

```
1 using System;
2 using System.Collections.Generic;
3 using System.Collections.ObjectModel;
4 using System.ComponentModel;
5 using System.Linq;
6 using System.Runtime.CompilerServices;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Input;
10 using System.Windows.Markup;
11 using Vertigo.DAL;
12 using Sfinx;
13
14 namespace Vertigo.ViewModel
15 {
16     public class UmowyKasaVM: ViewModelBase, IRequireViewIdentification
17     {
18         /// <summary>
19         /// Copyrights by KBSOFT
20         /// </summary>
21         public ObservableCollection<Umoney> gridData { get; set; }
22         private Umoney src = new Umoney();
23         private dbcn cn;
24
25         private Umoney _formData;
26         public Umoney formData
27         {
28             get { return _formData; }
29             set
30             {
31                 SetProperty(ref _formData, value);
32             }
33         }
34     }
35
36     private Umoney _selectedRow;
37     public Umoney SelectedRow
38     {
39         set
40         {
41             _selectedRow = value;
42             Extras.CopyPropertiesTo(value, formData);
43             UpdateProperty("formData");
44         }
45     }
46
47     private Guid _viewId;
48     public Guid ViewID
49     {
50         get { return _viewId; }
51     }
52
53     string _kind = "";
54     int _contract_id = 0;
55
56     public UmowyKasaVM()
```

```
57     {
58         _viewId = Guid.NewGuid();
59         _kind = (string)App.Current.Properties["UmowyKasaKind"];
60         _contract_id = (int)App.Current.Properties["UmowyKasaContractId"];
61         cn = new Proxy.DBConnection().Access
62             (Proxy.CreatorConnections.Tester);
63         LoadGridData();
64         formData = src;
65     }
66     void LoadGridData()
67     {
68         src = new Umoney($"where u_contract_id={_contract_id} AND
69             ext='{_kind}'", cn);
70         gridData = new ObservableCollection<Umoney>(src.Data as
71             List<Umoney>);
72     }
73     void RefreshGridData()
74     {
75         LoadGridData();
76         formData = src;
77         UpdateProperty("gridData");
78         UpdateProperty("formData");
79     }
80     public ICommand AddDataCommand
81     {
82         get { return new DelegateCommand((a) => AddData(a), (a) =>
83             IsValidAddData(a), this); }
84     }
85     private void AddData(object v)
86     {
87         UmowyKasaVM ins = (UmowyKasaVM)v;
88         ins.formData.Status = "A";
89         ins.formData.U_Contract_Id = _contract_id;
90         ins.formData.Ext = _kind;
91         int a = ins.formData.Insert(cn);
92         RefreshGridData();
93     }
94     private bool IsValidAddData(object v)
95     {
96         return true;
97     }
98
99
100    public ICommand DeleteDataCommand
101    {
102        get { return new DelegateCommand((a) => DeleteData(a), (a) =>
103            IsValidDeleteData(a), this); }
104    }
105    private void DeleteData(object v)
106    {
107        ((UmowyKasaVM)v)._selectedRow.Delete(cn);
```

```
108 RefreshGridData();
109 }
110
111 private bool IsValidDeleteData(object v)
112 {
113     if (((UmowyKasaVM)v)._selectedRow == null)
114     {
115         DialogBoxShow.ShowVQuick("Zaznacz na liście dane które chcesz
116         usunąć.", "Vertigo - usuwanie danych",
117         DialogBoxShow.DialogBoxButtons.btnExit);
118     }
119     return DialogBoxShow.ShowVQuick("Czy chcesz usunąć wskazane
120     dane?", "Vertigo - usuwanie danych",
121     DialogBoxShow.DialogBoxButtons.btnYesNo);
122 }
123
124 public ICommand UpdateDataCommand
125 {
126     get { return new DelegateCommand((a) => UpdateData(a), (a) =>
127     IsValidUpdateData(a), this); }
128 }
129
130 private void UpdateData(object v)
131 {
132     UmowyKasaVM ins = (UmowyKasaVM)v;
133     ins.formData.Status = ins._selectedRow.Status;
134     ins.formData.U_Contract_Id = ins._selectedRow.U_Contract_Id;
135     ins.formData.Id = ins._selectedRow.Id;
136     ins.formData.Update(cn);
137     RefreshGridData();
138 }
139
140 private bool IsValidUpdateData(object v)
141 {
142     if (((UmowyKasaVM)v)._selectedRow == null)
143     {
144         DialogBoxShow.ShowVQuick("Zaznacz na liście dane które chcesz
145         zmienić.", "Vertigo - zmiana danych",
146         DialogBoxShow.DialogBoxButtons.btnExit);
147     }
148     return true;
149 }
150
151 public ICommand QuitCommand
152 {
153     get { return new DelegateCommand((a) => Quit(a), (a) =>
154     IsValidQuit(a), this); }
155 }
156
157 private void Quit(object v)
158 {
159     WindowManager.CloseWindow(ViewID);
160 }
```

```
156     }  
157  
158     private bool IsValidQuit(object v)  
159     {  
160         return true;  
161     }  
162  
163  
164  
165  
166     }  
167  
168  
169 }  
170
```