AssertionConditionT IS_TRUE = 0 IS_FALSE = 1 IS_NULL = 2 IS_NOT_NULL = 3 ImgurApiResponse <t +="" :="" bo="" data="" get;="" imgurapiresponse()<="" set;="" success="" th="" tinn="" {="" }=""><th>CollectionAccessType None = 0 Read = 1 ModifyExistingContent = 2 UpdatedContent = 6 LimgurGaleryAsset contentImageThumbmacontentImageFull: Wea + Title { get; set; } : String + Description { get; set; } : + Timestamp { get; set; } : + Type { get; set; } : String + IsAnimated { get; set; } : + Width { get; set; } : Int32 + Height { get; set; } : Int32 + Views { get; set; } : Int32 + Views { get; set; } : Boolean + InGallery { get; set; } : B + Favorite { get; set; } : String + CommentCount { get; se + IsAlbum { get; set; } : B + Ratio { get; } : Double + ShouldDisplay { get; } : + ContentImageMedium { + ContentImageMedium { + ContentImageMedium { + ContentImageFull { get;}</th><th>### ImplicitUseTargetFlags Default = 1</th><th>ImplicitUseKindFlags Default = 7 Access = 1 Assign = 2 InstantiatedWithFixedConst InstantiatedNoFixedConstru + AccessToken { get; set; + ExpiresIn { get; set; } : + TokenType { get; set; } : + Scope { get; set; } : Object + RefreshToken { get; set; + AccountId { get; set; } : + AccountUsername { get; + ImgurAuthorisation()</th><th>+ Nsid { get; set; } : String + Username { get; set; } : + Fullname { get; set; } : S + User() RazorWriteMethodPa + RazorWriteMethodParam</th><th>Content + Value { get; set; } : String + Content() RazorWriteMethodAtt + RazorWriteMethodAttribu</th><th>Settings - SettingsDefault: String - SettingsKey: String = set - AppSettings { get: }: IS + GeneralSettings { get: se Settings() RazorWriteLiteralMet + RazorWriteLiteralMethod</th><th>FlickrBaseModel + Id { get; set; } : String + FlickrBaseModel() RazorLayoutAttribute + RazorLayoutAttribute()</th><th>FlickrApiResponse<t +="" :="" data="" flickrapiresponse()="" get;="" razorhelpercommon="" razorhelpercommonattri<="" set;="" st="" success="" th="" tinn="" {="" }=""><th>FlickrAuthorisation + Token { get; set; } : Con + Perms { get; set; } : Con + AuthenticatedUser { get; + FlickrAuthorisation() RazorDirectiveAttribute + Directive { get; set; } : S + RazorDirectiveAttribute(d</th><th>PersonConverter() + PersonConverter() + WriteJson(writer: JsonWr + ReadJson(reader: JsonRe + CanConvert(objectType: + Type { get; set; }: String + FieldName { get; set; }: + RazorInjectionAttribute(t</th><th>FlickrGaleryAsset contentImageThumbmacontentImageFull: Wea + Sources { get; set; } : Lis + Title { get; set; } : String + LinkThumbmail { get; set + Height { get; set; } : Int32 + Width { get; set; } : Int32 + ShouldDisplay { get; } : + Favorite { get; set; } : B + Ratio { get; } : Double + ContentImageMedium { + ContentImageFull { get; + FlickrGaleryAsset() - CreateMedium() : Image CreateFull() : ImageSource RazorImportNamespa + Name { get; jset; } : String + RazorImportNamespaceA</th><th>FlickrComment + AuthorIsDeleted { get; s + Author { get; set; } : String + Iconserver { get; set; } : + Datecreate { get; set; } : + Permalink { get; set; } : + Realname { get; set; } : + Comment { get; set; } : + FlickrComment() AspTypePropertyAttri + CreateConstructorRefere + AspTypePropertyAttribut</th><th>FlickrAssetSource + Label { get; set; } : String + Width { get; set; } : Int32 + Height { get; set; } : Stri + Url { get; set; } : String + Media { get; set; } : String + FlickrAssetSource() AspRequiredAttribute + Attribute { get; set; } : S + AspRequiredAttributeAttri</th><th> Hid { get; set; } : String + ImgurBaseModel() AspMethodPropertyA + AspMethodPropertyAttrib</th></t></th></t>	CollectionAccessType None = 0 Read = 1 ModifyExistingContent = 2 UpdatedContent = 6 LimgurGaleryAsset contentImageThumbmacontentImageFull: Wea + Title { get; set; } : String + Description { get; set; } : + Timestamp { get; set; } : + Type { get; set; } : String + IsAnimated { get; set; } : + Width { get; set; } : Int32 + Height { get; set; } : Int32 + Views { get; set; } : Int32 + Views { get; set; } : Boolean + InGallery { get; set; } : B + Favorite { get; set; } : String + CommentCount { get; se + IsAlbum { get; set; } : B + Ratio { get; } : Double + ShouldDisplay { get; } : + ContentImageMedium { + ContentImageMedium { + ContentImageMedium { + ContentImageFull { get;}	### ImplicitUseTargetFlags Default = 1	ImplicitUseKindFlags Default = 7 Access = 1 Assign = 2 InstantiatedWithFixedConst InstantiatedNoFixedConstru + AccessToken { get; set; + ExpiresIn { get; set; } : + TokenType { get; set; } : + Scope { get; set; } : Object + RefreshToken { get; set; + AccountId { get; set; } : + AccountUsername { get; + ImgurAuthorisation()	+ Nsid { get; set; } : String + Username { get; set; } : + Fullname { get; set; } : S + User() RazorWriteMethodPa + RazorWriteMethodParam	Content + Value { get; set; } : String + Content() RazorWriteMethodAtt + RazorWriteMethodAttribu	Settings - SettingsDefault: String - SettingsKey: String = set - AppSettings { get: }: IS + GeneralSettings { get: se Settings() RazorWriteLiteralMet + RazorWriteLiteralMethod	FlickrBaseModel + Id { get; set; } : String + FlickrBaseModel() RazorLayoutAttribute + RazorLayoutAttribute()	FlickrApiResponse <t +="" :="" data="" flickrapiresponse()="" get;="" razorhelpercommon="" razorhelpercommonattri<="" set;="" st="" success="" th="" tinn="" {="" }=""><th>FlickrAuthorisation + Token { get; set; } : Con + Perms { get; set; } : Con + AuthenticatedUser { get; + FlickrAuthorisation() RazorDirectiveAttribute + Directive { get; set; } : S + RazorDirectiveAttribute(d</th><th>PersonConverter() + PersonConverter() + WriteJson(writer: JsonWr + ReadJson(reader: JsonRe + CanConvert(objectType: + Type { get; set; }: String + FieldName { get; set; }: + RazorInjectionAttribute(t</th><th>FlickrGaleryAsset contentImageThumbmacontentImageFull: Wea + Sources { get; set; } : Lis + Title { get; set; } : String + LinkThumbmail { get; set + Height { get; set; } : Int32 + Width { get; set; } : Int32 + ShouldDisplay { get; } : + Favorite { get; set; } : B + Ratio { get; } : Double + ContentImageMedium { + ContentImageFull { get; + FlickrGaleryAsset() - CreateMedium() : Image CreateFull() : ImageSource RazorImportNamespa + Name { get; jset; } : String + RazorImportNamespaceA</th><th>FlickrComment + AuthorIsDeleted { get; s + Author { get; set; } : String + Iconserver { get; set; } : + Datecreate { get; set; } : + Permalink { get; set; } : + Realname { get; set; } : + Comment { get; set; } : + FlickrComment() AspTypePropertyAttri + CreateConstructorRefere + AspTypePropertyAttribut</th><th>FlickrAssetSource + Label { get; set; } : String + Width { get; set; } : Int32 + Height { get; set; } : Stri + Url { get; set; } : String + Media { get; set; } : String + FlickrAssetSource() AspRequiredAttribute + Attribute { get; set; } : S + AspRequiredAttributeAttri</th><th> Hid { get; set; } : String + ImgurBaseModel() AspMethodPropertyA + AspMethodPropertyAttrib</th></t>	FlickrAuthorisation + Token { get; set; } : Con + Perms { get; set; } : Con + AuthenticatedUser { get; + FlickrAuthorisation() RazorDirectiveAttribute + Directive { get; set; } : S + RazorDirectiveAttribute(d	PersonConverter() + PersonConverter() + WriteJson(writer: JsonWr + ReadJson(reader: JsonRe + CanConvert(objectType: + Type { get; set; }: String + FieldName { get; set; }: + RazorInjectionAttribute(t	FlickrGaleryAsset contentImageThumbmacontentImageFull: Wea + Sources { get; set; } : Lis + Title { get; set; } : String + LinkThumbmail { get; set + Height { get; set; } : Int32 + Width { get; set; } : Int32 + ShouldDisplay { get; } : + Favorite { get; set; } : B + Ratio { get; } : Double + ContentImageMedium { + ContentImageFull { get; + FlickrGaleryAsset() - CreateMedium() : Image CreateFull() : ImageSource RazorImportNamespa + Name { get; jset; } : String + RazorImportNamespaceA	FlickrComment + AuthorIsDeleted { get; s + Author { get; set; } : String + Iconserver { get; set; } : + Datecreate { get; set; } : + Permalink { get; set; } : + Realname { get; set; } : + Comment { get; set; } : + FlickrComment() AspTypePropertyAttri + CreateConstructorRefere + AspTypePropertyAttribut	FlickrAssetSource + Label { get; set; } : String + Width { get; set; } : Int32 + Height { get; set; } : Stri + Url { get; set; } : String + Media { get; set; } : String + FlickrAssetSource() AspRequiredAttribute + Attribute { get; set; } : S + AspRequiredAttributeAttri	Hid { get; set; } : String + ImgurBaseModel() AspMethodPropertyA + AspMethodPropertyAttrib
+ AspDataFieldsAttribute + AspDataFieldsAttribute() HtmlElementAttribute + Name { get; set; } : String + HtmlElementAttributesAtt + HtmlElementAttributesAtt	+ ImgurGaleryAsset() - CreateMedium(): Image CreateFull(): ImageSource + AspDataFieldAttribute + AspDataFieldAttribute() AspMvcActionSelect + AspMvcActionSelectorAttr	AspChildControlType + TagName { get; set; } : + ControlType { get; set; } + AspChildControlTypeAttri AspMvcViewCompon + AspMvcViewComponentV	**XamIltemBindingOfIt* + XamIItemBindingOfItems **AspMvcViewCompon* + AspMvcViewComponentA	+ XamlItemsControlAttribut + XamlItemsControlAttribut + AspMvcViewAttribute + AspMvcViewAttribute()	NoReorderAttribute + NoReorderAttribute() AspMvcTemplateAttri + AspMvcTemplateAttribute()	RegexPatternAttribute + RegexPatternAttribute() AspMvcEditorTempla + AspMvcEditorTemplateAt	NoEnumerationAttrib + NoEnumerationAttribute() AspMvcDisplayTempl + AspMvcDisplayTemplateA	LinqTunnelAttribute + LinqTunnelAttribute() AspMvcSuppressVie + AspMvcSuppressViewErro	TerminatesProgramA + TerminatesProgramAttrib + AspMvcPartialViewAt + AspMvcPartialViewAttribu	AssertionConditionAt + ConditionType { get; set; + AssertionConditionAttribu AspMvcModelTypeAt + AspMvcModelTypeAttribu	AssertionMethodAttri + AssertionMethodAttribute() AspMvcMasterAttribute + AspMvcMasterAttribute()	ļ	RazorSectionAttribute + RazorSectionAttribute() + RazorSectionAttribute()	+ AnonymousProperty { ge + AspMvcActionAttribute() + AspMvcActionAttribute() + AspMvcActionAttribute()
AspMvcViewLocation + Format { get; set; } : Stri + AspMvcViewLocationFor	AspMvcPartialViewLo + Format { get; set; } : Stri + AspMvcPartialViewLocati	+ Format { get; set; } : Stri + AspMvcMasterLocationFo	+ Format { get; set; } : Stri + AspMvcAreaViewLocation					+ BasePath { get; set; }: S + PathReferenceAttribute() + PathReferenceAttribute(b						
+ UseKindFlags { get; set; + TargetFlags { get; set; } + UsedImplicitlyAttribute() + UsedImplicitlyAttribute(ta + UsedImplicitlyAttribute(u	+ BaseType { get; set; }: + BaseTypeRequiredAttribu	+ CannotApplyEqualityOper	+ Required { get; set; } : B + LocalizationRequiredAttri + LocalizationRequiredAttri	+ Contract { get; set; } : St + ForceFullStates { get; set + ContractAnnotationAttrib + ContractAnnotationAttrib	NotifyPropertyChang + ParameterName { get; se + NotifyPropertyChangedIn + NotifyPropertyChangedIn	InvokerParameterNa + InvokerParameterNameA ImageDetailViewMod	+ Name { get; set; } : String + ValueProviderAttribute(n	+ FormatParameterName { + StringFormatMethodAttri	+ ItemCanBeNullAttribute() ContentPageBase <t< td=""><td>temNotNullAttribute + ItemNotNullAttribute() ImageDetailView<tse< td=""><td>+ NotNullAttribute() GaleryView<tservice></tservice></td><td>+ CanBeNullAttribute()</td><td>FlickrBaseClient client: HttpClient + ClientId: String = 37002 # Client { get; i} : HttpClient + IdentityProvider { get; } + FlickrBaseClient() # Execute(method: HttpMe # Execute(method: HttpMe # ExecuteNoReturn(metho # GetMethodDescription(m)</td><td>FlickrOauthIdentityPr client: HttpClient + SecretKey: String = bf6d + IdentityToken { get; set; + UserId { get; set; } : String + UserName { get; set; } : + NeedUserInput { get; } : Frob { get; set; } : String # Client { get; } : HttpClient + FlickrOauthIdentityProvid + GetAuthorisationUrl() : T + GetAuthenticationHeader + Authorize(pin: String) : T + ReAuthorize() : Task + Md5(value: String) : String</td></tse<></td></t<>	temNotNullAttribute + ItemNotNullAttribute() ImageDetailView <tse< td=""><td>+ NotNullAttribute() GaleryView<tservice></tservice></td><td>+ CanBeNullAttribute()</td><td>FlickrBaseClient client: HttpClient + ClientId: String = 37002 # Client { get; i} : HttpClient + IdentityProvider { get; } + FlickrBaseClient() # Execute(method: HttpMe # Execute(method: HttpMe # ExecuteNoReturn(metho # GetMethodDescription(m)</td><td>FlickrOauthIdentityPr client: HttpClient + SecretKey: String = bf6d + IdentityToken { get; set; + UserId { get; set; } : String + UserName { get; set; } : + NeedUserInput { get; } : Frob { get; set; } : String # Client { get; } : HttpClient + FlickrOauthIdentityProvid + GetAuthorisationUrl() : T + GetAuthenticationHeader + Authorize(pin: String) : T + ReAuthorize() : Task + Md5(value: String) : String</td></tse<>	+ NotNullAttribute() GaleryView <tservice></tservice>	+ CanBeNullAttribute()	FlickrBaseClient client: HttpClient + ClientId: String = 37002 # Client { get; i} : HttpClient + IdentityProvider { get; } + FlickrBaseClient() # Execute(method: HttpMe # Execute(method: HttpMe # ExecuteNoReturn(metho # GetMethodDescription(m)	FlickrOauthIdentityPr client: HttpClient + SecretKey: String = bf6d + IdentityToken { get; set; + UserId { get; set; } : String + UserName { get; set; } : + NeedUserInput { get; } : Frob { get; set; } : String # Client { get; } : HttpClient + FlickrOauthIdentityProvid + GetAuthorisationUrl() : T + GetAuthenticationHeader + Authorize(pin: String) : T + ReAuthorize() : Task + Md5(value: String) : String
+ FlickrClientService() + GetMainGalery(page: Int + SearchMainGalery(query: + GetGalleryAssetComment + FavoriteImage(asset: IIm + GetImage(assetId: String + UploadImage(image: Str GetImageSizes(asset: Fli *interface** IAssetComment + Comment { get; } : String + Author { get; } : String	1 1 1	client: HttpClient - SecretKey: String = fd65 + IdentityToken { get; set; + RefreshToken { get; set; + UserId.{_get; set; }:_String. + UserName { get; set; }: + NeedUserInput { get; }: # Client { get; }: HttpClient + ImgurOauthIdentityProvi + GetAuthorisationUrl(): T + GetAutherticationHeader + Authorize(pin: String): T + ReAuthorize()-:-Task	+ ImgurClientService() + GetMainGalery(page: Int + SearchMainGalery(query: + GetGalleryAssetComment + FavoriteImage(asset: IIm + GetImage(assetId: String + UploadImage(image: Str) «interface» IBaseClient + IdentityProvider { get; }	+ StartedToAssetValueCon + Convert(value: Object, ta + ConvertBack(value: Obje	isFetching: Boolean + Page { get; set; }: Page - IsFetching { get; set; }: + PropertyChanged: Prope ViewModelBase()	imgurAsset: IImageAssetisStared: BooleanassetId: String - + ImgurAsset { get; set; } + Comments { get; set; } : + ImageClientService { get + FetchComments { get; } + IsStared { get; set; } : B + StarCommand { get; } : I + ImageDetailViewModel() - GetComments() : Task	assets: ObservableCollesearchQuery: StringcurrentPage: Int32 mgurAsset + ImageClientService { get + SearchQuery { get; set; + SearchQuery { get; set; + CurrentPage { get; set; } + UploadFileCommand { ge + GalleryViewModel() + OnDisplay(): Void - UploadFile(file: MediaFile Fetch(): Task	+ RichImage() - RichImage()	viewModel: TViewModel # ViewModel { get; } : TVi + ContentPageBase()	+ ImageDetailView(assetId	stack: StackLayoutscrollView: ScrollView + GaleryView() - ScrollView_Scrolled(send = GetImage(asset: JImage	+ App() # OnStart(): Void # OnResume(): Void		+ ShouldDisplay { get; }: + Favorite { get; }: Boolean + Title { get; }: String + Ratio { get; }: Double + ContentImageMedium {+-ContentImageFull { get;