Table of Contents

Namespace PicScam

Classes

ActiveOfferViewModel

Data binds to one active (still available to buy) offer of the currently logged in user.

App

Interaction logic for App.xaml

BuyPicture

Buying picture - window

BuyPremium

Buying premium - window

CommentBox

CustomerSellerViewModel

ViewModel for info related to customer/seller selection in ReportQueryWindow.

Ftp

Class to handle uploading picture to FTP server

LoadPicsPage

Page in which pictrues are loaded to local database.

Login

Login page

MainPage

Interaction logic for MainPage.xaml

MainWindow

Interaction logic for MainWindow.xaml

NavButton

PdfPrintingComponent

Component for printing PDFs - especially the payment history report.

PhotoAdding

PhotoAdding.Adding

Pic

PicWindowInfo

General info about a singular Pic window. It is never changed through the course of the window's lifespan.

PurchasesPage

DONE Sell button inactive on local pictures with no server equivalent Download button inactive on NOT BOUGHT pictures

RecordedPaymentViewModel

ViewModel for displaying a single completed payment. It can be either a SALE or a PURCHASE.

Registration

Registration page

ReportQueryWindow

Window allowing the user to parametrize their payment history report.

Retrieve

Retrieving password page

SearchPage

Interaction logic for SearchPage.xaml

TagsPage

Interaction logic for TagsPage.xaml

UserSettingsPage

Page responsible for displaying and changing user data.

Class ActiveOfferViewModel

Data binds to one active (still available to buy) offer of the currently logged in user.

Inheritance

object

ActiveOfferViewModel

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class ActiveOfferViewModel
```

Constructors

ActiveOfferViewModel(int, string, int, string, decimal, int)

Declaration

```
public ActiveOfferViewModel(int oID, string DateCreated, int pID, string pName, decimal price, int
salesAmount)
```

Parameters

TYPE	NAME	DESCRIPTION
int	oID	
string	DateCreated	
int	pID	
string	pName	
decimal	price	
int	salesAmount	

Properties

DateCreated

DateTime in dd/MM/yyyy HH:mm format specifying when the row in Offers table has been created.

Declaration

```
public string DateCreated { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FontSize

Declaration

```
public static int FontSize { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

HeightBinding

Declaration

```
public static int HeightBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

OfferId

ID in the Offers table.

Declaration

```
public int OfferId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureId

Picture ID in the server database.

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureName

Full name of the picture.

Declaration

```
public string PictureName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Price

Paid money in PLN.

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

SalesAmount

How many times this offer has been bought by other users.

Declaration

```
public int SalesAmount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class App

Interaction logic for App.xaml

Inheritance

object
DispatcherObject
Application
App

Implements

IQueryAmbient

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class App : Application, IQueryAmbient
```

Implements

IQueryAmbient

Class BuyPicture

Buying picture - window

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
BuyPicture

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class BuyPicture : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

BuyPicture(int, int)

Constructor, initializes the window

Declaration

```
public BuyPicture(int userId, int offerId)
```

Parameters

TYPE	NAME	DESCRIPTION
int	userId	user ID
int	offerId	offer ID

Methods

WaitForPaymentEnd(string)

Waits for the payment to end

Declaration

```
public Task<bool> WaitForPaymentEnd(string email)
```

Parameters

TYPE	NAME	DESCRIPTION
string	email	user's e-mail

Returns

TYPE	DESCRIPTION
Task<bool>	nothing, async task

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Class BuyPremium

Buying premium - window

Inheritance

object

DispatcherObject

DependencyObject

Visual

UIElement

FrameworkElement

Control

ContentControl

Window

BuyPremium

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class BuyPremium : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

BuyPremium(int)

Constructor, initializes the window

Declaration

```
public BuyPremium(int id)
```

Parameters

TYPE	NAME	DESCRIPTION
int	id	user ID

Methods

WaitForPaymentEnd(string)

Waits for the payment to end

Declaration

```
public Task<bool> WaitForPaymentEnd(string email)
```

Parameters

TYPE	NAME	DESCRIPTION
string	email	user's e-mail

Returns

TYPE	DESCRIPTION
Task<bool>	nothing, async task

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Class CommentBox

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
CommentBox

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class CommentBox : Control, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient
```

Fields

CommentContentProperty

Declaration

```
public static readonly DependencyProperty CommentContentProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

DatePostedProperty

Declaration

```
public static readonly DependencyProperty DatePostedProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

UsernameProperty

Declaration

```
public static readonly DependencyProperty UsernameProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

Properties

CommentContent

Declaration

```
public string CommentContent { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DatePosted

Declaration

```
public string DatePosted { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Username

Declaration

```
public string Username { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

Class CustomerSellerViewModel

ViewModel for info related to customer/seller selection in ReportQueryWindow.

Inheritance

object

CustomerSellerViewModel

Implements

INotifyPropertyChanged

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class CustomerSellerViewModel : INotifyPropertyChanged
```

Constructors

CustomerSellerViewModel(int, string, string, bool)

Declaration

```
public CustomerSellerViewModel(int id, string nameSurname, string email, bool selected)
```

Parameters

TYPE	NAME	DESCRIPTION
int	id	
string	nameSurname	
string	email	
bool	selected	

Properties

Email

User full email.

Declaration

```
public string Email { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FontSize

Declaration

```
public static int FontSize { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

HeightBinding

Declaration

```
public static int HeightBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsSelected

Is this user selected to be included in the payment history report.

Declaration

```
public bool IsSelected { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

UserId

Customer/Seller ID.

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Username

User name + surname.

Declaration

```
public string Username { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Events

PropertyChanged

Occurs when a property value changes.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

INotifyPropertyChanged

Class Ftp

Class to handle uploading picture to FTP server

Inheritance

object

Ftp

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public static class Ftp
```

Methods

UploadPicture(string, string, string)

Method to upload picture

Declaration

```
public static bool UploadPicture(string fileName, string localPath, string onlinePath)
```

Parameters

TYPE	NAME	DESCRIPTION
string	fileName	file name
string	localPath	path to the picture on the computerS
string	onlinePath	destination

Returns

TYPE	DESCRIPTION
bool	true if success, otherwise false

Class LoadPicsPage

Page in which pictrues are loaded to local database.

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Page
LoadPicsPage

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class LoadPicsPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

LoadPicsPage()

Initializes the page.

Declaration

```
public LoadPicsPage()
```

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Class Login

Login page

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
Login

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class Login : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

Login()

Constructor, adds icon to window

Declaration

```
public Login()
```

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Class MainPage

Interaction logic for MainPage.xaml

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Page
MainPage

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class MainPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

MainPage()

Initializes database context and grid structure. The page must be refreshed later (in `Page_Loaded` method).

Declaration

```
public MainPage()
```

Methods

Refresh(object, EventArgs)

Refreshes Local, Server and Offer grid content.

Declaration

```
public void Refresh(object sender, EventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	This window or Pic window.
EventArgs	e	Empty.

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Class MainWindow

Interaction logic for MainWindow.xaml

Inheritance

object

MainWindow

Implements

INavigationWindow

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class MainWindow : INavigationWindow
```

Constructors

MainWindow()

Declaration

```
public MainWindow()
```

Methods

CloseWindow()

Triggers the command to close a window.

Declaration

```
public void CloseWindow()
```

GetFrame()

Provides direct access to the Frame used in navigation.

Declaration

```
public Frame GetFrame()
```

Returns

TYPE	DESCRIPTION
Frame	Instance of the Frame control.

GetNavigation()

Provides direct access to the control responsible for navigation.

Declaration

```
public INavigation GetNavigation()
```

Returns

TYPE	DESCRIPTION
INavigation	Instance of the Wpf.Ui.Controls.Interfaces.INavigation control.

Navigate(Type)

Lets you navigate to the selected page based on it's type. Should be used with Wpf.Ui.Mvvm.Contracts.IPageService.

Declaration

```
public bool Navigate(Type pageType)
```

Parameters

TYPE	NAME	DESCRIPTION
Type	pageType	`Type` of the page.

Returns

TYPE	DESCRIPTION
bool	

SetPageService(IPageService)

Lets you attach the service that delivers page instances to Wpf.Ui.Controls.Interfaces.INavigation.

Declaration

```
public void SetPageService(IPageService pageService)
```

Parameters

TYPE	NAME	DESCRIPTION
IPageService	pageService	Instance of the Wpf.Ui.Mvvm.Contracts.IPageService with attached service provider.

ShowWindow()

Triggers the command to open a window.

Declaration

```
public void ShowWindow()
```

Implements

Wpf.Ui.Mvvm.Contracts.INavigationWindow

Class NavButton

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
ButtonBase
NavButton

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild
ICommandSource

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class NavButton : ButtonBase, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild, ICommandSource
```

Fields

ImageSourceProperty

Declaration

```
public static readonly DependencyProperty ImageSourceProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

NavUriProperty

Declaration

```
public static readonly DependencyProperty NavUriProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

TextProperty

Declaration

```
public static readonly DependencyProperty TextProperty
```

Field Value

TYPE	DESCRIPTION
DependencyProperty	

Properties

ImageSource

Declaration

```
public ImageSource ImageSource { get; set; }
```

Property Value

TYPE	DESCRIPTION
ImageSource	

NavUri

Declaration

```
public Uri NavUri { get; set; }
```

Property Value

TYPE	DESCRIPTION
Uri	

Text

Declaration

```
public string Text { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

ICommandSource

Class PdfPrintingComponent

Component for printing PDFs - especially the payment history report.

Inheritance

object
BaseComponent
PdfPrintingComponent

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class PdfPrintingComponent : BaseComponent
```

Constructors

PdfPrintingComponent(IMediator)

Declaration

```
public PdfPrintingComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Properties

AllUsersAmount

Amount of all users that this user has made transactions with.

Declaration

```
public int AllUsersAmount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

ArePurchasesEnabled

Specifies if purchases should be listed in the report.

Declaration

```
public bool ArePurchasesEnabled { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

AreSalesEnabled

Specifies if sales should be listed in the report.

Declaration

```
public bool AreSalesEnabled { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

BeginDate

Minimum transaction date to be included in the report

Declaration

```
public DateTime? BeginDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

EndDate

Maximum transaction date to be included in the report

Declaration

```
public DateTime? EndDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

MaxValue

Maximum transaction Price to be included in the report.

Declaration

```
public decimal? MaxValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal?	

MinValue

Minimal transaction Price to be included in the report.

Declaration

```
public decimal? MinValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal?	

purchases

All of current user's purchases.

Declaration

```
public List<RecordedPaymentViewModel>? purchases { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<RecordedPaymentViewModel>	

sales

All of current user's sales.

Declaration

```
public List<RecordedPaymentViewModel>? sales { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<RecordedPaymentViewModel>	

selectedUsers

Users included in the payment report.

Declaration

```
public List<CustomerSellerViewModel> selectedUsers { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CustomerSellerViewModel>	

Methods

GenerateReportPdf()

Displays a SaveFileDialog, then creates and saves a PDF file if a location on disc has been selected.

Declaration

```
public void GenerateReportPdf()
```

SetupDataSources(List<CustomerSellerViewModel>, List<RecordedPaymentViewModel>?, List<RecordedPaymentViewModel>?)

Sets list content.

Declaration

```
public void SetupDataSources(List<CustomerSellerViewModel> selectedUsers, List<RecordedPaymentViewModel>?
purchases, List<RecordedPaymentViewModel>? sales)
```

Parameters

TYPE	NAME	DESCRIPTION
List<CustomerSellerViewModel>	selectedUsers	Users included in the payment report.
List<RecordedPaymentViewModel>	purchases	All of current user's purchases.
List<RecordedPaymentViewModel>	sales	All of current user's sales.

SetupSettings(bool, bool)

Set which parts of the report should be generated.

Declaration

```
public void SetupSettings(bool purchasesEnabled, bool salesEnabled)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	purchasesEnabled	True if purchases should be listed.
bool	salesEnabled	True if sales should be listed.

SetupValues(decimal?, decimal?, DateTime?, DateTime?, int)

Sets range limits and other parameter values.

Declaration

```
public void SetupValues(decimal? min, decimal? max, DateTime? begin, DateTime? end, int allUsers)
```

Parameters

TYPE	NAME	DESCRIPTION
decimal?	min	Minimal transaction Price to be included in the report.
decimal?	max	Maximum transaction Price to be included in the report.
DateTime?	begin	Minimum transaction date to be included in the report

TYPE	NAME	DESCRIPTION
DateTime?	end	Maximum transaction date to be included in the report
int	allUsers	Amount of all users that this user has made transactions with.

Class PhotoAdding

Inheritance

object
DbContext
PicscamContext
PhotoAdding

Implements

IInfrastructure<IServiceProvider>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class PhotoAdding : PicscamContext, IInfrastructure<IServiceProvider>, IDbContextDependencies,
IDbSetCache, IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable
```

Implements

IInfrastructure<T>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Class PhotoAdding.Adding

Inheritance

object

PhotoAdding.Adding

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class PhotoAdding.Adding
```

Methods

UploadImageToDatabase(string)

Declaration

```
public void UploadImageToDatabase(string imagePath)
```

Parameters

TYPE	NAME	DESCRIPTION
string	imagePath	

Class Pic

Inheritance

object

Pic

Implements

IMediator

INotifyPropertyChanged

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class Pic : IMediator, INotifyPropertyChanged
```

Constructors

Pic(int, bool)

Declaration

```
public Pic(int pictureId, bool isPictureLocal)
```

Parameters

TYPE	NAME	DESCRIPTION
int	pictureId	
bool	isPictureLocal	

Properties

CommentComponent

Component allowing the window to display, delete and post comments.

Declaration

```
public CommentComponent CommentComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
CommentComponent	

DescriptionEditComponent

Component in charge of TextBox responsible for modifying the picture's description.

Declaration

```
public ActivatedTextEditComponent DescriptionEditComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
ActivatedTextEditComponent	

LocalPicture

Declaration

```
public Picture? LocalPicture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

MessageComponent

Component enabling info/error message display.

Declaration

```
public MessageComponent MessageComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
MessageComponent	

NameEditComponent

Component in charge of TextBox responsible for modifying the picture's name.

Declaration

```
public ActivatedTextEditComponent NameEditComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
ActivatedTextEditComponent	

PicWindowInfo

General, unchangeable info about the current Pic window.

Declaration

```
public PicWindowInfo PicWindowInfo { get; }
```

Property Value

TYPE	DESCRIPTION
PicWindowInfo	

PictureDataDisplayComponent

Declaration

```
public PictureDataDisplayComponent PictureDataDisplayComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
PictureDataDisplayComponent	

PictureDisplayComponent

Component responsible for data binding the displayed WPF Image to a BitmapImage object. Has methods allowing the image to be loaded either from file or from a byte array.

Declaration

```
public PictureDisplayComponent PictureDisplayComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
PictureDisplayComponent	

PictureRemovalComponent

Component allowing for proper picture removal - takes care of all dependent entities. Is able to delete the picture data and/or remove it from PicScam system.

Declaration

```
public PictureRemovalComponent PictureRemovalComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
PictureRemovalComponent	

ServerLocalIntegrationComponent

Declaration

```
public ServerLocalIntegrationComponent ServerLocalIntegrationComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
ServerLocalIntegrationComponent	

ServerPicture

Declaration

```
public Picture? ServerPicture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

TagAssigmentDisplayComponent

Component enabling the display of the picture's tags.

Declaration

```
public TagAssigmentDisplayComponent TagAssigmentDisplayComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
TagAssigmentDisplayComponent	

UserComponent

Component enabling user identification. Makes it able to create buttons that are accessible only to the owner of the picture.

Declaration

```
public UserComponent UserComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
UserComponent	

_localContext

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

InitializeBuyComponent()

Declaration

```
public void InitializeBuyComponent()
```

Notify(object, string)

Handles messages coming in from the components.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	One of the components.
string	eventName	Type of the event.

Events

DatabaseModified

Raised when modifications to the database should cause a main menu refresh.

Declaration

```
public event EventHandler DatabaseModified
```

Event Type

TYPE	DESCRIPTION
EventHandler	

PictureEntryMissing

Raised when an entry is missing and main menu needs to display an appriopriate message.

Declaration

```
public event EventHandler PictureEntryMissing
```

Event Type

TYPE	DESCRIPTION
EventHandler	

PropertyChanged

Raised when a data binded value has been changed and GUI needs an update.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IMediator
INotifyPropertyChanged

Class PicWindowInfo

General info about a singular Pic window. It is never changed through the course of the window's lifespan.

Inheritance

object

PicWindowInfo

Implements

IEquatable<PicWindowInfo>

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public record PicWindowInfo : IEquatable<PicWindowInfo>
```

Properties

IsOwner

Declaration

```
public bool IsOwner { get; init; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsPictureLocal

Declaration

```
public bool IsPictureLocal { get; init; }
```

Property Value

TYPE	DESCRIPTION
bool	

LocalPictureId

Declaration

```
public int LocalPictureId { get; init; }
```

Property Value

TYPE	DESCRIPTION
int	

ServerPictureId

Declaration

```
public int ServerPictureId { get; init; }
```

Property Value

TYPE	DESCRIPTION
int	

Implements

IEquatable<T>

Class PurchasesPage

DONE Sell button inactive on local pictures with no server equivalent Download button inactive on NOT BOUGHT pictures

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Page
PurchasesPage

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild
IMediator

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class PurchasesPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild, IMediator
```

Constructors

PurchasesPage()

Initializes all ListView bindings.

Declaration

```
public PurchasesPage()
```

Properties

OffersBinding

Data binds to the viewed ACTIVE OFFERS list.

Declaration

```
public List<ActiveOfferViewModel> OffersBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<ActiveOfferViewModel>	

PurchasesBinding

Data binds to the viewed PURCHASES list.

Declaration

```
public List<RecordedPaymentViewModel> PurchasesBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<RecordedPaymentViewModel>	

SalesBinding

Data binds to the viewed SALES list.

Declaration

```
public List<RecordedPaymentViewModel> SalesBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<RecordedPaymentViewModel>	

_localContext

Local database context.

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Server database context

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

Notify(object, string)

Handles component messages.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	One of the components.
string	eventName	Empty.

PicLink_Click(object, EventArgs)

Opens a new Pic window with picture of the clicked ID.

Declaration

```
public void PicLink_Click(object sender, EventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	Clickable PictureID column.
EventArgs	e	Empty.

Events

PropertyChanged

Raised when a data binded value has been changed and GUI needs an update.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild
IMediator

Class RecordedPaymentViewModel

ViewModel for displaying a single completed payment. It can be either a SALE or a PURCHASE.

Inheritance

object

RecordedPaymentViewModel

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class RecordedPaymentViewModel
```

Constructors

RecordedPaymentViewModel(int, string, DateTime, int, string, string, int, string, decimal)

Declaration

```
public RecordedPaymentViewModel(int bID, string date, DateTime dateFull, int uID, string name, string email,
int pID, string pName, decimal price)
```

Parameters

TYPE	NAME	DESCRIPTION
int	bID	
string	date	
DateTime	dateFull	
int	uID	
string	name	
string	email	
int	pID	
string	pName	
decimal	price	

Properties

BoughtId

ID in the Bought table.

Declaration

```
public int BoughtId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Date

DateTime in dd/MM/yyyy HH:mm format specifying when the row in Bought table has been created.

Declaration

```
public string Date { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DateFull

Date as full DataTime type.

Declaration

```
public DateTime DateFull { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

FontSize

Declaration

```
public static int FontSize { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

HeightBinding

Declaration

```
public static int HeightBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureId

Picture ID in the server database.

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureName

Full name of the picture.

Declaration

```
public string PictureName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Price

Paid money in PLN.

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

UserEmail

Email of: customer in the case of SALE picture owner in the case of PURCHASE

Declaration

```
public string UserEmail { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

UserId

Customer/Seller ID.

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Username

Name + Surname of: customer in the case of SALE picture owner in the case of PURCHASE

Declaration

```
public string Username { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class Registration

Registration page

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
Registration

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class Registration : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

Registration()

Registration page constructor, sets icon

Declaration

```
public Registration()
```

Methods

Reset()

Empties all the fields

Declaration

```
public void Reset()
```

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Class ReportQueryWindow

Window allowing the user to parametrize their payment history report.

Inheritance

object

ReportQueryWindow

Implements

IMediator

INotifyPropertyChanged

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class ReportQueryWindow : IMediator, INotifyPropertyChanged
```

Constructors

ReportQueryWindow(List<RecordedPaymentViewModel>, List<RecordedPaymentViewModel>)

Fills available Customer/Seller List and saves purchases/sales list state.

Declaration

```
public ReportQueryWindow(List<RecordedPaymentViewModel> purchases, List<RecordedPaymentViewModel> sales)
```

Parameters

TYPE	NAME	DESCRIPTION
List<RecordedPaymentViewModel>	purchases	Current user's purchases.
List<RecordedPaymentViewModel>	sales	Current user's sales.

Properties

ArePurchasesIncluded

Specifies if purchases should be listed in the report.

Declaration

```
public bool ArePurchasesIncluded { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

AreSalesIncluded

Specifies if sales should be listed in the report.

Declaration

```
public bool AreSalesIncluded { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

BeginningTimeRangeFormatted

Minimum transaction date to be included in the report - formatted as a string containing only the date.

Declaration

```
public string BeginningTimeRangeFormatted { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CustomerSellersBinding

Binds all customers/sellers the current user made transactions with to the report filter.

Declaration

```
public List<CustomerSellerViewModel> CustomerSellersBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<CustomerSellerViewModel>	

EndingTimeRangeFormatted

Maximum transaction date to be included in the report - formatted as a string containing only the date.

Declaration

```
public string EndingTimeRangeFormatted { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

MaxPriceBinding

Maximum transaction Price to be included in the report.

Declaration

```
public string MaxPriceBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

MinPriceBinding

Minimal transaction Price to be included in the report.

Declaration

```
public string MinPriceBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

SelectedDate

Date currently selected in the GUI calendar.

Declaration

```
public DateTime? SelectedDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

_localContext

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

Notify(object, string)

Handles messages coming in from the components.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	
string	eventName	

ResetMinButton_Click(object, RoutedEventArgs)

Resets transaction minimum price.

Declaration

```
public void ResetMinButton_Click(object sender, RoutedEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	"Reset" button.
RoutedEventArgs	e	Empty.

Events

PropertyChanged

Raised when a data binded value has been changed and GUI needs an update.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IMediator
INotifyPropertyChanged

Class Retrieve

Retrieving password page

Inheritance

object

Retrieve

Implements

INavigationWindow

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class Retrieve : INavigationWindow
```

Constructors

Retrieve()

Default constructor, nothing interesting here

Declaration

```
public Retrieve()
```

Methods

CloseWindow()

Triggers the command to close a window.

Declaration

```
public void CloseWindow()
```

GetFrame()

Provides direct access to the Frame used in navigation.

Declaration

```
public Frame GetFrame()
```

Returns

TYPE	DESCRIPTION
Frame	Instance of the Frame control.

GetNavigation()

Provides direct access to the control responsible for navigation.

Declaration

```
public INavigation GetNavigation()
```

Returns

TYPE	DESCRIPTION
INavigation	Instance of the Wpf.Ui.Controls.Interfaces.INavigation control.

Navigate(Type)

Lets you navigate to the selected page based on it's type. Should be used with Wpf.Ui.Mvvm.Contracts.IPageService.

Declaration

```
public bool Navigate(Type pageType)
```

Parameters

TYPE	NAME	DESCRIPTION
Type	pageType	`Type` of the page.

Returns

TYPE	DESCRIPTION
bool	

SetPageService(IPageService)

Lets you attach the service that delivers page instances to Wpf.Ui.Controls.Interfaces.INavigation.

Declaration

```
public void SetPageService(IPageService pageService)
```

Parameters

TYPE	NAME	DESCRIPTION
IPageService	pageService	Instance of the Wpf.Ui.Mvvm.Contracts.IPageService with attached service provider.

ShowWindow()

Triggers the command to open a window.

Declaration

```
public void ShowWindow()
```

Implements

Wpf.Ui.Mvvm.Contracts.INavigationWindow

Class SearchPage

Interaction logic for SearchPage.xaml

Inheritance

object

DispatcherObject

DependencyObject

Visual

UIElement

FrameworkElement

Page

SearchPage

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class SearchPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

SearchPage(MainPage)

Declaration

```
public SearchPage(MainPage mainPg)
```

Parameters

TYPE	NAME	DESCRIPTION
MainPage	mainPg	

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Class TagsPage

Interaction logic for TagsPage.xaml

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Page
TagsPage

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class TagsPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

TagsPage()

constructor

Declaration

```
public TagsPage()
```

Methods

SelectionChanged(int)

manipulate the view of edit tags and delete tags buttons

Declaration

```
public void SelectionChanged(int add)
```

Parameters

TYPE	NAME	DESCRIPTION
int	add	change in number of selected items

Implements

IAnimatable
ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

Class UserSettingsPage

Page responsible for displaying and changing user data.

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Page
UserSettingsPage

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam

Assembly: PicScam.dll

Syntax

```
public class UserSettingsPage : Page, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

UserSettingsPage()

Initializes page - displays user data from the databse.

Declaration

```
public UserSettingsPage()
```

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace PicScam.Bases

Classes

BaseComponent

A `BaseComponent` class contains a reference to its mediator object.

Interfaces

IMediator

Class responsible for handling the components' messages.

Class BaseComponent

A `BaseComponent` class contains a reference to its mediator object.

Inheritance

object
BaseComponent
ActivatedTextEditComponent
MessageComponent
PictureDisplayComponent
UserComponent
PdfPrintingComponent
CommentComponent
PictureDataDisplayComponent
PictureRemovalComponent
ServerLocalIntegrationComponent
TagAssigmentDisplayComponent
TagAssigmentEditComponent

Namespace: PicScam.Bases

Assembly: PicScam.dll

Syntax

```
public class BaseComponent
```

Constructors

BaseComponent(IMediator)

Declaration

```
public BaseComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Fields

_mediator

Handles the component's messages.

Declaration

```
protected IMediator? _mediator
```

Field Value

TYPE	DESCRIPTION
IMediator	

Methods

SetMediator(IMediator)

Declaration

```
public void SetMediator(IMediator mediator)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Interface IMediator

Class responsible for handling the components' messages.

Namespace: PicScam.Bases

Assembly: PicScam.dll

Syntax

```
public interface IMediator
```

Properties

_localContext

Declaration

```
PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Declaration

```
KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

Notify(object, string)

For component -> mediator communication

Declaration

```
void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	The component object.
string	eventName	Name of type of message.

Namespace PicScam.CommonComponents

Classes

ActivatedTextEditComponent

Component responsible for managing editable TextBoxes that can be turned ON/OFF.

MessageComponent

A component able to display user info or error messages. Has two methods of display - MessageBox for confirmation messages and PopUp for notifications.

PictureDisplayComponent

Component responsible for displaying the image in GUI. Can load image from a file and from a byte array.

UserComponent

Component enabling user identification. Makes it able to create buttons that are accessible only to the owner of the picture.

Class ActivatedTextEditComponent

Component responsible for managing editable TextBoxes that can be turned ON/OFF.

Inheritance

object

BaseComponent

ActivatedTextEditComponent

Namespace: PicScam.CommonComponents

Assembly: PicScam.dll

Syntax

```
public class ActivatedTextEditComponent : BaseComponent
```

Constructors

ActivatedTextEditComponent(bool, IMediator)

Component initialization.

Declaration

```
public ActivatedTextEditComponent(bool canValueBeEmpty, IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	canValueBeEmpty	True if user should be able to save an empty value in the TextBox.
IMediator	mediator	The mediator handling this component's messages.

Properties

DisplayedTextBeforeEdit

Holds the value stored in the TextBox before current session of editing began.

Declaration

```
public string DisplayedTextBeforeEdit { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

EditButtonsVisibility

Specifies if the "Save" and "Discard" buttons are active or not.

Declaration

```
public Visibility EditButtonsVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

KeepBorder

Specifies if the TextBox should keep its border when editing is OFF.

Declaration

```
public bool KeepBorder { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

TextBinding

The TextBox's text content.

Declaration

```
public string TextBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Methods

DiscardEditingResult(TextBox)

Ends editing without saving.

Declaration

```
public void DiscardEditingResult(TextBox textBox)
```

Parameters

TYPE	NAME	DESCRIPTION
TextBox	textBox	The TextBox to turn OFF.

FillTextBox(string)

Update text displayed in the TextBox.

Declaration

```
public void FillTextBox(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
string	text	Text to be displayed.

SaveDataBeforeEditing()

Called when user clicks an "Edit..." button. Saves the previous content of the textbox.

Declaration

```
public void SaveDataBeforeEditing()
```

SaveEditingResult(TextBox)

Ends editing and returns the final string entered in the TextBox. If string is empty `CanValueBeEmpty` is set to false, returns null and notifies the mediator.

Declaration

```
public string? SaveEditingResult(TextBox textBox)
```

Parameters

TYPE	NAME	DESCRIPTION
TextBox	textBox	TextBox to get value from.

Returns

TYPE	DESCRIPTION
string	The displayed contents of the passed in TextBox or null if empty when it cannot be.

StartEditing(TextBox)

Prepare window for editing - enables the specified textbox and makes its "Save" and "Discard" buttons visible. Is called after user clicks the "Edit..." button.

Declaration

```
public void StartEditing(TextBox textBox)
```

Parameters

TYPE	NAME	DESCRIPTION
TextBox	textBox	TextBox that will be enabled for edit.

Class MessageComponent

A component able to display user info or error messages. Has two methods of display - MessageBox for confirmation messages and PopUp for notifications.

Inheritance

object
BaseComponent
MessageComponent

Namespace: PicScam.CommonComponents

Assembly: PicScam.dll

Syntax

```
public class MessageComponent : BaseComponent
```

Constructors

MessageComponent(IMediator)

Declaration

```
public MessageComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Methods

DisplayInfoMessageBox(string, string)

Displays a simple info message box with no defined action when user clicks "OK".

Declaration

```
public void DisplayInfoMessageBox(string title, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
string	title	Title in the bar of MessageBox.
string	message	The longer description.

DisplayInfoPopUp(string, string, Window, ToastPopUp?)

Displays a new popup with no defined clickable action.

Declaration

```
public void DisplayInfoPopUp(string title, string message, Window parentWindow, ToastPopUp? popup = null)
```

Parameters

TYPE	NAME	DESCRIPTION
string	title	Title of the PopUp.
string	message	Longer description.
Window	parentWindow	Parent window.
ToastPopUp	popup	Optional - reference to previous popup of the same kind.

Class PictureDisplayComponent

Component responsible for displaying the image in GUI. Can load image from a file and from a byte array.

Inheritance

object

BaseComponent

PictureDisplayComponent

Namespace: PicScam.CommonComponents

Assembly: PicScam.dll

Syntax

```
public class PictureDisplayComponent : BaseComponent
```

Constructors

PictureDisplayComponent(IMediator?)

Declaration

```
public PictureDisplayComponent(IMediator? mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Properties

PictureDisplaySourceBinding

Acts as a source of the image in the GUI.

Declaration

```
public ImageSource? PictureDisplaySourceBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
ImageSource	

Methods

DisplayImageFromByteArray(byte[])

Loads an image from byte array and displays it.

Declaration

```
public void DisplayImageFromByteArray(byte[] pictureArray)
```

Parameters

TYPE	NAME	DESCRIPTION
byte[]	pictureArray	Source byte array.

DisplayImageFromFile(string)

Loads an image from file and displays it.

Declaration

```
public void DisplayImageFromFile(string filePath)
```

Parameters

TYPE	NAME	DESCRIPTION
string	filePath	Path to the file on the local disc.

Class UserComponent

Component enabling user identification. Makes it able to create buttons that are accessible only to the owner of the picture.

Inheritance

object
BaseComponent
UserComponent

Namespace: PicScam.CommonComponents

Assembly: PicScam.dll

Syntax

```
public class UserComponent : BaseComponent
```

Constructors

UserComponent(IMediator)

Declaration

```
public UserComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Properties

EmailBinding

Binds to the displayed email.

Declaration

```
public string EmailBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Owner

The picture's owner entity.

Declaration

```
public User? Owner { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

OwnerButtonsVisibility

Specifies if owner-exclusive buttons should be accessible.

Declaration

```
public Visibility OwnerButtonsVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

UsernameBinding

Binds to the displayed username.

Declaration

```
public string UsernameBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Methods

CheckIfOwner(int)

Returns true if specified user is the owner.

Declaration

```
public bool CheckIfOwner(int userId)
```

Parameters

TYPE	NAME	DESCRIPTION
int	userId	Checked user ID.

Returns

TYPE	DESCRIPTION
bool	Returns true if passed in user ID is the same as owner ID.

DisplayOwnerButtonsIfCurrentUserIsOwner(int)

Checks if the passed in user ID is the same as owner ID. If it is, enables owner-exclusive buttons.

Declaration

```
public void DisplayOwnerButtonsIfCurrentUserIsOwner(int currentUserId)
```

Parameters

TYPE	NAME	DESCRIPTION
int	currentUserId	Id of currently logged user.

SetOwnerDisplayedInfo()

Displays user info in the `UsernameBinding` and `EmailBinding`.

Declaration

```
public void SetOwnerDisplayedInfo()
```

Namespace PicScam.Migrations

Classes

add_tag

A base class inherited by each EF Core migration.

Class add_tag

A base class inherited by each EF Core migration.

Inheritance

object

Migration

add_tag

Namespace: PicScam.Migrations

Assembly: PicScam.dll

Syntax

```
[DbContext(typeof(PicscamContext))]
[Migration("20230512210203_add_tag")]
public class add_tag : Migration
```

Remarks

See Database migrations for more information and examples.

Methods

BuildTargetModel(ModelBuilder)

Implemented to build the TargetModel.

Declaration

```
protected override void BuildTargetModel(ModelBuilder modelBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
ModelBuilder	modelBuilder	The ModelBuilder to use to build the model.

Overrides

Migration.BuildTargetModel(ModelBuilder)

Remarks

See Database migrations for more information and examples.

Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

Declaration

```
protected override void Down(MigrationBuilder migrationBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
MigrationBuilder	migrationBuilder	The MigrationBuilder that will build the operations.

Overrides

Migration.Down(MigrationBuilder)

Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from Migration if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See Database migrations for more information and examples.

Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

Declaration

```
protected override void Up(MigrationBuilder migrationBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
MigrationBuilder	migrationBuilder	The MigrationBuilder that will build the operations.

Overrides

Migration.Up(MigrationBuilder)

Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from Migration.

See Database migrations for more information and examples.

Namespace PicScam.Models

Classes

Bought

Comment

Offer

Owner

PicscamContext

Picture

Shared

Tag

TagAssigment

TestEntries1

TestEntries2

User

Class Bought

Inheritance

object

Bought

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Bought
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Offer

Declaration

```
public virtual Offer Offer { get; set; }
```

Property Value

TYPE	DESCRIPTION
Offer	

OfferId

Declaration

```
public int OfferId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Comment

Inheritance

object

Comment

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Comment
```

Properties

Content

Declaration

```
public string Content { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Offer

Inheritance

object

Offer

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Offer
```

Properties

Boughts

Declaration

```
public virtual ICollection<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Bought>	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Price

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Class Owner

Inheritance

object

Owner

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Owner
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class PicscamContext

Inheritance

object
DbContext
PicscamContext
PhotoAdding

Implements

IInfrastructure<IServiceProvider>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class PicscamContext : DbContext, IInfrastructure<IServiceProvider>, IDbContextDependencies,
IDbSetCache, IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable
```

Constructors

PicscamContext()

Declaration

```
public PicscamContext()
```

PicscamContext(DbContextOptions<PicscamContext>)

Declaration

```
public PicscamContext(DbContextOptions<PicscamContext> options)
```

Parameters

TYPE	NAME	DESCRIPTION
DbContextOptions<PicscamContext>	options	

Properties

Boughts

Declaration

```
public virtual DbSet<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Bought>	

Comments

Declaration

```
public virtual DbSet<Comment> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Comment>	

Offers

Declaration

```
public virtual DbSet<Offer> Offers { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Offer>	

Owners

Declaration

```
public virtual DbSet<Owner> Owners { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Owner>	

Pictures

Declaration

```
public virtual DbSet<Picture> Pictures { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Picture>	

Shareds

Declaration

```
public virtual DbSet<Shared> Shareds { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Shared>	

TagAssigments

Declaration

```
public virtual DbSet<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TagAssigment>	

Tags

Declaration

```
public virtual DbSet<Tag> Tags { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Tag>	

TestEntries1s

Declaration

```
public virtual DbSet<TestEntries1> TestEntries1s { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TestEntries1>	

TestEntries2s

Declaration

```
public virtual DbSet<TestEntries2> TestEntries2s { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TestEntries2>	

Users

Declaration

```
public virtual DbSet<User> Users { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<User>	

Methods

OnConfiguring(DbContextOptionsBuilder)

Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created. The base implementation does nothing.

Declaration

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
DbContextOptionsBuilder	optionsBuilder	A builder used to create or modify options for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure the context.

Overrides

DbContext.OnConfiguring(DbContextOptionsBuilder)

Remarks

In situations where an instance of DbContextOptions may or may not have been passed to the constructor, you can use IsConfigured to determine if the options have already been set, and skip some or all of the logic in OnConfiguring(DbContextOptionsBuilder).

See DbContext lifetime, configuration, and initialization for more information and examples.

OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in DbSet<TEntity> properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

Declaration

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
ModelBuilder	modelBuilder	The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

Overrides

DbContext.OnModelCreating(ModelBuilder)

Remarks

If a model is explicitly set on the options for this context (via UseModel(IModel)) then this method will not be run. However, it will still run when creating a compiled model.

See Modeling entity types and relationships for more information and examples.

Implements

IInfrastructure<T>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Class Picture

Inheritance

object

Picture

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Picture
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

FilePath

Declaration

```
public string FilePath { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Height

Declaration

```
public int Height { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

ServerPictureId

Declaration

```
public int ServerPictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Size

Declaration

```
public long Size { get; set; }
```

Property Value

TYPE	DESCRIPTION
long	

Thumbnail

Declaration

```
public byte[]? Thumbnail { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

Width

Declaration

```
public int Width { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Shared

Inheritance

object

Shared

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Shared
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Tag

Inheritance

object

Tag

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class Tag
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

InverseParentTag

Declaration

```
public virtual ICollection<Tag> InverseParentTag { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Tag>	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

ParentTag

Declaration

```
public virtual Tag? ParentTag { get; set; }
```

Property Value

TYPE	DESCRIPTION
Tag	

ParentTagId

Declaration

```
public int? ParentTagId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

TagAssigments

Declaration

```
public virtual ICollection<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TagAssigment>	

User

Declaration

```
public virtual User? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

Class TagAssigment

Inheritance

object

TagAssigment

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class TagAssigment
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Tag

Declaration

```
public virtual Tag Tag { get; set; }
```

Property Value

TYPE	DESCRIPTION
Tag	

TagId

Declaration

```
public int TagId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class TestEntries1

Inheritance

object

TestEntries1

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class TestEntries1
```

Properties

EntryName

Declaration

```
public string? EntryName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

TestEntries2s

Declaration

```
public virtual ICollection<TestEntries2> TestEntries2s { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TestEntries2>	

Class TestEntries2

Inheritance

object

TestEntries2

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class TestEntries2
```

Properties

EntryProperty

Declaration

```
public string? EntryProperty { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

TestForeignKey

Declaration

```
public int? TestForeignKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

TestForeignKeyNavigation

Declaration

```
public virtual TestEntries1? TestForeignKeyNavigation { get; set; }
```

Property Value

TYPE	DESCRIPTION
TestEntries1	

Class User

Inheritance

object

User

Namespace: PicScam.Models

Assembly: PicScam.dll

Syntax

```
public class User
```

Properties

Boughts

Declaration

```
public virtual ICollection<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Bought>	

Comments

Declaration

```
public virtual ICollection<Comment> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Comment>	

Email

Declaration

```
public string Email { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsPremium

Declaration

```
public byte IsPremium { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Owners

Declaration

```
public virtual ICollection<Owner> Owners { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Owner>	

Password

Declaration

```
public string Password { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

PhotoUrl

Declaration

```
public string? PhotoUrl { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

PremiumDate

Declaration

```
public DateTime? PremiumDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

Shareds

Declaration

```
public virtual ICollection<Shared> Shareds { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Shared>	

Surname

Declaration

```
public string Surname { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagAssigments

Declaration

```
public virtual ICollection<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TagAssigment>	

Tags

Declaration

```
public virtual ICollection<Tag> Tags { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Tag>	

Namespace PicScam.PicWindow

Classes

BingMapsGeocodeResponse

Response model for the Bing Maps Geocode API (JSON)

BingMapsGeocodeResponse.Address

BingMapsGeocodeResponse.Point

BingMapsGeocodeResponse.Resource

BingMapsGeocodeResponse.ResourceSet

CommentComponent

Component allowing the window to display, delete and post comments.

Geodata

Geodata window, allows the user to set the geolocation of a picture

PictureDataDisplayComponent

PictureRemovalComponent

Component allowing for proper picture removal - takes care of all dependent entities. Is able to delete the picture data and/or remove it from PicScam system.

SellPic

Window in which the user can add an offer to a picture or remove current active offer.

SellPic.OfferSimple

Simple class for displaying offers in the list view.

ServerLocalIntegrationComponent

SharePic

Window in which the owner of a picture can share access to that picture to other users.

SharePic.UserSimple

Simple user class for displaying users in the list view.

TagAssigmentDisplayComponent

Component enabling the display of the picture's tags.

TagAssigmentDisplayViewModel

ViewModel for displaying a single Tag.

TagAssigmentEdit

Window for adding and removing tags from a picture. Works both on universally seen tags and tags set to be seen only by the current user.

TagAssigmentEditComponent

Component for TagAssigment manipulation.

TagAssigmentEditViewModel

ViewModel for displaying a single editable TagAssigment.

Class BingMapsGeocodeResponse

Response model for the Bing Maps Geocode API (JSON)

Inheritance

object

BingMapsGeocodeResponse

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class BingMapsGeocodeResponse
```

Properties

AuthenticationResultCode

Declaration

```
[JsonProperty("authenticationResultCode")]
public string AuthenticationResultCode { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

BrandLogoUri

Declaration

```
[JsonProperty("brandLogoUri")]
public string BrandLogoUri { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Copyright

Declaration

```
[JsonProperty("copyright")]
public string Copyright { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

ResourceSets

Declaration

```
[JsonProperty("resourceSets")]
public List<BingMapsGeocodeResponse.ResourceSet> ResourceSets { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<BingMapsGeocodeResponse.ResourceSet>	

StatusCode

Declaration

```
[JsonProperty("statusCode")]
public int StatusCode { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

StatusDescription

Declaration

```
[JsonProperty("statusDescription")]
public string StatusDescription { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class BingMapsGeocodeResponse.Address

Inheritance

object

BingMapsGeocodeResponse.Address

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class BingMapsGeocodeResponse.Address
```

Properties

AddressLine

Declaration

```
[JsonProperty("addressLine")]
public string AddressLine { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

AdminDistrict

Declaration

```
[JsonProperty("adminDistrict")]
public string AdminDistrict { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

AdminDistrict2

Declaration

```
[JsonProperty("adminDistrict2")]
public string AdminDistrict2 { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CountryRegion

Declaration

```
[JsonProperty("countryRegion")]
public string CountryRegion { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FormattedAddress

Declaration

```
[JsonProperty("formattedAddress")]
public string FormattedAddress { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Locality

Declaration

```
[JsonProperty("locality")]
public string Locality { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

PostalCode

Declaration

```
[JsonProperty("postalCode")]
public string PostalCode { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class BingMapsGeocodeResponse.Point

Inheritance

object

BingMapsGeocodeResponse.Point

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class BingMapsGeocodeResponse.Point
```

Properties

Coordinates

Declaration

```
[JsonProperty("coordinates")]
public List<double> Coordinates { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<double>	

Type

Declaration

```
[JsonProperty("type")]
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class BingMapsGeocodeResponse.Resource

Inheritance

object

BingMapsGeocodeResponse.Resource

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class BingMapsGeocodeResponse.Resource
```

Properties

Address

Declaration

```
[JsonProperty("address")]
public BingMapsGeocodeResponse.Address Address { get; set; }
```

Property Value

TYPE	DESCRIPTION
BingMapsGeocodeResponse.Address	

BoundingBox

Declaration

```
[JsonProperty("bbox")]
public List<double> BoundingBox { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<double>	

Name

Declaration

```
[JsonProperty("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Point

Declaration

```
[JsonProperty("point")]
public BingMapsGeocodeResponse.Point Point { get; set; }
```

Property Value

TYPE	DESCRIPTION
BingMapsGeocodeResponse.Point	

Type

Declaration

```
[JsonProperty("__type")]
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class BingMapsGeocodeResponse.ResourceSet

Inheritance

object

BingMapsGeocodeResponse.ResourceSet

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class BingMapsGeocodeResponse.ResourceSet
```

Properties

EstimatedTotal

Declaration

```
[JsonProperty("estimatedTotal")]
public int EstimatedTotal { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Resources

Declaration

```
[JsonProperty("resources")]
public List<BingMapsGeocodeResponse.Resource> Resources { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<BingMapsGeocodeResponse.Resource>	

Class CommentComponent

Component allowing the window to display, delete and post comments.

Inheritance

object

BaseComponent

CommentComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class CommentComponent : BaseComponent
```

Constructors

CommentComponent(Pic)

Declaration

```
public CommentComponent(Pic mediator)
```

Parameters

TYPE	NAME	DESCRIPTION
Pic	mediator	

Fields

_mediator

Overrides the standard `_mediator` field found in components, because the comment component is always connected to a Pic window.

Declaration

```
public Pic _mediator
```

Field Value

TYPE	DESCRIPTION
Pic	

Properties

CharCount

Binds to the displayed current comment character count.

Declaration

```
public string CharCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CommentFontStyleBinding

Binds to the style of the displayed text in the new comment TextBox.

Declaration

```
public FontStyle CommentFontStyleBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
FontStyle	

CommentStackPanel

The GUI comment container.

Declaration

```
public StackPanel? CommentStackPanel { get; set; }
```

Property Value

TYPE	DESCRIPTION
StackPanel	

CommentTextBinding

Binds to the text visibile in the new comment.

Declaration

```
public string CommentTextBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

NewCommentBegan

Specifies if user began writing a new comment.

Declaration

```
public bool NewCommentBegan { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Methods

BeginNewCommentIfPossible()

Begins new comment if it hasn't already been started.

Declaration

```
public void BeginNewCommentIfPossible()
```

ClearNewComment()

Resets message displayed in the new comment TextBox.

Declaration

```
public void ClearNewComment()
```

CreateComment(string, int)

Returns a new Comment object with the specified text and userID.

Declaration

```
public Comment CreateComment(string text, int authorId)
```

Parameters

TYPE	NAME	DESCRIPTION
string	text	New comment text.
int	authorId	New comment author ID.

Returns

TYPE	DESCRIPTION
Comment	A new Comment object.

DeleteCommentButton_Click(object, RoutedEventArgs)

Displays a confirmation MessageBox and deletes comment if user clicks "YES".

Declaration

```
public void DeleteCommentButton_Click(object sender, RoutedEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	The "Delete" comment button.
RoutedEventArgs	e	Empty.

RefreshCommentSection()

Clears the current comment section and displays it again basing on data from database.

Declaration

```
public void RefreshCommentSection()
```

UpdateCharCount(string)

Updates the visible character count.

Declaration

```
public void UpdateCharCount(string textContent)
```

Parameters

TYPE	NAME	DESCRIPTION
string	textContent	The text of the current comment whose lenght should be displayed.

Class Geodata

Geodata window, allows the user to set the geolocation of a picture

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
Geodata

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class Geodata : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild
```

Constructors

Geodata(int, bool)

Constructor for the Geodata window, sets the map view to the geolocation of the picture (if available)

Declaration

```
public Geodata(int pictureId, bool isPictureLocal)
```

Parameters

TYPE	NAME	DESCRIPTION
int	pictureId	id of the picture in db
bool	isPictureLocal	true if picture is stored locally

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement

IQueryAmbient

IAddChild

Class PictureDataDisplayComponent

Inheritance

object
BaseComponent
PictureDataDisplayComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class PictureDataDisplayComponent : BaseComponent
```

Constructors

PictureDataDisplayComponent(IMediator)

Declaration

```
public PictureDataDisplayComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Properties

CreationDateBinding

Declaration

```
public string CreationDateBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DateTakenBinding

Declaration

```
public string DateTakenBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

DimensionsBinding

Declaration

```
public string DimensionsBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FileFormatBinding

Declaration

```
public string FileFormatBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

LatitudeBinding

Declaration

```
public string LatitudeBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

LongitudeBinding

Declaration

```
public string LongitudeBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

SizeBinding

Declaration

```
public string SizeBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

StorageTypeBinding

Declaration

```
public string StorageTypeBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Methods

DisplayServerExclusiveData(Picture)

Set bindings that are always dependent on the server storage. Bindings set: CreationDateBinding, DateTakenBinding, LatitudeBinding, LongitudeBinding.

Declaration

```
public void DisplayServerExclusiveData(Picture picture)
```

Parameters

TYPE	NAME	DESCRIPTION
Picture	picture	The server entity.

DisplaySourceSpecificData(long, int, int, string)

Set bindings that are dependent on the data in the local/server storage. Bindings set: DimensionsBinding, SizeBinding, FileFormatBinding.

Declaration

```
public void DisplaySourceSpecificData(long size, int width, int height, string fileFormat)
```

Parameters

TYPE	NAME	DESCRIPTION
long	size	File size.
int	width	Picture width.
int	height	Picture height.
string	fileFormat	Picture file format.

Class PictureRemovalComponent

Component allowing for proper picture removal - takes care of all dependent entities. Is able to delete the picture data and/or remove it from PicScam system.

Inheritance

object
BaseComponent
PictureRemovalComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class PictureRemovalComponent : BaseComponent
```

Constructors

PictureRemovalComponent(Pic)

Declaration

```
public PictureRemovalComponent(Pic mediator)
```

Parameters

TYPE	NAME	DESCRIPTION
Pic	mediator	

Fields

_mediator

Overrides _mediator field. (PictureRemovalComponent is used only in the Pic window)

Declaration

```
public Pic _mediator
```

Field Value

TYPE	DESCRIPTION
Pic	

Properties

RemovePictureEntryPopupBinding

Help message upon hover on the "Remove Entry" button.

Declaration

```
public string RemovePictureEntryPopupBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

RemovePictureFilePopupBinding

Help message upon hover on the "Remove File" button.

Declaration

```
public string RemovePictureFilePopupBinding { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Methods

DeletePictureData()

Deletes the picture data. If picture is server side: Removes picture file from server storage. If local copies still exist on any computer, tags and comments etc. are unaffected.

If picture is local: Removes picture from Picscam system and then deletes its file from disc. If a server copy or a local copy on any computer still exists, tags and comments etc. are unaffected

Declaration

```
public bool DeletePictureData()
```

Returns

TYPE	DESCRIPTION
bool	

DeletePictureEntry()

Deletes the picture entry from database. If picture is server side: Removes picture file stored on the server and deletes all additional info (tags, comments, etc.). Local copies are dissatached from the system.

If picture is local: Removes picture from Picscam system without deleting the file on local disc. If a server copy or a local copy on any computer still exists, tags and comments etc. are unaffected.

Declaration

```
public bool DeletePictureEntry()
```

Returns

TYPE	DESCRIPTION
bool	

SetRemovalButtonHelpText(bool)

Sets "Remove File" and "Remove Entry" button help messages.

Declaration

```
public void SetRemovalButtonHelpText(bool isLocal)
```

Parameters

TYPE	NAME	DESCRIPTION
bool	isLocal	Is picture from local storage.

Class SellPic

Window in which the user can add an offer to a picture or remove current active offer.

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
SellPic

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild
IMediator
INotifyPropertyChanged

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class SellPic : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild, IMediator, INotifyPropertyChanged
```

Constructors

SellPic(int)

Initializes the window. Loads offers for the picture and displays them in the list view.

Declaration

```
public SellPic(int pictureId)
```

Parameters

TYPE	NAME	DESCRIPTION
int	pictureId	id of the selled pic

Properties

_localContext

Database context for local database.

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Database context for server database.

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

FillListView(List<Offer>)

Fills the list view with offers.

Declaration

```
public void FillListView(List<Offer> myOffers)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Offer>	myOffers	list of offers to be displayed

Notify(object, string)

Handles messages coming in from the components.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	One of the components.
string	eventName	Type of the event.

Events

PropertyChanged

Raised when a data binded value has been changed and GUI needs an update.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

IMediator

INotifyPropertyChanged

Class SellPic.OfferSimple

Simple class for displaying offers in the list view.

Inheritance

object

SellPic.OfferSimple

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class SellPic.OfferSimple
```

Properties

Active

Declaration

```
public string Active { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CheckBoxVisibility

Declaration

```
public Visibility CheckBoxVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsChecked

Declaration

```
public bool IsChecked { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Message

Declaration

```
public string Message { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

NumOfSells

Declaration

```
public int NumOfSells { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Price

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Class ServerLocalIntegrationComponent

Inheritance

object

BaseComponent

ServerLocalIntegrationComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class ServerLocalIntegrationComponent : BaseComponent
```

Constructors

ServerLocalIntegrationComponent(Pic)

Declaration

```
public ServerLocalIntegrationComponent(Pic mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Pic	mediator	

Fields

_mediator

Overrides _mediator field. (PictureRemovalComponent is used only in the Pic window)

Declaration

```
public Pic _mediator
```

Field Value

TYPE	DESCRIPTION
Pic	

Properties

SellButtonVisibility

Specifies if "Upload to server/Download from server" button should be accessible.

Declaration

```
public Visibility SellButtonVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

SwitchSourceButtonText

Binds to text displayed on the "Go to server picture"/"Go to local picture" button.

Binds to text displayed on the "Go to server picture"/"Go to local picture" button.

Declaration

```
public string SwitchSourceButtonText { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

SwitchSourceButtonVisibility

Specifies if "Go to server picture"/"Go to local picture" button should be accessible.

Declaration

```
public Visibility SwitchSourceButtonVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

UploadButtonText

Binds to text displayed on the "Upload to server/Download from server" button.

Declaration

```
public string UploadButtonText { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

UploadButtonVisibility

Specifies if "Upload to server/Download from server" button should be accessible.

Declaration

```
public Visibility UploadButtonVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

Methods

DownloadFromServer(Picture)

Downloads server picture data referenced in mediator to the local disc folder.

Declaration

```
public void DownloadFromServer(Picture localPicture)
```

Parameters

TYPE	NAME	DESCRIPTION
Picture	localPicture	

Setup()

Sets visibility and text on buttons.

Declaration

```
public void Setup()
```

UploadToServer()

Uploads local picture data referenced in mediator to the server.

Declaration

```
public void UploadToServer()
```

Class SharePic

Window in which the owner of a picture can share access to that picture to other users.

Inheritance

object
DispatcherObject
DependencyObject
Visual
UIElement
FrameworkElement
Control
ContentControl
Window
SharePic

Implements

IAnimatable
ISupportInitialize
IFrameworkInputElement
IInputElement
IQueryAmbient
IAddChild
IMediator
INotifyPropertyChanged

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class SharePic : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement,
IQueryAmbient, IAddChild, IMediator, INotifyPropertyChanged
```

Constructors

SharePic(int)

Initializes the window - connects listview and combobox to their data sources and displays the users with whom we shared the photo.

Declaration

```
public SharePic(int pictureId)
```

Parameters

TYPE	NAME	DESCRIPTION
int	pictureId	id of the current picture

Properties

_localContext

Database context for local database.

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Database context for server database.

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

Notify(object, string)

Handles messages coming in from the components.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	One of the components.
string	eventName	Type of the event.

Events

PropertyChanged

Raised when a data binded value has been changed and GUI needs an update.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IAnimatable

ISupportInitialize

IFrameworkInputElement

IInputElement

IQueryAmbient

IAddChild

IMediator

INotifyPropertyChanged

Class SharePic.UserSimple

Simple user class for displaying users in the list view.

Inheritance

object

SharePic.UserSimple

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class SharePic.UserSimple
```

Properties

CheckBoxVisibility

Declaration

```
public Visibility CheckBoxVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

Email

Declaration

```
public string Email { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsBuyer

Declaration

```
public bool IsBuyer { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

IsChecked

Declaration

```
public bool IsChecked { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Message

Declaration

```
public string Message { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Class TagAssigmentDisplayComponent

Component enabling the display of the picture's tags.

Inheritance

object

BaseComponent

TagAssigmentDisplayComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class TagAssigmentDisplayComponent : BaseComponent
```

Constructors

TagAssigmentDisplayComponent(IMediator)

Declaration

```
public TagAssigmentDisplayComponent(IMediator mediator = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	

Properties

TagAssigmentDisplayViewModel

Current picture's tag list which is binded to the display.

Declaration

```
public List<TagAssigmentDisplayViewModel> TagAssigmentDisplayViewModel { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<TagAssigmentDisplayViewModel>	

Methods

RefreshDisplayViewModel(List<TagAssigment>, int)

Refreshes tag display.

Declaration

```
public void RefreshDisplayViewModel(List<TagAssigment> pictureTags, int userId)
```

Parameters

TYPE	NAME	DESCRIPTION
List<TagAssigment>	pictureTags	All the picture's tag entities.

TYPE	NAME	DESCRIPTION
int	userId	Current user id.

Class TagAssigmentDisplayViewModel

ViewModel for displaying a single Tag.

Inheritance

object

TagAssigmentDisplayViewModel

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class TagAssigmentDisplayViewModel
```

Properties

TagColor

The tag's background color.

Declaration

```
public Brush TagColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
Brush	

TagDisplayName

Name visible on the tag.

Declaration

```
public string TagDisplayName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagFullName

Full name visibile on hover.

Declaration

```
public string TagFullName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagVisibilityIcon

Binds to a globe icon for universally seen tags, binds to an user icon for tags seen only by the current user.

Declaration

```
public BitmapImage TagVisibilityIcon { get; set; }
```

Property Value

TYPE	DESCRIPTION
BitmapImage	

Class TagAssigmentEdit

Window for adding and removing tags from a picture. Works both on universally seen tags and tags set to be seen only by the current user.

Inheritance

object

TagAssigmentEdit

Implements

IMediator

INotifyPropertyChanged

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class TagAssigmentEdit : IMediator, INotifyPropertyChanged
```

Constructors

TagAssigmentEdit(List<Tag>, List<TagAssigment>, int, bool)

Initializes window by filling the displayed Tag list.

Declaration

```
public TagAssigmentEdit(List<Tag> userTags, List<TagAssigment> pictureTags, int pictureId, bool isOwner)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Tag>	userTags	Current user's tags.
List<TagAssigment>	pictureTags	Viewed picture's all tags.
int	pictureId	Viewed picture ID.
bool	isOwner	Is current user the owner of the picture.

Properties

TagAssigmentEditComponent

Component for TagAssigment manipulation.

Declaration

```
public TagAssigmentEditComponent TagAssigmentEditComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
TagAssigmentEditComponent	

_localContext

Declaration

```
public PicscamContext _localContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
PicscamContext	

_serverContext

Declaration

```
public KirschniPicscamContext _serverContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
KirschniPicscamContext	

Methods

Notify(object, string)

Handles messages coming in from the components.

Declaration

```
public void Notify(object sender, string eventName)
```

Parameters

TYPE	NAME	DESCRIPTION
object	sender	
string	eventName	

Events

NotifyParentWindow

Raised when this window's parent window should be notified.

Declaration

```
public event EventHandler NotifyParentWindow
```

Event Type

TYPE	DESCRIPTION
EventHandler	

PropertyChanged

Raised when modifications to the database should cause a main menu refresh.

Declaration

```
public event PropertyChangedEventHandler? PropertyChanged
```

Event Type

TYPE	DESCRIPTION
PropertyChangedEventHandler	

Implements

IMediator
INotifyPropertyChanged

Class TagAssigmentEditComponent

Component for TagAssigment manipulation.

Inheritance

object

BaseComponent

TagAssigmentEditComponent

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class TagAssigmentEditComponent : BaseComponent
```

Constructors

TagAssigmentEditComponent(IMediator, bool)

Declaration

```
public TagAssigmentEditComponent(IMediator mediator, bool AllowGlobalTagAssigment)
```

Parameters

TYPE	NAME	DESCRIPTION
IMediator	mediator	
bool	AllowGlobalTagAssigment	

Properties

AllowGlobalTagAssigment

Specifies if current user can modify global tags on the picture.

Declaration

```
public bool AllowGlobalTagAssigment { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

GlobalTagIconVisibility

Specifies if a global tag icon (Globe) should be displayed.

Declaration

```
public Visibility GlobalTagIconVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

TagAssigmentEditViewModel

Binds displayed GUI list to the ViewModel.

Declaration

```
public List<TagAssigmentEditViewModel> TagAssigmentEditViewModel { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<TagAssigmentEditViewModel>	

Methods

FillListView(List<Tag>, List<TagAssigment>)

Sets display and checkbox status for all user tags.

Declaration

```
public void FillListView(List<Tag> userTags, List<TagAssigment> pictureTags)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Tag>	userTags	All current user's tags.
List<TagAssigment>	pictureTags	All picture's tags.

SaveTagState(List<TagAssigment>, int, int, KirschniPicscamContext)

Creates/deletes TagAssigments for all user Tags depending on checkbox status.

Declaration

```
public void SaveTagState(List<TagAssigment> pictureTags, int pictureId, int currentUserId,
KirschniPicscamContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
List<TagAssigment>	pictureTags	All picture's tags.
int	pictureId	Server picture ID.
int	currentUserId	Current user ID.
KirschniPicscamContext	context	Server context to update.

Class TagAssigmentEditViewModel

ViewModel for displaying a single editable TagAssigment.

Inheritance

object

TagAssigmentEditViewModel

Namespace: PicScam.PicWindow

Assembly: PicScam.dll

Syntax

```
public class TagAssigmentEditViewModel
```

Properties

GlobalTagCheckboxVisibility

Specifies if this tag's global checkbox can be selected by current user.

Declaration

```
public Visibility GlobalTagCheckboxVisibility { get; set; }
```

Property Value

TYPE	DESCRIPTION
Visibility	

GlobalTagChecked

Specifies if this tag's global checkbox has been selected.

Declaration

```
public bool GlobalTagChecked { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

TagDisplayDescription

Tag description visible directly in the GUI.

Declaration

```
public string TagDisplayDescription { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagDisplayName

Tag name visible directly in the GUI.

Declaration

```
public string TagDisplayName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagFullDescription

Tag description visibile on hover.

Declaration

```
public string TagFullDescription { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagFullName

Tag name visible on hover.

Declaration

```
public string TagFullName { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagId

The TagAssigment's referenced TagId.

Declaration

```
public int TagId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

UserTagChecked

Specifies if this tag's user-specific checkbox has been selected.

Declaration

```
public bool UserTagChecked { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Namespace PicScam.Properties

Classes

Resources

A strongly-typed resource class, for looking up localized strings, etc.

Class Resources

A strongly-typed resource class, for looking up localized strings, etc.

Inheritance

object

Resources

Namespace: PicScam.Properties

Assembly: PicScam.dll

Syntax

```
[GeneratedCode("System.Resources.Tools.StronglyTypedResourceBuilder", "17.0.0.0")]
[DebuggerNonUserCode]
[CompilerGenerated]
public class Resources
```

Properties

Culture

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

Declaration

```
[EditorBrowsable(EditorBrowsableState.Advanced)]
public static CultureInfo Culture { get; set; }
```

Property Value

TYPE	DESCRIPTION
CultureInfo	

ResourceManager

Returns the cached ResourceManager instance used by this class.

Declaration

```
[EditorBrowsable(EditorBrowsableState.Advanced)]
public static ResourceManager ResourceManager { get; }
```

Property Value

TYPE	DESCRIPTION
ResourceManager	

Namespace PicScam.ServerModels

Classes

Bought

Comment

KirschniPicscamContext

Offer

Owner

PasswordReset

Picture

Shared

Tag

TagAssigment

User

Class Bought

Inheritance

object

Bought

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Bought
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Offer

Declaration

```
public virtual Offer Offer { get; set; }
```

Property Value

TYPE	DESCRIPTION
Offer	

OfferId

Declaration

```
public int OfferId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Comment

Inheritance

object

Comment

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Comment
```

Properties

Content

Declaration

```
public string Content { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Picture

Declaration

```
public virtual Picture Picture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class KirschniPicscamContext

Inheritance

object
DbContext
KirschniPicscamContext

Implements

IInfrastructure<IServiceProvider>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class KirschniPicscamContext : DbContext, IInfrastructure<IServiceProvider>, IDbContextDependencies,
IDbSetCache, IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable
```

Constructors

KirschniPicscamContext()

Declaration

```
public KirschniPicscamContext()
```

KirschniPicscamContext(DbContextOptions<KirschniPicscamContext>)

Declaration

```
public KirschniPicscamContext(DbContextOptions<KirschniPicscamContext> options)
```

Parameters

TYPE	NAME	DESCRIPTION
DbContextOptions<KirschniPicscamContext>	options	

Properties

Boughts

Declaration

```
public virtual DbSet<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Bought>	

Comments

Declaration

```
public virtual DbSet<Comment> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Comment>	

Offers

Declaration

```
public virtual DbSet<Offer> Offers { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Offer>	

Owners

Declaration

```
public virtual DbSet<Owner> Owners { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Owner>	

PasswordResets

Declaration

```
public virtual DbSet<PasswordReset> PasswordResets { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<PasswordReset>	

Pictures

Declaration

```
public virtual DbSet<Picture> Pictures { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Picture>	

Shareds

Declaration

```
public virtual DbSet<Shared> Shareds { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Shared>	

TagAssigments

Declaration

```
public virtual DbSet<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<TagAssigment>	

Tags

Declaration

```
public virtual DbSet<Tag> Tags { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<Tag>	

Users

Declaration

```
public virtual DbSet<User> Users { get; set; }
```

Property Value

TYPE	DESCRIPTION
DbSet<User>	

Methods

OnConfiguring(DbContextOptionsBuilder)

Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created. The base implementation does nothing.

Declaration

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
DbContextOptionsBuilder	optionsBuilder	A builder used to create or modify options for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure the context.

Overrides

DbContext.OnConfiguring(DbContextOptionsBuilder)

Remarks

In situations where an instance of DbContextOptions may or may not have been passed to the constructor, you can use IsConfigured to determine if the options have already been set, and skip some or all of the logic in OnConfiguring(DbContextOptionsBuilder).

See DbContext lifetime, configuration, and initialization for more information and examples.

OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in DbSet<TEntity> properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

Declaration

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

TYPE	NAME	DESCRIPTION
ModelBuilder	modelBuilder	The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

Overrides

DbContext.OnModelCreating(ModelBuilder)

Remarks

If a model is explicitly set on the options for this context (via UseModel(IModel)) then this method will not be run. However, it will still run when creating a compiled model.

See Modeling entity types and relationships for more information and examples.

Implements

IInfrastructure<T>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService
IDisposable
IAsyncDisposable

Class Offer

Inheritance

object

Offer

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Offer
```

Properties

Boughts

Declaration

```
public virtual ICollection<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Bought>	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsActive

Declaration

```
public ulong IsActive { get; set; }
```

Property Value

TYPE	DESCRIPTION
ulong	

Picture

Declaration

```
public virtual Picture Picture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Price

Declaration

```
public decimal Price { get; set; }
```

Property Value

TYPE	DESCRIPTION
decimal	

Class Owner

Inheritance

object

Owner

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Owner
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Picture

Declaration

```
public virtual Picture Picture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class PasswordReset

Inheritance

object

PasswordReset

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class PasswordReset
```

Properties

CodeHash

Declaration

```
public string CodeHash { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

ValidDate

Declaration

```
public DateTime ValidDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Class Picture

Inheritance

object

Picture

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Picture
```

Properties

Comments

Declaration

```
public virtual ICollection<Comment> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Comment>	

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

DateTaken

Declaration

```
public DateOnly? DateTaken { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateOnly?	

Description

Declaration

```
public string? Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FileExtension

Declaration

```
public string FileExtension { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

FullPicture

Declaration

```
public byte[]? FullPicture { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

Geolocation

Declaration

```
public Point? Geolocation { get; set; }
```

Property Value

TYPE	DESCRIPTION
Point	

Height

Declaration

```
public int Height { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

LocalPictureAmount

Declaration

```
public int LocalPictureAmount { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Offers

Declaration

```
public virtual ICollection<Offer> Offers { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Offer>	

Owners

Declaration

```
public virtual ICollection<Owner> Owners { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Owner>	

Shareds

Declaration

```
public virtual ICollection<Shared> Shareds { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Shared>	

Size

Declaration

```
public int Size { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

TagAssigments

Declaration

```
public virtual ICollection<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TagAssigment>	

Thumbnail

Declaration

```
public byte[]? Thumbnail { get; set; }
```

Property Value

TYPE	DESCRIPTION
byte[]	

Width

Declaration

```
public int Width { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Shared

Inheritance

object

Shared

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Shared
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Picture

Declaration

```
public virtual Picture Picture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class Tag

Inheritance

object

Tag

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class Tag
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Description

Declaration

```
public string? Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

InverseParentTag

Declaration

```
public virtual ICollection<Tag> InverseParentTag { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Tag>	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

ParentTag

Declaration

```
public virtual Tag? ParentTag { get; set; }
```

Property Value

TYPE	DESCRIPTION
Tag	

ParentTagId

Declaration

```
public int? ParentTagId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

TagAssigments

Declaration

```
public virtual ICollection<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TagAssigment>	

User

Declaration

```
public virtual User User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Class TagAssigment

Inheritance

object

TagAssigment

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class TagAssigment
```

Properties

CreationDate

Declaration

```
public DateTime CreationDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Picture

Declaration

```
public virtual Picture Picture { get; set; }
```

Property Value

TYPE	DESCRIPTION
Picture	

PictureId

Declaration

```
public int PictureId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

Tag

Declaration

```
public virtual Tag Tag { get; set; }
```

Property Value

TYPE	DESCRIPTION
Tag	

TagId

Declaration

```
public int TagId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

User

Declaration

```
public virtual User? User { get; set; }
```

Property Value

TYPE	DESCRIPTION
User	

UserId

Declaration

```
public int? UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
int?	

Class User

Inheritance

object

User

Namespace: PicScam.ServerModels

Assembly: PicScam.dll

Syntax

```
public class User
```

Properties

Boughts

Declaration

```
public virtual ICollection<Bought> Boughts { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Bought>	

Comments

Declaration

```
public virtual ICollection<Comment> Comments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Comment>	

Email

Declaration

```
public string Email { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Id

Declaration

```
public int Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
int	

IsPremium

Declaration

```
public bool IsPremium { get; set; }
```

Property Value

TYPE	DESCRIPTION
bool	

Name

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Owners

Declaration

```
public virtual ICollection<Owner> Owners { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Owner>	

Password

Declaration

```
public string Password { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

PasswordResets

Declaration

```
public virtual ICollection<PasswordReset> PasswordResets { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<PasswordReset>	

PremiumDate

Declaration

```
public DateTime? PremiumDate { get; set; }
```

Property Value

TYPE	DESCRIPTION
DateTime?	

ProfilePictureUrl

Declaration

```
public string? ProfilePictureUrl { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

Shareds

Declaration

```
public virtual ICollection<Shared> Shareds { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Shared>	

Surname

Declaration

```
public string Surname { get; set; }
```

Property Value

TYPE	DESCRIPTION
string	

TagAssigments

Declaration

```
public virtual ICollection<TagAssigment> TagAssigments { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<TagAssigment>	

Tags

Declaration

```
public virtual ICollection<Tag> Tags { get; set; }
```

Property Value

TYPE	DESCRIPTION
ICollection<Tag>	