoginViewModel, [52](#)
- ARFE.Models.ManageLoginsViewModel, [56](#)
- ARFE.Models.RegisterViewModel, [61](#)
- ARFE.Models.ResetPasswordViewModel, [62](#)
- ARFE.Models.SendCodeViewModel, [69](#)
- ARFE.Models.SetPasswordViewModel, [70](#)
- ARFE.Models.VerifyCodeViewModel, [90](#)
- ARFE.Models.VerifyPhoneNumberViewModel, [92](#)
- ARFE.MvcApplication, [59](#)
- ARFE.RouteConfig, [64](#)
- ARFE.SearchQueries, [65](#)
- ARFE.SmsService, [71](#)
- ARFE.Startup, [72](#)
- ARFE.Tests, [12](#)
- ARFE.Tests.BlobTests, [36](#)
- ARFE.Tests.Controllers, [12](#)
- ARFE.Tests.Controllers.HomeControllerTest, [50](#)
- ARFE.Tests.UploadedFileCacheTests, [86](#)
- ARFE.UploadedFileCache, [80](#)
- ARFE::ApplicationSignInManager
 - ApplicationSignInManager, [26](#)
 - Create, [26](#)
 - CreateUserIdentityAsync, [27](#)
- ARFE::ApplicationUserManager
 - ApplicationUserManager, [29](#)
 - Create, [29](#)
- ARFE::BlobManager
 - BlobManager, [31](#)
 - ConvertAndDownloadBlobFromUserContainer, [31](#)
 - DeleteBlobByNameInUserContainer, [31](#)
 - FormatBlobContainerName, [32](#)
 - GetOrCreateBlobContainer, [32](#)
 - ListBlobNamesToUrisInPublicContainer, [33](#)
 - ListBlobNamesToUrisInPublicContainerOwnedBy, [33](#)
 - ListBlobNamesToUrisInUserContainer, [33](#)
 - ListBlobUrisInPublicContainerOwnedBy, [34](#)
 - ListBlobUrisInUserContainer, [34](#)
 - ListPrivateBlobInfoForUI, [34](#)
 - ListPublicBlobInfoForUI, [35](#)
 - RemoveExpiredTemporaryBlobs, [35](#)
 - UploadBlobToPublicContainer, [35](#)
 - UploadBlobToUserContainer, [36](#)
- ARFE::BundleConfig
 - RegisterBundles, [38](#)
- ARFE::Controllers::AccountController
 - AccountController, [16](#), [17](#)
 - ConfirmEmail, [17](#)
 - Dispose, [17](#)
 - ForgotPassword, [18](#)
 - ForgotPasswordConfirmation, [18](#)
 - LogOff, [19](#)
 - Login, [18](#), [19](#)
 - Register, [19](#), [20](#)
 - ResetPassword, [20](#)
 - ResetPasswordConfirmation, [21](#)
 - SendCode, [21](#)
 - SignInManager, [23](#)
 - UserManager, [23](#)
 - VerifyCode, [22](#)
- ARFE::Controllers::FileTypeModel
 - FileType, [46](#)
 - FileTypes, [46](#)
- ARFE::Controllers::HelpController
 - Help, [50](#)
- ARFE::Controllers::ManageController
 - ManageController, [54](#), [55](#)
 - SignInManager, [55](#)
 - UserManager, [55](#)
- ARFE::Controllers::MobileAuthController
 - DownloadFile, [57](#)
 - ListFiles, [58](#)
 - MobileAuthController, [57](#)
 - RequestAuthToken, [58](#)

- UserManager, 59
 - ARFE::Controllers::PublicContentController
 - ContentSelect, 60
 - Index, 60
 - Search, 61
 - ARFE::Controllers::UploadController
 - Index, 78
 - OverWriteUpload, 79
 - UploadFile, 79
 - ARFE::Controllers::UserContentController
 - Index, 88
 - PrivateContentSelect, 88
 - PrivateSearch, 89
 - PublicContentSelect, 89
 - PublicSearch, 90
 - ARFE::EmailService
 - SendAsync, 41
 - ARFE::FileConverter
 - BasePath, 45
 - ConvertToDAE, 43
 - ConvertToFBX, 43
 - ConvertToOBJ, 43
 - ConvertToPLY, 44
 - ConvertToSTL, 44
 - FileConverter, 42
 - InputFolder, 45
 - OutputFolder, 45
 - ARFE::FilterConfig
 - RegisterGlobalFilters, 48
 - ARFE::Models::AddPhoneNumberViewModel
 - Number, 24
 - ARFE::Models::ApplicationDbContext
 - ApplicationDbContext, 25
 - Create, 25
 - ARFE::Models::ApplicationUser
 - GenerateUserIdentityAsync, 28
 - ARFE::Models::ChangePasswordViewModel
 - ConfirmPassword, 38
 - NewPassword, 39
 - OldPassword, 39
 - ARFE::Models::ConfigureTwoFactorViewModel
 - Providers, 40
 - SelectedProvider, 40
 - ARFE::Models::ForgotPasswordViewModel
 - Email, 49
 - ARFE::Models::IndexViewModel
 - BrowserRemembered, 51
 - HasPassword, 51
 - Logins, 51
 - PhoneNumber, 51
 - TwoFactor, 52
 - ARFE::Models::LoginViewModel
 - Email, 52
 - Password, 52
 - RememberMe, 53
 - ARFE::Models::ManageLoginsViewModel
 - CurrentLogins, 56
 - OtherLogins, 56
 - ARFE::Models::RegisterViewModel
 - ConfirmPassword, 62
 - Email, 62
 - Password, 62
 - ARFE::Models::ResetPasswordViewModel
 - Code, 63
 - ConfirmPassword, 63
 - Email, 63
 - Password, 63
 - ARFE::Models::SendCodeViewModel
 - Providers, 69
 - RememberMe, 69
 - ReturnUrl, 70
 - SelectedProvider, 70
 - ARFE::Models::SetPasswordViewModel
 - ConfirmPassword, 70
 - NewPassword, 71
 - ARFE::Models::VerifyCodeViewModel
 - Code, 91
 - Provider, 91
 - RememberBrowser, 91
 - RememberMe, 91
 - ReturnUrl, 91
 - ARFE::Models::VerifyPhoneNumberViewModel
 - Code, 92
 - PhoneNumber, 92
 - ARFE::RouteConfig
 - RegisterRoutes, 64
 - ARFE::SearchQueries
 - FilterByFileNameOrderByDateAscending, 65
 - FilterByFileNameOrderByDateDescending, 66
 - FilterByFileNameOrderByNameAscending, 66
 - FilterByFileNameOrderByNameDescending, 67
 - FilterDateOrderByDateAscending, 67
 - FilterDateOrderByDateDescending, 67
 - FilterDateOrderByNameAscending, 68
 - FilterDateOrderByNameDescending, 68
 - ARFE::SmsService
 - SendAsync, 71
 - ARFE::Startup
 - Configuration, 72
 - ConfigureAuth, 73
 - TimerInit, 73
 - ARFE::Tests::BlobTests
 - Test_Constructor, 37
 - Test_FormatBlobContainerName, 37
 - ARFE::Tests::UploadedFileCacheTests
 - Test_Init, 86
 - Test_PrivateConstructor, 87
 - ARFE::UploadedFileCache
 - BasePath, 86
 - DeleteAndRemoveFile, 81
 - DeleteOldFiles, 82
 - FindContainingFolderGUIDForFile, 82
 - GetFileDescription, 83
 - GetInstance, 83
 - IsSavedFilePublic, 83
 - MarkForDelete, 84
-

- RemoveFileTracking, 84
- SaveFile, 84, 85
- ARFE, 9
- AccountController
 - ARFE::Controllers::AccountController, 16, 17
- ApplicationDbContext
 - ARFE::Models::ApplicationDbContext, 25
- ApplicationSignInManager
 - ARFE::ApplicationSignInManager, 26
- ApplicationUserManager
 - ARFE::ApplicationUserManager, 29
- AuthenticationTokenCache, 13
- AuthenticationTokenCache.Tests, 13
- AuthenticationTokenCache.Tests.TokenCacheTests, 76
- AuthenticationTokenCache.TokenCache, 74
- AuthenticationTokenCache::Tests::TokenCacheTests
 - Test_BadTokenReturnsNull, 76
 - Test_Init, 76
 - Test_Insert, 76
 - Test_MultipleTokensToSameUser, 77
 - Test_PrivateConstructor, 77
 - Test_Singleton, 77
 - Test_ValidateMultiple, 77
 - Test_ValidateSingle, 77
- AuthenticationTokenCache::TokenCache
 - GenerateToken, 74
 - Init, 75
 - ValidateToken, 75
- Author
 - Common::FileUIInfo, 47
- BasePath
 - ARFE::FileConverter, 45
 - ARFE::UploadedFileCache, 86
- BlobManager
 - ARFE::BlobManager, 31
- BrowserRemembered
 - ARFE::Models::IndexViewModel, 51
- Code
 - ARFE::Models::ResetPasswordViewModel, 63
 - ARFE::Models::VerifyCodeViewModel, 91
 - ARFE::Models::VerifyPhoneNumberViewModel, 92
- Common, 13
- Common.FileUIInfo, 46
- Common.SupportedFileTypes, 73
- Common::FileUIInfo
 - Author, 47
 - Description, 47
 - FileName, 48
 - FileUIInfo, 47
 - UploadDate, 48
- Common::SupportedFileTypes
 - FileList, 73
- Configuration
 - ARFE::Startup, 72
- ConfigureAuth
 - ARFE::Startup, 73
- ConfirmEmail
 - ARFE::Controllers::AccountController, 17
- ConfirmPassword
 - ARFE::Models::ChangePasswordViewModel, 38
 - ARFE::Models::RegisterViewModel, 62
 - ARFE::Models::ResetPasswordViewModel, 63
 - ARFE::Models::SetPasswordViewModel, 70
- ContentSelect
 - ARFE::Controllers::PublicContentController, 60
- ConvertAndDownloadBlobFromUserContainer
 - ARFE::BlobManager, 31
- ConvertToDAE
 - ARFE::FileConverter, 43
- ConvertToFBX
 - ARFE::FileConverter, 43
- ConvertToOBJ
 - ARFE::FileConverter, 43
- ConvertToPLY
 - ARFE::FileConverter, 44
- ConvertToSTL
 - ARFE::FileConverter, 44
- Create
 - ARFE::ApplicationSignInManager, 26
 - ARFE::ApplicationUserManager, 29
 - ARFE::Models::ApplicationDbContext, 25
- CreateUserIdentityAsync
 - ARFE::ApplicationSignInManager, 27
- CurrentLogins
 - ARFE::Models::ManageLoginsViewModel, 56
- DeleteAndRemoveFile
 - ARFE::UploadedFileCache, 81
- DeleteBlobByNameInUserContainer
 - ARFE::BlobManager, 31
- DeleteOldFiles
 - ARFE::UploadedFileCache, 82
- Description
 - Common::FileUIInfo, 47
- Dispose
 - ARFE::Controllers::AccountController, 17
- DownloadFile
 - ARFE::Controllers::MobileAuthController, 57
- Email
 - ARFE::Models::ForgotPasswordViewModel, 49
 - ARFE::Models::LoginViewModel, 52
 - ARFE::Models::RegisterViewModel, 62
 - ARFE::Models::ResetPasswordViewModel, 63
- FileConverter
 - ARFE::FileConverter, 42
- FileList
 - Common::SupportedFileTypes, 73
- FileName
 - Common::FileUIInfo, 48
- FileType
 - ARFE::Controllers::FileTypeModel, 46
- FileTypeModel, 45
- FileTypes
 - ARFE::Controllers::FileTypeModel, 46

- FileUIInfo
 - Common::FileUIInfo, [47](#)
 - FilterByFileNameOrderByDateAscending
 - ARFE::SearchQueries, [65](#)
 - FilterByFileNameOrderByDateDescending
 - ARFE::SearchQueries, [66](#)
 - FilterByFileNameOrderByNameAscending
 - ARFE::SearchQueries, [66](#)
 - FilterByFileNameOrderByNameDescending
 - ARFE::SearchQueries, [67](#)
 - FilterDateOrderByDateAscending
 - ARFE::SearchQueries, [67](#)
 - FilterDateOrderByDateDescending
 - ARFE::SearchQueries, [67](#)
 - FilterDateOrderByNameAscending
 - ARFE::SearchQueries, [68](#)
 - FilterDateOrderByNameDescending
 - ARFE::SearchQueries, [68](#)
 - FindContainingFolderGUIDForFile
 - ARFE::UploadedFileCache, [82](#)
 - ForgotPassword
 - ARFE::Controllers::AccountController, [18](#)
 - ForgotPasswordConfirmation
 - ARFE::Controllers::AccountController, [18](#)
 - FormatBlobContainerName
 - ARFE::BlobManager, [32](#)
 - GenerateToken
 - AuthenticationTokenCache::TokenCache, [74](#)
 - GenerateUserIdentityAsync
 - ARFE::Models::ApplicationUser, [28](#)
 - GetFileDescription
 - ARFE::UploadedFileCache, [83](#)
 - GetInstance
 - ARFE::UploadedFileCache, [83](#)
 - GetOrCreateBlobContainer
 - ARFE::BlobManager, [32](#)
 - HasPassword
 - ARFE::Models::IndexViewModel, [51](#)
 - Help
 - ARFE::Controllers::HelpController, [50](#)
 - Index
 - ARFE::Controllers::PublicContentController, [60](#)
 - ARFE::Controllers::UploadController, [78](#)
 - ARFE::Controllers::UserContentController, [88](#)
 - Init
 - AuthenticationTokenCache::TokenCache, [75](#)
 - InputFolder
 - ARFE::FileConverter, [45](#)
 - IsSavedFilePublic
 - ARFE::UploadedFileCache, [83](#)
 - ListBlobNamesToUriInPublicContainer
 - ARFE::BlobManager, [33](#)
 - ListBlobNamesToUriInPublicContainerOwnedBy
 - ARFE::BlobManager, [33](#)
 - ListBlobNamesToUriInUserContainer
 - ARFE::BlobManager, [33](#)
 - ListBlobUriInPublicContainerOwnedBy
 - ARFE::BlobManager, [34](#)
 - ListBlobUriInUserContainer
 - ARFE::BlobManager, [34](#)
 - ListFiles
 - ARFE::Controllers::MobileAuthController, [58](#)
 - ListPrivateBlobInfoForUI
 - ARFE::BlobManager, [34](#)
 - ListPublicBlobInfoForUI
 - ARFE::BlobManager, [35](#)
 - LogOff
 - ARFE::Controllers::AccountController, [19](#)
 - Login
 - ARFE::Controllers::AccountController, [18, 19](#)
 - Logins
 - ARFE::Models::IndexViewModel, [51](#)
 - ManageController
 - ARFE::Controllers::ManageController, [54, 55](#)
 - MarkForDelete
 - ARFE::UploadedFileCache, [84](#)
 - MobileAuthController
 - ARFE::Controllers::MobileAuthController, [57](#)
 - NewPassword
 - ARFE::Models::ChangePasswordViewModel, [39](#)
 - ARFE::Models::SetPasswordViewModel, [71](#)
 - Number
 - ARFE::Models::AddPhoneNumberViewModel, [24](#)
 - OldPassword
 - ARFE::Models::ChangePasswordViewModel, [39](#)
 - OtherLogins
 - ARFE::Models::ManageLoginsViewModel, [56](#)
 - OutputFolder
 - ARFE::FileConverter, [45](#)
 - OverWriteUpload
 - ARFE::Controllers::UploadController, [79](#)
 - Password
 - ARFE::Models::LoginViewModel, [52](#)
 - ARFE::Models::RegisterViewModel, [62](#)
 - ARFE::Models::ResetPasswordViewModel, [63](#)
 - PhoneNumber
 - ARFE::Models::IndexViewModel, [51](#)
 - ARFE::Models::VerifyPhoneNumberViewModel, [92](#)
 - PrivateContentSelect
 - ARFE::Controllers::UserContentController, [88](#)
 - PrivateSearch
 - ARFE::Controllers::UserContentController, [89](#)
 - Provider
 - ARFE::Models::VerifyCodeViewModel, [91](#)
 - Providers
 - ARFE::Models::ConfigureTwoFactorViewModel, [40](#)
 - ARFE::Models::SendCodeViewModel, [69](#)
 - PublicContentSelect
 - ARFE::Controllers::UserContentController, [89](#)
 - PublicSearch
-

- ARFE::Controllers::UserController, 90
 - Register
 - ARFE::Controllers::AccountController, 19, 20
 - RegisterBundles
 - ARFE::BundleConfig, 38
 - RegisterGlobalFilters
 - ARFE::FilterConfig, 48
 - RegisterRoutes
 - ARFE::RouteConfig, 64
 - RememberBrowser
 - ARFE::Models::VerifyCodeViewModel, 91
 - RememberMe
 - ARFE::Models::LoginViewModel, 53
 - ARFE::Models::SendCodeViewModel, 69
 - ARFE::Models::VerifyCodeViewModel, 91
 - RemoveExpiredTemporaryBlobs
 - ARFE::BlobManager, 35
 - RemoveFileTracking
 - ARFE::UploadedFileCache, 84
 - RequestAuthToken
 - ARFE::Controllers::MobileAuthController, 58
 - ResetPassword
 - ARFE::Controllers::AccountController, 20
 - ResetPasswordConfirmation
 - ARFE::Controllers::AccountController, 21
 - ReturnUrl
 - ARFE::Models::SendCodeViewModel, 70
 - ARFE::Models::VerifyCodeViewModel, 91
 - SaveFile
 - ARFE::UploadedFileCache, 84, 85
 - Search
 - ARFE::Controllers::PublicContentController, 61
 - SelectedProvider
 - ARFE::Models::ConfigureTwoFactorViewModel, 40
 - ARFE::Models::SendCodeViewModel, 70
 - SendAsync
 - ARFE::EmailService, 41
 - ARFE::SmsService, 71
 - SendCode
 - ARFE::Controllers::AccountController, 21
 - SignInManager
 - ARFE::Controllers::AccountController, 23
 - ARFE::Controllers::ManageController, 55
 - Test_BadTokenReturnsNull
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 76
 - Test_Constructor
 - ARFE::Tests::BlobTests, 37
 - Test_FormatBlobContainerName
 - ARFE::Tests::BlobTests, 37
 - Test_Init
 - ARFE::Tests::UploadedFileCacheTests, 86
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 76
 - Test_Insert
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 76
 - Test_MultipleTokensToSameUser
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 77
 - Test_PrivateConstructor
 - ARFE::Tests::UploadedFileCacheTests, 87
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 77
 - Test_Singleton
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 77
 - Test_ValidateMultiple
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 77
 - Test_ValidateSingle
 - AuthenticationTokenCache::Tests::TokenCache↔
Tests, 77
 - TimerInit
 - ARFE::Startup, 73
 - TwoFactor
 - ARFE::Models::IndexViewModel, 52
 - UploadBlobToPublicContainer
 - ARFE::BlobManager, 35
 - UploadBlobToUserContainer
 - ARFE::BlobManager, 36
 - UploadDate
 - Common::FileUIInfo, 48
 - UploadFile
 - ARFE::Controllers::UploadController, 79
 - UserManager
 - ARFE::Controllers::AccountController, 23
 - ARFE::Controllers::ManageController, 55
 - ARFE::Controllers::MobileAuthController, 59
 - ValidateToken
 - AuthenticationTokenCache::TokenCache, 75
 - VerifyCode
 - ARFE::Controllers::AccountController, 22
-