

# Tourplanner

1.0

Generated by Doxygen 1.9.1



|  |           |
|--|-----------|
| <b>1 Namespace Index</b>   | <b>1</b>  |
| 1.1 Packages   | 1         |
| <b>2 Hierarchical Index</b>  | <b>3</b>  |
| 2.1 Class Hierarchy  | 3         |
| <b>3 Class Index</b>   | <b>7</b>  |
| 3.1 Class List   | 7         |
| <b>4 Namespace Documentation</b>                                     | <b>13</b> |
| 4.1 BusinessLogicLayer Namespace Reference                           | 13        |
| 4.2 BusinessLogicLayer.Exceptions Namespace Reference                | 13        |
| 4.3 BusinessLogicLayer.ExporterImporter Namespace Reference          | 13        |
| 4.4 BusinessLogicLayer.Factories Namespace Reference                 | 13        |
| 4.5 BusinessLogicLayer.PDFCreation Namespace Reference               | 14        |
| 4.6 Common Namespace Reference                                       | 14        |
| 4.7 Common.Config Namespace Reference                                | 14        |
| 4.8 Common.Entities Namespace Reference                              | 14        |
| 4.9 Common.Enums Namespace Reference                                 | 15        |
| 4.9.1 Enumeration Type Documentation                                 | 15        |
| 4.9.1.1 ERouteType   | 15        |
| 4.9.1.2 ETravelMethod  | 15        |
| 4.9.1.3 EWeather   | 15        |
| 4.10 Common.Exceptions Namespace Reference                           | 15        |
| 4.11 Common.Logging Namespace Reference                              | 16        |
| 4.12 Common.MapQuestClient Namespace Reference                       | 16        |
| 4.13 DataAccessLayer Namespace Reference                             | 16        |
| 4.14 DataAccessLayer.DBCommands Namespace Reference                  | 16        |
| 4.15 DataAccessLayer.DBCommands.ManeuverCommands Namespace Reference | 16        |
| 4.16 DataAccessLayer.DBCommands.TourCommands Namespace Reference     | 17        |
| 4.17 DataAccessLayer.DBCommands.TourLogCommands Namespace Reference  | 17        |
| 4.18 DataAccessLayer.DBConnection Namespace Reference                | 17        |
| 4.19 DataAccessLayer.Exceptions Namespace Reference                  | 18        |
| 4.20 DataAccessLayer.Repositories Namespace Reference                | 18        |
| 4.21 DataAccessLayer.UnitOfWork Namespace Reference                  | 18        |
| 4.22 SWE2_Tourplanner Namespace Reference                            | 19        |
| 4.23 SWE2_Tourplanner.Commands Namespace Reference                   | 19        |
| 4.24 SWE2_Tourplanner.Converters Namespace Reference                 | 19        |
| 4.25 SWE2_Tourplanner.Dialogs Namespace Reference                    | 20        |
| 4.26 SWE2_Tourplanner.ReusableComponents Namespace Reference         | 20        |
| 4.27 SWE2_Tourplanner.ValidationRules Namespace Reference            | 20        |
| 4.28 SWE2_Tourplanner.ViewModels Namespace Reference                 | 21        |
| 4.29 SWE2_Tourplanner_Tests Namespace Reference                      | 21        |

|   |           |
|---|-----------|
| 4.30 SWE2_Tourplanner_Tests.BLTests Namespace Reference                       | 21        |
| 4.31 SWE2_Tourplanner_Tests.CommonTests Namespace Reference                   | 21        |
| 4.32 SWE2_Tourplanner_Tests.DALTests Namespace Reference                      | 22        |
| 4.33 SWE2_Tourplanner_Tests.UITests Namespace Reference                       | 22        |
| 4.34 XamlGeneratedNamespace Namespace Reference                               | 22        |
| <b>5 Class Documentation</b>  | <b>23</b> |
| 5.1 SWE2_Tourplanner.App Class Reference                                      | 23        |
| 5.1.1 Detailed Description  | 24        |
| 5.1.2 Member Function Documentation   | 24        |
| 5.1.2.1 InitializeComponent() [1/4]   | 24        |
| 5.1.2.2 InitializeComponent() [2/4]   | 24        |
| 5.1.2.3 InitializeComponent() [3/4]   | 24        |
| 5.1.2.4 InitializeComponent() [4/4]   | 25        |
| 5.1.2.5 Main() [1/4]  | 25        |
| 5.1.2.6 Main() [2/4]  | 25        |
| 5.1.2.7 Main() [3/4]  | 25        |
| 5.1.2.8 Main() [4/4]  | 25        |
| 5.1.2.9 OnStartup()   | 25        |
| 5.2 SWE2_Tourplanner.ViewModels.BaseViewModel Class Reference                 | 26        |
| 5.2.1 Detailed Description  | 27        |
| 5.2.2 Member Function Documentation   | 27        |
| 5.2.2.1 OnPropertyChanged()   | 27        |
| 5.2.3 Event Documentation   | 27        |
| 5.2.3.1 PropertyChanged   | 27        |
| 5.3 BusinessLogicLayer.Exceptions.BLExporterImporterException Class Reference | 27        |
| 5.3.1 Detailed Description  | 28        |
| 5.3.2 Constructor & Destructor Documentation                                  | 28        |
| 5.3.2.1 BLExporterImporterException() [1/3]                                   | 28        |
| 5.3.2.2 BLExporterImporterException() [2/3]                                   | 28        |
| 5.3.2.3 BLExporterImporterException() [3/3]                                   | 28        |
| 5.4 BusinessLogicLayer.Exceptions.BLFactoryException Class Reference          | 29        |
| 5.4.1 Detailed Description  | 29        |
| 5.4.2 Constructor & Destructor Documentation                                  | 29        |
| 5.4.2.1 BLFactoryException() [1/3]  | 29        |
| 5.4.2.2 BLFactoryException() [2/3]  | 29        |
| 5.4.2.3 BLFactoryException() [3/3]  | 30        |
| 5.5 BusinessLogicLayer.Exceptions.BLPDFCreationException Class Reference      | 30        |
| 5.5.1 Detailed Description  | 31        |
| 5.5.2 Constructor & Destructor Documentation                                  | 31        |
| 5.5.2.1 BLPDFCreationException() [1/3]  | 31        |
| 5.5.2.2 BLPDFCreationException() [2/3]  | 31        |

|  |    |
|--|----|
| 5.5.2.3 BLPDFCreationException() [3/3]                                       | 31 |
| 5.6 Common.Exceptions.CommonConfigException Class Reference                  | 32 |
| 5.6.1 Detailed Description   | 32 |
| 5.6.2 Constructor & Destructor Documentation                                 | 32 |
| 5.6.2.1 CommonConfigException() [1/3]  | 32 |
| 5.6.2.2 CommonConfigException() [2/3]  | 32 |
| 5.6.2.3 CommonConfigException() [3/3]  | 33 |
| 5.7 SWE2_Tourplanner_Tests.UITests.ConverterTests Class Reference            | 33 |
| 5.8 SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel Class Reference | 33 |
| 5.8.1 Detailed Description   | 34 |
| 5.8.2 Constructor & Destructor Documentation                                 | 34 |
| 5.8.2.1 CreateUpdateTourLogViewModel()                                       | 34 |
| 5.8.3 Member Data Documentation  | 35 |
| 5.8.3.1 tourLog  | 35 |
| 5.8.4 Property Documentation   | 35 |
| 5.8.4.1 ConductCreateUpdateCommand   | 35 |
| 5.8.4.2 ConfirmButtonMessage   | 35 |
| 5.8.4.3 Distance   | 35 |
| 5.8.4.4 EndDate  | 35 |
| 5.8.4.5 ExitCommand  | 36 |
| 5.8.4.6 HeadingMessage   | 36 |
| 5.8.4.7 Rating   | 36 |
| 5.8.4.8 Report   | 36 |
| 5.8.4.9 StartDate  | 36 |
| 5.8.4.10 Temperature   | 36 |
| 5.8.4.11 TravelMethod  | 36 |
| 5.8.4.12 TravelMethods   | 37 |
| 5.8.4.13 Weather   | 37 |
| 5.8.4.14 WeatherTypes  | 37 |
| 5.8.5 Event Documentation  | 37 |
| 5.8.5.1 CloseRequested   | 37 |
| 5.9 SWE2_Tourplanner.CreateUpdateTourView Class Reference                    | 38 |
| 5.9.1 Detailed Description   | 39 |
| 5.9.2 Member Function Documentation  | 39 |
| 5.9.2.1 InitializeComponent() [1/4]  | 39 |
| 5.9.2.2 InitializeComponent() [2/4]  | 39 |
| 5.9.2.3 InitializeComponent() [3/4]  | 39 |
| 5.9.2.4 InitializeComponent() [4/4]  | 39 |
| 5.10 SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel Class Reference   | 40 |
| 5.10.1 Detailed Description  | 41 |
| 5.10.2 Constructor & Destructor Documentation                                | 41 |
| 5.10.2.1 CreateUpdateTourViewModel()   | 41 |

|   |    |
|---|----|
| 5.10.3 Member Data Documentation  | 41 |
| 5.10.3.1 maneuvers  | 41 |
| 5.10.3.2 maneuversViewModel   | 41 |
| 5.10.3.3 manualTour   | 42 |
| 5.10.3.4 tour   | 42 |
| 5.10.4 Property Documentation   | 42 |
| 5.10.4.1 CommitCommand  | 42 |
| 5.10.4.2 ConfirmButtonMessage   | 42 |
| 5.10.4.3 Description  | 42 |
| 5.10.4.4 Distance   | 42 |
| 5.10.4.5 EndLocation  | 42 |
| 5.10.4.6 ExitCommand  | 43 |
| 5.10.4.7 HeadingMessage   | 43 |
| 5.10.4.8 ManageManeuverCommand  | 43 |
| 5.10.4.9 Maneuvers  | 43 |
| 5.10.4.10 ManualTour  | 43 |
| 5.10.4.11 Name  | 43 |
| 5.10.4.12 RouteInfo   | 43 |
| 5.10.4.13 RouteType   | 44 |
| 5.10.4.14 RouteTypes  | 44 |
| 5.10.4.15 SelectRouteInfoCommand  | 44 |
| 5.10.4.16 StartLocation   | 44 |
| 5.10.5 Event Documentation  | 44 |
| 5.10.5.1 CloseRequested   | 44 |
| 5.11 DataAccessLayer.Exceptions.DALDBConnectionException Class Reference      | 45 |
| 5.11.1 Detailed Description   | 45 |
| 5.11.2 Constructor & Destructor Documentation                                 | 45 |
| 5.11.2.1 DALDBConnectionException() [1/3]                                     | 45 |
| 5.11.2.2 DALDBConnectionException() [2/3]                                     | 45 |
| 5.11.2.3 DALDBConnectionException() [3/3]                                     | 46 |
| 5.12 DataAccessLayer.Exceptions.DALParameterException Class Reference         | 46 |
| 5.12.1 Detailed Description   | 47 |
| 5.12.2 Constructor & Destructor Documentation                                 | 47 |
| 5.12.2.1 DALParameterException() [1/3]  | 47 |
| 5.12.2.2 DALParameterException() [2/3]  | 47 |
| 5.12.2.3 DALParameterException() [3/3]  | 47 |
| 5.13 DataAccessLayer.Exceptions.DALRepositoryCommandException Class Reference | 48 |
| 5.13.1 Detailed Description   | 48 |
| 5.13.2 Constructor & Destructor Documentation                                 | 48 |
| 5.13.2.1 DALRepositoryCommandException() [1/3]                                | 48 |
| 5.13.2.2 DALRepositoryCommandException() [2/3]                                | 48 |
| 5.13.2.3 DALRepositoryCommandException() [3/3]                                | 49 |

|  |    |
|--|----|
| 5.14 <a href="#">DataAccessLayer.Exceptions.DALUnitOfWorkException Class Reference</a>                 | 49 |
| 5.14.1 Detailed Description  | 50 |
| 5.14.2 Constructor & Destructor Documentation  | 50 |
| 5.14.2.1 DALUnitOfWorkException() [1/3]  | 50 |
| 5.14.2.2 DALUnitOfWorkException() [2/3]  | 50 |
| 5.14.2.3 DALUnitOfWorkException() [3/3]  | 50 |
| 5.15 <a href="#">DataAccessLayer.DBConnection.DatabaseConnection Class Reference</a>                   | 51 |
| 5.15.1 Detailed Description  | 52 |
| 5.15.2 Constructor & Destructor Documentation  | 52 |
| 5.15.2.1 DatabaseConnection() [1/2]  | 52 |
| 5.15.2.2 DatabaseConnection() [2/2]  | 52 |
| 5.15.3 Member Function Documentation   | 53 |
| 5.15.3.1 CloseConnection()   | 53 |
| 5.15.3.2 DeclareParameter()  | 53 |
| 5.15.3.3 DefineParameter()   | 53 |
| 5.15.3.4 ExecuteStatement()  | 54 |
| 5.15.3.5 GetDBConnection() [1/2]   | 54 |
| 5.15.3.6 GetDBConnection() [2/2]   | 55 |
| 5.15.3.7 OpenConnection()  | 55 |
| 5.15.3.8 QueryDatabase()   | 55 |
| 5.15.4 Member Data Documentation   | 56 |
| 5.15.4.1 config  | 56 |
| 5.15.4.2 dB  | 56 |
| 5.15.4.3 logger  | 56 |
| 5.15.4.4 npgsqlConnection  | 56 |
| 5.16 <a href="#">SWE2_Tourplanner_Tests.DALTests.DBConnectionTests Class Reference</a>                 | 57 |
| 5.17 <a href="#">DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand Class Reference</a> | 57 |
| 5.17.1 Detailed Description  | 58 |
| 5.17.2 Constructor & Destructor Documentation  | 58 |
| 5.17.2.1 DeleteManeuverCommand()   | 58 |
| 5.17.3 Member Function Documentation   | 58 |
| 5.17.3.1 Execute()   | 58 |
| 5.17.3.2 Undo()  | 59 |
| 5.17.4 Member Data Documentation   | 59 |
| 5.17.4.1 db  | 59 |
| 5.17.4.2 logger  | 59 |
| 5.17.4.3 maneuver  | 59 |
| 5.18 <a href="#">DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand Class Reference</a>         | 60 |
| 5.18.1 Detailed Description  | 60 |
| 5.18.2 Constructor & Destructor Documentation  | 60 |
| 5.18.2.1 DeleteTourCommand()   | 60 |
| 5.18.3 Member Function Documentation   | 61 |

|  |    |
|--|----|
| 5.18.3.1 Execute()   | 61 |
| 5.18.3.2 Undo()  | 61 |
| 5.18.4 Member Data Documentation   | 61 |
| 5.18.4.1 db  | 61 |
| 5.18.4.2 tour  | 62 |
| 5.19 DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand Class Reference | 62 |
| 5.19.1 Detailed Description  | 62 |
| 5.19.2 Constructor & Destructor Documentation  | 63 |
| 5.19.2.1 DeleteTourLogCommand()  | 63 |
| 5.19.3 Member Function Documentation   | 63 |
| 5.19.3.1 Execute()   | 63 |
| 5.19.3.2 Undo()  | 63 |
| 5.19.4 Member Data Documentation   | 64 |
| 5.19.4.1 db  | 64 |
| 5.19.4.2 logger  | 64 |
| 5.19.4.3 tourLog   | 64 |
| 5.20 SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs Class Reference          | 64 |
| 5.20.1 Detailed Description  | 65 |
| 5.20.2 Constructor & Destructor Documentation  | 65 |
| 5.20.2.1 DialogCloseRequestedEventArgs()   | 65 |
| 5.20.3 Property Documentation  | 65 |
| 5.20.3.1 DialogResult  | 65 |
| 5.21 SWE2_Tourplanner.Dialogs.DialogService Class Reference                          | 65 |
| 5.21.1 Detailed Description  | 66 |
| 5.21.2 Constructor & Destructor Documentation  | 66 |
| 5.21.2.1 DialogService()   | 66 |
| 5.21.3 Member Function Documentation   | 66 |
| 5.21.3.1 Register< TViewModel, TView >()   | 67 |
| 5.21.3.2 ShowDialog< TViewModel >()  | 67 |
| 5.21.4 Member Data Documentation   | 68 |
| 5.21.4.1 owner   | 68 |
| 5.21.5 Property Documentation  | 68 |
| 5.21.5.1 Mappings  | 68 |
| 5.22 SWE2_Tourplanner.ValidationRules.DoubleRangeRule Class Reference                | 68 |
| 5.22.1 Detailed Description  | 69 |
| 5.22.2 Member Function Documentation   | 69 |
| 5.22.2.1 Validate()  | 69 |
| 5.22.3 Property Documentation  | 69 |
| 5.22.3.1 Max   | 69 |
| 5.22.3.2 Min   | 69 |
| 5.23 SWE2_Tourplanner.ErrorView Class Reference                                      | 70 |
| 5.23.1 Detailed Description  | 71 |



|  |    |
|--|----|
| 5.23.2 Member Function Documentation   | 71 |
| 5.23.2.1 InitializeComponent() [1/4]   | 71 |
| 5.23.2.2 InitializeComponent() [2/4]   | 71 |
| 5.23.2.3 InitializeComponent() [3/4]   | 71 |
| 5.23.2.4 InitializeComponent() [4/4]   | 71 |
| 5.24 SWE2_Tourplanner.ViewModels.ErrorViewModel Class Reference                  | 72 |
| 5.24.1 Detailed Description  | 72 |
| 5.24.2 Constructor & Destructor Documentation                                    | 73 |
| 5.24.2.1 ErrorViewModel()  | 73 |
| 5.24.3 Member Data Documentation   | 74 |
| 5.24.3.1 errorHandlingMessage  | 74 |
| 5.24.3.2 errorType   | 74 |
| 5.24.4 Property Documentation  | 74 |
| 5.24.4.1 CloseCommand  | 74 |
| 5.24.4.2 ErrorHandlingMessage  | 74 |
| 5.24.4.3 ErrorType   | 74 |
| 5.24.5 Event Documentation   | 75 |
| 5.24.5.1 CloseRequested  | 75 |
| 5.25 SWE2_Tourplanner_Tests.CommonTests.ExtensionMethodsTest Class Reference     | 75 |
| 5.26 Common.Extensions Class Reference   | 75 |
| 5.26.1 Detailed Description  | 75 |
| 5.26.2 Member Function Documentation   | 75 |
| 5.26.2.1 CIContains()  | 75 |
| 5.27 SWE2_Tourplanner.ReusableComponents.ExternalResourceControl Class Reference | 76 |
| 5.27.1 Detailed Description  | 78 |
| 5.27.2 Constructor & Destructor Documentation                                    | 78 |
| 5.27.2.1 ExternalResourceControl()   | 78 |
| 5.27.3 Member Function Documentation   | 78 |
| 5.27.3.1 ExternalResource_MouseDown()  | 78 |
| 5.27.3.2 InitializeComponent() [1/4]   | 78 |
| 5.27.3.3 InitializeComponent() [2/4]   | 79 |
| 5.27.3.4 InitializeComponent() [3/4]   | 79 |
| 5.27.3.5 InitializeComponent() [4/4]   | 79 |
| 5.27.3.6 OnLinkChanged()   | 79 |
| 5.27.3.7 OnPropertyChanged()   | 79 |
| 5.27.4 Member Data Documentation   | 80 |
| 5.27.4.1 faviconLocation   | 80 |
| 5.27.4.2 googleFaviconCrawler  | 80 |
| 5.27.4.3 LinkProperty  | 80 |
| 5.27.4.4 LinkTextProperty  | 80 |
| 5.27.5 Property Documentation  | 81 |
| 5.27.5.1 Favicon   | 81 |

|   |    |
|---|----|
| 5.27.5.2 Link   | 81 |
| 5.27.5.3 LinkText   | 81 |
| 5.27.6 Event Documentation  | 81 |
| 5.27.6.1 PropertyChanged  | 81 |
| 5.28.XamlGeneratedNamespace.GeneratedInternalTypeHelper Class Reference | 82 |
| 5.28.1 Detailed Description   | 82 |
| 5.28.2 Member Function Documentation                                    | 82 |
| 5.28.2.1 AddEventHandler() [1/2]  | 83 |
| 5.28.2.2 AddEventHandler() [2/2]  | 83 |
| 5.28.2.3 CreateDelegate() [1/2]   | 83 |
| 5.28.2.4 CreateDelegate() [2/2]   | 83 |
| 5.28.2.5 CreateInstance() [1/2]   | 83 |
| 5.28.2.6 CreateInstance() [2/2]   | 84 |
| 5.28.2.7 GetPropertyValue() [1/2]                                       | 84 |
| 5.28.2.8 GetPropertyValue() [2/2]                                       | 84 |
| 5.28.2.9 SetPropertyValue() [1/2]                                       | 84 |
| 5.28.2.10 SetPropertyValue() [2/2]                                      | 84 |
| 5.29.DataAccessLayer.DBConnection.IDataReader Interface Reference       | 85 |
| 5.29.1 Detailed Description   | 85 |
| 5.29.2 Member Function Documentation                                    | 85 |
| 5.29.2.1 GetValue()   | 85 |
| 5.29.2.2 Read()   | 86 |
| 5.29.3 Property Documentation   | 86 |
| 5.29.3.1 FieldCount   | 86 |
| 5.30.DataAccessLayer.DBCommands.IDBCommand Interface Reference          | 86 |
| 5.30.1 Detailed Description   | 87 |
| 5.30.2 Member Function Documentation                                    | 87 |
| 5.30.2.1 Execute()  | 87 |
| 5.30.2.2 Undo()   | 87 |
| 5.31.DataAccessLayer.DBConnection.IDbCommand Interface Reference        | 88 |
| 5.31.1 Detailed Description   | 88 |
| 5.31.2 Member Function Documentation                                    | 88 |
| 5.31.2.1 ExecuteNonQuery()  | 88 |
| 5.31.2.2 ExecuteReader()  | 89 |
| 5.31.3 Property Documentation   | 89 |
| 5.31.3.1 Connection   | 89 |
| 5.31.3.2 Parameters   | 89 |
| 5.32.DataAccessLayer.DBConnection.IDBConnection Interface Reference     | 89 |
| 5.32.1 Detailed Description   | 90 |
| 5.32.2 Member Function Documentation                                    | 90 |
| 5.32.2.1 CloseConnection()  | 90 |
| 5.32.2.2 DeclareParameter()   | 90 |

|  |     |
|--|-----|
| 5.32.2.3 DefineParameter()   | 91  |
| 5.32.2.4 ExecuteStatement()  | 91  |
| 5.32.2.5 OpenConnection()  | 91  |
| 5.32.2.6 QueryDatabase()   | 92  |
| 5.33 SWE2_Tourplanner.Dialogs.IDialog Interface Reference                                    | 92  |
| 5.33.1 Detailed Description  | 93  |
| 5.33.2 Member Function Documentation   | 93  |
| 5.33.2.1 Close()   | 93  |
| 5.33.2.2 ShowDialog()  | 93  |
| 5.33.3 Property Documentation  | 93  |
| 5.33.3.1 DataContext   | 93  |
| 5.33.3.2 DialogResult  | 93  |
| 5.33.3.3 Owner   | 94  |
| 5.34 SWE2_Tourplanner.Dialogs.IDialogRequestClose Interface Reference                        | 94  |
| 5.34.1 Detailed Description  | 94  |
| 5.34.2 Event Documentation   | 94  |
| 5.34.2.1 CloseRequested  | 94  |
| 5.35 SWE2_Tourplanner.Dialogs.IDialogService Interface Reference                             | 95  |
| 5.35.1 Detailed Description  | 95  |
| 5.35.2 Member Function Documentation   | 95  |
| 5.35.2.1 Register< TViewModel, TView >()   | 95  |
| 5.35.2.2 ShowDialog< TViewModel >()  | 96  |
| 5.36 BusinessLogicLayer.ExporterImporter.IExporterImporter< T > Interface Template Reference | 96  |
| 5.36.1 Detailed Description  | 96  |
| 5.36.2 Member Function Documentation   | 97  |
| 5.36.2.1 Export()  | 97  |
| 5.36.2.2 Import()  | 97  |
| 5.37 DataAccessLayer.Repositories.IManeuverRepository Interface Reference                    | 98  |
| 5.37.1 Detailed Description  | 98  |
| 5.38 Common.MapQuestClient.IMapQuestClient Interface Reference                               | 98  |
| 5.38.1 Detailed Description  | 99  |
| 5.38.2 Member Function Documentation   | 99  |
| 5.38.2.1 GetRouteDataFromMapQuest()  | 99  |
| 5.38.2.2 RetrieveRouteImage()  | 99  |
| 5.39 SWE2_Tourplanner.ImportConfigView Class Reference                                       | 100 |
| 5.39.1 Detailed Description  | 101 |
| 5.39.2 Member Function Documentation   | 101 |
| 5.39.2.1 InitializeComponent() [1/4]   | 101 |
| 5.39.2.2 InitializeComponent() [2/4]   | 101 |
| 5.39.2.3 InitializeComponent() [3/4]   | 101 |
| 5.39.2.4 InitializeComponent() [4/4]   | 101 |
| 5.40 SWE2_Tourplanner.ViewModels.ImportConfigViewModel Class Reference                       | 102 |

|  |     |
|--|-----|
| 5.40.1 Detailed Description  | 102 |
| 5.40.2 Constructor & Destructor Documentation  | 102 |
| 5.40.2.1 ImportConfigViewModel()   | 103 |
| 5.40.3 Member Data Documentation   | 103 |
| 5.40.3.1 configPath  | 103 |
| 5.40.4 Property Documentation  | 103 |
| 5.40.4.1 ConfigPath  | 103 |
| 5.40.4.2 ConfirmCommand  | 103 |
| 5.40.4.3 ExitCommand   | 103 |
| 5.40.4.4 SelectConfigCommand   | 103 |
| 5.40.5 Event Documentation   | 104 |
| 5.40.5.1 CloseRequested  | 104 |
| 5.41 SWE2_Tourplanner.ImportView Class Reference                                       | 104 |
| 5.41.1 Detailed Description  | 105 |
| 5.41.2 Member Function Documentation   | 105 |
| 5.41.2.1 InitializeComponent() [1/4]   | 105 |
| 5.41.2.2 InitializeComponent() [2/4]   | 105 |
| 5.41.2.3 InitializeComponent() [3/4]   | 106 |
| 5.41.2.4 InitializeComponent() [4/4]   | 106 |
| 5.42 SWE2_Tourplanner.ViewModels.ImportViewModel Class Reference                       | 106 |
| 5.42.1 Detailed Description  | 107 |
| 5.42.2 Constructor & Destructor Documentation  | 107 |
| 5.42.2.1 ImportViewModel()   | 107 |
| 5.42.3 Member Data Documentation   | 107 |
| 5.42.3.1 config  | 107 |
| 5.42.3.2 jsonPaths   | 108 |
| 5.42.4 Property Documentation  | 108 |
| 5.42.4.1 ConfirmImportCommand  | 108 |
| 5.42.4.2 ExitCommand   | 108 |
| 5.42.4.3 JsonPaths   | 108 |
| 5.42.4.4 SelectPathsCommand  | 108 |
| 5.42.5 Event Documentation   | 108 |
| 5.42.5.1 CloseRequested  | 108 |
| 5.43 DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand Class Reference | 109 |
| 5.43.1 Detailed Description  | 109 |
| 5.43.2 Constructor & Destructor Documentation  | 109 |
| 5.43.2.1 InsertManeuverCommand()   | 109 |
| 5.43.3 Member Function Documentation   | 110 |
| 5.43.3.1 Execute()   | 110 |
| 5.43.3.2 Undo()  | 110 |
| 5.43.4 Member Data Documentation   | 110 |
| 5.43.4.1 db  | 110 |

|  |     |
|--|-----|
| 5.43.4.2 logger  | 111 |
| 5.43.4.3 maneuver  | 111 |
| 5.44 DataAccessLayer.DBCommands.TourCommands.InsertTourCommand Class Reference       | 111 |
| 5.44.1 Detailed Description  | 112 |
| 5.44.2 Constructor & Destructor Documentation  | 112 |
| 5.44.2.1 InsertTourCommand()   | 112 |
| 5.44.3 Member Function Documentation   | 112 |
| 5.44.3.1 Execute()   | 112 |
| 5.44.3.2 Undo()  | 112 |
| 5.44.4 Member Data Documentation   | 113 |
| 5.44.4.1 db  | 113 |
| 5.44.4.2 tour  | 113 |
| 5.45 DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand Class Reference | 113 |
| 5.45.1 Detailed Description  | 114 |
| 5.45.2 Constructor & Destructor Documentation  | 114 |
| 5.45.2.1 InsertTourLogCommand()  | 114 |
| 5.45.3 Member Function Documentation   | 114 |
| 5.45.3.1 Execute()   | 114 |
| 5.45.3.2 Undo()  | 115 |
| 5.45.4 Member Data Documentation   | 115 |
| 5.45.4.1 db  | 115 |
| 5.45.4.2 logger  | 115 |
| 5.45.4.3 tourLog   | 115 |
| 5.46 DataAccessLayer.Repositories.IRepository< T > Interface Template Reference      | 115 |
| 5.46.1 Detailed Description  | 116 |
| 5.46.2 Member Function Documentation   | 116 |
| 5.46.2.1 Delete()  | 116 |
| 5.46.2.2 Insert()  | 116 |
| 5.46.2.3 Read()  | 118 |
| 5.46.2.4 ReadAll()   | 118 |
| 5.46.2.5 Update()  | 118 |
| 5.47 BusinessLogicLayer.Factories.ITourFactory Interface Reference                   | 119 |
| 5.47.1 Detailed Description  | 119 |
| 5.47.2 Member Function Documentation   | 119 |
| 5.47.2.1 CreateMapQuestTour()  | 119 |
| 5.47.2.2 CreateTour()  | 120 |
| 5.47.2.3 DeleteTour()  | 120 |
| 5.47.2.4 GetTours()  | 121 |
| 5.47.2.5 UpdateMapQuestTour()  | 121 |
| 5.47.2.6 UpdateTour()  | 121 |
| 5.48 BusinessLogicLayer.Factories.ITourLogFactory Interface Reference                | 122 |
| 5.48.1 Detailed Description  | 122 |

|  |     |
|--|-----|
| 5.48.2 Member Function Documentation   | 122 |
| 5.48.2.1 CreateTourLog()   | 122 |
| 5.48.2.2 DeleteTourLog()   | 123 |
| 5.48.2.3 UpdateTourLog()   | 123 |
| 5.49 DataAccessLayer.Repositories.ITourLogRepository Interface Reference             | 124 |
| 5.49.1 Detailed Description  | 124 |
| 5.50 Common.Config.ITourPlannerConfig Interface Reference                            | 124 |
| 5.50.1 Detailed Description  | 125 |
| 5.50.2 Member Function Documentation   | 125 |
| 5.50.2.1 LoadConfigFromFile()  | 125 |
| 5.50.3 Property Documentation  | 125 |
| 5.50.3.1 DatabaseConnectionString  | 125 |
| 5.50.3.2 ExportsDirectory  | 125 |
| 5.50.3.3 MapQuestKey   | 126 |
| 5.50.3.4 PictureDirectory  | 126 |
| 5.51 BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator Interface Reference | 126 |
| 5.51.1 Detailed Description  | 126 |
| 5.52 DataAccessLayer.Repositories.ITourRepository Interface Reference                | 127 |
| 5.52.1 Detailed Description  | 127 |
| 5.53 DataAccessLayer.UnitOfWork.IUnitOfWork Interface Reference                      | 127 |
| 5.53.1 Member Function Documentation   | 128 |
| 5.53.1.1 Commit()  | 128 |
| 5.53.1.2 Rollback()  | 128 |
| 5.53.2 Property Documentation  | 128 |
| 5.53.2.1 ManeuverRepository  | 128 |
| 5.53.2.2 TourLogRepository   | 129 |
| 5.53.2.3 TourRepository  | 129 |
| 5.54 Common.Logging.LogHelper Class Reference  | 129 |
| 5.54.1 Detailed Description  | 129 |
| 5.54.2 Constructor & Destructor Documentation  | 129 |
| 5.54.2.1 LogHelper()   | 129 |
| 5.54.3 Member Function Documentation   | 130 |
| 5.54.3.1 GetLogger()   | 130 |
| 5.54.3.2 GetLogHelper()  | 130 |
| 5.54.4 Member Data Documentation   | 130 |
| 5.54.4.1 logHelper   | 130 |
| 5.55 SWE2_Tourplanner_Tests.CommonTests.LogHelperTests Class Reference               | 131 |
| 5.56 SWE2_Tourplanner.MainViewModel Class Reference                                  | 131 |
| 5.56.1 Detailed Description  | 132 |
| 5.56.2 Constructor & Destructor Documentation  | 132 |
| 5.56.2.1 MainViewModel()   | 132 |
| 5.56.3 Member Function Documentation   | 132 |

|  |     |
|--|-----|
| 5.56.3.1 ShowException()   | 133 |
| 5.56.4 Member Data Documentation                                     | 133 |
| 5.56.4.1 dialogService   | 133 |
| 5.56.4.2 exporterImporter  | 133 |
| 5.56.4.3 pdfGenerator  | 133 |
| 5.56.4.4 tourDetailViewModel   | 133 |
| 5.56.4.5 tourPlannerFactory  | 134 |
| 5.56.4.6 toursViewModel  | 134 |
| 5.56.5 Property Documentation  | 134 |
| 5.56.5.1 AddTourCommand  | 134 |
| 5.56.5.2 ConductImportCommand  | 134 |
| 5.56.5.3 EditTourCommand   | 134 |
| 5.56.5.4 GenerateJSONExportCommand                                   | 134 |
| 5.56.5.5 GenerateSummaryReportCommand                                | 134 |
| 5.56.5.6 GenerateTourReportCommand                                   | 135 |
| 5.56.5.7 LoadConfigCommand   | 135 |
| 5.56.5.8 RemoveTourCommand   | 135 |
| 5.56.5.9 TourDetailViewModel   | 135 |
| 5.56.5.10 ToursViewModel   | 135 |
| 5.57 SWE2_Tourplanner.MainWindow Class Reference                     | 136 |
| 5.57.1 Detailed Description  | 137 |
| 5.57.2 Member Function Documentation                                 | 137 |
| 5.57.2.1 InitializeComponent() [1/4]                                 | 137 |
| 5.57.2.2 InitializeComponent() [2/4]                                 | 137 |
| 5.57.2.3 InitializeComponent() [3/4]                                 | 137 |
| 5.57.2.4 InitializeComponent() [4/4]                                 | 137 |
| 5.58 Common.Entities.Maneuver Class Reference                        | 138 |
| 5.58.1 Detailed Description  | 138 |
| 5.59 SWE2_Tourplanner.ManeuverManagementWindow Class Reference       | 138 |
| 5.59.1 Detailed Description  | 139 |
| 5.59.2 Member Function Documentation                                 | 140 |
| 5.59.2.1 InitializeComponent() [1/4]                                 | 140 |
| 5.59.2.2 InitializeComponent() [2/4]                                 | 140 |
| 5.59.2.3 InitializeComponent() [3/4]                                 | 140 |
| 5.59.2.4 InitializeComponent() [4/4]                                 | 140 |
| 5.60 DataAccessLayer.Repositories.ManeuverRepository Class Reference | 141 |
| 5.60.1 Detailed Description  | 142 |
| 5.60.2 Constructor & Destructor Documentation                        | 142 |
| 5.60.2.1 ManeuverRepository() [1/2]                                  | 142 |
| 5.60.2.2 ManeuverRepository() [2/2]                                  | 142 |
| 5.60.3 Member Function Documentation                                 | 142 |
| 5.60.3.1 CheckDBConstraints()  | 142 |

|  |     |
|--|-----|
| 5.60.3.2 ConvertToManeuver()   | 143 |
| 5.60.3.3 Delete()  | 143 |
| 5.60.3.4 Insert()  | 143 |
| 5.60.3.5 Read()  | 144 |
| 5.60.3.6 ReadAll()   | 144 |
| 5.60.3.7 Update()  | 144 |
| 5.60.4 Member Data Documentation   | 145 |
| 5.60.4.1 commitCommands  | 145 |
| 5.60.4.2 db  | 145 |
| 5.60.4.3 logger  | 145 |
| 5.61 SWE2_Tourplanner_Tests.DALTests.ManeuverRepositoryTests Class Reference | 145 |
| 5.62 SWE2_Tourplanner.ViewModels.ManeuversViewModel Class Reference          | 146 |
| 5.62.1 Detailed Description  | 147 |
| 5.62.2 Constructor & Destructor Documentation                                | 147 |
| 5.62.2.1 ManeuversViewModel()  | 147 |
| 5.62.3 Member Data Documentation   | 147 |
| 5.62.3.1 currentManeuver   | 147 |
| 5.62.3.2 maneuvers   | 147 |
| 5.62.4 Property Documentation  | 147 |
| 5.62.4.1 AddManeuverCommand  | 148 |
| 5.62.4.2 ConfirmCommand  | 148 |
| 5.62.4.3 CurrentManeuver   | 148 |
| 5.62.4.4 ExitCommand   | 148 |
| 5.62.4.5 Maneuvers   | 148 |
| 5.62.4.6 RemoveManeuverCommand   | 148 |
| 5.62.5 Event Documentation   | 148 |
| 5.62.5.1 CloseRequested  | 149 |
| 5.63 SWE2_Tourplanner.ValidationRules.ManeuverValidationRule Class Reference | 149 |
| 5.63.1 Detailed Description  | 149 |
| 5.63.2 Member Function Documentation   | 149 |
| 5.63.2.1 Validate()  | 149 |
| 5.64 Common.MapQuestClient.MapQuestClient Class Reference                    | 150 |
| 5.64.1 Detailed Description  | 151 |
| 5.64.2 Constructor & Destructor Documentation                                | 151 |
| 5.64.2.1 MapQuestClient()  | 151 |
| 5.64.3 Member Function Documentation   | 151 |
| 5.64.3.1 GetRouteDataFromMapQuest()  | 151 |
| 5.64.3.2 RetrieveRouteImage()  | 152 |
| 5.64.4 Member Data Documentation   | 152 |
| 5.64.4.1 client  | 152 |
| 5.64.4.2 config  | 153 |
| 5.64.4.3 logger  | 153 |



|   |     |
|---|-----|
| 5.64.4.4 pictureHeight . . . . .  | 153 |
| 5.64.4.5 pictureWidth . . . . .   | 153 |
| 5.64.4.6 routeBaseURL . . . . .   | 153 |
| 5.64.4.7 staticmapBaseURL . . . . .   | 153 |
| 5.65 SWE2_Tourplanner.ValidationRules.NoNegativeDistancesRule Class Reference . . . . . | 154 |
| 5.65.1 Detailed Description . . . . .   | 154 |
| 5.65.2 Member Function Documentation . . . . .  | 154 |
| 5.65.2.1 Validate() . . . . .   | 154 |
| 5.66 DataAccessLayer.DBConnection.NpgsqlCommand Class Reference . . . . .               | 155 |
| 5.66.1 Detailed Description . . . . .   | 155 |
| 5.66.2 Constructor & Destructor Documentation . . . . .                                 | 155 |
| 5.66.2.1 NpgsqlCommand() . . . . .  | 155 |
| 5.66.3 Member Function Documentation . . . . .  | 156 |
| 5.66.3.1 ExecuteNonQuery() . . . . .  | 156 |
| 5.66.3.2 ExecuteReader() . . . . .  | 156 |
| 5.66.4 Member Data Documentation . . . . .  | 156 |
| 5.66.4.1 npgsqlCommand . . . . .  | 156 |
| 5.66.5 Property Documentation . . . . .   | 157 |
| 5.66.5.1 Connection . . . . .   | 157 |
| 5.66.5.2 Parameters . . . . .   | 157 |
| 5.67 DataAccessLayer.DBConnection.NpgsqlDataReader Class Reference . . . . .            | 157 |
| 5.67.1 Detailed Description . . . . .   | 158 |
| 5.67.2 Constructor & Destructor Documentation . . . . .                                 | 158 |
| 5.67.2.1 NpgsqlDataReader() . . . . .   | 158 |
| 5.67.3 Member Function Documentation . . . . .  | 158 |
| 5.67.3.1 Dispose() . . . . .  | 158 |
| 5.67.3.2 GetValue() . . . . .   | 158 |
| 5.67.3.3 Read() . . . . .   | 159 |
| 5.67.4 Member Data Documentation . . . . .  | 159 |
| 5.67.4.1 npgsqlDataReader . . . . .   | 159 |
| 5.67.5 Property Documentation . . . . .   | 159 |
| 5.67.5.1 FieldCount . . . . .   | 159 |
| 5.68 SWE2_Tourplanner.Commands.RelayCommand Class Reference . . . . .                   | 160 |
| 5.68.1 Detailed Description . . . . .   | 160 |
| 5.68.2 Constructor & Destructor Documentation . . . . .                                 | 160 |
| 5.68.2.1 RelayCommand() . . . . .   | 160 |
| 5.68.3 Member Function Documentation . . . . .  | 161 |
| 5.68.3.1 CanExecute() . . . . .   | 161 |
| 5.68.3.2 Execute() . . . . .  | 161 |
| 5.68.4 Member Data Documentation . . . . .  | 161 |
| 5.68.4.1 _canExecute . . . . .  | 161 |
| 5.68.4.2 _execute . . . . .   | 162 |

|   |     |
|---|-----|
| 5.68.5 Property Documentation . . . . .   | 162 |
| 5.68.5.1 CanExecuteChanged . . . . .  | 162 |
| 5.69 SWE2_Tourplanner.Converters.RouteInfoImageConverter Class Reference . . . . .      | 162 |
| 5.69.1 Detailed Description . . . . .   | 163 |
| 5.69.2 Member Function Documentation . . . . .  | 163 |
| 5.69.2.1 Convert() . . . . .  | 163 |
| 5.69.2.2 ConvertBack() . . . . .  | 163 |
| 5.70 SWE2_Tourplanner.Converters.StringToDateTimeConverter Class Reference . . . . .    | 164 |
| 5.70.1 Detailed Description . . . . .   | 164 |
| 5.70.2 Member Function Documentation . . . . .  | 164 |
| 5.70.2.1 Convert() . . . . .  | 164 |
| 5.70.2.2 ConvertBack() . . . . .  | 165 |
| 5.71 BusinessLogicLayer.PDFCreation.SummaryReport Class Reference . . . . .             | 165 |
| 5.71.1 Detailed Description . . . . .   | 166 |
| 5.71.2 Constructor & Destructor Documentation . . . . .                                 | 166 |
| 5.71.2.1 SummaryReport() . . . . .  | 166 |
| 5.71.3 Member Function Documentation . . . . .  | 167 |
| 5.71.3.1 Compose() . . . . .  | 167 |
| 5.71.3.2 ComposeContent() . . . . .   | 167 |
| 5.71.3.3 ComposeHeader() . . . . .  | 167 |
| 5.71.3.4 ComposeManeuverStats() . . . . .   | 167 |
| 5.71.3.5 ComposeTourLogStats() . . . . .  | 168 |
| 5.71.3.6 ComposeTourStats() . . . . .   | 168 |
| 5.71.3.7 GetMetadata() . . . . .  | 168 |
| 5.71.4 Property Documentation . . . . .   | 168 |
| 5.71.4.1 Tours . . . . .  | 169 |
| 5.72 SWE2_Tourplanner.ReusableComponents.TemperatureSliders Class Reference . . . . .   | 169 |
| 5.72.1 Detailed Description . . . . .   | 169 |
| 5.72.2 Member Function Documentation . . . . .  | 169 |
| 5.72.2.1 InitializeComponent() . . . . .  | 170 |
| 5.73 SWE2_Tourplanner.ViewModels.TemperatureSlidersViewModel Class Reference . . . . .  | 170 |
| 5.73.1 Detailed Description . . . . .   | 170 |
| 5.73.2 Member Function Documentation . . . . .  | 170 |
| 5.73.2.1 InitializeComponent() . . . . .  | 171 |
| 5.74 SWE2_Tourplanner.ValidationRules.TextInputValidationRule Class Reference . . . . . | 171 |
| 5.74.1 Detailed Description . . . . .   | 171 |
| 5.74.2 Member Function Documentation . . . . .  | 171 |
| 5.74.2.1 Validate() . . . . .   | 171 |
| 5.74.3 Property Documentation . . . . .   | 172 |
| 5.74.3.1 Max . . . . .  | 172 |
| 5.74.3.2 Min . . . . .  | 172 |
| 5.75 Common.Entities.Tour Class Reference . . . . .                                     | 172 |

|   |     |
|---|-----|
| 5.75.1 Detailed Description . . . . .   | 173 |
| 5.76 SWE2_Tourplanner.ViewModels.TourDetailViewModel Class Reference . . . . .          | 173 |
| 5.76.1 Detailed Description . . . . .   | 174 |
| 5.76.2 Constructor & Destructor Documentation . . . . .                                 | 174 |
| 5.76.2.1 TourDetailViewModel() . . . . .  | 174 |
| 5.76.3 Member Function Documentation . . . . .  | 174 |
| 5.76.3.1 ShowException() . . . . .  | 174 |
| 5.76.4 Member Data Documentation . . . . .  | 175 |
| 5.76.4.1 _selectedTour . . . . .  | 175 |
| 5.76.4.2 currentTourLogs . . . . .  | 175 |
| 5.76.4.3 dialogService . . . . .  | 175 |
| 5.76.4.4 selectedTourLog . . . . .  | 175 |
| 5.76.4.5 tourLogFactory . . . . .   | 175 |
| 5.76.5 Property Documentation . . . . .   | 176 |
| 5.76.5.1 AddLogCommand . . . . .  | 176 |
| 5.76.5.2 CurrentTourLogs . . . . .  | 176 |
| 5.76.5.3 EditLogCommand . . . . .   | 176 |
| 5.76.5.4 RemoveLogCommand . . . . .   | 176 |
| 5.76.5.5 SelectedTour . . . . .   | 176 |
| 5.76.5.6 SelectedTourLog . . . . .  | 176 |
| 5.77 BusinessLogicLayer.ExporterImporter.TourExporterImporter Class Reference . . . . . | 177 |
| 5.77.1 Detailed Description . . . . .   | 177 |
| 5.77.2 Constructor & Destructor Documentation . . . . .                                 | 177 |
| 5.77.2.1 TourExporterImporter() . . . . .   | 177 |
| 5.77.3 Member Function Documentation . . . . .  | 178 |
| 5.77.3.1 Export() . . . . .   | 178 |
| 5.77.3.2 ExportPicture() . . . . .  | 178 |
| 5.77.3.3 Import() . . . . .   | 178 |
| 5.77.4 Member Data Documentation . . . . .  | 179 |
| 5.77.4.1 logger . . . . .   | 179 |
| 5.77.4.2 tourPlannerConfig . . . . .  | 179 |
| 5.78 BusinessLogicLayer.Factories.TourFactory Class Reference . . . . .                 | 179 |
| 5.78.1 Detailed Description . . . . .   | 180 |
| 5.78.2 Constructor & Destructor Documentation . . . . .                                 | 180 |
| 5.78.2.1 TourFactory() . . . . .  | 180 |
| 5.78.3 Member Function Documentation . . . . .  | 181 |
| 5.78.3.1 CreateMapQuestTour() . . . . .   | 181 |
| 5.78.3.2 CreateTour() . . . . .   | 181 |
| 5.78.3.3 DeleteTour() . . . . .   | 182 |
| 5.78.3.4 GetTours() . . . . .   | 182 |
| 5.78.3.5 RetrieveMapQuestData() . . . . .   | 182 |
| 5.78.3.6 UpdateMapQuestTour() . . . . .   | 183 |

|  |     |
|--|-----|
| 5.78.3.7 UpdateTour()  | 183 |
| 5.78.4 Member Data Documentation                                       | 184 |
| 5.78.4.1 logger  | 184 |
| 5.78.4.2 mapQuestClient  | 184 |
| 5.78.4.3 uow   | 184 |
| 5.79 SWE2_Tourplanner_Tests.BLTests.TourFactoryTests Class Reference   | 184 |
| 5.80 Common.Entities.TourLog Class Reference                           | 185 |
| 5.80.1 Detailed Description  | 185 |
| 5.81 SWE2_Tourplanner.TourLogCreateUpdateView Class Reference          | 186 |
| 5.81.1 Detailed Description  | 187 |
| 5.81.2 Member Function Documentation                                   | 187 |
| 5.81.2.1 InitializeComponent() [1/4]                                   | 187 |
| 5.81.2.2 InitializeComponent() [2/4]                                   | 187 |
| 5.81.2.3 InitializeComponent() [3/4]                                   | 187 |
| 5.81.2.4 InitializeComponent() [4/4]                                   | 188 |
| 5.82 BusinessLogicLayer.Factories.TourLogFactory Class Reference       | 188 |
| 5.82.1 Detailed Description  | 188 |
| 5.82.2 Constructor & Destructor Documentation                          | 189 |
| 5.82.2.1 TourLogFactory()  | 189 |
| 5.82.3 Member Function Documentation                                   | 189 |
| 5.82.3.1 CreateTourLog()   | 189 |
| 5.82.3.2 DeleteTourLog()   | 189 |
| 5.82.3.3 UpdateTourLog()   | 190 |
| 5.82.4 Member Data Documentation                                       | 190 |
| 5.82.4.1 logger  | 190 |
| 5.82.4.2 uow   | 191 |
| 5.83 SWE2_Tourplanner_Tests.BLTests.TourLogFactoryTest Class Reference | 191 |
| 5.84 DataAccessLayer.Repositories.TourLogRepository Class Reference    | 191 |
| 5.84.1 Detailed Description  | 192 |
| 5.84.2 Constructor & Destructor Documentation                          | 192 |
| 5.84.2.1 TourLogRepository() [1/2]                                     | 193 |
| 5.84.2.2 TourLogRepository() [2/2]                                     | 193 |
| 5.84.3 Member Function Documentation                                   | 193 |
| 5.84.3.1 CheckDBConstraints()  | 193 |
| 5.84.3.2 ConvertToTourLog()  | 193 |
| 5.84.3.3 Delete()  | 194 |
| 5.84.3.4 Insert()  | 194 |
| 5.84.3.5 Read()  | 194 |
| 5.84.3.6 ReadAll()   | 195 |
| 5.84.3.7 Update()  | 195 |
| 5.84.4 Member Data Documentation                                       | 195 |
| 5.84.4.1 commitCommands  | 196 |

|   |     |
|---|-----|
| 5.84.4.2 db   | 196 |
| 5.84.4.3 logger   | 196 |
| 5.85 SWE2_Tourplanner_Tests.DALTests.TourLogRepositoryTests Class Reference     | 196 |
| 5.86 Common.Config.TourPlannerConfig Class Reference                            | 197 |
| 5.86.1 Detailed Description   | 198 |
| 5.86.2 Constructor & Destructor Documentation                                   | 198 |
| 5.86.2.1 TourPlannerConfig()  | 198 |
| 5.86.3 Member Function Documentation  | 198 |
| 5.86.3.1 GetTourPlannerConfig()   | 198 |
| 5.86.3.2 LoadConfigFromFile()   | 198 |
| 5.86.3.3 SetUpEnvironment()   | 199 |
| 5.86.4 Member Data Documentation  | 199 |
| 5.86.4.1 logger   | 199 |
| 5.86.4.2 originalConfigPath   | 199 |
| 5.86.4.3 tourPlannerConfig  | 200 |
| 5.87 SWE2_Tourplanner_Tests.CommonTests.TourPlannerConfigTests Class Reference  | 200 |
| 5.88 BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator Class Reference | 200 |
| 5.88.1 Detailed Description   | 201 |
| 5.88.2 Constructor & Destructor Documentation                                   | 201 |
| 5.88.2.1 TourPlannerReportsGenerator()  | 201 |
| 5.88.3 Member Function Documentation  | 201 |
| 5.88.3.1 GenerateSummaryReport()  | 201 |
| 5.88.3.2 GenerateTourReport()   | 202 |
| 5.88.4 Member Data Documentation  | 202 |
| 5.88.4.1 config   | 202 |
| 5.88.4.2 logger   | 202 |
| 5.89 BusinessLogicLayer.PDFCreation.TourReport Class Reference                  | 203 |
| 5.89.1 Detailed Description   | 203 |
| 5.89.2 Constructor & Destructor Documentation                                   | 203 |
| 5.89.2.1 TourReport()   | 203 |
| 5.89.3 Member Function Documentation  | 204 |
| 5.89.3.1 Compose()  | 204 |
| 5.89.3.2 ComposeContent()   | 204 |
| 5.89.3.3 ComposeDetails()   | 204 |
| 5.89.3.4 ComposeHeader()  | 205 |
| 5.89.3.5 ComposeManeuvers()   | 205 |
| 5.89.3.6 ComposeReports()   | 205 |
| 5.89.3.7 ComposeTourLogs()  | 205 |
| 5.89.3.8 GetMetadata()  | 206 |
| 5.89.4 Property Documentation   | 206 |
| 5.89.4.1 Tour   | 206 |
| 5.90 DataAccessLayer.Repositories.TourRepository Class Reference                | 206 |

|  |     |
|--|-----|
| 5.90.1 Detailed Description  | 207 |
| 5.90.2 Constructor & Destructor Documentation                            | 207 |
| 5.90.2.1 TourRepository() [1/3]  | 208 |
| 5.90.2.2 TourRepository() [2/3]  | 208 |
| 5.90.2.3 TourRepository() [3/3]  | 208 |
| 5.90.3 Member Function Documentation                                     | 208 |
| 5.90.3.1 CheckDBConstraints()  | 208 |
| 5.90.3.2 ConvertToTour()   | 210 |
| 5.90.3.3 Delete()  | 210 |
| 5.90.3.4 Insert()  | 210 |
| 5.90.3.5 Read()  | 211 |
| 5.90.3.6 ReadAll()   | 211 |
| 5.90.3.7 Update()  | 211 |
| 5.90.4 Member Data Documentation   | 212 |
| 5.90.4.1 commitCommands  | 212 |
| 5.90.4.2 db  | 212 |
| 5.90.4.3 logger  | 212 |
| 5.90.4.4 maneuverRepository  | 212 |
| 5.90.4.5 tourLogRepository   | 213 |
| 5.91 SWE2_Tourplanner_Tests.DALTests.TourRepositoryTests Class Reference | 213 |
| 5.92 SWE2_Tourplanner.ViewModels.ToursViewModel Class Reference          | 213 |
| 5.92.1 Detailed Description  | 214 |
| 5.92.2 Constructor & Destructor Documentation                            | 214 |
| 5.92.2.1 ToursViewModel()  | 214 |
| 5.92.3 Member Data Documentation   | 215 |
| 5.92.3.1 _tours  | 215 |
| 5.92.3.2 dialogService   | 215 |
| 5.92.3.3 filteredTours   | 215 |
| 5.92.3.4 searchString  | 215 |
| 5.92.3.5 tourPlannerFactory  | 215 |
| 5.92.4 Property Documentation  | 215 |
| 5.92.4.1 FilteredTours   | 216 |
| 5.92.4.2 SearchString  | 216 |
| 5.92.4.3 SearchToursCommand  | 216 |
| 5.92.4.4 Tours   | 216 |
| 5.93 DataAccessLayer.UnitOfWork.UnitOfWork Class Reference               | 216 |
| 5.93.1 Detailed Description  | 217 |
| 5.93.2 Constructor & Destructor Documentation                            | 217 |
| 5.93.2.1 UnitOfWork() [1/2]  | 218 |
| 5.93.2.2 UnitOfWork() [2/2]  | 218 |
| 5.93.3 Member Function Documentation                                     | 218 |
| 5.93.3.1 Commit()  | 218 |

|  |     |
|--|-----|
| 5.93.3.2 Dispose()   | 219 |
| 5.93.3.3 Rollback()  | 219 |
| 5.93.4 Member Data Documentation   | 219 |
| 5.93.4.1 commitCommands  | 219 |
| 5.93.4.2 db  | 219 |
| 5.93.4.3 logger  | 220 |
| 5.93.4.4 maneuverRepository  | 220 |
| 5.93.4.5 rollbackCommands  | 220 |
| 5.93.4.6 tourLogRepository   | 220 |
| 5.93.4.7 tourRepository  | 220 |
| 5.93.5 Property Documentation  | 220 |
| 5.93.5.1 ManeuverRepository  | 221 |
| 5.93.5.2 TourLogRepository   | 221 |
| 5.93.5.3 TourRepository  | 221 |
| 5.94 SWE2_Tourplanner_Tests.DALTests.UnitOfWorkTests Class Reference                   | 221 |
| 5.95 DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand Class Reference | 222 |
| 5.95.1 Detailed Description  | 222 |
| 5.95.2 Constructor & Destructor Documentation  | 222 |
| 5.95.2.1 UpdateManeuverCommand()   | 222 |
| 5.95.3 Member Function Documentation   | 223 |
| 5.95.3.1 Execute()   | 223 |
| 5.95.3.2 Undo()  | 223 |
| 5.95.4 Member Data Documentation   | 223 |
| 5.95.4.1 db  | 223 |
| 5.95.4.2 logger  | 224 |
| 5.95.4.3 maneuver  | 224 |
| 5.95.4.4 oldManeuver   | 224 |
| 5.96 DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand Class Reference         | 224 |
| 5.96.1 Detailed Description  | 225 |
| 5.96.2 Constructor & Destructor Documentation  | 225 |
| 5.96.2.1 UpdateTourCommand()   | 225 |
| 5.96.3 Member Function Documentation   | 225 |
| 5.96.3.1 Execute()   | 225 |
| 5.96.3.2 Undo()  | 226 |
| 5.96.4 Member Data Documentation   | 226 |
| 5.96.4.1 db  | 226 |
| 5.96.4.2 oldTour   | 226 |
| 5.96.4.3 tour  | 226 |
| 5.97 DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand Class Reference   | 227 |
| 5.97.1 Detailed Description  | 227 |
| 5.97.2 Constructor & Destructor Documentation  | 227 |
| 5.97.2.1 UpdateTourLogCommand()  | 227 |

|  |            |
|--|------------|
| 5.97.3 Member Function Documentation . . . . .                                     | 228        |
| 5.97.3.1 Execute() . . . . .   | 228        |
| 5.97.3.2 Undo() . . . . .  | 228        |
| 5.97.4 Member Data Documentation . . . . .   | 228        |
| 5.97.4.1 db . . . . .  | 228        |
| 5.97.4.2 logger . . . . .  | 229        |
| 5.97.4.3 oldTourLog . . . . .  | 229        |
| 5.97.4.4 tourLog . . . . .   | 229        |
| 5.98 SWE2_Tourplanner_Tests.UITests.ValidationRulesTests Class Reference . . . . . | 229        |
| <b>Index</b>   | <b>231</b> |



# Chapter 1

## Namespace Index

### 1.1 Packages

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

|   |    |
|---|----|
| <a href="#">BusinessLogicLayer</a>                          | 13 |
| <a href="#">BusinessLogicLayer.Exceptions</a>               | 13 |
| <a href="#">BusinessLogicLayer.ExporterImporter</a>         | 13 |
| <a href="#">BusinessLogicLayer.Factories</a>                | 13 |
| <a href="#">BusinessLogicLayer.PDFCreation</a>              | 14 |
| <a href="#">Common</a>                                      | 14 |
| <a href="#">Common.Config</a>                               | 14 |
| <a href="#">Common.Entities</a>                             | 14 |
| <a href="#">Common.Enums</a>                                | 15 |
| <a href="#">Common.Exceptions</a>                           | 15 |
| <a href="#">Common.Logging</a>                              | 16 |
| <a href="#">Common.MapQuestClient</a>                       | 16 |
| <a href="#">DataAccessLayer</a>                             | 16 |
| <a href="#">DataAccessLayer.DBCommands</a>                  | 16 |
| <a href="#">DataAccessLayer.DBCommands.ManeuverCommands</a> | 16 |
| <a href="#">DataAccessLayer.DBCommands.TourCommands</a>     | 17 |
| <a href="#">DataAccessLayer.DBCommands.TourLogCommands</a>  | 17 |
| <a href="#">DataAccessLayer.DBConnection</a>                | 17 |
| <a href="#">DataAccessLayer.Exceptions</a>                  | 18 |
| <a href="#">DataAccessLayer.Repositories</a>                | 18 |
| <a href="#">DataAccessLayer.UnitOfWork</a>                  | 18 |
| <a href="#">SWE2_Tourplanner</a>                            | 19 |
| <a href="#">SWE2_Tourplanner.Commands</a>                   | 19 |
| <a href="#">SWE2_Tourplanner.Converters</a>                 | 19 |
| <a href="#">SWE2_Tourplanner.Dialogs</a>                    | 20 |
| <a href="#">SWE2_Tourplanner.ReusableComponents</a>         | 20 |
| <a href="#">SWE2_Tourplanner.ValidationRules</a>            | 20 |
| <a href="#">SWE2_Tourplanner.ViewModels</a>                 | 21 |
| <a href="#">SWE2_Tourplanner_Tests</a>                      | 21 |
| <a href="#">SWE2_Tourplanner_Tests.BLTests</a>              | 21 |
| <a href="#">SWE2_Tourplanner_Tests.CommonTests</a>          | 21 |
| <a href="#">SWE2_Tourplanner_Tests.DALTests</a>             | 22 |
| <a href="#">SWE2_Tourplanner_Tests.UITests</a>              | 22 |
| <a href="#">XamlGeneratedNamespace</a>                      | 22 |



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

|   |     |
|---|-----|
| Application   |     |
| SWE2_Tourplanner.App . . . . .                                      | 23  |
| System.Windows.Application  |     |
| SWE2_Tourplanner.App . . . . .                                      | 23  |
| SWE2_Tourplanner.App . . . . .                                      | 23  |
| SWE2_Tourplanner.App . . . . .                                      | 23  |
| SWE2_Tourplanner.App . . . . .                                      | 23  |
| SWE2_Tourplanner_Tests.UITests.ConverterTests . . . . .             | 33  |
| SWE2_Tourplanner_Tests.DALTests.DBConnectionTests . . . . .         | 57  |
| EventArgs   |     |
| SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs . . . . .    | 64  |
| Exception   |     |
| BusinessLogicLayer.Exceptions.BLExporterImporterException . . . . . | 27  |
| BusinessLogicLayer.Exceptions.BLFactoryException . . . . .          | 29  |
| BusinessLogicLayer.Exceptions.BLPDFCreationException . . . . .      | 30  |
| Common.Exceptions.CommonConfigException . . . . .                   | 32  |
| DataAccessLayer.Exceptions.DALDBConnectionException . . . . .       | 45  |
| DataAccessLayer.Exceptions.DALParameterException . . . . .          | 46  |
| DataAccessLayer.Exceptions.DALRepositoryCommandException . . . . .  | 48  |
| DataAccessLayer.Exceptions.DALUnitOfWorkException . . . . .         | 49  |
| SWE2_Tourplanner_Tests.CommonTests.ExtensionMethodsTest . . . . .   | 75  |
| Common.Extensions . . . . .   | 75  |
| ICommand  |     |
| SWE2_Tourplanner.Commands.RelayCommand . . . . .                    | 160 |
| System.Windows.Markup.IComponentConnector                           |     |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                     | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                     | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                     | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                     | 38  |
| SWE2_Tourplanner.ErrorView . . . . .                                | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                | 70  |
| SWE2_Tourplanner.ImportConfigView . . . . .                         | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                         | 100 |

|   |     |
|---|-----|
| SWE2_Tourplanner.ImportConfigView . . . . .                                 | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                                 | 100 |
| SWE2_Tourplanner.ImportView . . . . .                                       | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                       | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                       | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                       | 104 |
| SWE2_Tourplanner.MainWindow . . . . .                                       | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                       | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                       | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                       | 136 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                         | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                         | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                         | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                         | 138 |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . .       | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . .       | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . .       | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . .       | 76  |
| SWE2_Tourplanner.ReusableComponents.TemperatureSliders . . . . .            | 169 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                          | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                          | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                          | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                          | 186 |
| SWE2_Tourplanner.ViewModels.TemperatureSlidersViewModel . . . . .           | 170 |
| DataAccessLayer.DBCommands.IDBCommand . . . . .                             | 86  |
| DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand . . . . . | 57  |
| DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand . . . . . | 109 |
| DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand . . . . . | 222 |
| DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand . . . . .         | 60  |
| DataAccessLayer.DBCommands.TourCommands.InsertTourCommand . . . . .         | 111 |
| DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand . . . . .         | 224 |
| DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand . . . . .   | 62  |
| DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand . . . . .   | 113 |
| DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand . . . . .   | 227 |
| DataAccessLayer.DBConnection.IDbCommand . . . . .                           | 88  |
| DataAccessLayer.DBConnection.NpgsqlCommand . . . . .                        | 155 |
| DataAccessLayer.DBConnection.IDBConnection . . . . .                        | 89  |
| DataAccessLayer.DBConnection.DatabaseConnection . . . . .                   | 51  |
| IDialog   |     |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                             | 38  |
| SWE2_Tourplanner.ErrorView . . . . .  | 70  |
| SWE2_Tourplanner.ImportConfigView . . . . .                                 | 100 |
| SWE2_Tourplanner.ImportView . . . . .                                       | 104 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                         | 138 |
| SWE2_Tourplanner.Dialogs.IDialog . . . . .                                  | 92  |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                          | 186 |
| SWE2_Tourplanner.Dialogs.IDialogRequestClose . . . . .                      | 94  |
| SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel . . . . .          | 33  |
| SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel . . . . .             | 40  |
| SWE2_Tourplanner.ViewModels.ErrorViewModel . . . . .                        | 72  |
| SWE2_Tourplanner.ViewModels.ImportConfigViewModel . . . . .                 | 102 |
| SWE2_Tourplanner.ViewModels.ImportViewModel . . . . .                       | 106 |
| SWE2_Tourplanner.ViewModels.ManeuversViewModel . . . . .                    | 146 |
| SWE2_Tourplanner.Dialogs.IDialogService . . . . .                           | 95  |
| SWE2_Tourplanner.Dialogs.DialogService . . . . .                            | 65  |
| IDisposable   |     |
| DataAccessLayer.DBConnection.IDataReader . . . . .                          | 85  |

|   |     |
|---|-----|
| DataAccessLayer.DBConnection.NpgsqlDataReader . . . . .                 | 157 |
| DataAccessLayer.UnitOfWork.IUnitOfWork . . . . .                        | 127 |
| DataAccessLayer.UnitOfWork.UnitOfWork . . . . .                         | 216 |
| IDocument   |     |
| BusinessLogicLayer.PDFCreation.SummaryReport . . . . .                  | 165 |
| BusinessLogicLayer.PDFCreation.TourReport . . . . .                     | 203 |
| BusinessLogicLayer.ExporterImporter.IExporterImporter< T > . . . . .    | 96  |
| BusinessLogicLayer.ExporterImporter.IExporterImporter< Tour > . . . . . | 96  |
| BusinessLogicLayer.ExporterImporter.TourExporterImporter . . . . .      | 177 |
| Common.MapQuestClient.IMapQuestClient . . . . .                         | 98  |
| Common.MapQuestClient.MapQuestClient . . . . .                          | 150 |
| INotifyPropertyChanged  |     |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . .   | 76  |
| SWE2_Tourplanner.ViewModels.BaseViewModel . . . . .                     | 26  |
| SWE2_Tourplanner.MainViewModel . . . . .                                | 131 |
| SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel . . . . .      | 33  |
| SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel . . . . .         | 40  |
| SWE2_Tourplanner.ViewModels.ErrorViewModel . . . . .                    | 72  |
| SWE2_Tourplanner.ViewModels.ImportConfigViewModel . . . . .             | 102 |
| SWE2_Tourplanner.ViewModels.ImportViewModel . . . . .                   | 106 |
| SWE2_Tourplanner.ViewModels.ManeuversViewModel . . . . .                | 146 |
| SWE2_Tourplanner.ViewModels.TourDetailViewModel . . . . .               | 173 |
| SWE2_Tourplanner.ViewModels.ToursViewModel . . . . .                    | 213 |
| System.Windows.Markup.InternalTypeHelper                                |     |
| XamlGeneratedNamespace.GeneratedInternalTypeHelper . . . . .            | 82  |
| XamlGeneratedNamespace.GeneratedInternalTypeHelper . . . . .            | 82  |
| DataAccessLayer.Repositories.IRepository< T > . . . . .                 | 115 |
| DataAccessLayer.Repositories.IRepository< Maneuver > . . . . .          | 115 |
| DataAccessLayer.Repositories.IManeuverRepository . . . . .              | 98  |
| DataAccessLayer.Repositories.ManeuverRepository . . . . .               | 141 |
| DataAccessLayer.Repositories.IRepository< Tour > . . . . .              | 115 |
| DataAccessLayer.Repositories.ITourRepository . . . . .                  | 127 |
| DataAccessLayer.Repositories.TourRepository . . . . .                   | 206 |
| DataAccessLayer.Repositories.IRepository< TourLog > . . . . .           | 115 |
| DataAccessLayer.Repositories.ITourLogRepository . . . . .               | 124 |
| DataAccessLayer.Repositories.TourLogRepository . . . . .                | 191 |
| BusinessLogicLayer.Factories.ITourFactory . . . . .                     | 119 |
| BusinessLogicLayer.Factories.TourFactory . . . . .                      | 179 |
| BusinessLogicLayer.Factories.ITourLogFactory . . . . .                  | 122 |
| BusinessLogicLayer.Factories.TourLogFactory . . . . .                   | 188 |
| Common.Config.ITourPlannerConfig . . . . .                              | 124 |
| Common.Config.TourPlannerConfig . . . . .                               | 197 |
| BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator . . . . .   | 126 |
| BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator . . . . .    | 200 |
| IValueConverter   |     |
| SWE2_Tourplanner.Converters.RouteInfoImageConverter . . . . .           | 162 |
| SWE2_Tourplanner.Converters.StringToDateTimeConverter . . . . .         | 164 |
| Common.Logging.LogHelper . . . . .                                      | 129 |
| SWE2_Tourplanner_Tests.CommonTests.LogHelperTests . . . . .             | 131 |
| Common.Entities.Maneuver . . . . .                                      | 138 |
| SWE2_Tourplanner_Tests.DALTests.ManeuverRepositoryTests . . . . .       | 145 |
| Common.Entities.Tour . . . . .  | 172 |
| SWE2_Tourplanner_Tests.BLTests.TourFactoryTests . . . . .               | 184 |
| Common.Entities.TourLog . . . . .                                       | 185 |
| SWE2_Tourplanner_Tests.BLTests.TourLogFactoryTest . . . . .             | 191 |

|   |     |
|---|-----|
| SWE2_Tourplanner_Tests.DALTests.TourLogRepositoryTests . . . . .      | 196 |
| SWE2_Tourplanner_Tests.CommonTests.TourPlannerConfigTests . . . . .   | 200 |
| SWE2_Tourplanner_Tests.DALTests.TourRepositoryTests . . . . .         | 213 |
| SWE2_Tourplanner_Tests.DALTests.UnitOfWorkTests . . . . .             | 221 |
| System.Windows.Controls.UserControl                                   |     |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . . | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . . | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . . | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . . | 76  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl . . . . . | 76  |
| SWE2_Tourplanner.ReusableComponents.TemperatureSliders . . . . .      | 169 |
| SWE2_Tourplanner.ViewModels.TemperatureSlidersViewModel . . . . .     | 170 |
| ValidationRule  |     |
| SWE2_Tourplanner.ValidationRules.DoubleRangeRule . . . . .            | 68  |
| SWE2_Tourplanner.ValidationRules.ManeuverValidationRule . . . . .     | 149 |
| SWE2_Tourplanner.ValidationRules.NoNegativeDistancesRule . . . . .    | 154 |
| SWE2_Tourplanner.ValidationRules.TextInputValidationRule . . . . .    | 171 |
| SWE2_Tourplanner_Tests.UITests.ValidationRulesTests . . . . .         | 229 |
| System.Windows.Window   |     |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                       | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                       | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                       | 38  |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                       | 38  |
| SWE2_Tourplanner.ErrorView . . . . .                                  | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                  | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                  | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                  | 70  |
| SWE2_Tourplanner.ErrorView . . . . .                                  | 70  |
| SWE2_Tourplanner.ImportConfigView . . . . .                           | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                           | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                           | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                           | 100 |
| SWE2_Tourplanner.ImportConfigView . . . . .                           | 100 |
| SWE2_Tourplanner.ImportView . . . . .                                 | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                 | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                 | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                 | 104 |
| SWE2_Tourplanner.ImportView . . . . .                                 | 104 |
| SWE2_Tourplanner.MainWindow . . . . .                                 | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                 | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                 | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                 | 136 |
| SWE2_Tourplanner.MainWindow . . . . .                                 | 136 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                   | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                   | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                   | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                   | 138 |
| SWE2_Tourplanner.ManeuverManagementWindow . . . . .                   | 138 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                    | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                    | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                    | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                    | 186 |
| SWE2_Tourplanner.TourLogCreateUpdateView . . . . .                    | 186 |
| Window  |     |
| SWE2_Tourplanner.CreateUpdateTourView . . . . .                       | 38  |

## Chapter 3

# Class Index

### 3.1 Class List

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

|   |    |
|---|----|
| <a href="#">SWE2_Tourplanner.App</a>  | 23 |
| Interaction logic for App.xaml . . . . .  |    |
| <a href="#">SWE2_Tourplanner.ViewModels.BaseViewModel</a>   | 26 |
| <a href="#">BaseViewModel</a> provides inheriting classes with PropertyChanged and OnPropertyChanged. It allows inheriting <a href="#">ViewModels</a> to notify the UI about data changes . . . . .                   |    |
| <a href="#">BusinessLogicLayer.Exceptions.BLExporterImporterException</a>   | 27 |
| An exception that gets thrown if errors occur during the import / export process of tour data . . . . .   |    |
| <a href="#">BusinessLogicLayer.Exceptions.BLFactoryException</a>  | 29 |
| An exception that gets thrown if errors occur in factory classes . . . . .  |    |
| <a href="#">BusinessLogicLayer.Exceptions.BLPDFCreationException</a>  | 30 |
| An exception that gets thrown if errors occur during the pdf creation process . . . . .   |    |
| <a href="#">Common.Exceptions.CommonConfigException</a>   | 32 |
| An exception that gets thrown when errors occur in TourPlannerConfig . . . . .  |    |
| <a href="#">SWE2_Tourplanner_Tests.UITests.ConverterTests</a>   | 33 |
| <a href="#">SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel</a>  | 33 |
| <a href="#">CreateUpdateTourLogViewModel</a> is used for creating and updating TourLog entities. To be precise, it is used for setting the values and initiating creation / update process in another class . . . . . |    |
| <a href="#">SWE2_Tourplanner.CreateUpdateTourView</a>   | 38 |
| Interaction logic for <a href="#">CreateUpdateTourView</a> . . . . .  |    |
| <a href="#">SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel</a>   | 40 |
| <a href="#">CreateUpdateTourViewModel</a> is used for creating / updating tours with MapQuest or manually. To be precise, it can start the creation / update process of tours . . . . .                               |    |
| <a href="#">DataAccessLayer.Exceptions.DALDBConnectionException</a>   | 45 |
| An exception that gets thrown when errors regarding database connection / queries / execution of statements occur . . . . .   |    |
| <a href="#">DataAccessLayer.Exceptions.DALParameterException</a>  | 46 |
| An exception that gets thrown when errors occur whilst declaring / setting parameters of database commands . . . . .  |    |
| <a href="#">DataAccessLayer.Exceptions.DALRepositoryCommandException</a>  | 48 |
| An exception that gets thrown when repositories can't create insert / update / delete commands . . . . .  |    |
| <a href="#">DataAccessLayer.Exceptions.DALUnitOfWorkException</a>   | 49 |
| An exception that gets thrown when something goes wrong during commits or rollbacks of <a href="#">IUnitOfWork</a> classes . . . . .  |    |
| <a href="#">DataAccessLayer.DBConnection.DatabaseConnection</a>   | 51 |
| <a href="#">DatabaseConnection</a> implements <a href="#">IDBConnection</a> and is the DBConnection-Class for Npgsql. Is a singleton . . . . .  |    |

|   |     |
|---|-----|
| SWE2_Tourplanner_Tests.DALTests.DBConnectionTests   | 57  |
| DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand   |     |
| The DeleteManeuverCommand is used for deleting specific maneuvers   | 57  |
| DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand   |     |
| The DeleteTourCommand is used for deleting tours  | 60  |
| DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand   |     |
| DeleteTourLogCommand is used for creating new logs of tours   | 62  |
| SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs  |     |
| DialogCloseRequestedEventArgs is an EventArgs used for abstracting / closing Dialogs  | 64  |
| SWE2_Tourplanner.Dialogs.DialogService  |     |
| Concrete implementation of IDialogService   | 65  |
| SWE2_Tourplanner.ValidationRules.DoubleRangeRule  |     |
| DoubleRangeRule is a ValidationRule for validating whether a string converted to a double value is within a Min and Max Value   | 68  |
| SWE2_Tourplanner.ErrorView  |     |
| Interaction logic for ErrorView   | 70  |
| SWE2_Tourplanner.ViewModels.ErrorViewModel  |     |
| ErrorViewModel is used for informing the user about exceptions / errors that occurred during execution of other methods   | 72  |
| SWE2_Tourplanner_Tests.CommonTests.ExtensionMethodsTest   | 75  |
| Common.Extensions   |     |
| Class Extensions is used for providing new extension methods to the application   | 75  |
| SWE2_Tourplanner.ReusableComponents.ExternalResourceControl   |     |
| ExternalResourceControl   | 76  |
| XamlGeneratedNamespace.GeneratedInternalTypeHelper  |     |
| GeneratedInternalTypeHelper   | 82  |
| DataAccessLayer.DBConnection.IDataReader  |     |
| Interface used for exposing only necessary methods / attributes of DataReaders to the application   | 85  |
| DataAccessLayer.DBCommands.IDBCommand   |     |
| The IDBCommand interfaces defines two functions - Execute() and Undo() - which are needed for commits and rollbacks. In a commit of a IUnitOfWork, the Execute() function of every issued command is called. In a rollback, the Undo() function is executed. Is the interface for the command pattern | 86  |
| DataAccessLayer.DBConnection.IDbCommand   |     |
| Interface used for exposing only necessary methods / properties to the application  | 88  |
| DataAccessLayer.DBConnection.IDBConnection  |     |
| The IDbConnection interface defines several methods for setting parameters in IDbCommand instances, opening/closing the database connection and for querying / executing statements   | 89  |
| SWE2_Tourplanner.Dialogs.IDialog  |     |
| IDialog only defines necessary properties / methods for windows to be opened as dialogs   | 92  |
| SWE2_Tourplanner.Dialogs.IDialogRequestClose  |     |
| IDialogRequestClose defines an event for implementing classes, to ensure that the associated dialog gets closed with the according result   | 94  |
| SWE2_Tourplanner.Dialogs.IDialogService   |     |
| IDialogService defines the necessary methods for managing dialogs and the associated ViewModels   | 95  |
| BusinessLogicLayer.ExporterImporter.IExporterImporter< T >  |     |
| A generic interface that defines basic methods for importing and exporting entities   | 96  |
| DataAccessLayer.Repositories.IManeuverRepository  |     |
| The IManeuverRepository interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from IRepository<Maneuver>   | 98  |
| Common.MapQuestClient.IMapQuestClient   |     |
| The IMapQuestClient interface defines two methods for retrieving data from the MapQuest API   | 98  |
| SWE2_Tourplanner.ImportConfigView   |     |
| Interaction logic for ImportConfigView  | 100 |



|   |     |
|---|-----|
| <a href="#">SWE2_Tourplanner.ViewModels.ImportConfigViewModel</a>   |     |
| <a href="#">ImportConfigViewModel</a> is used for importing / selecting new configuration files during the execution . . . . .  | 102 |
| <a href="#">SWE2_Tourplanner.ImportView</a>   |     |
| Interaction logic for <a href="#">ImportView</a> . . . . .  | 104 |
| <a href="#">SWE2_Tourplanner.ViewModels.ImportViewModel</a>   |     |
| <a href="#">ImportViewModel</a> is used for selecting JSON files for the import process . . . . .   | 106 |
| <a href="#">DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand</a>   |     |
| The <a href="#">InsertManeuverCommand</a> is used for inserting new maneuvers . . . . .   | 109 |
| <a href="#">DataAccessLayer.DBCommands.TourCommands.InsertTourCommand</a>   |     |
| <a href="#">InsertTourCommand</a> is used for inserting new tours . . . . .   | 111 |
| <a href="#">DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand</a>   |     |
| <a href="#">InsertTourLogCommand</a> is used for inserting new logs for tours . . . . .   | 113 |
| <a href="#">DataAccessLayer.Repositories.IRepository&lt; T &gt;</a>   |     |
| Generic repository that defines basic functions for querying, inserting, updating and deleting data   | 115 |
| <a href="#">BusinessLogicLayer.Factories.ITourFactory</a>   |     |
| <a href="#">ITourFactory</a> defines several methods for managing and retrieving Tour data from datastore . . .   | 119 |
| <a href="#">BusinessLogicLayer.Factories.ITourLogFactory</a>  |     |
| <a href="#">ITourLogFactory</a> defines several methods for managing TourLog entities . . . . .   | 122 |
| <a href="#">DataAccessLayer.Repositories.ITourLogRepository</a>   |     |
| The <a href="#">ITourLogRepository</a> interface is a place, where specific functions for tourlogs can be defined.  |     |
| The basic methods of repositories are derived from <a href="#">IRepository&lt;TourLog&gt;</a> . . . . .   | 124 |
| <a href="#">Common.Config.ITourPlannerConfig</a>  |     |
| The <a href="#">ITourPlannerConfig</a> interface ensures that all derived TourPlannerConfigs provide a Map<QuestKey, a PictureDirectory, an ExportsDirectory, a DatabaseConnectionString and the option to load another config file during the execution of the program . . . . .   | 124 |
| <a href="#">BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator</a>   |     |
| An interface that defines methods for generating different reports . . . . .  | 126 |
| <a href="#">DataAccessLayer.Repositories.ITourRepository</a>  |     |
| The <a href="#">ITourRepository</a> interface is a place, where specific functions for maneuvers can be defined.  |     |
| The basic methods of repositories are derived from <a href="#">IRepository&lt;Tour&gt;</a> . . . . .  | 127 |
| <a href="#">DataAccessLayer.UnitOfWork.IUnitOfWork</a> . . . . .  | 127 |
| <a href="#">Common.Logging.LogHelper</a>  |     |
| The <a href="#">LogHelper</a> class is responsible for creating ILog objects. Those objects are used throughout the different layers of the application for logging purposes. Is a singleton . . . . .  | 129 |
| <a href="#">SWE2_Tourplanner_Tests.CommonTests.LogHelperTests</a> . . . . .   | 131 |
| <a href="#">SWE2_Tourplanner.MainViewModel</a>  |     |
| <a href="#">MainViewModel</a> is core of the application and is used for executing several different actions triggered by other <a href="#">ViewModels</a> . . . . .  | 131 |
| <a href="#">SWE2_Tourplanner.MainWindow</a>   |     |
| Interaction logic for <a href="#">MainWindow</a> . . . . .  | 136 |
| <a href="#">Common.Entities.Maneuver</a>  |     |
| The maneuver entity is basically a single part of the overall route description of a tour. It consists of a narrative, describing the part of the route in detail, and the length (km) of the section . . .   | 138 |
| <a href="#">SWE2_Tourplanner.ManeuverManagementWindow</a>   |     |
| Interaction logic for <a href="#">ManeuverManagementWindow</a> . . . . .  | 138 |
| <a href="#">DataAccessLayer.Repositories.ManeuverRepository</a>   |     |
| ManeuverRepositories are used for querying, inserting, updating and deleting maneuvers. DML commands are not executed in this class. The only way to truly insert, update and delete maneuvers is through the <a href="#">UnitOfWork</a> class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere . . . . . | 141 |
| <a href="#">SWE2_Tourplanner_Tests.DALTests.ManeuverRepositoryTests</a> . . . . .   | 145 |
| <a href="#">SWE2_Tourplanner.ViewModels.ManeuversViewModel</a>  |     |
| <a href="#">ManeuversViewModel</a> is used for managing Maneuver entities of a Tour . . . . .   | 146 |
| <a href="#">SWE2_Tourplanner.ValidationRules.ManeuverValidationRule</a>   |     |
| <a href="#">ManeuverValidationRule</a> is a ValidationRule for Maneuver entities inside a Datagrid . . . . .  | 149 |
| <a href="#">Common.MapQuestClient.MapQuestClient</a>  |     |
| Concrete implementation of interface <a href="#">IMapQuestClient</a> . . . . .  | 150 |

|  |     |
|--|-----|
| <a href="#">SWE2_Tourplanner.ValidationRules.NoNegativeDistancesRule</a>   |     |
| <a href="#">NoNegativeDistancesRule</a> is a <a href="#">ValidationRule</a> for validating that double values used for distance-like properties aren't negative  | 154 |
| <a href="#">DataAccessLayer.DBConnection.NpgsqlCommand</a>   |     |
| <a href="#">NpgsqlCommand</a> is the command class for NPGSQL commands and it implements the functions defined in the <a href="#">IDbCommand</a> interface   | 155 |
| <a href="#">DataAccessLayer.DBConnection.NpgsqlDataReader</a>  |     |
| <a href="#">NpgsqlDataReader</a> is the concrete <a href="#">IDataReader</a> implementation for Npgsql package. All functions/properties direct requests to a <a href="#">Npgsql.NpgsqlDataReader</a> object   | 157 |
| <a href="#">SWE2_Tourplanner.Commands.RelayCommand</a>   |     |
| <a href="#">RelayCommand</a> is a concrete implementation of <a href="#">ICommand</a> and allows us to create new commands, without having to create new concrete classes  | 160 |
| <a href="#">SWE2_Tourplanner.Converters.RouteInfoImageConverter</a>  |     |
| <a href="#">RouteInfoImageConverter</a> is a used for string - <a href="#">BitmapImage</a> conversion  | 162 |
| <a href="#">SWE2_Tourplanner.Converters.StringToDateTimeConverter</a>  |     |
| <a href="#">StringToDateTimeConverter</a> is used for string - <a href="#">DateTime</a> conversion   | 164 |
| <a href="#">BusinessLogicLayer.PDFCreation.SummaryReport</a>   |     |
| <a href="#">SummaryReport</a> defines the generation process of Tour Summary Reports as PDF  | 165 |
| <a href="#">SWE2_Tourplanner.ReusableComponents.TemperatureSliders</a>   |     |
| <a href="#">TemperatureSliders</a>   | 169 |
| <a href="#">SWE2_Tourplanner.ViewModels.TemperatureSlidersViewModel</a>  |     |
| <a href="#">TemperatureSlidersViewModel</a>  | 170 |
| <a href="#">SWE2_Tourplanner.ValidationRules.TextInputValidationRule</a>   |     |
| <a href="#">TextInputValidationRule</a> is a <a href="#">ValidationRule</a> for the length of text input   | 171 |
| <a href="#">Common.Entities.Tour</a>   |     |
| The tour is the most important entity of the project. It represents a route  | 172 |
| <a href="#">SWE2_Tourplanner.ViewModels.TourDetailViewModel</a>  |     |
| <a href="#">TourDetailViewModel</a> provides a <a href="#">Tour</a> and commands for managing its <a href="#">TourLogs</a>   | 173 |
| <a href="#">BusinessLogicLayer.ExporterImporter.TourExporterImporter</a>   |     |
| Concrete implementation of <a href="#">IExporterImporter</a> for <a href="#">Tour</a> entities   | 177 |
| <a href="#">BusinessLogicLayer.Factories.TourFactory</a>   |     |
| <a href="#">TourFactory</a> is a concrete implementation of <a href="#">ITourFactory</a>   | 179 |
| <a href="#">SWE2_Tourplanner_Tests.BLTests.TourFactoryTests</a>  | 184 |
| <a href="#">Common.Entities.TourLog</a>  |     |
| The tourlog is an entry from a person, who participated in a tour. It's basically a short review of the tour   | 185 |
| <a href="#">SWE2_Tourplanner.TourLogCreateUpdateView</a>   |     |
| <a href="#">TourLogCreateUpdateView</a>  | 186 |
| <a href="#">BusinessLogicLayer.Factories.TourLogFactory</a>  |     |
| <a href="#">TourLogFactory</a> is a concrete implementation of <a href="#">ITourLogFactory</a>   | 188 |
| <a href="#">SWE2_Tourplanner_Tests.BLTests.TourLogFactoryTest</a>  | 191 |
| <a href="#">DataAccessLayer.Repositories.TourLogRepository</a>   |     |
| <a href="#">TourLogRepositories</a> are used for querying, inserting, updating and deleting tourlogs. DML commands are not executed in this class. The only way to truly insert, update and delete tourlogs is through the <a href="#">UnitOfWork</a> class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere | 191 |
| <a href="#">SWE2_Tourplanner_Tests.DALTests.TourLogRepositoryTests</a>   | 196 |
| <a href="#">Common.Config.TourPlannerConfig</a>  |     |
| <a href="#">TourPlannerConfig</a> is a concrete implementation of the <a href="#">ITourPlannerConfig</a> interface. It is a singleton and provides the application with vital config data  | 197 |
| <a href="#">SWE2_Tourplanner_Tests.CommonTests.TourPlannerConfigTests</a>  | 200 |
| <a href="#">BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator</a>   |     |
| <a href="#">TourPlannerReportsGenerator</a> is a concrete implementation of <a href="#">ITourPlannerReportsGenerator</a> . It generates tour- and summary reports as PDF files   | 200 |
| <a href="#">BusinessLogicLayer.PDFCreation.TourReport</a>  |     |
| <a href="#">TourReport</a> defines the creation process of tour reports (PDF)  | 203 |

|   |     |
|---|-----|
| <a href="#">DataAccessLayer.Repositories.TourRepository</a>   |     |
| TourRepositories are used for querying, inserting, updating and deleting tours. DML commands are not executed in this class. The only way to truly insert, update and delete tours is through the <a href="#">UnitOfWork</a> class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere . . . . . | 206 |
| <a href="#">SWE2_Tourplanner_Tests.DALTests.TourRepositoryTests</a> . . . . .   | 213 |
| <a href="#">SWE2_Tourplanner.ViewModels.ToursViewModel</a>  |     |
| TourViewModel provides tours and a way to search / filter tours . . . . .   | 213 |
| <a href="#">DataAccessLayer.UnitOfWork.UnitOfWork</a>   |     |
| Concrete implementation of the <a href="#">IUnitOfWork</a> interface . . . . .  | 216 |
| <a href="#">SWE2_Tourplanner_Tests.DALTests.UnitOfWorkTests</a> . . . . .   | 221 |
| <a href="#">DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand</a>   |     |
| UpdateManeuverCommand is used for updating maneuvers . . . . .  | 222 |
| <a href="#">DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand</a>   |     |
| UpdateTourCommand is used for updating tours . . . . .  | 224 |
| <a href="#">DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand</a>   |     |
| UpdateTourLogCommand is used for updating tourlogs . . . . .  | 227 |
| <a href="#">SWE2_Tourplanner_Tests.UITests.ValidationRulesTests</a> . . . . .   | 229 |



## Chapter 4

# Namespace Documentation

### 4.1 BusinessLogicLayer Namespace Reference

### 4.2 BusinessLogicLayer.Exceptions Namespace Reference

#### Classes

- class [BLExporterImporterException](#)  
*An exception that gets thrown if errors occur during the import / export process of tour data*
- class [BLFactoryException](#)  
*An exception that gets thrown if errors occur in factory classes*
- class [BLPDFCreationException](#)  
*An exception that gets thrown if errors occur during the pdf creation process*

### 4.3 BusinessLogicLayer.ExporterImporter Namespace Reference

#### Classes

- interface [IExporterImporter](#)  
*A generic interface that defines basic methods for importing and exporting entities*
- class [TourExporterImporter](#)  
*Concrete implementation of [IExporterImporter](#) for Tour entities*

### 4.4 BusinessLogicLayer.Factories Namespace Reference

#### Classes

- interface [ITourFactory](#)  
*[ITourFactory](#) defines several methods for managing and retrieving Tour data from datastore*
- interface [ITourLogFactory](#)  
*[ITourLogFactory](#) defines several methods for managing TourLog entities*
- class [TourFactory](#)  
*[TourFactory](#) is a concrete implementation of [ITourFactory](#).*
- class [TourLogFactory](#)  
*[TourLogFactory](#) is a concrete implementation of [ITourLogFactory](#).*

## 4.5 BusinessLogicLayer.PDFCreation Namespace Reference

### Classes

- interface [ITourPlannerReportsGenerator](#)  
*An interface that defines methods for generating different reports*
- class [SummaryReport](#)  
*[SummaryReport](#) defines the generation process of Tour Summary Reports as PDF*
- class [TourPlannerReportsGenerator](#)  
*[TourPlannerReportsGenerator](#) is a concrete implementation of [ITourPlannerReportsGenerator](#). It generates tour- and summary reports as PDF files.*
- class [TourReport](#)  
*[TourReport](#) defines the creation process of tour reports (PDF)*

## 4.6 Common Namespace Reference

### Classes

- class [Extensions](#)  
*Class [Extensions](#) is used for providing new extension methods to the application.*

## 4.7 Common.Config Namespace Reference

### Classes

- interface [ITourPlannerConfig](#)  
*The [ITourPlannerConfig](#) interface ensures that all derived [TourPlannerConfigs](#) provide a [MapQuestKey](#), a [Picture↔Directory](#), an [ExportsDirectory](#), a [DatabaseConnectionString](#) and the option to load another config file during the execution of the program.*
- class [TourPlannerConfig](#)  
*[TourPlannerConfig](#) is a concrete implementation of the [ITourPlannerConfig](#) interface. It is a singleton and provides the application with vital config data.*

## 4.8 Common.Entities Namespace Reference

### Classes

- class [Maneuver](#)  
*The maneuver entity is basically a single part of the overall route description of a tour. It consists of a narrative, describing the part of the route in detail, and the length (km) of the section.*
- class [Tour](#)  
*The tour is the most important entity of the project. It represents a route.*
- class [TourLog](#)  
*The tourlog is an entry from a person, who participated in a tour. It's basically a short review of the tour.*

## 4.9 Common.Enums Namespace Reference

### Enumerations

- enum class [ERouteType](#) {  
    **FASTEST** , **SHORTEST** , **PEDESTRIAN** , **MULTIMODAL** ,  
    **BICYCLE** }  
*Enum for the possible routetypes in mapquest. The route type influences the route itself.*
- enum class [ETravelMethod](#) { **RUNNING** , **HIKING** , **BIKING** }  
*Enum for the different travel methods of users (in tourlogs it is specified how the user travelled. e.g.: Hiking tour, Bike tour, ...).*
- enum class [EWeather](#) {  
    **SUNNY** , **RAINY** , **CLOUDY** , **HAIL** ,  
    **SNOWY** , **STORMY** }  
*Enum for the weather of tour logs.*

### 4.9.1 Enumeration Type Documentation

#### 4.9.1.1 ERouteType

```
enum Common.Enums.ERouteType [strong]
```

Enum for the possible routetypes in mapquest. The route type influences the route itself.

#### 4.9.1.2 ETravelMethod

```
enum Common.Enums.ETravelMethod [strong]
```

Enum for the different travel methods of users (in tourlogs it is specified how the user travelled. e.g.: Hiking tour, Bike tour, ...).

#### 4.9.1.3 EWeather

```
enum Common.Enums.EWeather [strong]
```

Enum for the weather of tour logs.

## 4.10 Common.Exceptions Namespace Reference

### Classes

- class [CommonConfigException](#)  
*An exception that gets thrown when errors occur in TourPlannerConfig*

## 4.11 Common.Logging Namespace Reference

### Classes

- class [LogHelper](#)

*The [LogHelper](#) class is responsible for creating ILog objects. Those objects are used throughout the different layers of the application for logging purposes. Is a singleton.*

## 4.12 Common.MapQuestClient Namespace Reference

### Classes

- interface [IMapQuestClient](#)

*The [IMapQuestClient](#) interface defines two methods for retrieving data from the MapQuest API*

- class [MapQuestClient](#)

*Concrete implementation of interface [IMapQuestClient](#)*

## 4.13 DataAccessLayer Namespace Reference

## 4.14 DataAccessLayer.DBCommands Namespace Reference

### Classes

- interface [IDBCommand](#)

*The [IDBCommand](#) interfaces defines two functions - [Execute\(\)](#) and [Undo\(\)](#) - which are needed for commits and rollbacks. In a commit of a IUnitOfWork, the [Execute\(\)](#) function of every issued command is called. In a rollback, the [Undo\(\)](#) function is executed. Is the interface for the command pattern.*

## 4.15 DataAccessLayer.DBCommands.ManeuverCommands Namespace Reference

### Classes

- class [DeleteManeuverCommand](#)

*The [DeleteManeuverCommand](#) is used for deleting specific maneuvers.*

- class [InsertManeuverCommand](#)

*The [InsertManeuverCommand](#) is used for inserting new maneuvers.*

- class [UpdateManeuverCommand](#)

*[UpdateManeuverCommand](#) is used for updating maneuvers.*



## 4.16 DataAccessLayer.DBCommands.TourCommands Namespace Reference

### Classes

- class [DeleteTourCommand](#)  
*The [DeleteTourCommand](#) is used for deleting tours.*
- class [InsertTourCommand](#)  
*[InsertTourCommand](#) is used for inserting new tours.*
- class [UpdateTourCommand](#)  
*[UpdateTourCommand](#) is used for updating tours.*

## 4.17 DataAccessLayer.DBCommands.TourLogCommands Namespace Reference

### Classes

- class [DeleteTourLogCommand](#)  
*[DeleteTourLogCommand](#) is used for creating new logs of tours.*
- class [InsertTourLogCommand](#)  
*[InsertTourLogCommand](#) is used for inserting new logs for tours.*
- class [UpdateTourLogCommand](#)  
*[UpdateTourLogCommand](#) is used for updating tourlogs.*

## 4.18 DataAccessLayer.DBConnection Namespace Reference

### Classes

- class [DatabaseConnection](#)  
*[DatabaseConnection](#) implements [IDBConnection](#) and is the DBConnection-Class for Npgsql. Is a singleton.*
- interface [IDataReader](#)  
*Interface used for exposing only necessary methods / attributes of DataReaders to the application*
- interface [IDbCommand](#)  
*Interface used for exposing only necessary methods / properties to the application.*
- interface [IDBConnection](#)  
*The IDbConnection interface defines several methods for setting parameters in IDbCommand instances, opening/closing the database connection and for querying / executing statements.*
- class [NpgsqlCommand](#)  
*[NpgsqlCommand](#) is the command class for NPGSQL commands and it implements the functions defined in the [IDbCommand](#) interface.*
- class [NpgsqlDataReader](#)  
*[NpgsqlDataReader](#) is the concrete [IDataReader](#) implementation for Npgsql package. All functions/properties direct requests to a [Npgsql.NpgsqlDataReader](#) object.*

## 4.19 DataAccessLayer.Exceptions Namespace Reference

### Classes

- class [DALDBConnectionException](#)  
*An exception that gets thrown when errors regarding database connection / queries / execution of statements occur*
- class [DALParameterException](#)  
*An exception that gets thrown when errors occur whilst declaring / setting parameters of database commands*
- class [DALRepositoryCommandException](#)  
*An exception that gets thrown when repositories can't create insert / update / delete commands*
- class [DALUnitOfWorkException](#)  
*An exception that gets thrown when something goes wrong during commits or rollbacks of IUnitOfWork classes*

## 4.20 DataAccessLayer.Repositories Namespace Reference

### Classes

- interface [IManeuverRepository](#)  
*The [IManeuverRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Maneuver>](#)*
- interface [IRepository](#)  
*Generic repository that defines basic functions for querying, inserting, updating and deleting data.*
- interface [ITourLogRepository](#)  
*The [ITourLogRepository](#) interface is a place, where specific functions for tourlogs can be defined. The basic methods of repositories are derived from [IRepository<TourLog>](#)*
- interface [ITourRepository](#)  
*The [ITourRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Tour>](#)*
- class [ManeuverRepository](#)  
*ManeuverRepositories are used for querying, inserting, updating and deleting maneuvers. DML commands are not executed in this class. The only way to truly insert, update and delete maneuvers is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.*
- class [TourLogRepository](#)  
*TourLogRepositories are used for querying, inserting, updating and deleting tourlogs. DML commands are not executed in this class. The only way to truly insert, update and delete tourlogs is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.*
- class [TourRepository](#)  
*TourRepositories are used for querying, inserting, updating and deleting tours. DML commands are not executed in this class. The only way to truly insert, update and delete tours is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.*

## 4.21 DataAccessLayer.UnitOfWork Namespace Reference

### Classes

- interface [IUnitOfWork](#)
- class [UnitOfWork](#)  
*Concrete implementation of the [IUnitOfWork](#) interface.*

## 4.22 SWE2\_Tourplanner Namespace Reference

### Classes

- class [App](#)  
*Interaction logic for App.xaml*
- class [CreateUpdateTourView](#)  
*Interaction logic for CreateUpdateTourView*
- class [ErrorView](#)  
*Interaction logic for ErrorView*
- class [ImportConfigView](#)  
*Interaction logic for ImportConfigView*
- class [ImportView](#)  
*Interaction logic for ImportView*
- class [MainWindow](#)  
*Interaction logic for MainWindow*
- class [ManeuverManagementWindow](#)  
*Interaction logic for ManeuverManagementWindow*
- class [TourLogCreateUpdateView](#)  
*TourLogCreateUpdateView*
- class [MainViewModel](#)  
*MainViewModel is core of the application and is used for executing several different actions triggered by other ViewModels*

## 4.23 SWE2\_Tourplanner.Commands Namespace Reference

### Classes

- class [RelayCommand](#)  
*RelayCommand is a concrete implementation of ICommand and allows us to create new commands, without having to create new concrete classes.*

## 4.24 SWE2\_Tourplanner.Converters Namespace Reference

### Classes

- class [RouteInfoImageConverter](#)  
*RouteInfoImageConverter is used for string - BitmapImage conversion.*
- class [StringToDateTimeConverter](#)  
*StringToDateTimeConverter is used for string - DateTime conversion.*

## 4.25 SWE2\_Tourplanner.Dialogs Namespace Reference

### Classes

- class [DialogCloseRequestedEventArgs](#)  
*DialogCloseRequestedEventArgs is an EventArgs used for abstracting / closing Dialogs*
- class [DialogService](#)  
*Concrete implementation of IDialogService*
- interface [IDialog](#)  
*IDialog only defines necessary properties / methods for windows to be opened as dialogs.*
- interface [IDialogRequestClose](#)  
*IDialogRequestClose defines an event for implementing classes, to ensure that the associated dialog gets closed with the according result.*
- interface [IDialogService](#)  
*IDialogService defines the necessary methods for managing dialogs and the associated ViewModels*

## 4.26 SWE2\_Tourplanner.ReusableComponents Namespace Reference

### Classes

- class [ExternalResourceControl](#)  
*ExternalResourceControl*
- class [TemperatureSliders](#)  
*TemperatureSliders*

## 4.27 SWE2\_Tourplanner.ValidationRules Namespace Reference

### Classes

- class [DoubleRangeRule](#)  
*DoubleRangeRule is a ValidationRule for validating whether a string converted to a double value is within a Min and Max Value*
- class [ManeuverValidationRule](#)  
*ManeuverValidationRule is a ValidationRule for Maneuver entities inside a Datagrid.*
- class [NoNegativeDistancesRule](#)  
*NoNegativeDistancesRule is a ValidationRule for validating that double values used for distance-like properties aren't negative*
- class [TextInputValidationRule](#)  
*TextInputValidationRule is a ValidationRule for the length of text input*

## 4.28 SWE2\_Tourplanner.ViewModels Namespace Reference

### Classes

- class [TemperatureSlidersViewModel](#)  
*TemperatureSlidersViewModel*
- class [BaseViewModel](#)  
*BaseViewModel* provides inheriting classes with *PropertyChanged* and *OnPropertyChanged*. It allows inheriting *ViewModels* to notify the UI about data changes.
- class [CreateUpdateTourLogViewModel](#)  
*CreateUpdateTourLogViewModel* is used for creating and updating *TourLog* entities. To be precise, it is used for setting the values and initiating creation / update process in another class.
- class [CreateUpdateTourViewModel](#)  
*CreateUpdateTourViewModel* is used for creating / updating tours with *MapQuest* or manually. To be precise, it can start the creation / update process of tours.
- class [ErrorViewModel](#)  
*ErrorViewModel* is used for informing the user about exceptions / errors that occurred during execution of other methods
- class [ImportConfigViewModel](#)  
*ImportConfigViewModel* is used for importing / selecting new configuration files during the execution
- class [ImportViewModel](#)  
*ImportViewModel* is used for selecting JSON files for the import process
- class [ManeuversViewModel](#)  
*ManeuversViewModel* is used for managing *Maneuver* entities of a *Tour*
- class [TourDetailViewModel](#)  
*TourDetailViewModel* provides a *Tour* and commands for managing its *TourLogs*
- class [ToursViewModel](#)  
*ToursViewModel* provides tours and a way to search / filter tours

## 4.29 SWE2\_Tourplanner\_Tests Namespace Reference

## 4.30 SWE2\_Tourplanner\_Tests.BLTests Namespace Reference

### Classes

- class [TourFactoryTests](#)
- class [TourLogFactoryTest](#)

## 4.31 SWE2\_Tourplanner\_Tests.CommonTests Namespace Reference

### Classes

- class [ExtensionMethodsTest](#)
- class [LogHelperTests](#)
- class [TourPlannerConfigTests](#)

## 4.32 SWE2\_Tourplanner\_Tests.DALTests Namespace Reference

### Classes

- class [DBConnectionTests](#)
- class [ManeuverRepositoryTests](#)
- class [TourLogRepositoryTests](#)
- class [TourRepositoryTests](#)
- class [UnitOfWorkTests](#)

## 4.33 SWE2\_Tourplanner\_Tests.UITests Namespace Reference

### Classes

- class [ConverterTests](#)
- class [ValidationRulesTests](#)

## 4.34 XamlGeneratedNamespace Namespace Reference

### Classes

- class [GeneratedInternalTypeHelper](#)  
*[GeneratedInternalTypeHelper](#)*

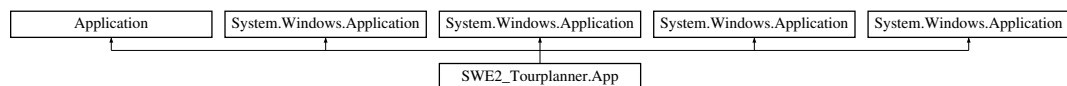
## Chapter 5

# Class Documentation

### 5.1 SWE2\_Tourplanner.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for SWE2\_Tourplanner.App:



#### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

#### Static Public Member Functions

- static void [Main](#) ()  
*Application Entry Point.*
- static void [Main](#) ()  
*Application Entry Point.*
- static void [Main](#) ()  
*Application Entry Point.*
- static void [Main](#) ()  
*Application Entry Point.*

## Protected Member Functions

- override void [OnStartup](#) (StartupEventArgs e)  
*OnStartup registers [ViewModels](#) and their associated Views. It also opens a new [MainWindow](#) instance*

## Private Attributes

- bool `_contentLoaded`

### 5.1.1 Detailed Description

Interaction logic for App.xaml

[App](#)

### 5.1.2 Member Function Documentation

#### 5.1.2.1 `InitializeComponent()` [1/4]

```
void SWE2_Tourplanner.App.InitializeComponent ( )
```

InitializeComponent

#### 5.1.2.2 `InitializeComponent()` [2/4]

```
void SWE2_Tourplanner.App.InitializeComponent ( )
```

InitializeComponent

#### 5.1.2.3 `InitializeComponent()` [3/4]

```
void SWE2_Tourplanner.App.InitializeComponent ( )
```

InitializeComponent



#### 5.1.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.App.InitializeComponent ( )
```

InitializeComponent

#### 5.1.2.5 Main() [1/4]

```
static void SWE2_Tourplanner.App.Main ( ) [static]
```

Application Entry Point.

#### 5.1.2.6 Main() [2/4]

```
static void SWE2_Tourplanner.App.Main ( ) [static]
```

Application Entry Point.

#### 5.1.2.7 Main() [3/4]

```
static void SWE2_Tourplanner.App.Main ( ) [static]
```

Application Entry Point.

#### 5.1.2.8 Main() [4/4]

```
static void SWE2_Tourplanner.App.Main ( ) [static]
```

Application Entry Point.

#### 5.1.2.9 OnStartup()

```
override void SWE2_Tourplanner.App.OnStartup (
    StartupEventArgs e ) [protected]
```

OnStartup registers [ViewModels](#) and their associated Views. It also opens a new [MainWindow](#) instance

## Parameters

|   |   |
|---|---|
| e | EventArgs related to the startup of the application |
|---|---|

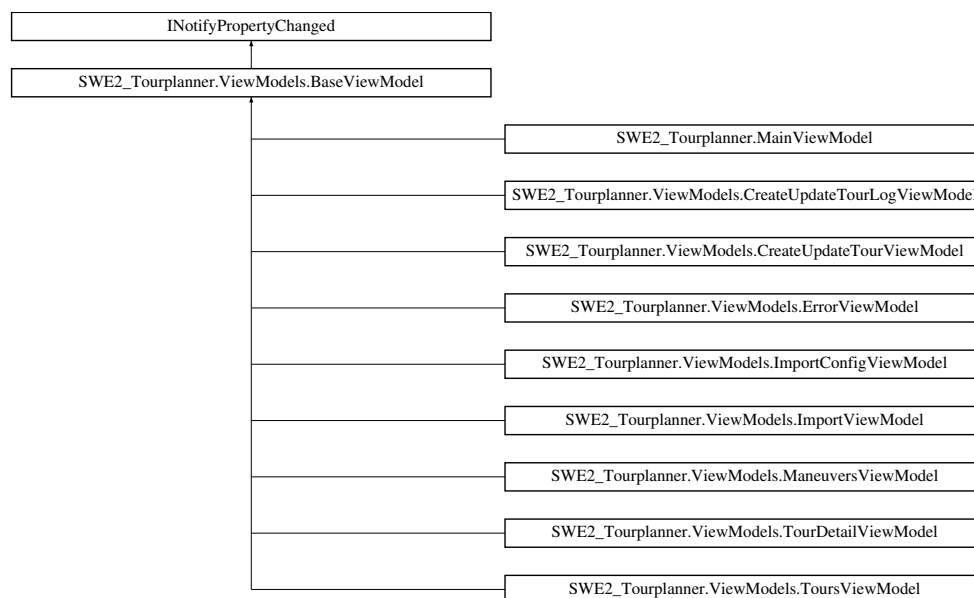
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/App.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/App.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/App.g.i.cs

## 5.2 SWE2\_Tourplanner.ViewModels.BaseViewModel Class Reference

[BaseViewModel](#) provides inheriting classes with `PropertyChanged` and `OnPropertyChanged`. It allows inheriting [ViewModels](#) to notify the UI about data changes.

Inheritance diagram for `SWE2_Tourplanner.ViewModels.BaseViewModel`:



### Protected Member Functions

- virtual void [OnPropertyChanged](#) ([CallerMemberName] string propertyName=null)  
*OnPropertyChanged invokes PropertyChanged and notifies other parts of the application about changes*

### Events

- PropertyChangedEventHandler [PropertyChanged](#)  
*EventHandler used for notifying the UI aspect of the application about data changes*

### 5.2.1 Detailed Description

[BaseViewModel](#) provides inheriting classes with `PropertyChanged` and `OnPropertyChanged`. It allows inheriting [ViewModels](#) to notify the UI about data changes.

### 5.2.2 Member Function Documentation

#### 5.2.2.1 OnPropertyChanged()

```
virtual void SWE2_Tourplanner.ViewModels.BaseViewModel.OnPropertyChanged (
    [CallerMemberName] string propertyName = null ) [protected], [virtual]
```

`OnPropertyChanged` invokes `PropertyChanged` and notifies other parts of the application about changes

Parameters

|                     |                      |
|---------------------|----------------------|
| <i>propertyName</i> | Name of the property |
|---------------------|----------------------|

### 5.2.3 Event Documentation

#### 5.2.3.1 PropertyChanged

```
PropertyChangedEventHandler SWE2_Tourplanner.ViewModels.BaseViewModel.PropertyChanged
```

EventHandler used for notifying the UI aspect of the application about data changes

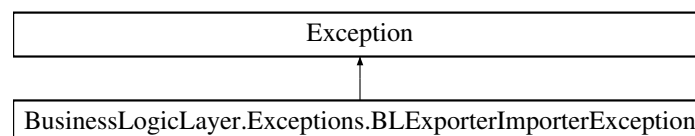
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/BaseViewModel.cs

## 5.3 BusinessLogicLayer.Exceptions.BLExporterImporterException Class Reference

An exception that gets thrown if errors occur during the import / export process of tour data

Inheritance diagram for `BusinessLogicLayer.Exceptions.BLExporterImporterException`:



## Public Member Functions

- [BLExporterImporterException](#) ()  
*Default constructor of [BLExporterImporterException](#)*
- [BLExporterImporterException](#) (string message)  
*A constructor that takes a message as a string parameter and passes it to the base constructor*
- [BLExporterImporterException](#) (string message, Exception inner)  
*A constructor that takes a message and exception and passes it to the base constructor*

### 5.3.1 Detailed Description

An exception that gets thrown if errors occur during the import / export process of tour data

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 [BLExporterImporterException](#)() [1/3]

```
BusinessLogicLayer.Exceptions.BLExporterImporterException.BLExporterImporterException ( )
```

Default constructor of [BLExporterImporterException](#)

#### 5.3.2.2 [BLExporterImporterException](#)() [2/3]

```
BusinessLogicLayer.Exceptions.BLExporterImporterException.BLExporterImporterException (
    string message )
```

A constructor that takes a message as a string parameter and passes it to the base constructor

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

#### 5.3.2.3 [BLExporterImporterException](#)() [3/3]

```
BusinessLogicLayer.Exceptions.BLExporterImporterException.BLExporterImporterException (
    string message,
    Exception inner )
```

A constructor that takes a message and exception and passes it to the base constructor

## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
| <i>inner</i>   | Inner exception          |

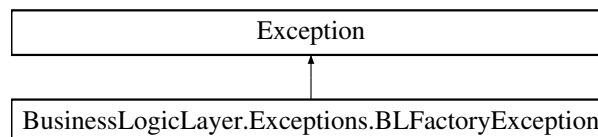
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Exceptions/BLExporterImporterException.cs

## 5.4 BusinessLogicLayer.Exceptions.BLFactoryException Class Reference

An exception that gets thrown if errors occur in factory classes

Inheritance diagram for BusinessLogicLayer.Exceptions.BLFactoryException:



### Public Member Functions

- [BLFactoryException\(\)](#)  
*Default constructor of [BLFactoryException](#)*
- [BLFactoryException\(string message\)](#)  
*A constructor that takes a message and passes it to the base constructor*
- [BLFactoryException\(string message, Exception inner\)](#)  
*A constructor that takes a message and exception and passes them to the base constructor*

### 5.4.1 Detailed Description

An exception that gets thrown if errors occur in factory classes

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 BLFactoryException() [1/3]

```
BusinessLogicLayer.Exceptions.BLFactoryException.BLFactoryException ( )
```

Default constructor of [BLFactoryException](#)

#### 5.4.2.2 BLFactoryException() [2/3]

```
BusinessLogicLayer.Exceptions.BLFactoryException.BLFactoryException (
    string message )
```

A constructor that takes a message and passes it to the base constructor

## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

**5.4.2.3 BLFactoryException() [3/3]**

```
BusinessLogicLayer.Exceptions.BLFactoryException.BLFactoryException (
    string message,
    Exception inner )
```

A constructor that takes a message and exception and passes them to the base constructor

## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
| <i>inner</i>   | Inner exception          |

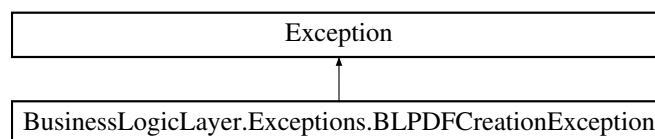
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Exceptions/BLFactoryException.cs

## 5.5 BusinessLogicLayer.Exceptions.BLPDFCreationException Class Reference

An exception that gets thrown if errors occur during the pdf creation process

Inheritance diagram for BusinessLogicLayer.Exceptions.BLPDFCreationException:

**Public Member Functions**

- [BLPDFCreationException \(\)](#)  
*Default constructor of BLPDFCreationException*
- [BLPDFCreationException \(string message\)](#)  
*A constructor that takes a message and passes it to the base constructor*
- [BLPDFCreationException \(string message, Exception inner\)](#)  
*A constructor that takes a message and exception and passes them to the base constructor*

### 5.5.1 Detailed Description

An exception that gets thrown if errors occur during the pdf creation process

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 BLPDFCreationException() [1/3]

```
BusinessLogicLayer.Exceptions.BLPDFCreationException.BLPDFCreationException ( )
```

Default constructor of [BLPDFCreationException](#)

#### 5.5.2.2 BLPDFCreationException() [2/3]

```
BusinessLogicLayer.Exceptions.BLPDFCreationException.BLPDFCreationException (
    string message )
```

A constructor that takes a message and passes it to the base constructor

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

#### 5.5.2.3 BLPDFCreationException() [3/3]

```
BusinessLogicLayer.Exceptions.BLPDFCreationException.BLPDFCreationException (
    string message,
    Exception inner )
```

A constructor that takes a message and exception and passes them to the base constructor

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
| <i>inner</i>   | Inner exception          |

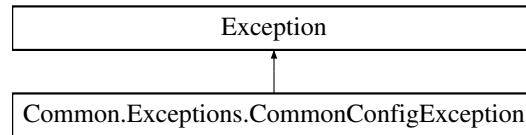
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Exceptions/BLPDFCreationException.cs

## 5.6 Common.Exceptions.CommonConfigException Class Reference

An exception that gets thrown when errors occur in TourPlannerConfig

Inheritance diagram for Common.Exceptions.CommonConfigException:



### Public Member Functions

- [CommonConfigException](#) ()  
*Default constructor for [CommonConfigException](#).*
- [CommonConfigException](#) (string message)  
*Constructor that takes an exception message and passes it to the base constructor of Exception class.*
- [CommonConfigException](#) (string message, Exception inner)  
*Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.*

### 5.6.1 Detailed Description

An exception that gets thrown when errors occur in TourPlannerConfig

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 CommonConfigException() [1/3]

```
Common.Exceptions.CommonConfigException.CommonConfigException ( )
```

Default constructor for [CommonConfigException](#).

#### 5.6.2.2 CommonConfigException() [2/3]

```
Common.Exceptions.CommonConfigException.CommonConfigException (
    string message )
```

Constructor that takes an exception message and passes it to the base constructor of Exception class.



## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

## 5.6.2.3 CommonConfigException() [3/3]

```
Common.Exceptions.CommonConfigException.CommonConfigException (
    string message,
    Exception inner )
```

Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.

## Parameters

|                |  |
|----------------|--|
| <i>message</i> | Message of the exception.                                |
| <i>inner</i>   | Inner exception of <a href="#">CommonConfigException</a> |

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

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Exceptions/CommonConfigException.cs

## 5.7 SWE2\_Tourplanner\_Tests.UITests.ConverterTests Class Reference

## Public Member Functions

- void **StringToDateTimeConverterTest** ()
- void **RouteInfoImageConverter** ()

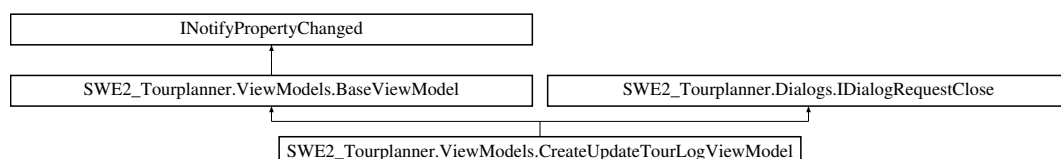
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/UITests/ConverterTests.cs

## 5.8 SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel Class Reference

[CreateUpdateTourLogViewModel](#) is used for creating and updating TourLog entities. To be precise, it is used for setting the values and initiating creation / update process in another class.

Inheritance diagram for SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel:



## Public Member Functions

- [CreateUpdateTourLogViewModel](#) ([TourLog](#) tourLog)  
*Default constructor of [CreateUpdateTourLogViewModel](#)*

## Properties

- DateTime [StartDate](#) [get, set]
- DateTime [EndDate](#) [get, set]
- double [Distance](#) [get, set]
- double [Temperature](#) [get, set]
- double [Rating](#) [get, set]
- [EWeather](#) [Weather](#) [get, set]
- [ETravelMethod](#) [TravelMethod](#) [get, set]
- string [Report](#) [get, set]
- string [ConfirmButtonMessage](#) [get]
- string [HeadingMessage](#) [get]
- IEnumerable< [EWeather](#) > [WeatherTypes](#) [get]
- IEnumerable< [ETravelMethod](#) > [TravelMethods](#) [get]
- ICommand [ExitCommand](#) [get]
- ICommand [ConductCreateUpdateCommand](#) [get]

## Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Event for closing the Dialog*

## Private Attributes

- [TourLog](#) tourLog  
*The TourLog entity, whose values need to be set.*

## Additional Inherited Members

### 5.8.1 Detailed Description

[CreateUpdateTourLogViewModel](#) is used for creating and updating TourLog entities. To be precise, it is used for setting the values and initiating creation / update process in another class.

### 5.8.2 Constructor & Destructor Documentation

#### 5.8.2.1 CreateUpdateTourLogViewModel()

```
SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.CreateUpdateTourLogViewModel (
    TourLog tourLog )
```

Default constructor of [CreateUpdateTourLogViewModel](#)

## Parameters

|                |                                 |
|----------------|---------------------------------|
| <i>tourLog</i> | TourLog to be created / updated |
|----------------|---------------------------------|

## 5.8.3 Member Data Documentation

### 5.8.3.1 tourLog

`TourLog` SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.tourLog [private]

The TourLog entity, whose values need to be set.

## 5.8.4 Property Documentation

### 5.8.4.1 ConductCreateUpdateCommand

`ICommand` SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.ConductCreateUpdateCommand [get]

Command to start create / update process.

### 5.8.4.2 ConfirmButtonMessage

`string` SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.ConfirmButtonMessage [get]

Text on the Confirm Button

### 5.8.4.3 Distance

`double` SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Distance [get], [set]

Distance is the amount of km travelled.

### 5.8.4.4 EndDate

`DateTime` SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.EndDate [get], [set]

DateTime value when the TourLog ended.

#### 5.8.4.5 ExitCommand

`ICommand SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.ExitCommand [get]`

Command to stop the create / update process.

#### 5.8.4.6 HeadingMessage

`string SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.HeadingMessage [get]`

Text in the heading.

#### 5.8.4.7 Rating

`double SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Rating [get], [set]`

Rating of the tour from 0 - 10

#### 5.8.4.8 Report

`string SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Report [get], [set]`

Additional comment on the tour.

#### 5.8.4.9 StartDate

`DateTime SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.StartDate [get], [set]`

DateTime value when the TourLog started.

#### 5.8.4.10 Temperature

`double SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Temperature [get], [set]`

Temperature in Celsius during the tour.

#### 5.8.4.11 TravelMethod

`ETravelMethod SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.TravelMethod [get], [set]`

Travel method used for the tour.

#### 5.8.4.12 TravelMethods

```
IEnumerable<ETravelMethod> SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Travel↔  
Methods [get]
```

All available travel methods.

#### 5.8.4.13 Weather

```
EWeather SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.Weather [get], [set]
```

Weather during the tour.

#### 5.8.4.14 WeatherTypes

```
IEnumerable<EWeather> SWE2_Tourplanner.ViewModels.CreateUpdateTourLogViewModel.WeatherTypes  
[get]
```

All available weather values.

### 5.8.5 Event Documentation

#### 5.8.5.1 CloseRequested

```
EventHandler<DialogCloseRequestedEventArgs> SWE2_Tourplanner.ViewModels.CreateUpdateTourLog↔  
ViewModel.CloseRequested
```

Event for closing the Dialog

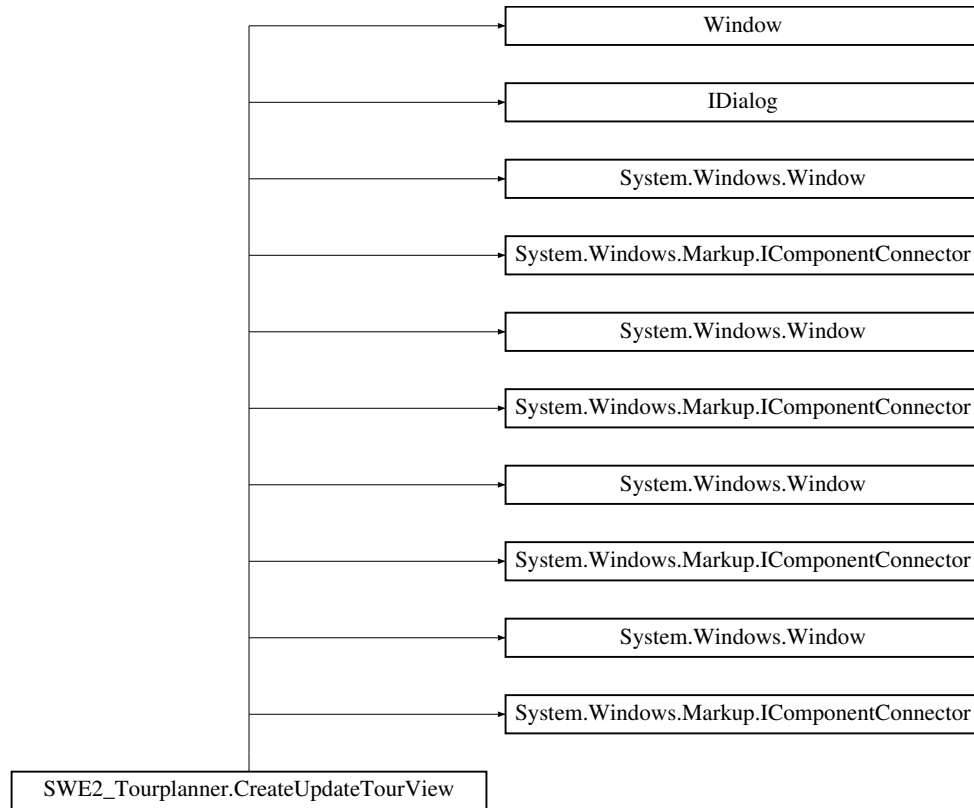
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/CreateUpdateTourLogView↔  
Model.cs

## 5.9 SWE2\_Tourplanner.CreateUpdateTourView Class Reference

Interaction logic for [CreateUpdateTourView](#)

Inheritance diagram for SWE2\_Tourplanner.CreateUpdateTourView:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

## Private Attributes

- `bool _contentLoaded`

### 5.9.1 Detailed Description

Interaction logic for [CreateUpdateTourView](#)

[CreateUpdateTourView](#)

### 5.9.2 Member Function Documentation

#### 5.9.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.CreateUpdateTourView.InitializeComponent ( )
```

InitializeComponent

#### 5.9.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.CreateUpdateTourView.InitializeComponent ( )
```

InitializeComponent

#### 5.9.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.CreateUpdateTourView.InitializeComponent ( )
```

InitializeComponent

#### 5.9.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.CreateUpdateTourView.InitializeComponent ( )
```

InitializeComponent

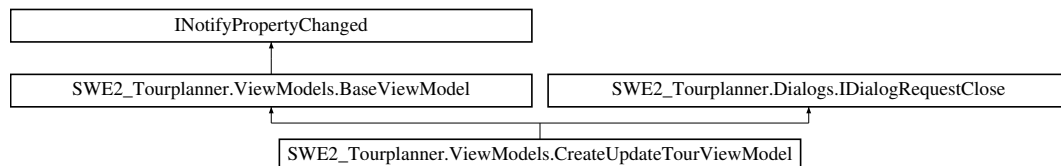
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/CreateUpdateTourView.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/CreateUpdateTour↔View.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/CreateUpdateTour↔View.g.i.cs

## 5.10 SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel Class Reference

[CreateUpdateTourViewModel](#) is used for creating / updating tours with MapQuest or manually. To be precise, it can start the creation / update process of tours.

Inheritance diagram for SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel:



### Public Member Functions

- [CreateUpdateTourViewModel](#) ([Tour](#) tour, [IDialogService](#) dialogService)  
*Default constructor of [CreateUpdateTourViewModel](#)*

### Properties

- string [Name](#) [get, set]
- string [StartLocation](#) [get, set]
- string [EndLocation](#) [get, set]
- string [Description](#) [get, set]
- [ERouteType](#) [RouteType](#) [get, set]
- double [Distance](#) [get, set]
- [ObservableCollection](#)< [Maneuver](#) > [Maneuvers](#) [get, set]
- string [RouteInfo](#) [get, set]
- [IEnumerable](#)< [ERouteType](#) > [RouteTypes](#) [get]
- string [HeadingMessage](#) [get]
- string [ConfirmButtonMessage](#) [get]
- bool [ManualTour](#) [get, set]
- [ICommand](#) [ExitCommand](#) [get]
- [ICommand](#) [CommitCommand](#) [get]
- [ICommand](#) [ManageManeuverCommand](#) [get]
- [ICommand](#) [SelectRouteInfoCommand](#) [get]

### Events

- [EventHandler](#)< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Event for closing the dialog*

### Private Attributes

- [Tour](#) tour  
*Tour to be created / updated*
- bool [manualTour](#) = false  
*Flag that determines if Tour will be created / updated manually or with MapQuest*
- [ObservableCollection](#)< [Maneuver](#) > [maneuvers](#)  
*Collection of maneuvers of the tour*
- [ManeuversViewModel](#) [maneuversViewModel](#)  
*ViewModel for the managing maneuvers*



## Additional Inherited Members

### 5.10.1 Detailed Description

[CreateUpdateTourViewModel](#) is used for creating / updating tours with MapQuest or manually. To be precise, it can start the creation / update process of tours.

### 5.10.2 Constructor & Destructor Documentation

#### 5.10.2.1 CreateUpdateTourViewModel()

```
SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.CreateUpdateTourViewModel (
    Tour tour,
    IDialogService dialogService )
```

Default constructor of [CreateUpdateTourViewModel](#)

##### Parameters

|                      |   |
|----------------------|---|
| <i>tour</i>          | Tour to be updated / created  |
| <i>dialogService</i> | DialogService used for opening <a href="#">ManeuverManagementWindow</a> |

### 5.10.3 Member Data Documentation

#### 5.10.3.1 maneuvers

```
ObservableCollection<Maneuver> SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.maneuvers
[private]
```

Collection of maneuvers of the tour

#### 5.10.3.2 maneuversViewModel

```
ManeuversViewModel SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.maneuversViewModel
[private]
```

ViewModel for the managing maneuvers

### 5.10.3.3 manualTour

```
bool SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.manualTour = false [private]
```

Flag that determines if Tour will be created / updated manually or with MapQuest

### 5.10.3.4 tour

```
Tour SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.tour [private]
```

Tour to be created / updated

## 5.10.4 Property Documentation

### 5.10.4.1 CommitCommand

```
 ICommand SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.CommitCommand [get]
```

Command to initiate the creation / update process

### 5.10.4.2 ConfirmButtonMessage

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.ConfirmButtonMessage [get]
```

Text on the confirm button

### 5.10.4.3 Description

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.Description [get], [set]
```

Description of the tour

### 5.10.4.4 Distance

```
double SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.Distance [get], [set]
```

Length (km) of the tour

### 5.10.4.5 EndLocation

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.EndLocation [get], [set]
```

End location of the tour

#### 5.10.4.6 ExitCommand

```
ICommand SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.ExitCommand [get]
```

Command to exit the dialog

#### 5.10.4.7 HeadingMessage

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.HeadingMessage [get]
```

Text in the heading

#### 5.10.4.8 ManageManeuverCommand

```
ICommand SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.ManageManeuverCommand [get]
```

Command to open ManeuverManagement Dialog

#### 5.10.4.9 Maneuvers

```
ObservableCollection<Maneuver> SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.Maneuvers  
[get], [set]
```

Maneuvers of the tour

#### 5.10.4.10 ManualTour

```
bool SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.ManualTour [get], [set]
```

Flag that indicates whether the tour should be created manually or not

#### 5.10.4.11 Name

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.Name [get], [set]
```

Name of the tour

#### 5.10.4.12 RouteInfo

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.RouteInfo [get], [set]
```

Path to the map image of the tour

#### 5.10.4.13 RouteType

```
ERouteType SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.RouteType [get], [set]
```

Route type of the tour

#### 5.10.4.14 RouteTypes

```
IEnumerable<ERouteType> SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.RouteTypes  
[get]
```

All available route types

#### 5.10.4.15 SelectRouteInfoCommand

```
ICommand SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.SelectRouteInfoCommand [get]
```

Command to select path to map image

#### 5.10.4.16 StartLocation

```
string SWE2_Tourplanner.ViewModels.CreateUpdateTourViewModel.StartLocation [get], [set]
```

Start location of the tour

### 5.10.5 Event Documentation

#### 5.10.5.1 CloseRequested

```
EventHandler<DialogCloseRequestedEventArgs> SWE2_Tourplanner.ViewModels.CreateUpdateTourView↔  
Model.CloseRequested
```

Event for closing the dialog

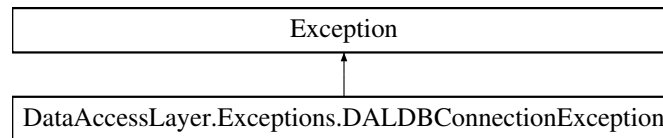
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/CreateUpdateTourViewModel.cs

## 5.11 DataAccessLayer.Exceptions.DALDBConnectionException Class Reference

An exception that gets thrown when errors regarding database connection / queries / execution of statements occur

Inheritance diagram for DataAccessLayer.Exceptions.DALDBConnectionException:



### Public Member Functions

- [DALDBConnectionException](#) ()  
*Default constructor for [DALDBConnectionException](#).*
- [DALDBConnectionException](#) (string message)  
*Constructor that takes an exception message and passes it to the base constructor of Exception class.*
- [DALDBConnectionException](#) (string message, Exception inner)  
*Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.*

#### 5.11.1 Detailed Description

An exception that gets thrown when errors regarding database connection / queries / execution of statements occur

#### 5.11.2 Constructor & Destructor Documentation

##### 5.11.2.1 DALDBConnectionException() [1/3]

```
DataAccessLayer.Exceptions.DALDBConnectionException.DALDBConnectionException ( )
```

Default constructor for [DALDBConnectionException](#).

##### 5.11.2.2 DALDBConnectionException() [2/3]

```
DataAccessLayer.Exceptions.DALDBConnectionException.DALDBConnectionException (
    string message )
```

Constructor that takes an exception message and passes it to the base constructor of Exception class.

## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

## 5.11.2.3 DALDBConnectionException() [3/3]

```

DataAccessLayer.Exceptions.DALDBConnectionException.DALDBConnectionException (
    string message,
    Exception inner )

```

Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.

## Parameters

|                |   |
|----------------|---|
| <i>message</i> | Message of the exception.                                   |
| <i>inner</i>   | Inner exception of <a href="#">DALDBConnectionException</a> |

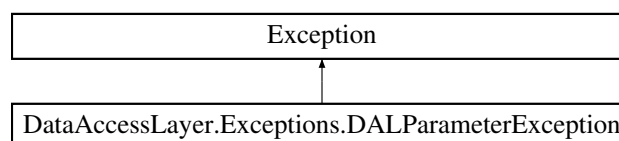
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Exceptions/DALDBConnectionException.cs

## 5.12 DataAccessLayer.Exceptions.DALParameterException Class Reference

An exception that gets thrown when errors occur whilst declaring / setting parameters of database commands

Inheritance diagram for DataAccessLayer.Exceptions.DALParameterException:



### Public Member Functions

- [DALParameterException](#) ()  
*Default constructor for [DALParameterException](#).*
- [DALParameterException](#) (string message)  
*Constructor that takes an exception message and passes it to the base constructor of Exception class.*
- [DALParameterException](#) (string message, Exception inner)  
*Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.*

### 5.12.1 Detailed Description

An exception that gets thrown when errors occur whilst declaring / setting parameters of database commands

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 DALParameterException() [1/3]

```
DataAccessLayer.Exceptions.DALParameterException.DALParameterException ( )
```

Default constructor for [DALParameterException](#).

#### 5.12.2.2 DALParameterException() [2/3]

```
DataAccessLayer.Exceptions.DALParameterException.DALParameterException (
    string message )
```

Constructor that takes an exception message and passes it to the base constructor of Exception class.

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

#### 5.12.2.3 DALParameterException() [3/3]

```
DataAccessLayer.Exceptions.DALParameterException.DALParameterException (
    string message,
    Exception inner )
```

Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.

##### Parameters

|                |  |
|----------------|--|
| <i>message</i> | Message of the exception.                                |
| <i>inner</i>   | Inner exception of <a href="#">DALParameterException</a> |

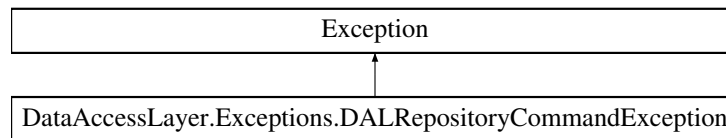
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Exceptions/DALParameterException.cs

## 5.13 DataAccessLayer.Exceptions.DALRepositoryCommandException Class Reference

An exception that gets thrown when repositories can't create insert / update / delete commands

Inheritance diagram for `DataAccessLayer.Exceptions.DALRepositoryCommandException`:



### Public Member Functions

- [DALRepositoryCommandException](#) ()  
*Default constructor for [DALRepositoryCommandException](#).*
- [DALRepositoryCommandException](#) (string message)  
*Constructor that takes an exception message and passes it to the base constructor of `Exception` class.*
- [DALRepositoryCommandException](#) (string message, `Exception` inner)  
*Constructor that takes an exception message and an inner exception and passes it to the base constructor of `Exception` class.*

#### 5.13.1 Detailed Description

An exception that gets thrown when repositories can't create insert / update / delete commands

#### 5.13.2 Constructor & Destructor Documentation

##### 5.13.2.1 DALRepositoryCommandException() [1/3]

```
DataAccessLayer.Exceptions.DALRepositoryCommandException.DALRepositoryCommandException ( )
```

Default constructor for [DALRepositoryCommandException](#).

##### 5.13.2.2 DALRepositoryCommandException() [2/3]

```
DataAccessLayer.Exceptions.DALRepositoryCommandException.DALRepositoryCommandException (
    string message )
```

Constructor that takes an exception message and passes it to the base constructor of `Exception` class.



## Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

## 5.13.2.3 DALRepositoryCommandException() [3/3]

```
DataAccessLayer.Exceptions.DALRepositoryCommandException.DALRepositoryCommandException (
    string message,
    Exception inner )
```

Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.

## Parameters

|                |  |
|----------------|--|
| <i>message</i> | Message of the exception.  |
| <i>inner</i>   | Inner exception of <a href="#">DALRepositoryCommandException</a> |

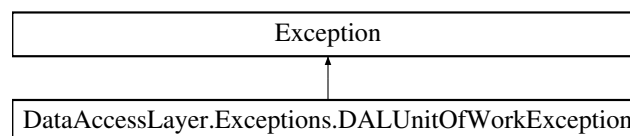
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Exceptions/DALRepositoryCommandException.cs

## 5.14 DataAccessLayer.Exceptions.DALUnitOfWorkException Class Reference

An exception that gets thrown when something goes wrong during commits or rollbacks of IUnitOfWork classes

Inheritance diagram for DataAccessLayer.Exceptions.DALUnitOfWorkException:



### Public Member Functions

- [DALUnitOfWorkException](#) ()  
*Default constructor for [DALUnitOfWorkException](#).*
- [DALUnitOfWorkException](#) (string message)  
*Constructor that takes an exception message and passes it to the base constructor of Exception class.*
- [DALUnitOfWorkException](#) (string message, Exception inner)  
*Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.*

### 5.14.1 Detailed Description

An exception that gets thrown when something goes wrong during commits or rollbacks of IUnitOfWork classes

### 5.14.2 Constructor & Destructor Documentation

#### 5.14.2.1 DALUnitOfWorkException() [1/3]

```
DataAccessLayer.Exceptions.DALUnitOfWorkException.DALUnitOfWorkException ( )
```

Default constructor for [DALUnitOfWorkException](#).

#### 5.14.2.2 DALUnitOfWorkException() [2/3]

```
DataAccessLayer.Exceptions.DALUnitOfWorkException.DALUnitOfWorkException (
    string message )
```

Constructor that takes an exception message and passes it to the base constructor of Exception class.

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>message</i> | Message of the exception |
|----------------|--------------------------|

#### 5.14.2.3 DALUnitOfWorkException() [3/3]

```
DataAccessLayer.Exceptions.DALUnitOfWorkException.DALUnitOfWorkException (
    string message,
    Exception inner )
```

Constructor that takes an exception message and an inner exception and passes it to the base constructor of Exception class.

##### Parameters

|                |   |
|----------------|---|
| <i>message</i> | Message of the exception.                                 |
| <i>inner</i>   | Inner exception of <a href="#">DALUnitOfWorkException</a> |

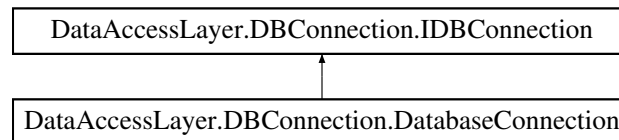
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Exceptions/DALUnitOfWorkException.cs

## 5.15 DataAccessLayer.DBConnection.DatabaseConnection Class Reference

[DatabaseConnection](#) implements [IDBConnection](#) and is the DBConnection-Class for Npgsql. Is a singleton.

Inheritance diagram for DataAccessLayer.DBConnection.DatabaseConnection:



### Public Member Functions

- int [ExecuteStatement](#) ([IDbCommand](#) command)  
*Executes a DML statement and returns the amount of affected rows*
- List< object[] > [QueryDatabase](#) ([IDbCommand](#) command)  
*Executes a query and retrieves the results*
- void [OpenConnection](#) ()  
*Opens the database connection*
- void [CloseConnection](#) ()  
*Closes the database connection*
- int [DeclareParameter](#) ([IDbCommand](#) command, string name, System.Data.DbType type)  
*Declares a parameter in a command*
- void [DefineParameter](#) ([IDbCommand](#) command, string name, System.Data.DbType type, object value)  
*Declares and defines a parameter in a command.*

### Static Public Member Functions

- static [IDBConnection GetDBConnection](#) ()  
*Ensures that the [DatabaseConnection](#) constructor only gets called once. Provides the [DatabaseConnection](#) instance.*
- static [IDBConnection GetDBConnection](#) ([ITourPlannerConfig](#) testConfig)  
*Ensures that the [DatabaseConnection](#) constructor only gets called once. Used for testing purposes. The test configuration needs to be passed*

### Private Member Functions

- [DatabaseConnection](#) ()  
*Creates a [DatabaseConnection](#) instance. Is only called once due to [GetDBConnection\(\)](#) and the private access modifier.*
- [DatabaseConnection](#) ([ITourPlannerConfig](#) config)  
*Creates a [DatabaseConnection](#) instance. Is only called once due to [GetDBConnection\(ITourPlannerConfig testConfig\)](#) and the private access modifier.*

## Private Attributes

- Npgsql.NpgsqlConnection [nppgsqlConnection](#)  
*Connection to the Postgres database*
- ITourPlannerConfig [config](#)  
*Tourplanner config object. Used to retrieve the newest config data.*
- log4net.ILog [logger](#)  
*ILog object used for logging.*

## Static Private Attributes

- static [DatabaseConnection](#) [dB](#)  
*Only instance of the [DatabaseConnection](#) class. Gets set once.*

### 5.15.1 Detailed Description

[DatabaseConnection](#) implements [IDBConnection](#) and is the DBConnection-Class for Npgsql. Is a singleton.

### 5.15.2 Constructor & Destructor Documentation

#### 5.15.2.1 DatabaseConnection() [1/2]

```
.DataAccessLayer.DBConnection.DatabaseConnection.DatabaseConnection ( ) [private]
```

Creates a [DatabaseConnection](#) instance. Is only called once due to [GetDBConnection\(\)](#) and the private access modifier.

#### 5.15.2.2 DatabaseConnection() [2/2]

```
.DataAccessLayer.DBConnection.DatabaseConnection.DatabaseConnection (
    ITourPlannerConfig config ) [private]
```

Creates a [DatabaseConnection](#) instance. Is only called once due to [GetDBConnection\(ITourPlannerConfig testConfig\)](#) and the private access modifier.

#### Parameters

|               |   |
|---------------|---|
| <i>config</i> | Tourplanner configuration used for establishing database connection |
|---------------|---|

### 5.15.3 Member Function Documentation

#### 5.15.3.1 CloseConnection()

```
void DataAccessLayer.DBConnection.DatabaseConnection.CloseConnection ( )
```

Closes the database connection

Implements [DataAccessLayer.DBConnection.IDBConnection](#).

#### 5.15.3.2 DeclareParameter()

```
int DataAccessLayer.DBConnection.DatabaseConnection.DeclareParameter (
    IDbCommand command,
    string name,
    System.Data.DbType type )
```

Declares a parameter in a command

##### Parameters

|                |   |
|----------------|---|
| <i>command</i> | Command where the parameter gets declared |
| <i>name</i>    | Name of the parameter                     |
| <i>type</i>    | Type of the parameter                     |

##### Returns

Parameter index

##### Exceptions

|                              |  |
|------------------------------|--|
| <i>DALParameterException</i> | Thrown, when parameter can't be declared |
|------------------------------|--|

#### 5.15.3.3 DefineParameter()

```
void DataAccessLayer.DBConnection.DatabaseConnection.DefineParameter (
    IDbCommand command,
    string name,
    System.Data.DbType type,
    object value )
```

Declares and defines a parameter in a command.

## Parameters

|                |  |
|----------------|--|
| <i>command</i> | Command where the parameter gets defined |
| <i>name</i>    | Name of the parameter                    |
| <i>type</i>    | Type of the parameter                    |
| <i>value</i>   | Value of the parameter                   |

## Exceptions

|                              |  |
|------------------------------|--|
| <i>DALParameterException</i> | Thrown, when declaration / setting of parameter can't be conducted |
|------------------------------|--|

#### 5.15.3.4 ExecuteStatement()

```
int DataAccessLayer.DBConnection.DatabaseConnection.ExecuteStatement (
    IDbCommand command )
```

Executes a DML statement and returns the amount of affected rows

## Parameters

|                |                              |
|----------------|------------------------------|
| <i>command</i> | DML statement to be executed |
|----------------|------------------------------|

## Returns

Amount of affected rows

## Exceptions

|                                 |   |
|---------------------------------|---|
| <i>DALDBConnectionException</i> | Thrown, when the command can't be executed properly |
|---------------------------------|---|

Implements [DataAccessLayer.DBConnection.IDBConnection](#).

#### 5.15.3.5 GetDBConnection() [1/2]

```
static IDbConnection DataAccessLayer.DBConnection.DatabaseConnection.GetDBConnection ( ) [static]
```

Ensures that the [DatabaseConnection](#) constructor only gets called once. Provides the [DatabaseConnection](#) instance.

## Returns

The only [DatabaseConnection](#) instance.

### 5.15.3.6 GetDBConnection() [2/2]

```
static IDbConnection DataAccessLayer.DBConnection.DatabaseConnection.GetDBConnection (
    ITourPlannerConfig testConfig ) [static]
```

Ensures that the [DatabaseConnection](#) constructor only gets called once. Used for testing purposes. The test configuration needs to be passed

#### Parameters

|                   |   |
|-------------------|---|
| <i>testConfig</i> | Tourplanner configuration used for establishing database connection. Most of the time just some test data |
|-------------------|---|

#### Returns

The only [DatabaseConnection](#) instance.

### 5.15.3.7 OpenConnection()

```
void DataAccessLayer.DBConnection.DatabaseConnection.OpenConnection ( )
```

Opens the database connection

#### Exceptions

|                                 |   |
|---------------------------------|---|
| <i>DALDBConnectionException</i> | Thrown, when connection to the database can't be opened |
|---------------------------------|---|

Implements [DataAccessLayer.DBConnection.IDbConnection](#).

### 5.15.3.8 QueryDatabase()

```
List<object[]> DataAccessLayer.DBConnection.DatabaseConnection.QueryDatabase (
    IDbCommand command )
```

Executes a query and retrieves the results

#### Parameters

|                |                      |
|----------------|----------------------|
| <i>command</i> | Query to be executed |
|----------------|----------------------|

#### Returns

Results of the query as a collection of object arrays

## Exceptions

|                                       |  |
|---------------------------------------|--|
| <code>DALDBConnectionException</code> | Thrown, when error occurs whilst querying the database |
|---------------------------------------|--|

Implements [DataAccessLayer.DBConnection.IDBConnection](#).

## 5.15.4 Member Data Documentation

### 5.15.4.1 config

`ITourPlannerConfig` `DataAccessLayer.DBConnection.DatabaseConnection.config` [private]

Tourplanner config object. Used to retrieve the newest config data.

### 5.15.4.2 dB

`DatabaseConnection` `DataAccessLayer.DBConnection.DatabaseConnection.dB` [static], [private]

Only instance of the [DatabaseConnection](#) class. Gets set once.

### 5.15.4.3 logger

`log4net.ILog` `DataAccessLayer.DBConnection.DatabaseConnection.logger` [private]

ILog object used for logging.

### 5.15.4.4 npgsqlConnection

`Npgsql.NpgsqlConnection` `DataAccessLayer.DBConnection.DatabaseConnection.npgsqlConnection` [private]

Connection to the Postgres database

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/DatabaseConnection.cs



## 5.16 SWE2\_Tourplanner\_Tests.DALTests.DBConnectionTests Class Reference

### Public Member Functions

- void **Setup** ()
- void **ExecuteStatementCorrectCallsMock** ()
- void **QueryDatabaseCorrectCallsMock** ()
- void **DefineParameterArgumentExceptionThrownTest** ()
- void **OpenConnectionPostgresExceptionThrownTest** ()

### Private Attributes

- [IDBConnection](#) **db**
- Mock< [IDbCommand](#) > **mockNpgsqlCommand**
- Mock< [IDataReader](#) > **mockNpgsqlDataReader**
- Mock< [ITourPlannerConfig](#) > **mockConfig**

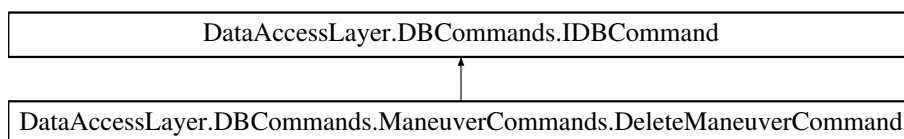
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/DALTests/DBConnectionTests.cs

## 5.17 DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand Class Reference

The [DeleteManeuverCommand](#) is used for deleting specific maneuvers.

Inheritance diagram for DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand:



### Public Member Functions

- [DeleteManeuverCommand](#) ([IDBConnection](#) db, [Maneuver](#) maneuver)  
*Creates the new [DeleteManeuverCommand](#) and sets its values.*
- int [Execute](#) ()  
*Removes a specific maneuver from the maneuver-table.*
- int [Undo](#) ()  
*Inserts the previously deleted maneuver into the database again.*

## Private Attributes

- [IDBConnection db](#)  
*The connection to the database.*
- [Maneuver maneuver](#)  
*The maneuver to be deleted/inserted again.*
- [log4net.ILog logger](#)  
*ILog instance used for logging errors, warnings etc.*

### 5.17.1 Detailed Description

The [DeleteManeuverCommand](#) is used for deleting specific maneuvers.

### 5.17.2 Constructor & Destructor Documentation

#### 5.17.2.1 DeleteManeuverCommand()

```
.DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.DeleteManeuverCommand (
    IDBConnection db,
    Maneuver maneuver )
```

Creates the new [DeleteManeuverCommand](#) and sets its values.

##### Parameters

|                 |                            |
|-----------------|----------------------------|
| <i>db</i>       | Database connection.       |
| <i>maneuver</i> | The maneuver to be deleted |

### 5.17.3 Member Function Documentation

#### 5.17.3.1 Execute()

```
int DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.Execute ( )
```

Removes a specific maneuver from the maneuver-table.

##### Returns

Amount of rows affected by the delete-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.17.3.2 Undo()

```
int DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.Undo ( )
```

Inserts the previously deleted maneuver into the database again.

#### Returns

Amount of rows affected by the insert of the previously deleted data. Normally expected value: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

## 5.17.4 Member Data Documentation

### 5.17.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.db [private]
```

The connection to the database.

### 5.17.4.2 logger

```
log4net.ILog DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.logger [private]
```

ILog instance used for logging errors, warnings etc.

### 5.17.4.3 maneuver

```
Maneuver DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand.maneuver [private]
```

The maneuver to be deleted/inserted again.

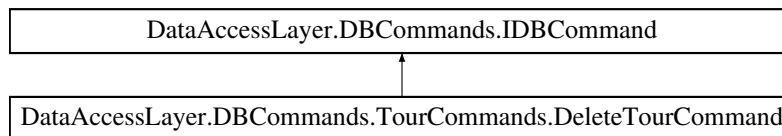
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/ManeuverCommands/Delete↵  
ManeuverCommand.cs

## 5.18 DataAccessLayer.DBCommands.TourCommands.DeleteTour↔ Command Class Reference

The [DeleteTourCommand](#) is used for deleting tours.

Inheritance diagram for DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand:



### Public Member Functions

- [DeleteTourCommand](#) ([IDBConnection](#) db, [Tour](#) tour)  
*Creates the [DeleteTourCommand](#) instance.*
- int [Execute](#) ()  
*Deletes the specified tour.*
- int [Undo](#) ()  
*Inserts the previously deleted tour again (and the previously associated data).*

### Private Attributes

- [IDBConnection](#) db  
*Connection to the database*
- [Tour](#) tour  
*Tour to be deleted/inserted again.*

#### 5.18.1 Detailed Description

The [DeleteTourCommand](#) is used for deleting tours.

#### 5.18.2 Constructor & Destructor Documentation

##### 5.18.2.1 DeleteTourCommand()

```

DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand.DeleteTourCommand (
    IDBConnection db,
    Tour tour )
  
```

Creates the [DeleteTourCommand](#) instance.

## Parameters

|             |                            |
|-------------|----------------------------|
| <i>db</i>   | Connection to the database |
| <i>tour</i> | Tour to be deleted         |

### 5.18.3 Member Function Documentation

#### 5.18.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand.Execute ( )
```

Deletes the specified tour.

## Returns

Amount of rows affected by the delete-statement. Expected: 1 + amount of associated maneuvers + amount of associated tourlogs

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

#### 5.18.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand.Undo ( )
```

Inserts the previously deleted tour again (and the previously associated data).

## Returns

Amount of rows affected by the insert-statement(s). Expected: 1 + amount of associated maneuvers + amount of associated tourlogs

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.18.4 Member Data Documentation

#### 5.18.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand.db [private]
```

Connection to the database

#### 5.18.4.2 tour

[Tour](#) `DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand.tour` [private]

Tour to be deleted/inserted again.

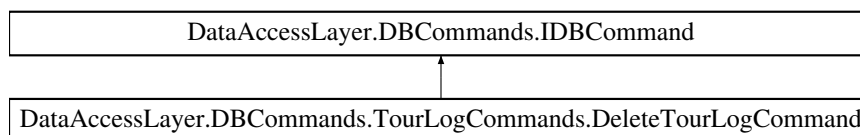
The documentation for this class was generated from the following file:

- `SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourCommands/DeleteTourCommand.cs`

### 5.19 DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand Class Reference

[DeleteTourLogCommand](#) is used for creating new logs of tours.

Inheritance diagram for `DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand`:



#### Public Member Functions

- [DeleteTourLogCommand](#) (`IDBConnection db`, `TourLog tourLog`)  
*Creates the [DeleteTourLogCommand](#) instance.*
- `int` [Execute](#) ()  
*Deletes the log of a tour from the tourlog table.*
- `int` [Undo](#) ()  
*Inserts the previously deleted log back into the tourlog table.*

#### Private Attributes

- [IDBConnection db](#)  
*Connection to the database.*
- [TourLog tourLog](#)  
*Log of a tour to be deleted.*
- `log4net.ILog` [logger](#)  
*ILog instance used for logging errors, warnings etc.*

#### 5.19.1 Detailed Description

[DeleteTourLogCommand](#) is used for creating new logs of tours.

## 5.19.2 Constructor & Destructor Documentation

### 5.19.2.1 DeleteTourLogCommand()

```
DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.DeleteTourLogCommand (
    IDBConnection db,
    TourLog tourLog )
```

Creates the [DeleteTourLogCommand](#) instance.

#### Parameters

|                |                            |
|----------------|----------------------------|
| <i>db</i>      | Connection to the database |
| <i>tourLog</i> | Log to be deleted.         |

## 5.19.3 Member Function Documentation

### 5.19.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.Execute ( )
```

Deletes the log of a tour from the tourlog table.

#### Returns

Amount of rows affected by the delete-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.19.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.Undo ( )
```

Inserts the previously deleted log back into the tourlog table.

#### Returns

Amount of rows affected by the insert-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

## 5.19.4 Member Data Documentation

### 5.19.4.1 db

`IDBConnection` `DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.db` [private]

Connection to the database.

### 5.19.4.2 logger

`log4net.ILog` `DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.logger` [private]

ILog instance used for logging errors, warnings etc.

### 5.19.4.3 tourLog

`TourLog` `DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand.tourLog` [private]

Log of a tour to be deleted.

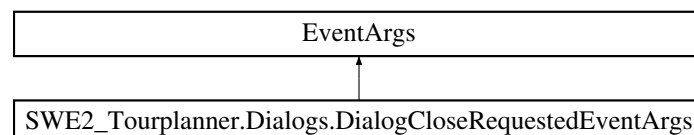
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourLogCommands/DeleteTourLogCommand.cs

## 5.20 SWE2\_Tourplanner.Dialogs.DialogCloseRequestedEventArgs Class Reference

`DialogCloseRequestedEventArgs` is an `EventArgs` used for abstracting / closing `Dialogs`

Inheritance diagram for `SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs`:



### Public Member Functions

- `DialogCloseRequestedEventArgs` (bool? dialogResult)  
Default constructor of `DialogCloseRequestedEventArgs`



## Properties

- bool? [DialogResult](#) [get]  
Result of the dialog. Can be null if used just closes the dialog

### 5.20.1 Detailed Description

[DialogCloseRequestedEventArgs](#) is an EventArgs used for abstracting / closing [Dialogs](#)

### 5.20.2 Constructor & Destructor Documentation

#### 5.20.2.1 DialogCloseRequestedEventArgs()

```
SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs.DialogCloseRequestedEventArgs (
    bool? dialogResult )
```

Default constructor of [DialogCloseRequestedEventArgs](#)

#### Parameters

|                     |  |
|---------------------|--|
| <i>dialogResult</i> | Result of the dialog: true if user clicks OK-, commit-buttons etc., false if user exits the dialog |
|---------------------|--|

### 5.20.3 Property Documentation

#### 5.20.3.1 DialogResult

```
bool? SWE2_Tourplanner.Dialogs.DialogCloseRequestedEventArgs.DialogResult [get]
```

Result of the dialog. Can be null if used just closes the dialog

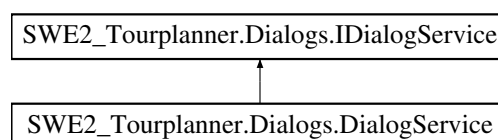
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Dialogs/DialogCloseRequestedEventArgs.cs

## 5.21 SWE2\_Tourplanner.Dialogs.DialogService Class Reference

Concrete implementation of [IDialogService](#)

Inheritance diagram for SWE2\_Tourplanner.Dialogs.DialogService:



## Public Member Functions

- [DialogService](#) (Window [owner](#))  
*Default constructor of [DialogService](#)*
- void [Register](#)< TViewModel, TView > ()  
*Registers a viewmodel together with a view*
- bool? [ShowDialog](#)< TViewModel > (TViewModel viewModel)  
*Opens a dialog view for the registered ViewModel*

## Properties

- IDictionary< Type, Type > [Mappings](#) [get]  
*Registered View, ViewModel mappings*

## Private Attributes

- readonly Window [owner](#)  
*Owner of [DialogService](#)*

### 5.21.1 Detailed Description

Concrete implementation of [IDialogService](#)

### 5.21.2 Constructor & Destructor Documentation

#### 5.21.2.1 DialogService()

```
SWE2_Tourplanner.Dialogs.DialogService.DialogService (  
    Window owner )
```

Default constructor of [DialogService](#)

#### Parameters

|                       |   |
|-----------------------|---|
| <a href="#">owner</a> | Owner of the <a href="#">DialogService</a> instance |
|-----------------------|---|

### 5.21.3 Member Function Documentation

### 5.21.3.1 Register< TViewModel, TView >()

```
void SWE2_Tourplanner.Dialogs.DialogService.Register< TViewModel, TView > ( )
```

Registers a viewmodel together with a view

#### Template Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>TViewModel</i> | ViewModel type to be registered |
| <i>TView</i>      | View to be registered           |

#### Exceptions

|                          |  |
|--------------------------|--|
| <i>ArgumentException</i> | Thrown, when ViewModel is already registered |
|--------------------------|--|

Implements [SWE2\\_Tourplanner.Dialogs.IDialogService](#).

#### Type Constraints

***TViewModel* : *IDialogRequestClose***

***TView* : *IDialog***

### 5.21.3.2 ShowDialog< TViewModel >()

```
bool? SWE2_Tourplanner.Dialogs.DialogService.ShowDialog< TViewModel > (
    TViewModel viewModel )
```

Opens a dialog view for the registered ViewModel

#### Template Parameters

|                   |                       |
|-------------------|-----------------------|
| <i>TViewModel</i> | Type of the ViewModel |
|-------------------|-----------------------|

#### Parameters

|                  |  |
|------------------|--|
| <i>viewModel</i> | ViewModel that needs the registered View to open as a Dialog |
|------------------|--|

#### Returns

Result of the dialog. Can be null if the dialog gets closed

Implements [SWE2\\_Tourplanner.Dialogs.IDialogService](#).

#### Type Constraints

***TViewModel* : *IDialogRequestClose***

## 5.21.4 Member Data Documentation

### 5.21.4.1 owner

```
readonly Window SWE2_Tourplanner.Dialogs.DialogService.owner [private]
```

Owner of [DialogService](#)

## 5.21.5 Property Documentation

### 5.21.5.1 Mappings

```
IDictionary<Type, Type> SWE2_Tourplanner.Dialogs.DialogService.Mappings [get]
```

Registered View, ViewModel mappings

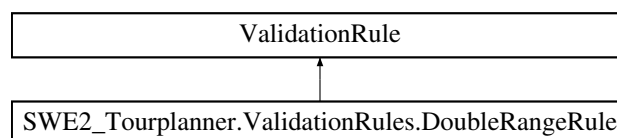
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Dialogs/DialogService.cs

## 5.22 SWE2\_Tourplanner.ValidationRules.DoubleRangeRule Class Reference

[DoubleRangeRule](#) is a ValidationRule for validating whether a string converted to a double value is within a Min and Max Value

Inheritance diagram for SWE2\_Tourplanner.ValidationRules.DoubleRangeRule:



## Public Member Functions

- override ValidationResult [Validate](#) (object value, CultureInfo cultureInfo)  
*Validates if the input value is between Min and Max*

## Properties

- double [Min](#) [get, set]
- double [Max](#) [get, set]

### 5.22.1 Detailed Description

[DoubleRangeRule](#) is a ValidationRule for validating whether a string converted to a double value is within a Min and Max Value

### 5.22.2 Member Function Documentation

#### 5.22.2.1 Validate()

```
override ValidationResult SWE2_Tourplanner.ValidationRules.DoubleRangeRule.Validate (
    object value,
    CultureInfo cultureInfo )
```

Validates if the input value is between Min and Max

##### Parameters

|                    |  |
|--------------------|--|
| <i>value</i>       | Input value                              |
| <i>cultureInfo</i> | CultureInfo used for convertig the value |

##### Returns

ValidationResult. Returns valid result if value is within bounds, invalid result if value is out of bounds

### 5.22.3 Property Documentation

#### 5.22.3.1 Max

```
double SWE2_Tourplanner.ValidationRules.DoubleRangeRule.Max [get], [set]
```

Upper Bound of the validation rule

#### 5.22.3.2 Min

```
double SWE2_Tourplanner.ValidationRules.DoubleRangeRule.Min [get], [set]
```

Lower Bound of the validation rule

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ValidationRules/DoubleRangeRule.cs



## Private Attributes

- `bool _contentLoaded`

### 5.23.1 Detailed Description

Interaction logic for [ErrorView](#)

[ErrorView](#)

### 5.23.2 Member Function Documentation

#### 5.23.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.ErrorView.InitializeComponent ( )
```

InitializeComponent

#### 5.23.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.ErrorView.InitializeComponent ( )
```

InitializeComponent

#### 5.23.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.ErrorView.InitializeComponent ( )
```

InitializeComponent

#### 5.23.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.ErrorView.InitializeComponent ( )
```

InitializeComponent

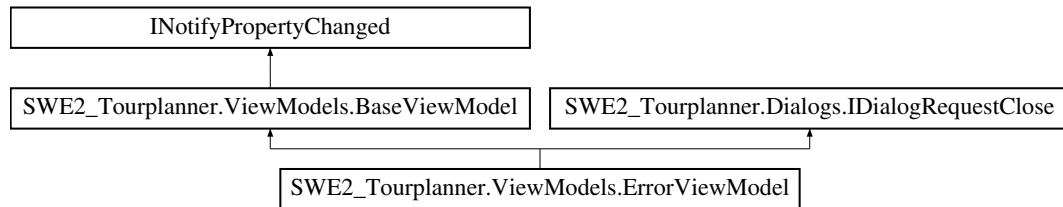
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ErrorView.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ErrorView.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ErrorView.g.i.cs

## 5.24 SWE2\_Tourplanner.ViewModels.ErrorViewModel Class Reference

[ErrorViewModel](#) is used for informing the user about exceptions / errors that occurred during execution of other methods

Inheritance diagram for SWE2\_Tourplanner.ViewModels.ErrorViewModel:



### Public Member Functions

- [ErrorViewModel](#) (string [errorHandlingMessage](#), string [errorType](#))

*Default constructor of [ErrorViewModel](#)*

### Properties

- ICommand [CloseCommand](#) [get]
- string [ErrorHandlingMessage](#) [get]
- string [ErrorType](#) [get]

### Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)

*Event used for closing the dialog*

### Private Attributes

- string [errorHandlingMessage](#)  
*Exception / Error message*
- string [errorType](#)  
*Type of error / exception*

### Additional Inherited Members

#### 5.24.1 Detailed Description

[ErrorViewModel](#) is used for informing the user about exceptions / errors that occurred during execution of other methods



## 5.24.2 Constructor & Destructor Documentation

### 5.24.2.1 ErrorViewModel()

```
SWE2_Tourplanner.ViewModels.ErrorViewModel.ErrorViewModel (
    string errorHandlingMessage,
    string errorType )
```

Default constructor of [ErrorViewModel](#)

## Parameters

|                             |               |
|-----------------------------|---------------|
| <i>errorHandlingMessage</i> | Error message |
| <i>errorType</i>            | Error type    |

## 5.24.3 Member Data Documentation

### 5.24.3.1 `errorHandlingMessage`

```
string SWE2_Tourplanner.ViewModels.ErrorViewModel.errorHandlingMessage [private]
```

Exception / Error message

### 5.24.3.2 `errorType`

```
string SWE2_Tourplanner.ViewModels.ErrorViewModel.errorType [private]
```

Type of error / exception

## 5.24.4 Property Documentation

### 5.24.4.1 `CloseCommand`

```
 ICommand SWE2_Tourplanner.ViewModels.ErrorViewModel.CloseCommand [get]
```

Command used for closing the dialog

### 5.24.4.2 `ErrorHandlingMessage`

```
string SWE2_Tourplanner.ViewModels.ErrorViewModel.ErrorHandlingMessage [get]
```

Exception / Error message

### 5.24.4.3 `ErrorType`

```
string SWE2_Tourplanner.ViewModels.ErrorViewModel.ErrorType [get]
```

Exception / Error type

### 5.24.5 Event Documentation

#### 5.24.5.1 CloseRequested

EventHandler<DialogCloseRequestedEventArgs> SWE2\_Tourplanner.ViewModels.ErrorViewModel.CloseRequested

Event used for closing the dialog

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/ErrorViewModel.cs

## 5.25 SWE2\_Tourplanner\_Tests.CommonTests.ExtensionMethodsTest Class Reference

### Public Member Functions

- void **CIContainsTest** ()

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/CommonTests/ExtensionMethodsTest.cs

## 5.26 Common.Extensions Class Reference

Class [Extensions](#) is used for providing new extension methods to the application.

### Static Public Member Functions

- static bool [CIContains](#) (this string text, string value, StringComparison stringComparison=StringComparison.CurrentCultureIgnoreCase)  
*CIContains checks if a string contains another string (case insensitive).*

#### 5.26.1 Detailed Description

Class [Extensions](#) is used for providing new extension methods to the application.

#### 5.26.2 Member Function Documentation

##### 5.26.2.1 CIContains()

```
static bool Common.Extensions.CIContains (
    this string text,
    string value,
    StringComparison stringComparison = StringComparison.CurrentCultureIgnoreCase )
[static]
```

CIContains checks if a string contains another string (case insensitive).

## Parameters

|                         |  |
|-------------------------|--|
| <i>text</i>             | Some string that may contain parameter value           |
| <i>value</i>            | String that may be part of parameter text              |
| <i>stringComparison</i> | String comparison rules. In this case case is ignored. |

## Returns

True if value is contained in text, false if not

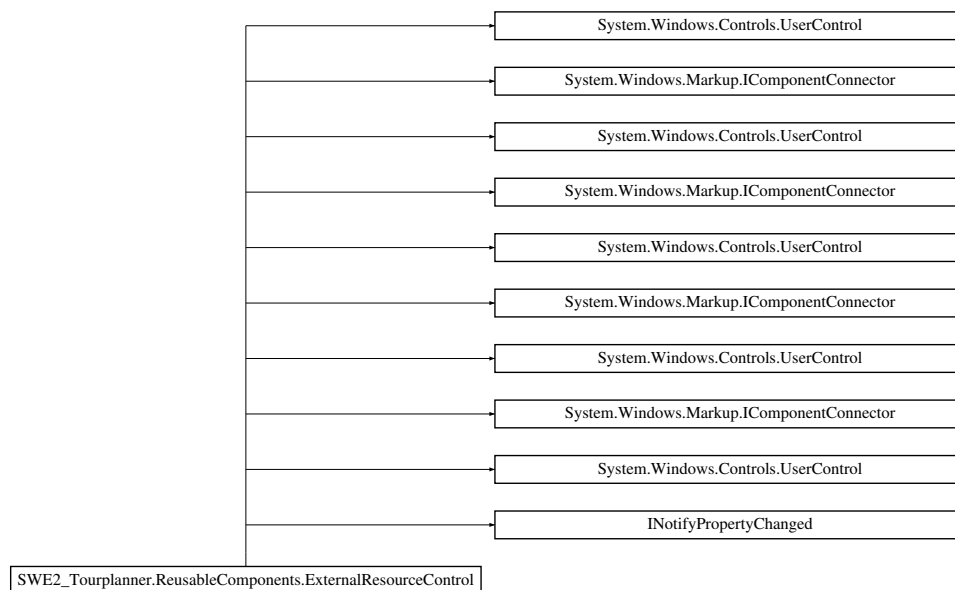
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/ExtensionMethods.cs

## 5.27 SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl Class Reference

### ExternalResourceControl

Inheritance diagram for SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- [ExternalResourceControl](#) ()  
*Default constructor of [ExternalResourceControl](#). Initializes the component*

## Static Public Attributes

- static readonly DependencyProperty [LinkTextProperty](#)  
*DependencyProperty as the backing store for LinkText. Enables Binding*
- static readonly DependencyProperty [LinkProperty](#)  
*DependencyProperty as the backing store of Link. Enables Binding*

## Protected Member Functions

- virtual void [OnPropertyChanged](#) ([CallerMemberName] string propertyName=null)  
*OnPropertyChanged notifies the UI aspect of the UserControl about changes of a specific property*

## Properties

- string [LinkText](#) [get, set]
- string [Link](#) [get, set]
- object [Favicon](#) [get, set]

## Events

- PropertyChangedEventHandler [PropertyChanged](#)  
*EventHandler used for notifying the UI aspect of the UserControl about data changes*

## Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void [ExternalResource\\_MouseDown](#) (object sender, MouseButtonEventArgs e)  
*Click event of [ExternalResourceControl](#). Opens Link in the default browser.*

## Static Private Member Functions

- static void [OnLinkChanged](#) (DependencyObject d, DependencyPropertyChangedEventArgs e)  
*Callback method for the DependencyProperty LinkProperty. Updates the Favicon value according to the changes in the Binding*

## Private Attributes

- bool [\\_contentLoaded](#)
- string [faviconLocation](#)  
*Location of the favicon. Is a URL based on the Google Favicon API / Crawler URL*

## Static Private Attributes

- static readonly string [googleFaviconCrawler](#) = @"https://www.google.com/s2/favicons?domain"  
*Base URL of the Google Favicon API / Crawler*

### 5.27.1 Detailed Description

[ExternalResourceControl](#)

[ExternalResourceControl](#) is UserControl used for linking external resources like websites

### 5.27.2 Constructor & Destructor Documentation

#### 5.27.2.1 ExternalResourceControl()

```
SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.ExternalResourceControl ( )
```

Default constructor of [ExternalResourceControl](#). Initializes the component

### 5.27.3 Member Function Documentation

#### 5.27.3.1 ExternalResource\_MouseDown()

```
void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.ExternalResource_MouseDown (
    object sender,
    MouseButtonEventArgs e ) [private]
```

Click event of [ExternalResourceControl](#). Opens Link in the default browser.

##### Parameters

|               |  |
|---------------|--|
| <i>sender</i> | Sender object                                  |
| <i>e</i>      | MouseButtonEventArgs associated with the click |

#### 5.27.3.2 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.InitializeComponent ( )
```

InitializeComponent

### 5.27.3.3 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.InitializeComponent ( )
```

InitializeComponent

### 5.27.3.4 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.InitializeComponent ( )
```

InitializeComponent

### 5.27.3.5 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.InitializeComponent ( )
```

InitializeComponent

### 5.27.3.6 OnLinkChanged()

```
static void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.OnLinkChanged (
    DependencyObject d,
    DependencyPropertyChangedEventArgs e ) [static], [private]
```

Callback method for the DependencyProperty LinkProperty. Updates the Favicon value according to the changes in the Binding

#### Parameters

|          |   |
|----------|---|
| <i>d</i> | DependencyObject that calls the callback method             |
| <i>e</i> | DependencyPropertyChangedEventArgs associated with the call |

### 5.27.3.7 OnPropertyChanged()

```
virtual void SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.OnPropertyChanged (
    [CallerMemberName] string propertyName = null ) [protected], [virtual]
```

OnPropertyChanged notifies the UI aspect of the UserControl about changes of a specific property

## Parameters

|                     |                              |
|---------------------|------------------------------|
| <i>propertyName</i> | Name of the changed property |
|---------------------|------------------------------|

## 5.27.4 Member Data Documentation

### 5.27.4.1 faviconLocation

```
string SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.faviconLocation [private]
```

Location of the favicon. Is a URL based on the Google Favicon API / Crawler URL

### 5.27.4.2 googleFaviconCrawler

```
readonly string SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.googleFavicon↵
Crawler = @"https://www.google.com/s2/favicons?domain" [static], [private]
```

Base URL of the Google Favicon API / Crawler

### 5.27.4.3 LinkProperty

```
readonly DependencyProperty SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.Link↵
Property [static]
```

#### Initial value:

```
=
    DependencyProperty.Register("Link", typeof(string), typeof(ExternalResourceControl), new
    PropertyMetadata("", new PropertyChangedCallback(OnLinkChanged)))
```

DependencyProperty as the backing store of Link. Enables Binding

### 5.27.4.4 LinkTextProperty

```
readonly DependencyProperty SWE2_Tourplanner.ReusableComponents.ExternalResourceControl.Link↵
TextProperty [static]
```

#### Initial value:

```
=
    DependencyProperty.Register("LinkText", typeof(string), typeof(ExternalResourceControl), new
    PropertyMetadata(""))
```

DependencyProperty as the backing store for LinkText. Enables Binding



## 5.27.5 Property Documentation

### 5.27.5.1 Favicon

object SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl.Favicon [get], [set]

Location of the favicon. Is a URL based on the Google Favicon API / Crawler URL. Can return DependencyProperty.UnsetValue to avoid Binding Errors

### 5.27.5.2 Link

string SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl.Link [get], [set]

Link is the URL of the external resource

### 5.27.5.3 LinkText

string SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl.LinkText [get], [set]

Text that is displayed instead of the Link itself

## 5.27.6 Event Documentation

### 5.27.6.1 PropertyChanged

PropertyChangedEventHandler SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl.PropertyChanged

EventHandler used for notifying the UI aspect of the UserControl about data changes

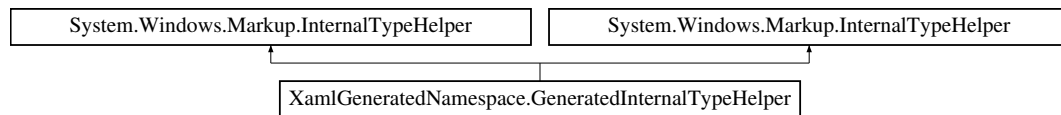
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ReusableComponents/ExternalResourceControl.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ReusableComponents/ExternalResourceControl.g.i.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ReusableComponents/ExternalResourceControl.xaml.cs

## 5.28.XamlGeneratedNamespace.GeneratedInternalTypeHelper Class Reference

### GeneratedInternalTypeHelper

Inheritance diagram for XamlGeneratedNamespace.GeneratedInternalTypeHelper:



### Protected Member Functions

- override object [CreateInstance](#) (System.Type type, System.Globalization.CultureInfo culture)  
*CreateInstance*
- override object [GetPropertyValue](#) (System.Reflection.PropertyInfo propertyInfo, object target, System.Globalization.CultureInfo culture)  
*GetPropertyValue*
- override void [SetPropertyValue](#) (System.Reflection.PropertyInfo propertyInfo, object target, object value, System.Globalization.CultureInfo culture)  
*SetPropertyValue*
- override System.Delegate [CreateDelegate](#) (System.Type delegateType, object target, string handler)  
*CreateDelegate*
- override void [AddEventHandler](#) (System.Reflection.EventInfo eventInfo, object target, System.Delegate handler)  
*AddEventHandler*
- override object [CreateInstance](#) (System.Type type, System.Globalization.CultureInfo culture)  
*CreateInstance*
- override object [GetPropertyValue](#) (System.Reflection.PropertyInfo propertyInfo, object target, System.Globalization.CultureInfo culture)  
*GetPropertyValue*
- override void [SetPropertyValue](#) (System.Reflection.PropertyInfo propertyInfo, object target, object value, System.Globalization.CultureInfo culture)  
*SetPropertyValue*
- override System.Delegate [CreateDelegate](#) (System.Type delegateType, object target, string handler)  
*CreateDelegate*
- override void [AddEventHandler](#) (System.Reflection.EventInfo eventInfo, object target, System.Delegate handler)  
*AddEventHandler*

### 5.28.1 Detailed Description

### GeneratedInternalTypeHelper

### 5.28.2 Member Function Documentation

### 5.28.2.1 AddEventHandler() [1/2]

```
override void XamlGeneratedNamespace.GeneratedInternalTypeHelper.AddEventHandler (
    System.Reflection.EventInfo eventInfo,
    object target,
    System.Delegate handler ) [protected]
```

AddEventHandler

### 5.28.2.2 AddEventHandler() [2/2]

```
override void XamlGeneratedNamespace.GeneratedInternalTypeHelper.AddEventHandler (
    System.Reflection.EventInfo eventInfo,
    object target,
    System.Delegate handler ) [protected]
```

AddEventHandler

### 5.28.2.3 CreateDelegate() [1/2]

```
override System.Delegate XamlGeneratedNamespace.GeneratedInternalTypeHelper.CreateDelegate (
    System.Type delegateType,
    object target,
    string handler ) [protected]
```

CreateDelegate

### 5.28.2.4 CreateDelegate() [2/2]

```
override System.Delegate XamlGeneratedNamespace.GeneratedInternalTypeHelper.CreateDelegate (
    System.Type delegateType,
    object target,
    string handler ) [protected]
```

CreateDelegate

### 5.28.2.5 CreateInstance() [1/2]

```
override object XamlGeneratedNamespace.GeneratedInternalTypeHelper.CreateInstance (
    System.Type type,
    System.Globalization.CultureInfo culture ) [protected]
```

CreateInstance

**5.28.2.6 CreateInstance() [2/2]**

```
override object XamlGeneratedNamespace.GeneratedInternalTypeHelper.CreateInstance (
    System.Type type,
    System.Globalization.CultureInfo culture ) [protected]
```

CreateInstance

**5.28.2.7 GetPropertyValue() [1/2]**

```
override object XamlGeneratedNamespace.GeneratedInternalTypeHelper.GetPropertyValue (
    System.Reflection.PropertyInfo propertyInfo,
    object target,
    System.Globalization.CultureInfo culture ) [protected]
```

GetPropertyValue

**5.28.2.8 GetPropertyValue() [2/2]**

```
override object XamlGeneratedNamespace.GeneratedInternalTypeHelper.GetPropertyValue (
    System.Reflection.PropertyInfo propertyInfo,
    object target,
    System.Globalization.CultureInfo culture ) [protected]
```

GetPropertyValue

**5.28.2.9 SetPropertyValue() [1/2]**

```
override void XamlGeneratedNamespace.GeneratedInternalTypeHelper.SetPropertyValue (
    System.Reflection.PropertyInfo propertyInfo,
    object target,
    object value,
    System.Globalization.CultureInfo culture ) [protected]
```

SetPropertyValue

**5.28.2.10 SetPropertyValue() [2/2]**

```
override void XamlGeneratedNamespace.GeneratedInternalTypeHelper.SetPropertyValue (
    System.Reflection.PropertyInfo propertyInfo,
    object target,
    object value,
    System.Globalization.CultureInfo culture ) [protected]
```

SetPropertyValue

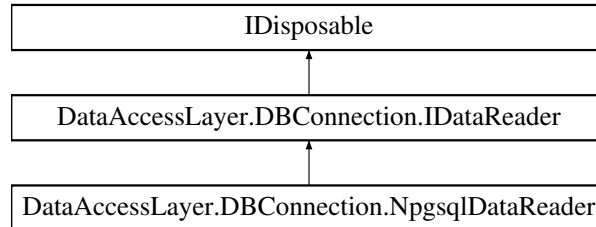
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/GeneratedInternal↔  
TypeHelper.g.i.cs

## 5.29 DataAccessLayer.DBConnection.IDataReader Interface Reference

Interface used for exposing only necessary methods / attributes of DataReaders to the application

Inheritance diagram for DataAccessLayer.DBConnection.IDataReader:



### Public Member Functions

- bool [Read](#) ()  
*[Read\(\)](#) checks if rows are still left or not.*
- object [GetValue](#) (int i)  
*Returns the value of a specific field.*

### Properties

- int [FieldCount](#) [get]

#### 5.29.1 Detailed Description

Interface used for exposing only necessary methods / attributes of DataReaders to the application

#### 5.29.2 Member Function Documentation

##### 5.29.2.1 GetValue()

```
object DataAccessLayer.DBConnection.IDataReader.GetValue (
    int i )
```

Returns the value of a specific field.

##### Parameters

|          |                         |
|----------|-------------------------|
| <i>i</i> | Position / Field number |
|----------|-------------------------|

**Returns**

Value of the field i

Implemented in [DataAccessLayer.DBConnection.NpgsqlDataReader](#).

**5.29.2.2 Read()**

```
bool DataAccessLayer.DBConnection.IDataReader.Read ( )
```

[Read\(\)](#) checks if rows are still left or not.

**Returns**

True if rows are still left, false if no rows are left.

Implemented in [DataAccessLayer.DBConnection.NpgsqlDataReader](#).

**5.29.3 Property Documentation****5.29.3.1 FieldCount**

```
int DataAccessLayer.DBConnection.IDataReader.FieldCount [get]
```

Returns the amount of fields of a row.

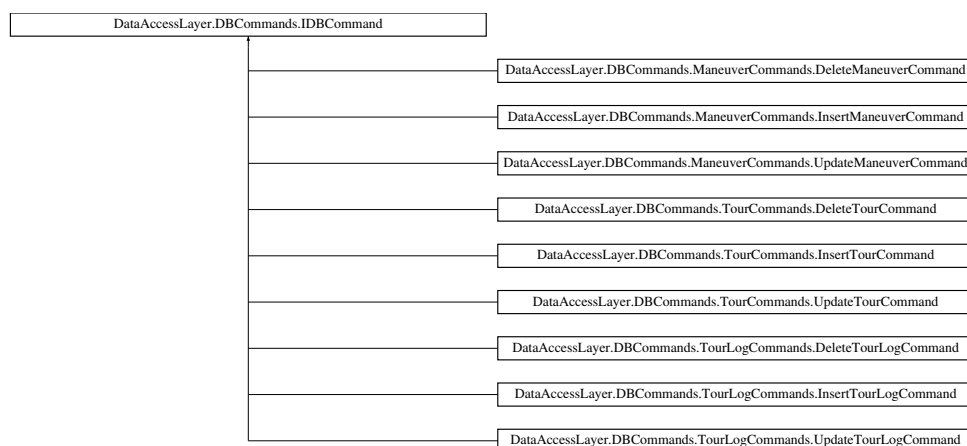
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/IDataReader.cs

**5.30 DataAccessLayer.DBCommands.IDBCommand Interface Reference**

The [IDBCommand](#) interfaces defines two functions - [Execute\(\)](#) and [Undo\(\)](#) - which are needed for commits and rollbacks. In a commit of a `IUnitOfWork`, the [Execute\(\)](#) function of every issued command is called. In a rollback, the [Undo\(\)](#) function is executed. Is the interface for the command pattern.

Inheritance diagram for `DataAccessLayer.DBCommands.IDBCommand`:



## Public Member Functions

- int [Execute](#) ()  
*In the [Execute\(\)](#) method, the data in the database is altered or new data gets inserted.*
- int [Undo](#) ()  
*The [Undo\(\)](#) function is basically the opposite of the [Execute\(\)](#) method functionality-wise.*

### 5.30.1 Detailed Description

The [IDBCommand](#) interfaces defines two functions - [Execute\(\)](#) and [Undo\(\)](#) - which are needed for commits and rollbacks. In a commit of a [IUnitOfWork](#), the [Execute\(\)](#) function of every issued command is called. In a rollback, the [Undo\(\)](#) function is executed. Is the interface for the command pattern.

### 5.30.2 Member Function Documentation

#### 5.30.2.1 [Execute\(\)](#)

```
int DataAccessLayer.DBCommands.IDBCommand.Execute ( )
```

In the [Execute\(\)](#) method, the data in the database is altered or new data gets inserted.

##### Returns

Amount of rows affected by an [IDBCommand](#) instance's [Execute\(\)](#) method.

Implemented in [DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand](#), [DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand](#), [DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.InsertTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand](#), [DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand](#), [DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand](#) and [DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand](#).

#### 5.30.2.2 [Undo\(\)](#)

```
int DataAccessLayer.DBCommands.IDBCommand.Undo ( )
```

The [Undo\(\)](#) function is basically the opposite of the [Execute\(\)](#) method functionality-wise.

##### Returns

Amount of rows affected by an [IDBCommand](#) instance's [Undo\(\)](#) method.

Implemented in [DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand](#), [DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand](#), [DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.InsertTourCommand](#), [DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand](#), [DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand](#), [DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand](#) and [DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand](#).

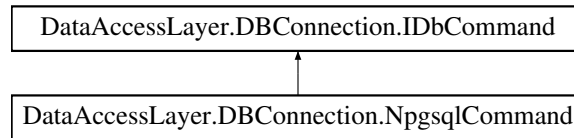
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/IDBCommand.cs

## 5.31 DataAccessLayer.DBConnection.IDbCommand Interface Reference

Interface used for exposing only necessary methods / properties to the application.

Inheritance diagram for DataAccessLayer.DBConnection.IDbCommand:



### Public Member Functions

- `int ExecuteNonQuery ()`  
*Executes the command (DML - INSERT, UPDATE, DELETE)*
- `IDataReader ExecuteReader ()`  
*Executes the command (query) and returns a reader for retrieving the results.*

### Properties

- `DbParameterCollection Parameters` [get]
- `DbConnection Connection` [get, set]

### 5.31.1 Detailed Description

Interface used for exposing only necessary methods / properties to the application.

### 5.31.2 Member Function Documentation

#### 5.31.2.1 ExecuteNonQuery()

```
int DataAccessLayer.DBConnection.IDbCommand.ExecuteNonQuery ( )
```

Executes the command (DML - INSERT, UPDATE, DELETE)

#### Returns

Amount of affected rows.

Implemented in [DataAccessLayer.DBConnection.NpgsqlCommand](#).



### 5.31.2.2 ExecuteReader()

`IDataReader` `DataAccessLayer.DBConnection.IDbCommand.ExecuteReader ( )`

Executes the command (query) and returns a reader for retrieving the results.

#### Returns

`DataReader` which can be used for getting the results of a query.

Implemented in [DataAccessLayer.DBConnection.NpgsqlCommand](#).

### 5.31.3 Property Documentation

#### 5.31.3.1 Connection

`DbConnection` `DataAccessLayer.DBConnection.IDbCommand.Connection [get], [set]`

The connection to the database, where the command gets executed.

#### 5.31.3.2 Parameters

`DbParameterCollection` `DataAccessLayer.DBConnection.IDbCommand.Parameters [get]`

Parameters of the command.

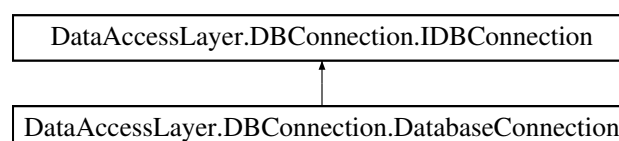
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/IDbCommand.cs

## 5.32 DataAccessLayer.DBConnection.IDBConnection Interface Reference

The `IDbConnection` interface defines several methods for setting parameters in `IDbCommand` instances, opening/closing the database connection and for querying / executing statements.

Inheritance diagram for `DataAccessLayer.DBConnection.IDBConnection`:



## Public Member Functions

- int [ExecuteStatement](#) ([IDbCommand](#) command)  
*Executes a statement (!=Query).*
- List< object[] > [QueryDatabase](#) ([IDbCommand](#) command)  
*Executes a query.*
- int [DeclareParameter](#) ([IDbCommand](#) command, string name, DbType type)  
*Declares a parameter of the command*
- void [DefineParameter](#) ([IDbCommand](#) command, string name, DbType type, object value)  
*Declares a parameter and defines it afterwards.*
- void [OpenConnection](#) ()  
*Opens the connection to the database.*
- void [CloseConnection](#) ()  
*Closes the connection to the database.*

### 5.32.1 Detailed Description

The IDbConnection interface defines several methods for setting parameters in IDbCommand instances, opening/closing the database connection and for querying / executing statements.

### 5.32.2 Member Function Documentation

#### 5.32.2.1 CloseConnection()

```
void DataAccessLayer.DBConnection.IDbConnection.CloseConnection ( )
```

Closes the connection to the database.

Implemented in [DataAccessLayer.DBConnection.DatabaseConnection](#).

#### 5.32.2.2 DeclareParameter()

```
int DataAccessLayer.DBConnection.IDbConnection.DeclareParameter (
    IDbCommand command,
    string name,
    DbType type )
```

Declares a parameter of the command

#### Parameters

|                |   |
|----------------|---|
| <i>command</i> | Command where the parameter is being set. |
| <i>name</i>    | Name of the parameter.                    |
| <i>type</i>    | Type of the new parameter.                |

Returns

### 5.32.2.3 DefineParameter()

```
void DataAccessLayer.DBConnection.IDBConnection.DefineParameter (
    IDbCommand command,
    string name,
    DbType type,
    object value )
```

Declares a parameter and defines it afterwards.

Parameters

|                |   |
|----------------|---|
| <i>command</i> | Command where the parameter is being set. |
| <i>name</i>    | Name of the parameter.                    |
| <i>type</i>    | Type of the parameter.                    |
| <i>value</i>   | Value to be inserted.                     |

### 5.32.2.4 ExecuteStatement()

```
int DataAccessLayer.DBConnection.IDBConnection.ExecuteStatement (
    IDbCommand command )
```

Executes a statement (!=Query).

Parameters

|                |                         |
|----------------|-------------------------|
| <i>command</i> | Command to be executed. |
|----------------|-------------------------|

Returns

Amount of rows affected by the statement.

Implemented in [DataAccessLayer.DBConnection.DatabaseConnection](#).

### 5.32.2.5 OpenConnection()

```
void DataAccessLayer.DBConnection.IDBConnection.OpenConnection ( )
```

Opens the connection to the database.

Implemented in [DataAccessLayer.DBConnection.DatabaseConnection](#).

### 5.32.2.6 QueryDatabase()

```
List<object [ ]> DataAccessLayer.DBConnection.IDBConnection.QueryDatabase (
    IDbCommand command )
```

Executes a query.

#### Parameters

|                |                       |
|----------------|-----------------------|
| <i>command</i> | Query to be executed. |
|----------------|-----------------------|

#### Returns

Collection of data(=rows) returned by the query.

Implemented in [DataAccessLayer.DBConnection.DatabaseConnection](#).

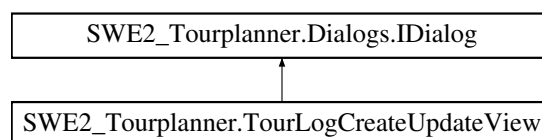
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/IDBConnection.cs

## 5.33 SWE2\_Tourplanner.Dialogs.IDialog Interface Reference

[IDialog](#) only defines necessary properties / methods for windows to be opened as dialogs.

Inheritance diagram for SWE2\_Tourplanner.Dialogs.IDialog:



### Public Member Functions

- void [Close](#) ()  
*Closes the dialog*
- bool? [ShowDialog](#) ()  
*Open a Dialog of the specific [IDialog](#) class and return result of Dialog*

### Properties

- object [DataContext](#) [get, set]
- bool? [DialogResult](#) [get, set]
- Window [Owner](#) [get, set]

### 5.33.1 Detailed Description

[IDialog](#) only defines necessary properties / methods for windows to be opened as dialogs.

### 5.33.2 Member Function Documentation

#### 5.33.2.1 Close()

```
void SWE2_Tourplanner.Dialogs.IDialog.Close ( )
```

Closes the dialog

#### 5.33.2.2 ShowDialog()

```
bool? SWE2_Tourplanner.Dialogs.IDialog.ShowDialog ( )
```

Open a Dialog of the specific [IDialog](#) class and return result of Dialog

##### Returns

Result of the dialog. True if successful, otherwise false. Can return null if Dialog just gets closed like any other window

### 5.33.3 Property Documentation

#### 5.33.3.1 DataContext

```
object SWE2_Tourplanner.Dialogs.IDialog.DataContext [get], [set]
```

DataContext of the dialog

#### 5.33.3.2 DialogResult

```
bool? SWE2_Tourplanner.Dialogs.IDialog.DialogResult [get], [set]
```

Result of the dialog

### 5.33.3.3 Owner

Window SWE2\_Tourplanner.Dialogs.IDialog.Owner [get], [set]

Owner of the dialog

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Dialogs/IDialog.cs

## 5.34 SWE2\_Tourplanner.Dialogs.IDialogRequestClose Interface Reference

[IDialogRequestClose](#) defines an event for implementing classes, to ensure that the associated dialog gets closed with the according result.

Inheritance diagram for SWE2\_Tourplanner.Dialogs.IDialogRequestClose:



### Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Close event of the dialog.*

### 5.34.1 Detailed Description

[IDialogRequestClose](#) defines an event for implementing classes, to ensure that the associated dialog gets closed with the according result.

### 5.34.2 Event Documentation

#### 5.34.2.1 CloseRequested

EventHandler<[DialogCloseRequestedEventArgs](#)> SWE2\_Tourplanner.Dialogs.IDialogRequestClose.  
[CloseRequested](#)

Close event of the dialog.

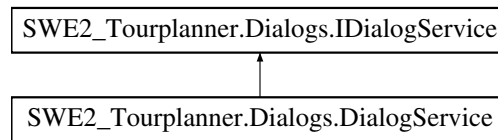
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Dialogs/IDialogRequestClose.cs

## 5.35 SWE2\_Tourplanner.Dialogs.IDialogService Interface Reference

[IDialogService](#) defines the necessary methods for managing dialogs and the associated [ViewModels](#)

Inheritance diagram for SWE2\_Tourplanner.Dialogs.IDialogService:



### Public Member Functions

- void [Register](#)< TViewModel, TView > ()  
*Registers a ViewModel and the associated View*
- bool? [ShowDialog](#)< TViewModel > (TViewModel viewModel)  
*Opens a dialog and returns the result of the dialog*

#### 5.35.1 Detailed Description

[IDialogService](#) defines the necessary methods for managing dialogs and the associated [ViewModels](#)

#### 5.35.2 Member Function Documentation

##### 5.35.2.1 Register< TViewModel, TView >()

```
void SWE2_Tourplanner.Dialogs.IDialogService.Register< TViewModel, TView > ( )
```

Registers a ViewModel and the associated View

Template Parameters

|                   |                       |
|-------------------|-----------------------|
| <i>TViewModel</i> | Type of the ViewModel |
| <i>TView</i>      | Type of the View      |

Implemented in [SWE2\\_Tourplanner.Dialogs.DialogService](#).

Type Constraints

***TViewModel : IDialogRequestClose***

***TView : IDialog***

### 5.35.2.2 ShowDialog< TViewModel >()

```
bool? SWE2_Tourplanner.Dialogs.IDialogService.ShowDialog< TViewModel > (
    TViewModel viewModel )
```

Opens a dialog and returns the result of the dialog

#### Template Parameters

|                   |                       |
|-------------------|-----------------------|
| <i>TViewModel</i> | Type of the ViewModel |
|-------------------|-----------------------|

#### Parameters

|                  |  |
|------------------|--|
| <i>viewModel</i> | ViewModel that needs the registered View to open |
|------------------|--|

#### Returns

Result of the dialog.

Implemented in [SWE2\\_Tourplanner.Dialogs.DialogService](#).

#### Type Constraints

***TViewModel : IDialogRequestClose***

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Dialogs/IDialogService.cs

## 5.36 BusinessLogicLayer.ExporterImporter.IExporterImporter< T > Interface Template Reference

A generic interface that defines basic methods for importing and exporting entities

### Public Member Functions

- Task< List< T > > [Import](#) (string filePath)  
*Method for importing data from JSON file*
- Task [Export](#) (List< T > entites)  
*Method for exporting data as JSON file*

### 5.36.1 Detailed Description

A generic interface that defines basic methods for importing and exporting entities



## Template Parameters

|          |  |
|----------|--|
| <i>T</i> | Type of data that gets imported / exported |
|----------|--|

## 5.36.2 Member Function Documentation

### 5.36.2.1 Export()

```
Task BusinessLogicLayer.ExporterImporter.IExporterImporter< T >.Export (
    List< T > entites )
```

Method for exporting data as JSON file

## Parameters

|                |                                 |
|----------------|---------------------------------|
| <i>entites</i> | Entities which will be exported |
|----------------|---------------------------------|

## Returns

Task, which exports entities as JSON

### 5.36.2.2 Import()

```
Task<List<T> > BusinessLogicLayer.ExporterImporter.IExporterImporter< T >.Import (
    string filePath )
```

Method for importing data from JSON file

## Parameters

|                 |                                     |
|-----------------|-------------------------------------|
| <i>filePath</i> | Path to the file that gets imported |
|-----------------|-------------------------------------|

## Returns

Task, that returns list of entities from file

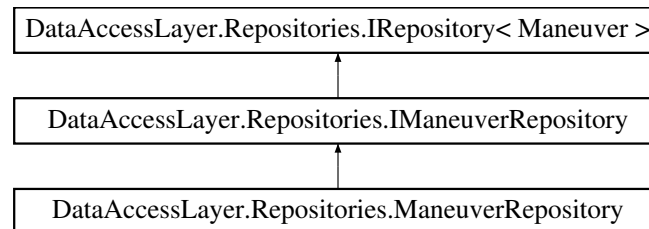
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/ExporterImporter/IExporterImporter.cs

## 5.37 DataAccessLayer.Repositories.IManeuverRepository Interface Reference

The [IManeuverRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Maneuver>](#)

Inheritance diagram for DataAccessLayer.Repositories.IManeuverRepository:



### Additional Inherited Members

#### 5.37.1 Detailed Description

The [IManeuverRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Maneuver>](#)

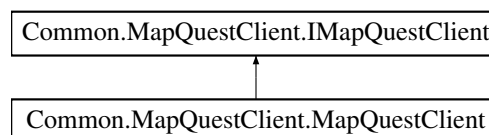
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/IManeuverRepository.cs

## 5.38 Common.MapQuestClient.IMapQuestClient Interface Reference

The [IMapQuestClient](#) interface defines two methods for retrieving data from the MapQuest API

Inheritance diagram for Common.MapQuestClient.IMapQuestClient:



### Public Member Functions

- Task< string > [RetrieveRouteImage](#) (string storeId, string sessionId, string boundingBox)  
*RetrieveRouteImage retrieves the associated map image of a previous MapQuest API request.*
- Task [GetRouteDataFromMapQuest](#) (Tour tour)  
*Retrieves data from the MapQuest API for a specific tour and sets the data in the tour.*

### 5.38.1 Detailed Description

The [IMapQuestClient](#) interface defines two methods for retrieving data from the MapQuest API

### 5.38.2 Member Function Documentation

#### 5.38.2.1 GetRouteDataFromMapQuest()

```
Task Common.MapQuestClient.IMapQuestClient.GetRouteDataFromMapQuest (
    Tour tour )
```

Retrieves data from the MapQuest API for a specific tour and sets the data in the tour.

##### Parameters

|             |  |
|-------------|--|
| <i>tour</i> | Tour that needs data from the MapQuest API |
|-------------|--|

##### Returns

Task, where important data for the tour is retrieved from the MapQuest API and inserted into the tour.

Implemented in [Common.MapQuestClient.MapQuestClient](#).

#### 5.38.2.2 RetrieveRouteImage()

```
Task<string> Common.MapQuestClient.IMapQuestClient.RetrieveRouteImage (
    string storeId,
    string sessionId,
    string boundingBox )
```

RetrieveRouteImage retrieves the associated map image of a previous MapQuest API request.

##### Parameters

|                    |   |
|--------------------|---|
| <i>storeId</i>     | Name of the route image file                          |
| <i>sessionId</i>   | SessionId of the previous request to the MapQuest API |
| <i>boundingBox</i> | BoundingBox of the map                                |

##### Returns

Task, where a route image is retrieved from MapQuest

Implemented in [Common.MapQuestClient.MapQuestClient](#).

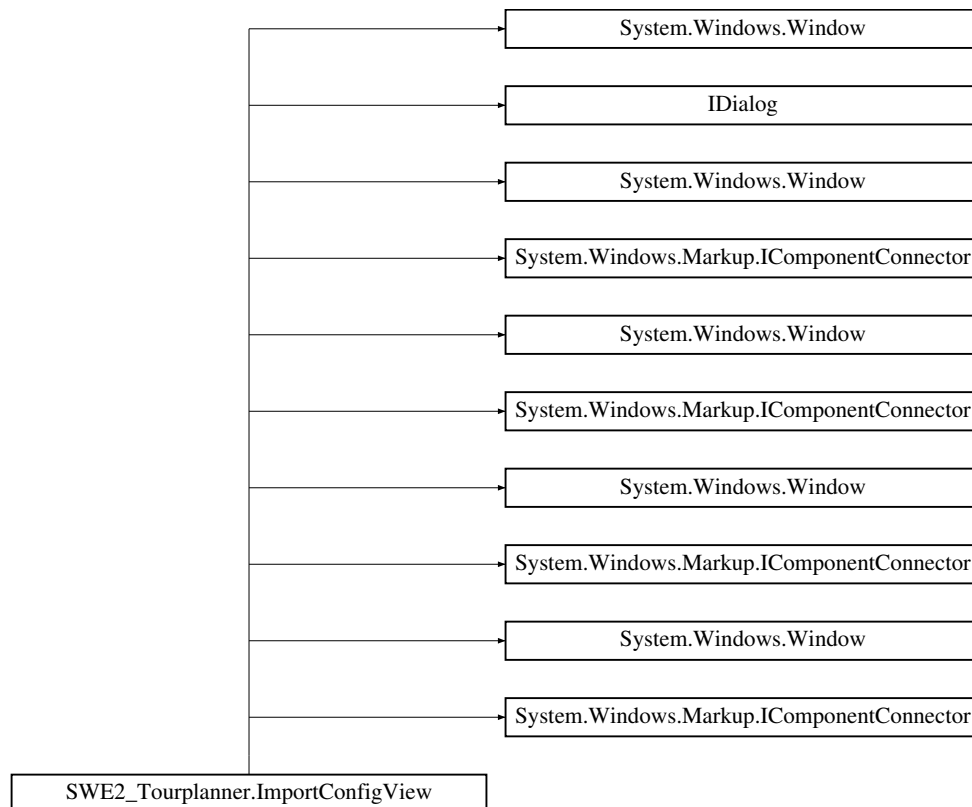
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/MapQuestClient/IMapQuestClient.cs

## 5.39 SWE2\_Tourplanner.ImportConfigView Class Reference

Interaction logic for [ImportConfigView](#)

Inheritance diagram for SWE2\_Tourplanner.ImportConfigView:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

## Private Attributes

- `bool _contentLoaded`

### 5.39.1 Detailed Description

Interaction logic for [ImportConfigView](#)

[ImportConfigView](#)

### 5.39.2 Member Function Documentation

#### 5.39.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.ImportConfigView.InitializeComponent ( )
```

InitializeComponent

#### 5.39.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.ImportConfigView.InitializeComponent ( )
```

InitializeComponent

#### 5.39.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.ImportConfigView.InitializeComponent ( )
```

InitializeComponent

#### 5.39.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.ImportConfigView.InitializeComponent ( )
```

InitializeComponent

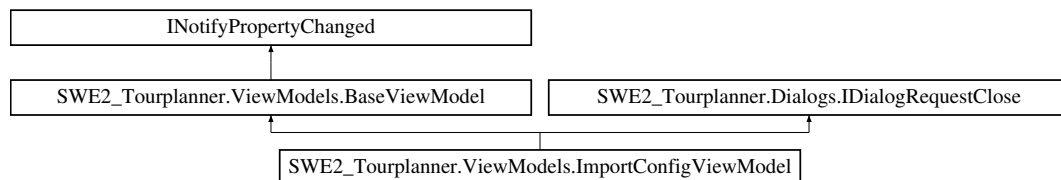
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ImportConfigView.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ImportConfigView.↔g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ImportConfigView.↔g.i.cs

## 5.40 SWE2\_Tourplanner.ViewModels.ImportConfigViewModel Class Reference

[ImportConfigViewModel](#) is used for importing / selecting new configuration files during the execution

Inheritance diagram for SWE2\_Tourplanner.ViewModels.ImportConfigViewModel:



### Public Member Functions

- [ImportConfigViewModel](#) ()  
*Default constructor of [ImportConfigViewModel](#)*

### Properties

- string [ConfigPath](#) [get, set]
- ICommand [SelectConfigCommand](#) [get]
- ICommand [ExitCommand](#) [get]
- ICommand [ConfirmCommand](#) [get]  
*Command for initiating the load config process*

### Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Even used for closing the dialog*

### Private Attributes

- string [configPath](#)  
*Path to the new configuration*

### Additional Inherited Members

#### 5.40.1 Detailed Description

[ImportConfigViewModel](#) is used for importing / selecting new configuration files during the execution

#### 5.40.2 Constructor & Destructor Documentation

### 5.40.2.1 ImportConfigViewModel()

```
SWE2_Tourplanner.ViewModels.ImportConfigViewModel.ImportConfigViewModel ( )
```

Default constructor of [ImportConfigViewModel](#)

## 5.40.3 Member Data Documentation

### 5.40.3.1 configPath

```
string SWE2_Tourplanner.ViewModels.ImportConfigViewModel.configPath [private]
```

Path to the new configuration

## 5.40.4 Property Documentation

### 5.40.4.1 ConfigPath

```
string SWE2_Tourplanner.ViewModels.ImportConfigViewModel.ConfigPath [get], [set]
```

Path to the new configuration

### 5.40.4.2 ConfirmCommand

```
 ICommand SWE2_Tourplanner.ViewModels.ImportConfigViewModel.ConfirmCommand [get]
```

Command for initiating the load config process

### 5.40.4.3 ExitCommand

```
 ICommand SWE2_Tourplanner.ViewModels.ImportConfigViewModel.ExitCommand [get]
```

Command for closing the dialog

### 5.40.4.4 SelectConfigCommand

```
 ICommand SWE2_Tourplanner.ViewModels.ImportConfigViewModel.SelectConfigCommand [get]
```

Command for selecting a new configuration

## 5.40.5 Event Documentation

### 5.40.5.1 CloseRequested

```
EventHandler<DialogCloseRequestedEventArgs> SWE2_Tourplanner.ViewModels.ImportConfigView↔  
Model.CloseRequested
```

Even used for closing the dialog

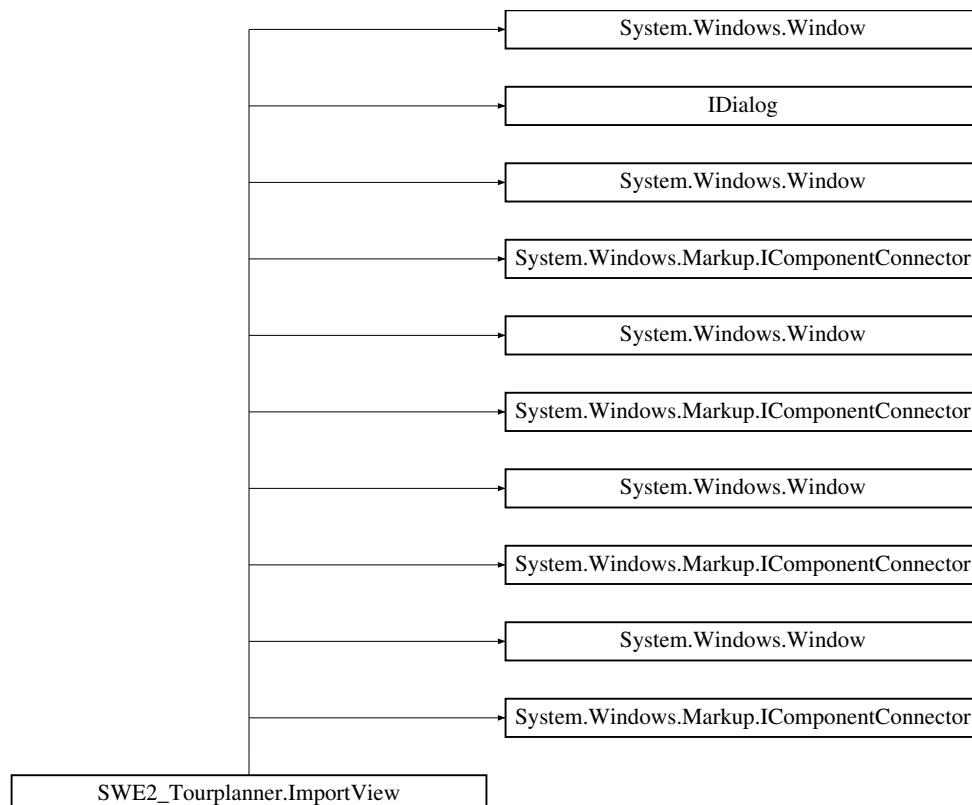
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/ImportConfigViewModel.cs

## 5.41 SWE2\_Tourplanner.ImportView Class Reference

Interaction logic for [ImportView](#)

Inheritance diagram for SWE2\_Tourplanner.ImportView:





## Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

## Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

## Private Attributes

- bool **\_contentLoaded**

### 5.41.1 Detailed Description

Interaction logic for [ImportView](#)

[ImportView](#)

### 5.41.2 Member Function Documentation

#### 5.41.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.ImportView.InitializeComponent ( )
```

InitializeComponent

#### 5.41.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.ImportView.InitializeComponent ( )
```

InitializeComponent

### 5.41.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.ImportView.InitializeComponent ( )
```

InitializeComponent

### 5.41.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.ImportView.InitializeComponent ( )
```

InitializeComponent

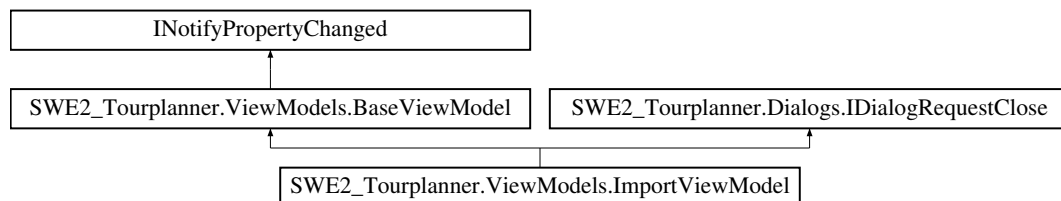
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ImportView.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ImportView.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ImportView.g.i.cs

## 5.42 SWE2\_Tourplanner.ViewModels.ImportViewModel Class Reference

[ImportViewModel](#) is used for selecting JSON files for the import process

Inheritance diagram for SWE2\_Tourplanner.ViewModels.ImportViewModel:



### Public Member Functions

- [ImportViewModel](#) (ITourPlannerConfig config)  
Default constructor for [ImportViewModel](#)

### Properties

- ObservableCollection< string > [JsonPaths](#) [get, set]
- ICommand [SelectPathsCommand](#) [get]
- ICommand [ExitCommand](#) [get]
- ICommand [ConfirmImportCommand](#) [get]

## Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Event used for closing the dialog*

## Private Attributes

- ObservableCollection< string > [jsonPaths](#)  
*All selected JSON files for the import process*
- [ITourPlannerConfig](#) [config](#)  
*ITourPlannerConfig instance for starting the OpenFileDialog in an appropriate folder*

## Additional Inherited Members

### 5.42.1 Detailed Description

[ImportViewModel](#) is used for selecting JSON files for the import process

### 5.42.2 Constructor & Destructor Documentation

#### 5.42.2.1 ImportViewModel()

```
SWE2_Tourplanner.ViewModels.ImportViewModel.ImportViewModel (
    ITourPlannerConfig config )
```

Default constructor for [ImportViewModel](#)

#### Parameters

|               |  |
|---------------|--|
| <i>config</i> | Configuration instance used for OpenFileDialog |
|---------------|--|

### 5.42.3 Member Data Documentation

#### 5.42.3.1 config

```
ITourPlannerConfig SWE2_Tourplanner.ViewModels.ImportViewModel.config [private]
```

[ITourPlannerConfig](#) instance for starting the OpenFileDialog in an appropriate folder

### 5.42.3.2 jsonPaths

`ObservableCollection<string> SWE2_Tourplanner.ViewModels.ImportViewModel.jsonPaths [private]`

All selected JSON files for the import process

## 5.42.4 Property Documentation

### 5.42.4.1 ConfirmImportCommand

`ICommand SWE2_Tourplanner.ViewModels.ImportViewModel.ConfirmImportCommand [get]`

Command for starting the import process

### 5.42.4.2 ExitCommand

`ICommand SWE2_Tourplanner.ViewModels.ImportViewModel.ExitCommand [get]`

Command for closing the dialog

### 5.42.4.3 JsonPaths

`ObservableCollection<string> SWE2_Tourplanner.ViewModels.ImportViewModel.JsonPaths [get], [set]`

All selected JSON files for the import process

### 5.42.4.4 SelectPathsCommand

`ICommand SWE2_Tourplanner.ViewModels.ImportViewModel.SelectPathsCommand [get]`

Command for selecting JSON files

## 5.42.5 Event Documentation

### 5.42.5.1 CloseRequested

`EventHandler<DialogCloseRequestedEventArgs> SWE2_Tourplanner.ViewModels.ImportViewModel.CloseRequested ↵`

Event used for closing the dialog

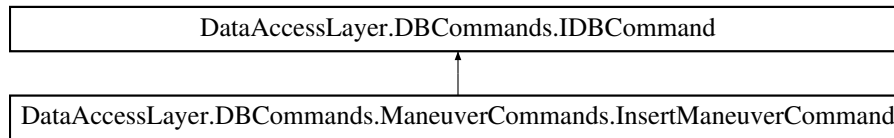
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/ImportViewModel.cs

## 5.43 DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand Class Reference

The [InsertManeuverCommand](#) is used for inserting new maneuvers.

Inheritance diagram for `DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand`:



### Public Member Functions

- [InsertManeuverCommand](#) ([IDBConnection](#) db, [Maneuver](#) maneuver)  
*Creates a new [InsertManeuverCommand](#) instance.*
- [Execute](#) ()  
*Inserts the new maneuver into the maneuver table.*
- [Undo](#) ()  
*Removes the previously inserted maneuver from the maneuver table.*

### Private Attributes

- [IDBConnection](#) db  
*The connection to the database.*
- [Maneuver](#) maneuver  
*The maneuver to be created/deleted.*
- [log4net.ILog](#) logger  
*ILog instance used for logging errors, warnings etc.*

#### 5.43.1 Detailed Description

The [InsertManeuverCommand](#) is used for inserting new maneuvers.

#### 5.43.2 Constructor & Destructor Documentation

##### 5.43.2.1 InsertManeuverCommand()

```

DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.InsertManeuverCommand (
    IDBConnection db,
    Maneuver maneuver )
  
```

Creates a new [InsertManeuverCommand](#) instance.

**Parameters**

|                 |                            |
|-----------------|----------------------------|
| <i>maneuver</i> | Maneuver to be inserted    |
| <i>db</i>       | Connection to the database |

### 5.43.3 Member Function Documentation

#### 5.43.3.1 Execute()

```
int DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.Execute ( )
```

Inserts the new maneuver into the maneuver table.

**Returns**

Amount of rows affected by the insert of the new maneuver. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

#### 5.43.3.2 Undo()

```
int DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.Undo ( )
```

Removes the previously inserted maneuver from the maneuver table.

**Returns**

Amount of rows affected by the delete-statement of the previously inserted maneuver

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.43.4 Member Data Documentation

#### 5.43.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.db [private]
```

The connection to the database.

#### 5.43.4.2 logger

`log4net.ILog DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.logger [private]`

ILog instance used for logging errors, warnings etc.

#### 5.43.4.3 maneuver

`Maneuver DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand.maneuver [private]`

The maneuver to be created/deleted.

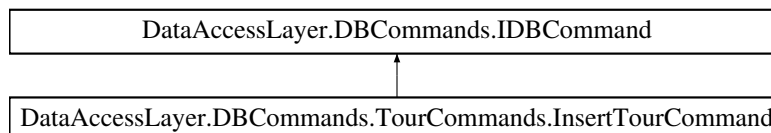
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/ManeuverCommands/InsertManeuverCommand.cs

## 5.44 DataAccessLayer.DBCommands.TourCommands.InsertTourCommand Class Reference

`InsertTourCommand` is used for inserting new tours.

Inheritance diagram for `DataAccessLayer.DBCommands.TourCommands.InsertTourCommand`:



### Public Member Functions

- `InsertTourCommand (IDBConnection db, Tour tour)`  
*Create the `InsertTourCommand` instance.*
- `int Execute ()`  
*Inserts a new tour into the tour table.*
- `int Undo ()`  
*Deletes the previously inserted tour.*

### Private Attributes

- `IDBConnection db`  
*The connection to the database.*
- `Tour tour`  
*The tour to be inserted.*

### 5.44.1 Detailed Description

[InsertTourCommand](#) is used for inserting new tours.

### 5.44.2 Constructor & Destructor Documentation

#### 5.44.2.1 InsertTourCommand()

```
.DataAccessLayer.DBCommands.TourCommands.InsertTourCommand.InsertTourCommand (
    IDBConnection db,
    Tour tour )
```

Create the [InsertTourCommand](#) instance.

##### Parameters

|             |                             |
|-------------|-----------------------------|
| <i>db</i>   | Connection to the database. |
| <i>tour</i> | Tour to be inserted         |

### 5.44.3 Member Function Documentation

#### 5.44.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourCommands.InsertTourCommand.Execute ( )
```

Inserts a new tour into the tour table.

##### Returns

Amount of rows affected by the insert statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

#### 5.44.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourCommands.InsertTourCommand.Undo ( )
```

Deletes the previously inserted tour.

##### Returns

Amount of rows affected by the delete statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).



## 5.44.4 Member Data Documentation

### 5.44.4.1 db

`IDBConnection` `DataAccessLayer.DBCommands.TourCommands.InsertTourCommand.db` [private]

The connection to the database.

### 5.44.4.2 tour

`Tour` `DataAccessLayer.DBCommands.TourCommands.InsertTourCommand.tour` [private]

The tour to be inserted.

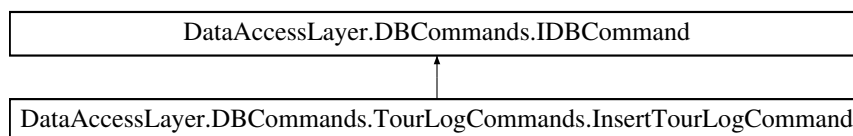
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourCommands/InsertTour↵  
Command.cs

## 5.45 DataAccessLayer.DBCommands.TourLogCommands.InsertTour↵ LogCommand Class Reference

`InsertTourLogCommand` is used for inserting new logs for tours.

Inheritance diagram for `DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand`:



## Public Member Functions

- `InsertTourLogCommand` (`IDBConnection db`, `TourLog tourLog`)  
*Creates a new `InsertTourLogCommand` instance.*
- `int Execute` ()  
*Inserts a new log of a tour into the tourlog table*
- `int Undo` ()  
*Deletes the previously inserted tourlog.*

## Private Attributes

- [IDBConnection db](#)  
*Connection to the database.*
- [TourLog tourLog](#)  
*The new log to be inserted.*
- [log4net.ILog logger](#)  
*ILog instance used for logging errors, warnings etc.*

### 5.45.1 Detailed Description

[InsertTourLogCommand](#) is used for inserting new logs for tours.

### 5.45.2 Constructor & Destructor Documentation

#### 5.45.2.1 InsertTourLogCommand()

```
.DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.InsertTourLogCommand (
    IDBConnection db,
    TourLog tourLog )
```

Creates a new [InsertTourLogCommand](#) instance.

##### Parameters

|                |                             |
|----------------|-----------------------------|
| <i>db</i>      | Connection to the database. |
| <i>tourLog</i> | Log to be inserted          |

### 5.45.3 Member Function Documentation

#### 5.45.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.Execute ( )
```

Inserts a new log of a tour into the tourlog table

##### Returns

Amount of rows affected by the insert statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.45.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.Undo ( )
```

Deletes the previously inserted tourlog.

#### Returns

Amount of rows affected by the delete-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

## 5.45.4 Member Data Documentation

### 5.45.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.db [private]
```

Connection to the database.

### 5.45.4.2 logger

```
log4net.ILog DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.logger [private]
```

ILog instance used for logging errors, warnings etc.

### 5.45.4.3 tourLog

```
TourLog DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand.tourLog [private]
```

The new log to be inserted.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourLogCommands/InsertTour↵  
LogCommand.cs

## 5.46 DataAccessLayer.Repositories.IRepository< T > Interface Template Reference

Generic repository that defines basic functions for querying, inserting, updating and deleting data.

## Public Member Functions

- void [Insert](#) (T entity)  
*Inserts a new entity into the corresponding table.*
- T [Read](#) (int id)  
*Retrieves an entity with the specified Id*
- List< T > [ReadAll](#) ()  
*Retrieves every entity of the type T.*
- void [Update](#) (T entity)  
*Updates the data in the respective table according to the new state.*
- void [Delete](#) (int id)  
*Removes an entity with the specified id from the T table.*

### 5.46.1 Detailed Description

Generic repository that defines basic functions for querying, inserting, updating and deleting data.

#### Template Parameters

|          |  |
|----------|--|
| <i>T</i> | Class from Entities-Namespace: Tour, TourLog, Maneuver |
|----------|--|

#### Type Constraints

*T*: **class**

### 5.46.2 Member Function Documentation

#### 5.46.2.1 Delete()

```
void DataAccessLayer.Repositories.IRepository< T >.Delete (
    int id )
```

Removes an entity with the specified id from the T table.

#### Parameters

|           |  |
|-----------|--|
| <i>id</i> | Id of the entity that is supposed to be deleted. |
|-----------|--|

#### 5.46.2.2 Insert()

```
void DataAccessLayer.Repositories.IRepository< T >.Insert (
    T entity )
```

Inserts a new entity into the corresponding table.

**Parameters**

|               |                       |
|---------------|-----------------------|
| <i>entity</i> | Entity to be inserted |
|---------------|-----------------------|

**5.46.2.3 Read()**

```
T DataAccessLayer.Repositories.IRepository< T >.Read (
    int id )
```

Retrieves an entity with the specified Id

**Parameters**

|           |                         |
|-----------|-------------------------|
| <i>id</i> | Id of the wanted entity |
|-----------|-------------------------|

**Returns**

Entity with the specified id or null if no such entity exists.

**5.46.2.4 ReadAll()**

```
List<T> DataAccessLayer.Repositories.IRepository< T >.ReadAll ( )
```

Retrieves every entity of the type T.

**Returns**

Collection of every entity in the corresponding table.

**5.46.2.5 Update()**

```
void DataAccessLayer.Repositories.IRepository< T >.Update (
    T entity )
```

Updates the data in the respective table according to the new state.

**Parameters**

|               |                         |
|---------------|-------------------------|
| <i>entity</i> | New state of the entity |
|---------------|-------------------------|

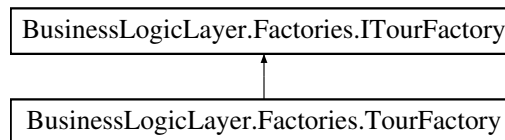
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/IRepository.cs

## 5.47 BusinessLogicLayer.Factories.ITourFactory Interface Reference

[ITourFactory](#) defines several methods for managing and retrieving Tour data from datastore

Inheritance diagram for BusinessLogicLayer.Factories.ITourFactory:



### Public Member Functions

- Task [CreateMapQuestTour](#) ([Tour](#) tour)  
*CreateMapQuestTour is a method for creating new Tours with data from MapQuest*
- Task [CreateTour](#) ([Tour](#) tour)  
*CreateTour is a method for inserting new Tour entities in the datastore*
- Task [DeleteTour](#) ([Tour](#) tour)  
*DeleteTour is a method for deleting Tour entities in the datastore*
- Task [UpdateMapQuestTour](#) ([Tour](#) tour)  
*UpdateMapQuestTour is a method for updating a MapQuest Tour*
- Task [UpdateTour](#) ([Tour](#) tour)  
*UpdateTour is a method for updating Tour entities in the datastore*
- List< [Tour](#) > [GetTours](#) ()  
*GetTours is a method for retrieving Tour entities*

### 5.47.1 Detailed Description

[ITourFactory](#) defines several methods for managing and retrieving Tour data from datastore

### 5.47.2 Member Function Documentation

#### 5.47.2.1 CreateMapQuestTour()

```
Task BusinessLogicLayer.Factories.ITourFactory.CreateMapQuestTour (
    Tour tour )
```

CreateMapQuestTour is a method for creating new Tours with data from MapQuest

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>tour</i> | Tour to be created |
|-------------|--------------------|

**Returns**

Task, which retrieves data from MapQuest API and creates a Tour

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).

**5.47.2.2 CreateTour()**

```
Task BusinessLogicLayer.Factories.ITourFactory.CreateTour (
    Tour tour )
```

CreateTour is a method for inserting new Tour entities in the datastore

**Parameters**

|             |                                     |
|-------------|-------------------------------------|
| <i>tour</i> | Tour that is supposed to be created |
|-------------|-------------------------------------|

**Returns**

Task, which inserts a new Tour entity in the datastore

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).

**5.47.2.3 DeleteTour()**

```
Task BusinessLogicLayer.Factories.ITourFactory.DeleteTour (
    Tour tour )
```

DeleteTour is a method for deleting Tour entities in the datastore

**Parameters**

|             |                                     |
|-------------|-------------------------------------|
| <i>tour</i> | Tour that is supposed to be deleted |
|-------------|-------------------------------------|

**Returns**

Task, which deletes a Tour in the datastore

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).



#### 5.47.2.4 GetTours()

```
List<Tour> BusinessLogicLayer.Factories.ITourFactory.GetTours ( )
```

GetTours is a method for retrieving Tour entities

##### Returns

All Tour entities from the datastore

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).

#### 5.47.2.5 UpdateMapQuestTour()

```
Task BusinessLogicLayer.Factories.ITourFactory.UpdateMapQuestTour (
    Tour tour )
```

UpdateMapQuestTour is a method for updating a MapQuest Tour

##### Parameters

|             |              |
|-------------|--------------|
| <i>tour</i> | Updated tour |
|-------------|--------------|

##### Returns

Task, which retrieves data from MapQuest API and updates Tour entity in datastore

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).

#### 5.47.2.6 UpdateTour()

```
Task BusinessLogicLayer.Factories.ITourFactory.UpdateTour (
    Tour tour )
```

UpdateTour is a method for updating Tour entities in the datastore

##### Parameters

|             |                   |
|-------------|-------------------|
| <i>tour</i> | Updated tour data |
|-------------|-------------------|

##### Returns

Task, which updates a Tour entity

Implemented in [BusinessLogicLayer.Factories.TourFactory](#).

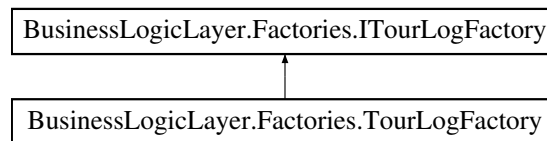
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Factories/ITourFactory.cs

## 5.48 BusinessLogicLayer.Factories.ITourLogFactory Interface Reference

[ITourLogFactory](#) defines several methods for managing TourLog entities

Inheritance diagram for BusinessLogicLayer.Factories.ITourLogFactory:



### Public Member Functions

- Task [CreateTourLog](#) ([TourLog](#) tourLog)  
*CreateTourLog is a method for crating new TourLog entities in the datastore*
- Task [UpdateTourLog](#) ([TourLog](#) tourLog)  
*UpdateTourLog is a method for updating TourLog entities in the datastore*
- Task [DeleteTourLog](#) ([TourLog](#) tourLog)  
*DeleteTourLog is a method for deleting TourLog entities in the datastore*

### 5.48.1 Detailed Description

[ITourLogFactory](#) defines several methods for managing TourLog entities

### 5.48.2 Member Function Documentation

#### 5.48.2.1 CreateTourLog()

```
Task BusinessLogicLayer.Factories.ITourLogFactory.CreateTourLog (
    TourLog tourLog )
```

CreateTourLog is a method for crating new TourLog entities in the datastore

Parameters

|                |                       |
|----------------|-----------------------|
| <i>tourLog</i> | TourLog to be created |
|----------------|-----------------------|

**Returns**

Task, which creates a new TourLog entity in the datastore

Implemented in [BusinessLogicLayer.Factories.TourLogFactory](#).

**5.48.2.2 DeleteTourLog()**

```
Task BusinessLogicLayer.Factories.ITourLogFactory.DeleteTourLog (
    TourLog tourLog )
```

DeleteTourLog is a method for deleting TourLog entities in the datastore

**Parameters**

|                |                       |
|----------------|-----------------------|
| <i>tourLog</i> | TourLog to be deleted |
|----------------|-----------------------|

**Returns**

Task, which deletes a TourLog entity in the datastore

Implemented in [BusinessLogicLayer.Factories.TourLogFactory](#).

**5.48.2.3 UpdateTourLog()**

```
Task BusinessLogicLayer.Factories.ITourLogFactory.UpdateTourLog (
    TourLog tourLog )
```

UpdateTourLog is a method for updating TourLog entities in the datastore

**Parameters**

|                |                        |
|----------------|------------------------|
| <i>tourLog</i> | Updated TourLog entity |
|----------------|------------------------|

**Returns**

Task, which updates a TourLog entity in the database

Implemented in [BusinessLogicLayer.Factories.TourLogFactory](#).

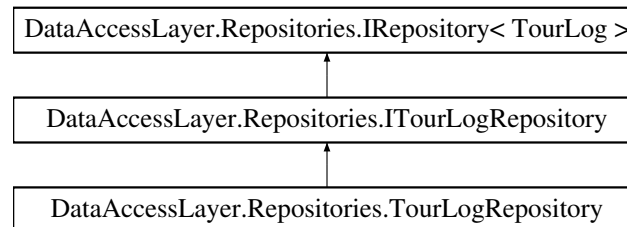
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Factories/ITourLogFactory.cs

## 5.49 DataAccessLayer.Repositories.ITourLogRepository Interface Reference

The [ITourLogRepository](#) interface is a place, where specific functions for tourlogs can be defined. The basic methods of repositories are derived from [IRepository<TourLog>](#)

Inheritance diagram for DataAccessLayer.Repositories.ITourLogRepository:



### Additional Inherited Members

#### 5.49.1 Detailed Description

The [ITourLogRepository](#) interface is a place, where specific functions for tourlogs can be defined. The basic methods of repositories are derived from [IRepository<TourLog>](#)

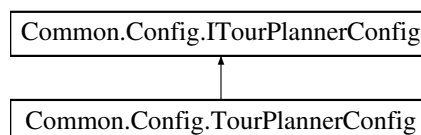
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/ITourLogRepository.cs

## 5.50 Common.Config.ITourPlannerConfig Interface Reference

The [ITourPlannerConfig](#) interface ensures that all derived TourPlannerConfigs provide a MapQuestKey, a Picture↔ Directory, an ExportsDirectory, a DatabaseConnectionString and the option to load another config file during the execution of the program.

Inheritance diagram for Common.Config.ITourPlannerConfig:



### Public Member Functions

- void [LoadConfigFromFile](#) (string configPath)  
*Method used for loading different configurations during the execution of the program.*

## Properties

- string [MapQuestKey](#) [get]
- string [PictureDirectory](#) [get]
- string [DatabaseConnectionString](#) [get]
- string [ExportsDirectory](#) [get]

### 5.50.1 Detailed Description

The [ITourPlannerConfig](#) interface ensures that all derived TourPlannerConfigs provide a MapQuestKey, a Picture↔Directory, an ExportsDirectory, a DatabaseConnectionString and the option to load another config file during the execution of the program.

### 5.50.2 Member Function Documentation

#### 5.50.2.1 LoadConfigFromFile()

```
void Common.Config.ITourPlannerConfig.LoadConfigFromFile (
    string configPath )
```

Method used for loading different configurations during the execution of the program.

##### Parameters

|                   |                         |
|-------------------|-------------------------|
| <i>configPath</i> | Path to the config file |
|-------------------|-------------------------|

Implemented in [Common.Config.TourPlannerConfig](#).

### 5.50.3 Property Documentation

#### 5.50.3.1 DatabaseConnectionString

```
string Common.Config.ITourPlannerConfig.DatabaseConnectionString [get]
```

The DatabaseConnectionString defines the database where the data is going to be stored.

#### 5.50.3.2 ExportsDirectory

```
string Common.Config.ITourPlannerConfig.ExportsDirectory [get]
```

ExportsDirectory is a path to a directory where exports and associated images are saved.

### 5.50.3.3 MapQuestKey

```
string Common.Config.ITourPlannerConfig.MapQuestKey [get]
```

The MapQuestKey is used to retrieve data about tours from mapquest.

### 5.50.3.4 PictureDirectory

```
string Common.Config.ITourPlannerConfig.PictureDirectory [get]
```

PictureDirectory is a path to a directory where MapQuest maps can be saved.

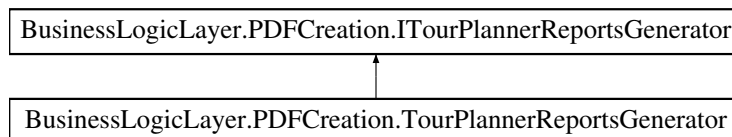
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Config/ITourPlannerConfig.cs

## 5.51 BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator Interface Reference

An interface that defines methods for generating different reports

Inheritance diagram for BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator:



### Public Member Functions

- Task **GenerateTourReport** ([Tour](#) tour)
- Task **GenerateSummaryReport** (List< [Tour](#) > tours)

### 5.51.1 Detailed Description

An interface that defines methods for generating different reports

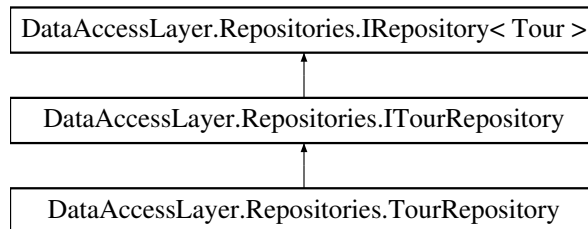
The documentation for this interface was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/PDFCreation/ITourPlannerReportsGenerator.cs

## 5.52 DataAccessLayer.Repositories.ITourRepository Interface Reference

The [ITourRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Tour>](#)

Inheritance diagram for DataAccessLayer.Repositories.ITourRepository:



### Additional Inherited Members

#### 5.52.1 Detailed Description

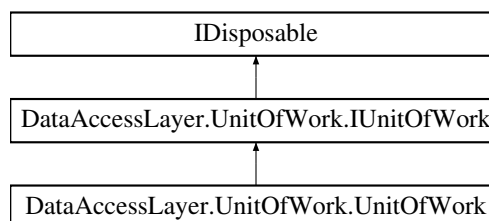
The [ITourRepository](#) interface is a place, where specific functions for maneuvers can be defined. The basic methods of repositories are derived from [IRepository<Tour>](#)

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/ITourRepository.cs

## 5.53 DataAccessLayer.UnitOfWork.IUnitOfWork Interface Reference

Inheritance diagram for DataAccessLayer.UnitOfWork.IUnitOfWork:



### Public Member Functions

- int [Commit](#) ()  
*The commit function takes every issued command and executes them together. That way, the database connection is only established once.*
- int [Rollback](#) ()  
*The rollback function reverts the data store to it's state before the commit. Like the [Commit\(\)](#) method, only one database connection is established for every command in the rollback.*

## Properties

- [ITourRepository](#) [TourRepository](#) [get, set]
- [ITourLogRepository](#) [TourLogRepository](#) [get, set]
- [IManeuverRepository](#) [ManeuverRepository](#) [get, set]

### 5.53.1 Member Function Documentation

#### 5.53.1.1 Commit()

```
int DataAccessLayer.UnitOfWork.IUnitOfWork.Commit ( )
```

The commit function takes every issued command and executes them together. That way, the database connection is only established once.

##### Returns

Amount of affected rows by the commit. Can be used to check if the commit affected the expected amount of rows.

Implemented in [DataAccessLayer.UnitOfWork.UnitOfWork](#).

#### 5.53.1.2 Rollback()

```
int DataAccessLayer.UnitOfWork.IUnitOfWork.Rollback ( )
```

The rollback function reverts the data store to it's state before the commit. Like the [Commit\(\)](#) method, only one database connection is established for every command in the rollback.

##### Returns

Amount of affected rows by the rollback.

Implemented in [DataAccessLayer.UnitOfWork.UnitOfWork](#).

### 5.53.2 Property Documentation

#### 5.53.2.1 ManeuverRepository

```
IManeuverRepository DataAccessLayer.UnitOfWork.IUnitOfWork.ManeuverRepository [get], [set]
```

Provides the user with a maneuver repository.



### 5.53.2.2 TourLogRepository

`ITourLogRepository` `DataAccessLayer.UnitOfWork.IUnitOfWork.TourLogRepository` [get], [set]

Provides the user with a tourlog repository.

### 5.53.2.3 TourRepository

`ITourRepository` `DataAccessLayer.UnitOfWork.IUnitOfWork.TourRepository` [get], [set]

Provides the user with a tour repository.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/UnitOfWork/IUnitOfWork.cs

## 5.54 Common.Logging.LogHelper Class Reference

The `LogHelper` class is responsible for creating `ILog` objects. Those objects are used throughout the different layers of the application for logging purposes. Is a singleton.

### Public Member Functions

- `log4net.ILog` `GetLogger` ([CallerFilePath] string filename="")  
*In order to get an already configured ILog instance, the GetLogger method can be called.*

### Static Public Member Functions

- static `LogHelper` `GetLogHelper` ()  
*Global access point to the only instance of LogHelper.*

### Private Member Functions

- `LogHelper` ([CallerFilePath] string filename="")  
*Private constructor that gets called only once.*

### Static Private Attributes

- static `LogHelper` `logHelper`  
*One and only instance of LogHelper during execution.*

### 5.54.1 Detailed Description

The `LogHelper` class is responsible for creating `ILog` objects. Those objects are used throughout the different layers of the application for logging purposes. Is a singleton.

### 5.54.2 Constructor & Destructor Documentation

#### 5.54.2.1 LogHelper()

```
Common.Logging.LogHelper.LogHelper (
    [CallerFilePath] string filename = "" ) [private]
```

Private constructor that gets called only once.

## Parameters

|                 |   |
|-----------------|---|
| <i>filename</i> | Filename of the code that calls the constructor |
|-----------------|---|

### 5.54.3 Member Function Documentation

#### 5.54.3.1 GetLogger()

```
log4net.ILog Common.Logging.LogHelper.GetLogger (
    [CallerFilePath] string filename = "" )
```

In order to get an already configured ILog instance, the GetLogger method can be called.

## Parameters

|                 |   |
|-----------------|---|
| <i>filename</i> | Name of the file where the ILog instance is requested. This parameter doesn't need to be passed due to CallerFilePath |
|-----------------|---|

## Returns

Properly configured ILog instance for corresponding files/classes

#### 5.54.3.2 GetLogHelper()

```
static LogHelper Common.Logging.LogHelper.GetLogHelper ( ) [static]
```

Global access point to the only instance of [LogHelper](#).

## Returns

Only instance of [LogHelper](#)

### 5.54.4 Member Data Documentation

#### 5.54.4.1 logHelper

```
LogHelper Common.Logging.LogHelper.logHelper [static], [private]
```

One and only instance of [LogHelper](#) during execution.

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

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Logging/LogHelper.cs

## 5.55 SWE2\_Tourplanner\_Tests.CommonTests.LogHelperTests Class Reference

### Public Member Functions

- void **GetLogHelperTest** ()
- void **GetLoggerTest** ()

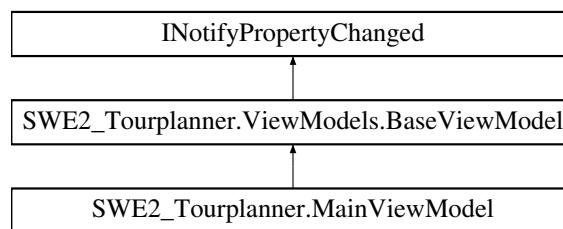
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/CommonTests/LogHelperTests.cs

## 5.56 SWE2\_Tourplanner.MainViewModel Class Reference

[MainViewModel](#) is core of the application and is used for executing several different actions triggered by other [ViewModels](#)

Inheritance diagram for SWE2\_Tourplanner.MainViewModel:



### Public Member Functions

- [MainViewModel](#) (IDialogService dialogService)  
*Default constructor for [MainViewModel](#)*

### Properties

- [TourDetailViewModel](#) [TourDetailViewModel](#) [get]
- [ToursViewModel](#) [ToursViewModel](#) [get]
- ICommand [AddTourCommand](#) [get]
- ICommand [RemoveTourCommand](#) [get]
- ICommand [EditTourCommand](#) [get]
- ICommand [GenerateTourReportCommand](#) [get]
- ICommand [GenerateSummaryReportCommand](#) [get]
- ICommand [GenerateJSONExportCommand](#) [get]
- ICommand [ConductImportCommand](#) [get]
- ICommand [LoadConfigCommand](#) [get]

## Private Member Functions

- void [ShowException](#) (Exception e)  
*Helper method for displaying exceptions*

## Private Attributes

- [TourDetailViewModel](#) `tourDetailViewModel`  
*TourDetailViewModel contains the currently selected Tour and several different commands for managing its TourLog entities*
- [ToursViewModel](#) `toursViewModel`  
*ToursViewModel provides Tour entities and a search command*
- [ITourFactory](#) `tourPlannerFactory`  
*Object used for creating, updating and deleting tours.*
- readonly [IDialogService](#) `dialogService`  
*Used for opening dialogs and handling the results*
- [ITourPlannerReportsGenerator](#) `pdfGenerator`  
*PDF Generator for Summary and Tour Reports*
- [IExporterImporter< Tour >](#) `exporterImporter`  
*Used for exporting / importing data to / from JSON files*

## Additional Inherited Members

### 5.56.1 Detailed Description

[MainViewModel](#) is core of the application and is used for executing several different actions triggered by other [ViewModels](#)

### 5.56.2 Constructor & Destructor Documentation

#### 5.56.2.1 MainViewModel()

```
SWE2_Tourplanner.MainViewModel.MainViewModel (
    IDialogService dialogService )
```

Default constructor for [MainViewModel](#)

Parameters

|                            |   |
|----------------------------|---|
| <code>dialogService</code> | Service used for handling <a href="#">Dialogs</a> and their results |
|----------------------------|---|

### 5.56.3 Member Function Documentation

### 5.56.3.1 ShowException()

```
void SWE2_Tourplanner.MainViewModel.ShowException (
    Exception e ) [private]
```

Helper method for displaying exceptions

#### Parameters

|          |                  |
|----------|------------------|
| <i>e</i> | Thrown Exception |
|----------|------------------|

## 5.56.4 Member Data Documentation

### 5.56.4.1 dialogService

```
readonly IDialogService SWE2_Tourplanner.MainViewModel.dialogService [private]
```

Used for opening dialogs and handling the results

### 5.56.4.2 exporterImporter

```
IExporterImporter<Tour> SWE2_Tourplanner.MainViewModel.exporterImporter [private]
```

Used for exporting / importing data to / from JSON files

### 5.56.4.3 pdfGenerator

```
ITourPlannerReportsGenerator SWE2_Tourplanner.MainViewModel.pdfGenerator [private]
```

PDF Generator for Summary and Tour Reports

### 5.56.4.4 tourDetailViewModel

```
TourDetailViewModel SWE2_Tourplanner.MainViewModel.tourDetailViewModel [private]
```

TourDetailViewModel contains the currently selected Tour and several different commands for managing its TourLog entities

#### 5.56.4.5 tourPlannerFactory

`ITourFactory SWE2_Tourplanner.MainViewModel.tourPlannerFactory [private]`

Object used for creating, updating and deleting tours.

#### 5.56.4.6 toursViewModel

`ToursViewModel SWE2_Tourplanner.MainViewModel.toursViewModel [private]`

ToursViewModel provides Tour entities and a search command

### 5.56.5 Property Documentation

#### 5.56.5.1 AddTourCommand

`ICommand SWE2_Tourplanner.MainViewModel.AddTourCommand [get]`

Command for creating new tours

#### 5.56.5.2 ConductImportCommand

`ICommand SWE2_Tourplanner.MainViewModel.ConductImportCommand [get]`

Command for importing data from JSON files

#### 5.56.5.3 EditTourCommand

`ICommand SWE2_Tourplanner.MainViewModel.EditTourCommand [get]`

TCommand for updating tours

#### 5.56.5.4 GenerateJSONExportCommand

`ICommand SWE2_Tourplanner.MainViewModel.GenerateJSONExportCommand [get]`

Command for exporting data as JSON file

#### 5.56.5.5 GenerateSummaryReportCommand

`ICommand SWE2_Tourplanner.MainViewModel.GenerateSummaryReportCommand [get]`

Command for generating a Summary report

#### 5.56.5.6 GenerateTourReportCommand

`ICommand SWE2_Tourplanner.MainViewModel.GenerateTourReportCommand [get]`

Command for generating a Tour report

#### 5.56.5.7 LoadConfigCommand

`ICommand SWE2_Tourplanner.MainViewModel.LoadConfigCommand [get]`

Command for loading a new configuration during execution

#### 5.56.5.8 RemoveTourCommand

`ICommand SWE2_Tourplanner.MainViewModel.RemoveTourCommand [get]`

Command for deleting tours

#### 5.56.5.9 TourDetailViewModel

`TourDetailViewModel SWE2_Tourplanner.MainViewModel.TourDetailViewModel [get]`

TourDetailViewModel contains the currently selected Tour and several different commands for managing its TourLog entities

#### 5.56.5.10 ToursViewModel

`ToursViewModel SWE2_Tourplanner.MainViewModel.ToursViewModel [get]`

ToursViewModel provides Tour entities and a search command

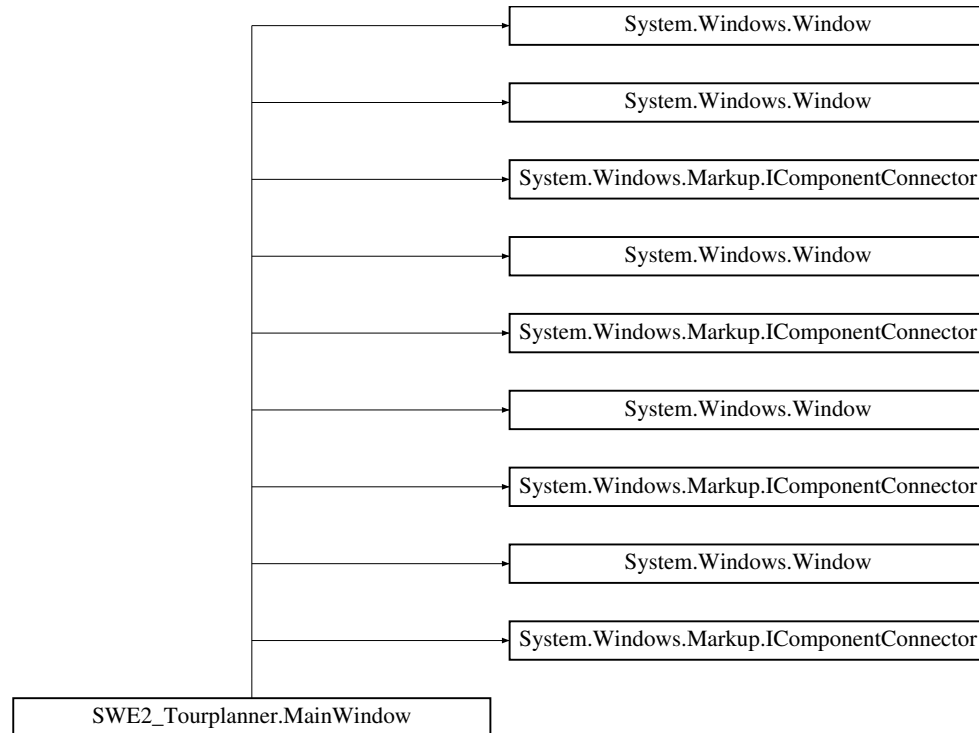
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/MainViewModel.cs

## 5.57 SWE2\_Tourplanner.MainWindow Class Reference

Interaction logic for [MainWindow](#)

Inheritance diagram for SWE2\_Tourplanner.MainWindow:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

### Private Attributes

- bool **\_contentLoaded**



### 5.57.1 Detailed Description

Interaction logic for [MainWindow](#)

[MainWindow](#)

### 5.57.2 Member Function Documentation

#### 5.57.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.MainWindow.InitializeComponent ( )
```

InitializeComponent

#### 5.57.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.MainWindow.InitializeComponent ( )
```

InitializeComponent

#### 5.57.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.MainWindow.InitializeComponent ( )
```

InitializeComponent

#### 5.57.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.MainWindow.InitializeComponent ( )
```

InitializeComponent

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/MainWindow.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/MainWindow.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/MainWindow.g.i.cs

## 5.58 Common.Entities.Maneuver Class Reference

The maneuver entity is basically a single part of the overall route description of a tour. It consists of a narrative, describing the part of the route in detail, and the length (km) of the section.

### Public Member Functions

- override bool **Equals** (object obj)
- override int **GetHashCode** ()

### Properties

- int **Id** [get, set]  
*Id of the maneuver.*
- int **TourId** [get, set]  
*Id of the associated tour.*
- string **Narrative** [get, set]  
*Description of the section*
- double **Distance** [get, set]  
*Length of the section. Unit is km.*

### 5.58.1 Detailed Description

The maneuver entity is basically a single part of the overall route description of a tour. It consists of a narrative, describing the part of the route in detail, and the length (km) of the section.

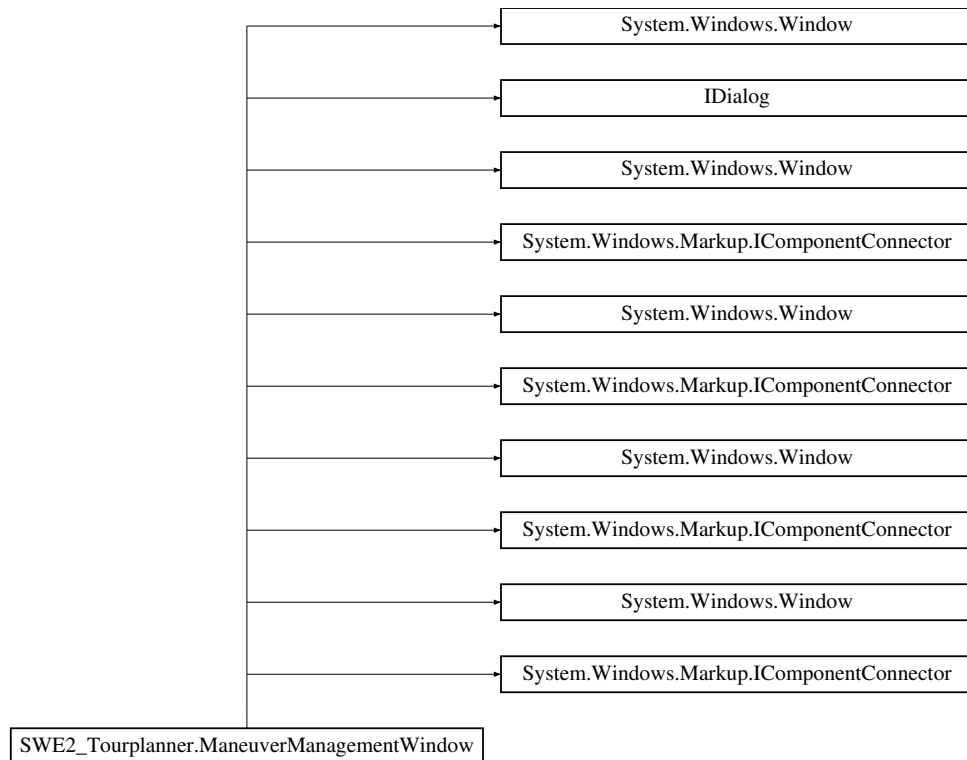
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Entities/Maneuver.cs

## 5.59 SWE2\_Tourplanner.ManeuverManagementWindow Class Reference

Interaction logic for [ManeuverManagementWindow](#)

Inheritance diagram for SWE2\_Tourplanner.ManeuverManagementWindow:



## Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

## Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

## Private Attributes

- bool **\_contentLoaded**

### 5.59.1 Detailed Description

Interaction logic for [ManeuverManagementWindow](#)

[ManeuverManagementWindow](#)

## 5.59.2 Member Function Documentation

### 5.59.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.ManeuverManagementWindow.InitializeComponent ( )
```

InitializeComponent

### 5.59.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.ManeuverManagementWindow.InitializeComponent ( )
```

InitializeComponent

### 5.59.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.ManeuverManagementWindow.InitializeComponent ( )
```

InitializeComponent

### 5.59.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.ManeuverManagementWindow.InitializeComponent ( )
```

InitializeComponent

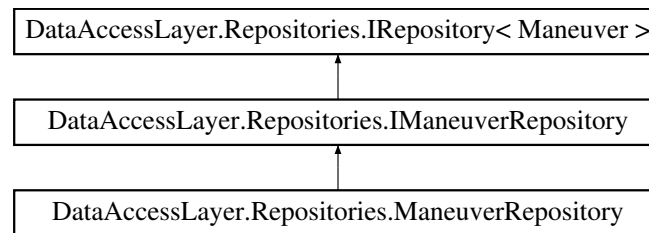
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ManeuverManagementWindow.xaml.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ManeuverManagement↵Window.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ManeuverManagement↵Window.g.i.cs

## 5.60 DataAccessLayer.Repositories.ManeuverRepository Class Reference

ManeuverRepositories are used for querying, inserting, updating and deleting maneuvers. DML commands are not executed in this class. The only way to truly insert, update and delete maneuvers is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

Inheritance diagram for DataAccessLayer.Repositories.ManeuverRepository:



### Public Member Functions

- [ManeuverRepository](#) ()  
*Creates the [ManeuverRepository](#) instance.*
- [ManeuverRepository](#) (IDBConnection db, List< IDBCommand > commitCommands)  
*Creates the [ManeuverRepository](#) instance and "connects" it to the [UnitOfWork](#) class*
- void [Delete](#) (int id)  
*Deletes a maneuver with a specific id.*
- void [Insert](#) ([Maneuver](#) entity)  
*Checks if properties are ok. If so, creates a InsertManeuverCommand instance with the specified data.*
- [Maneuver Read](#) (int id)  
*Function for retrieving a maneuver with a specific id.*
- List< [Maneuver](#) > [ReadAll](#) ()  
*Function for retrieval of all available maneuvers.*
- void [Update](#) ([Maneuver](#) entity)  
*Checks whether a maneuver with the same id as the parameter exists. If so, creates an UpdateManeuverCommand with the old and new state of the maneuver.*

### Private Member Functions

- [Maneuver ConvertToManeuver](#) (object[] row)  
*Converts object arrays to maneuvers.*
- bool [CheckDBConstraints](#) ([Maneuver](#) maneuver)  
*CheckDBConstraints is used to check whether the db constraints are complied with or not.*

### Private Attributes

- IDBConnection db  
*Database connection*
- List< IDBCommand > commitCommands  
*Commands to be committed to the database.*
- log4net.ILog logger  
*ILog object used for logging errors etc.*

### 5.60.1 Detailed Description

ManeuverRepositories are used for querying, inserting, updating and deleting maneuvers. DML commands are not executed in this class. The only way to truly insert, update and delete maneuvers is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

### 5.60.2 Constructor & Destructor Documentation

#### 5.60.2.1 ManeuverRepository() [1/2]

```
DataAccessLayer.Repositories.ManeuverRepository.ManeuverRepository ( )
```

Creates the [ManeuverRepository](#) instance.

#### 5.60.2.2 ManeuverRepository() [2/2]

```
DataAccessLayer.Repositories.ManeuverRepository.ManeuverRepository (
    IDbConnection db,
    List< IDbCommand > commitCommands )
```

Creates the [ManeuverRepository](#) instance and "connects" it to the [UnitOfWork](#) class

##### Parameters

|                       |  |
|-----------------------|--|
| <i>db</i>             | Database connection                                  |
| <i>commitCommands</i> | Commands for a commit ( <a href="#">UnitOfWork</a> ) |

### 5.60.3 Member Function Documentation

#### 5.60.3.1 CheckDBConstraints()

```
bool DataAccessLayer.Repositories.ManeuverRepository.CheckDBConstraints (
    Maneuver maneuver ) [private]
```

CheckDBConstraints is used to check whether the db constraints are complied with or not.

##### Parameters

|                 |  |
|-----------------|--|
| <i>maneuver</i> | The maneuver that needs to be checked. |
|-----------------|--|

**Returns**

True if constraints are adhered to, false if constraints are not complied with.

**5.60.3.2 ConvertToManeuver()**

```
Maneuver DataAccessLayer.Repositories.ManeuverRepository.ConvertToManeuver (
    object[] row ) [private]
```

Converts object arrays to maneuvers.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>row</i> | Result of a query |
|------------|-------------------|

**Returns**

Converted maneuver.

**5.60.3.3 Delete()**

```
void DataAccessLayer.Repositories.ManeuverRepository.Delete (
    int id )
```

Deletes a maneuver with a specific id.

**Parameters**

|           |                                  |
|-----------|----------------------------------|
| <i>id</i> | Id of the maneuver to be deleted |
|-----------|----------------------------------|

**Exceptions**

|                                      |  |
|--------------------------------------|--|
| <i>DALRepositoryCommandException</i> | Thrown, when DeleteManeuverCommand can't be created! |
|--------------------------------------|--|

**5.60.3.4 Insert()**

```
void DataAccessLayer.Repositories.ManeuverRepository.Insert (
    Maneuver entity )
```

Checks if properties are ok. If so, creates a InsertManeuverCommand instance with the specified data.

**Parameters**

|               |                        |
|---------------|------------------------|
| <i>entity</i> | Maneuver to be created |
|---------------|------------------------|

**Exceptions**

|                                      |   |
|--------------------------------------|---|
| <i>DALRepositoryCommandException</i> | Thrown, when InsertManeuverCommand can't be created |
|--------------------------------------|---|

**5.60.3.5 Read()**

```
Maneuver DataAccessLayer.Repositories.ManeuverRepository.Read (
    int id )
```

Function for retrieving a maneuver with a specific id.

**Parameters**

|           |                            |
|-----------|----------------------------|
| <i>id</i> | Id of the wanted maneuver. |
|-----------|----------------------------|

**Returns**

Maneuver with the specified id.

**5.60.3.6 ReadAll()**

```
List<Maneuver> DataAccessLayer.Repositories.ManeuverRepository.ReadAll ( )
```

Function for retrieval of all available maneuvers.

**Returns**

All maneuvers in the maneuver table

**5.60.3.7 Update()**

```
void DataAccessLayer.Repositories.ManeuverRepository.Update (
    Maneuver entity )
```

Checks whether a maneuver with the same id as the parameter exists. If so, creates an UpdateManeuverCommand with the old and new state of the maneuver.



## Parameters

|               |                              |
|---------------|------------------------------|
| <i>entity</i> | The new state of a maneuver. |
|---------------|------------------------------|

## Exceptions

|                                      |   |
|--------------------------------------|---|
| <i>DALRepositoryCommandException</i> | Thrown, when UpdateManeuverCommand can't be created |
|--------------------------------------|---|

## 5.60.4 Member Data Documentation

### 5.60.4.1 commitCommands

`List<IDBCommand> DataAccessLayer.Repositories.ManeuverRepository.commitCommands [private]`

Commands to be committed to the database.

### 5.60.4.2 db

`IDBConnection DataAccessLayer.Repositories.ManeuverRepository.db [private]`

Database connection

### 5.60.4.3 logger

`log4net.ILog DataAccessLayer.Repositories.ManeuverRepository.logger [private]`

ILog object used for logging errors etc.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/ManeuverRepository.cs

## 5.61 SWE2\_Tourplanner\_Tests.DALTests.ManeuverRepositoryTests Class Reference

### Public Member Functions

- void **Setup** ()
- void **InsertCorrectCallsMock** ()
- void **UpdateCorrectCallsMock** ()
- void **DeleteCorrectCallsMock** ()
- void **ReadCorrectCallsMock** ()
- void **ReadAllCorrectCallsMock** ()

## Private Attributes

- Mock< [IDBConnection](#) > **mockDb**
- List< [IDBCommand](#) > **mockCommandList**
- [IManeuverRepository](#) **maneuverRepository**

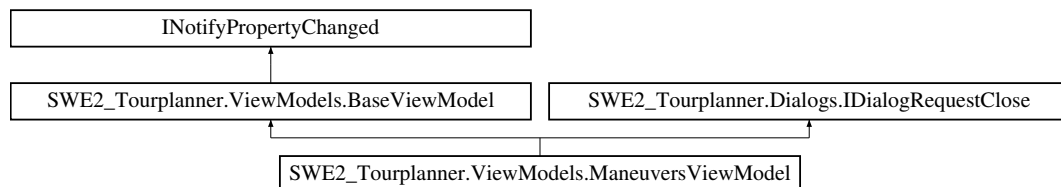
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/DALTests/ManeuverRepositoryTests.cs

## 5.62 SWE2\_Tourplanner.ViewModels.ManeuversViewModel Class Reference

[ManeuversViewModel](#) is used for managing Maneuver entities of a Tour

Inheritance diagram for SWE2\_Tourplanner.ViewModels.ManeuversViewModel:



## Public Member Functions

- [ManeuversViewModel](#) (List< [Maneuver](#) > maneuvers)  
*Default constructor of [ManeuversViewModel](#)*

## Properties

- ObservableCollection< [Maneuver](#) > [Maneuvers](#) [get, set]
- [Maneuver](#) [CurrentManeuver](#) [get, set]
- ICommand [AddManeuverCommand](#) [get]
- ICommand [RemoveManeuverCommand](#) [get]
- ICommand [ConfirmCommand](#) [get]
- ICommand [ExitCommand](#) [get]

## Events

- EventHandler< [DialogCloseRequestedEventArgs](#) > [CloseRequested](#)  
*Event used for closing the dialog*

## Private Attributes

- ObservableCollection< [Maneuver](#) > [maneuvers](#)  
*Maneuvers of the Tour*
- [Maneuver](#) [currentManeuver](#)  
*Currently selected Maneuver*

## Additional Inherited Members

### 5.62.1 Detailed Description

[ManeuversViewModel](#) is used for managing Maneuver entities of a Tour

### 5.62.2 Constructor & Destructor Documentation

#### 5.62.2.1 ManeuversViewModel()

```
SWE2_Tourplanner.ViewModels.ManeuversViewModel.ManeuversViewModel (
    List< Maneuver > maneuvers )
```

Default constructor of [ManeuversViewModel](#)

##### Parameters

|                  |                               |
|------------------|-------------------------------|
| <i>maneuvers</i> | Current maneuvers of the tour |
|------------------|-------------------------------|

### 5.62.3 Member Data Documentation

#### 5.62.3.1 currentManeuver

[Maneuver](#) SWE2\_Tourplanner.ViewModels.ManeuversViewModel.currentManeuver [private]

Currently selected Maneuver

#### 5.62.3.2 maneuvers

ObservableCollection<[Maneuver](#)> SWE2\_Tourplanner.ViewModels.ManeuversViewModel.maneuvers [private]

Maneuvers of the Tour

### 5.62.4 Property Documentation

#### 5.62.4.1 AddManeuverCommand

`ICommand SWE2_Tourplanner.ViewModels.ManeuversViewModel.AddManeuverCommand [get]`

Command for adding new maneuvers

#### 5.62.4.2 ConfirmCommand

`ICommand SWE2_Tourplanner.ViewModels.ManeuversViewModel.ConfirmCommand [get]`

Command for confirming the maneuvers of the tour

#### 5.62.4.3 CurrentManeuver

`Maneuver SWE2_Tourplanner.ViewModels.ManeuversViewModel.CurrentManeuver [get], [set]`

Currently selected Maneuver

#### 5.62.4.4 ExitCommand

`ICommand SWE2_Tourplanner.ViewModels.ManeuversViewModel.ExitCommand [get]`

Command for closing the dialog

#### 5.62.4.5 Maneuvers

`ObservableCollection<Maneuver> SWE2_Tourplanner.ViewModels.ManeuversViewModel.Maneuvers [get], [set]`

Maneuvers of the Tour

#### 5.62.4.6 RemoveManeuverCommand

`ICommand SWE2_Tourplanner.ViewModels.ManeuversViewModel.RemoveManeuverCommand [get]`

Command for removing currently selected maneuver

### 5.62.5 Event Documentation

### 5.62.5.1 CloseRequested

EventHandler<DialogCloseRequestedEventArgs> SWE2\_Tourplanner.ViewModels.ManeuversViewModel.↔  
CloseRequested

Event used for closing the dialog

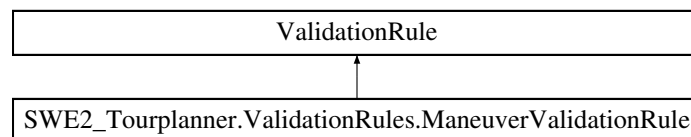
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/ManeuversViewModel.cs

## 5.63 SWE2\_Tourplanner.ValidationRules.ManeuverValidationRule Class Reference

[ManeuverValidationRule](#) is a ValidationRule for Maneuver entities inside a Datagrid.

Inheritance diagram for SWE2\_Tourplanner.ValidationRules.ManeuverValidationRule:



### Public Member Functions

- override ValidationResult [Validate](#) (object value, CultureInfo cultureInfo)  
*Validates whether a Maneuver has a Narrative and positive Distance or not.*

### 5.63.1 Detailed Description

[ManeuverValidationRule](#) is a ValidationRule for Maneuver entities inside a Datagrid.

### 5.63.2 Member Function Documentation

#### 5.63.2.1 Validate()

```

override ValidationResult SWE2_Tourplanner.ValidationRules.ManeuverValidationRule.Validate (
    object value,
    CultureInfo cultureInfo )
  
```

Validates whether a Maneuver has a Narrative and positive Distance or not.

## Parameters

|                    |                                     |
|--------------------|-------------------------------------|
| <i>value</i>       | Maneuver to be validated            |
| <i>cultureInfo</i> | CultureInfo used for the conversion |

## Returns

Returns valid result if Narrative isn't empty, null or only whitespace and if Distance isn't negative. Returns invalid result if those conditions aren't fulfilled

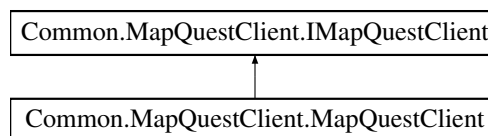
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ValidationRules/ManeuverValidationRule.cs

## 5.64 Common.MapQuestClient.MapQuestClient Class Reference

Concrete implementation of interface [IMapQuestClient](#)

Inheritance diagram for Common.MapQuestClient.MapQuestClient:



### Public Member Functions

- [MapQuestClient](#) ()  
*Default [MapQuestClient](#) constructor*
- async Task [GetRouteDataFromMapQuest](#) ([Tour](#) tour)  
*Retrieves data from the MapQuest API for a specific tour and sets the data in the tour.*
- async Task< string > [RetrieveRouteImage](#) (string storeId, string sessionId, string boundingBox)  
*[RetrieveRouteImage](#) retrieves the associated map image of a previous MapQuest API request.*

### Private Attributes

- HttpClient [client](#)  
*HttpClient used for sending requests and receiving responses from MapQuest API*
- [ITourPlannerConfig](#) [config](#)  
*TourPlannerConfig object that provides [MapQuestClient](#) with MapQuestKey and PictureDirectory values*
- ILog [logger](#)  
*ILog instance used for logging errors, warnings etc.*

## Static Private Attributes

- static string `routeBaseURL` = "http://www.mapquestapi.com/directions/v2/route"  
*Directions base URL used for retrieving tour data*
- static string `staticmapBaseURL` = "http://www.mapquestapi.com/staticmap/v5/map"  
*Staticmap base URL used for retrieving map pictures*
- static int `pictureWidth` = 1920  
*Width of map pictures*
- static int `pictureHeight` = 1080  
*Height of map pictures*

### 5.64.1 Detailed Description

Concrete implementation of interface [IMapQuestClient](#)

### 5.64.2 Constructor & Destructor Documentation

#### 5.64.2.1 MapQuestClient()

```
Common.MapQuestClient.MapQuestClient.MapQuestClient ( )
```

Default [MapQuestClient](#) constructor

### 5.64.3 Member Function Documentation

#### 5.64.3.1 GetRouteDataFromMapQuest()

```
async Task Common.MapQuestClient.MapQuestClient.GetRouteDataFromMapQuest (
    Tour tour )
```

Retrieves data from the MapQuest API for a specific tour and sets the data in the tour.

#### Parameters

|             |  |
|-------------|--|
| <i>tour</i> | Tour that needs data from the MapQuest API |
|-------------|--|

#### Returns

Task, where important data for the tour is retrieved from the MapQuest API and inserted into the tour.

**Exceptions**

|                             |  |
|-----------------------------|--|
| <i>ArgumentException</i>    | Thrown, when MapQuest can't create tour with passed parameters |
| <i>HttpRequestException</i> | Thrown, when the request doesn't have success status code      |

Implements [Common.MapQuestClient.IMapQuestClient](#).

**5.64.3.2 RetrieveRouteImage()**

```
async Task<string> Common.MapQuestClient.MapQuestClient.RetrieveRouteImage (
    string storeId,
    string sessionId,
    string boundingBox )
```

RetrieveRouteImage retrieves the associated map image of a previous MapQuest API request.

**Parameters**

|                    |   |
|--------------------|---|
| <i>storeId</i>     | Name of the route image file                          |
| <i>sessionId</i>   | SessionId of the previous request to the MapQuest API |
| <i>boundingBox</i> | BoundingBox of the map                                |

**Returns**

Task, where a route image is retrieved from MapQuest

**Exceptions**

|                             |   |
|-----------------------------|---|
| <i>ArgumentException</i>    | Thrown, when the MapQuest Request with parameters boundingBox, sessionId and size is not successful |
| <i>HttpRequestException</i> | Thrown, when the request doesn't have success status code   |

Implements [Common.MapQuestClient.IMapQuestClient](#).

**5.64.4 Member Data Documentation****5.64.4.1 client**

```
HttpClient Common.MapQuestClient.MapQuestClient.client [private]
```

HttpClient used for sending requests and receiving responses from MapQuest API



#### 5.64.4.2 config

`ITourPlannerConfig` `Common.MapQuestClient.MapQuestClient.config` [private]

TourPlannerConfig object that provides [MapQuestClient](#) with MapQuestKey and PictureDirectory values

#### 5.64.4.3 logger

`ILog` `Common.MapQuestClient.MapQuestClient.logger` [private]

ILog instance used for logging errors, warnings etc.

#### 5.64.4.4 pictureHeight

`int` `Common.MapQuestClient.MapQuestClient.pictureHeight` = 1080 [static], [private]

Height of map pictures

#### 5.64.4.5 pictureWidth

`int` `Common.MapQuestClient.MapQuestClient.pictureWidth` = 1920 [static], [private]

Width of map pictures

#### 5.64.4.6 routeBaseURL

`string` `Common.MapQuestClient.MapQuestClient.routeBaseURL` = "http://www.mapquestapi.com/directions/v2/route" [static], [private]

Directions base URL used for retrieving tour data

#### 5.64.4.7 staticmapBaseURL

`string` `Common.MapQuestClient.MapQuestClient.staticmapBaseURL` = "http://www.mapquestapi.com/staticmap/v5/map" [static], [private]

Staticmap base URL used for retrieving map pictures

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

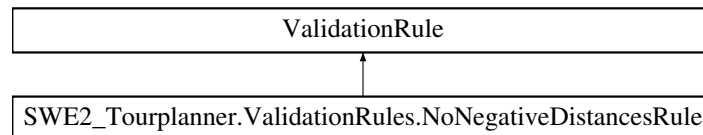
- SWE2-Tourplanner/SWE2-Tourplanner/Common/MapQuestClient/MapQuestClient.cs

## 5.65 SWE2\_Tourplanner.ValidationRules.NoNegativeDistancesRule

### Class Reference

[NoNegativeDistancesRule](#) is a [ValidationRule](#) for validating that double values used for distance-like properties aren't negative

Inheritance diagram for SWE2\_Tourplanner.ValidationRules.NoNegativeDistancesRule:



### Public Member Functions

- override ValidationResult [Validate](#) (object value, CultureInfo cultureInfo)

*Validates whether a distance (double) is negative or not.*

#### 5.65.1 Detailed Description

[NoNegativeDistancesRule](#) is a [ValidationRule](#) for validating that double values used for distance-like properties aren't negative

#### 5.65.2 Member Function Documentation

##### 5.65.2.1 Validate()

```

override ValidationResult SWE2_Tourplanner.ValidationRules.NoNegativeDistancesRule.Validate (
    object value,
    CultureInfo cultureInfo )
  
```

Validates whether a distance (double) is negative or not.

##### Parameters

|                    |                                 |
|--------------------|---------------------------------|
| <i>value</i>       | Distance to be validated        |
| <i>cultureInfo</i> | CultureInfo used for conversion |

##### Returns

Valid result if value is greater than or equal 0, invalid result if value is negative

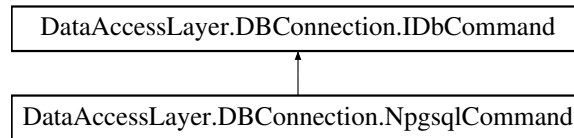
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ValidationRules/NoNegativeDistancesRule.cs

## 5.66 DataAccessLayer.DBConnection.NpgsqlCommand Class Reference

NpgsqlCommand is the command class for NPGSQL commands and it implements the functions defined in the [IDbCommand](#) interface.

Inheritance diagram for DataAccessLayer.DBConnection.NpgsqlCommand:



### Public Member Functions

- int [ExecuteNonQuery](#) ()  
*Executes the command (DML)*
- [IDataReader ExecuteReader](#) ()  
*Executes the query*
- [NpgsqlCommand](#) (string cmdText)  
*Creates an [NpgsqlCommand](#) with the specified text.*

### Properties

- DbParameterCollection [Parameters](#) [get]
- DbConnection [Connection](#) [get, set]

### Private Attributes

- Npgsql.NpgsqlCommand [nppgsqlCommand](#)  
*The actual command. All functions basically direct requests to this object.*

#### 5.66.1 Detailed Description

NpgsqlCommand is the command class for NPGSQL commands and it implements the functions defined in the [IDbCommand](#) interface.

#### 5.66.2 Constructor & Destructor Documentation

##### 5.66.2.1 NpgsqlCommand()

```
DataAccessLayer.DBConnection.NpgsqlCommand.NpgsqlCommand (
    string cmdText )
```

Creates an [NpgsqlCommand](#) with the specified text.

**Parameters**

|                |                      |
|----------------|----------------------|
| <i>cmdText</i> | Text of the command. |
|----------------|----------------------|

### 5.66.3 Member Function Documentation

#### 5.66.3.1 ExecuteNonQuery()

```
int DataAccessLayer.DBConnection.NpgsqlCommand.ExecuteNonQuery ( )
```

Executes the command (DML)

**Returns**

Amount of affected rows

Implements [DataAccessLayer.DBConnection.IDbCommand](#).

#### 5.66.3.2 ExecuteReader()

```
IDataReader DataAccessLayer.DBConnection.NpgsqlCommand.ExecuteReader ( )
```

Executes the query

**Returns**

DataReader for retrieval of results

Implements [DataAccessLayer.DBConnection.IDbCommand](#).

### 5.66.4 Member Data Documentation

#### 5.66.4.1 npgsqlCommand

```
Npgsql.NpgsqlCommand DataAccessLayer.DBConnection.NpgsqlCommand.npgsqlCommand [private]
```

The actual command. All functions basically direct requests to this object.

### 5.66.5 Property Documentation

#### 5.66.5.1 Connection

`DbConnection DataAccessLayer.DBConnection.NpgsqlCommand.Connection [get], [set]`

Connection to the database where the command gets executed.

#### 5.66.5.2 Parameters

`DbParameterCollection DataAccessLayer.DBConnection.NpgsqlCommand.Parameters [get]`

Parameters of the command.

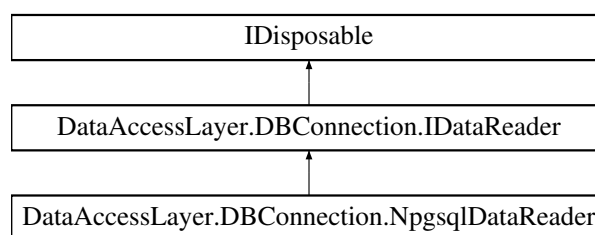
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/NpgsqlCommand.cs

## 5.67 DataAccessLayer.DBConnection.NpgsqlDataReader Class Reference

[NpgsqlDataReader](#) is the concrete [IDataReader](#) implementation for Npgsql package. All functions/properties direct requests to a [Npgsql.NpgsqlDataReader](#) object.

Inheritance diagram for `DataAccessLayer.DBConnection.NpgsqlDataReader`:



### Public Member Functions

- [NpgsqlDataReader](#) ([Npgsql.NpgsqlDataReader npgsqlDataReader](#))  
*Creates the [NpgsqlDataReader](#) object*
- bool [Read](#) ()  
*Checks if reading is possible or not*
- object [GetValue](#) (int i)  
*Retrieves the value stored in the specified field*
- void [Dispose](#) ()  
*Disposal of the [npgsqlDataReader](#) object.*

## Properties

- int [FieldCount](#) [get]

## Private Attributes

- Npgsql.NpgsqlDataReader [npgsqldatareader](#)

*The actual datareader. All requests are redirected to this object.*

### 5.67.1 Detailed Description

[NpgsqlDataReader](#) is the concrete [IDataReader](#) implementation for Npgsql package. All functions/properties direct requests to a `Npgsql.NpgsqlDataReader` object.

### 5.67.2 Constructor & Destructor Documentation

#### 5.67.2.1 NpgsqlDataReader()

```

DataAccessLayer.DBConnection.NpgsqlDataReader.NpgsqlDataReader (
    Npgsql.NpgsqlDataReader npgsqldatareader )

```

Creates the [NpgsqlDataReader](#) object

##### Parameters

|                         |                       |
|-------------------------|-----------------------|
| <i>npgsqldatareader</i> | The actual datareader |
|-------------------------|-----------------------|

### 5.67.3 Member Function Documentation

#### 5.67.3.1 Dispose()

```

void DataAccessLayer.DBConnection.NpgsqlDataReader.Dispose ( )

```

Disposal of the `npgsqldatareader` object.

#### 5.67.3.2 GetValue()

```

object DataAccessLayer.DBConnection.NpgsqlDataReader.GetValue (
    int i )

```

Retrieves the value stored in the specified field

**Parameters**

|          |              |
|----------|--------------|
| <i>i</i> | Field number |
|----------|--------------|

**Returns**

Value in the specified field

Implements [DataAccessLayer.DBConnection.IDataReader](#).

**5.67.3.3 Read()**

```
bool DataAccessLayer.DBConnection.NpgsqlDataReader.Read ( )
```

Checks if reading is possible or not

**Returns**

True if reading is possible, else returns false

Implements [DataAccessLayer.DBConnection.IDataReader](#).

**5.67.4 Member Data Documentation****5.67.4.1 npgsqlDataReader**

```
Npgsql.NpgsqlDataReader DataAccessLayer.DBConnection.NpgsqlDataReader.npgsqlDataReader [private]
```

The actual datareader. All requests are redirected to this object.

**5.67.5 Property Documentation****5.67.5.1 FieldCount**

```
int DataAccessLayer.DBConnection.NpgsqlDataReader.FieldCount [get]
```

Specifies how many fields the rows have

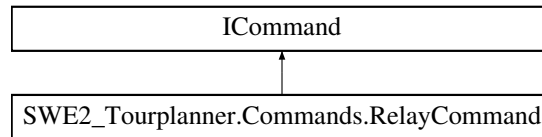
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBConnection/NpgsqlDataReader.cs

## 5.68 SWE2\_Tourplanner.Commands.RelayCommand Class Reference

[RelayCommand](#) is a concrete implementation of [ICommand](#) and allows us to create new commands, without having to create new concrete classes.

Inheritance diagram for [SWE2\\_Tourplanner.Commands.RelayCommand](#):



### Public Member Functions

- [RelayCommand](#) (Action< object > execute, Predicate< object > canExecute=null)  
*Default constructor of [RelayCommand](#).*
- bool [CanExecute](#) (object parameter)  
*Checks if command can be executed or not.*
- virtual void [Execute](#) (object parameter)  
*Executes the Action of the command.*

### Properties

- EventHandler [CanExecuteChanged](#)  
*EventHandler that manages Requery if CanExecute changes*

### Private Attributes

- Action< object > [\\_execute](#)  
*Action that gets executed by the command. Can be anonymous.*
- Predicate< object > [\\_canExecute](#)  
*Condition, whether the command can be executed or not.*

#### 5.68.1 Detailed Description

[RelayCommand](#) is a concrete implementation of [ICommand](#) and allows us to create new commands, without having to create new concrete classes.

#### 5.68.2 Constructor & Destructor Documentation

##### 5.68.2.1 RelayCommand()

```

SWE2_Tourplanner.Commands.RelayCommand.RelayCommand (
    Action< object > execute,
    Predicate< object > canExecute = null )
  
```

Default constructor of [RelayCommand](#).



## Parameters

|                   |  |
|-------------------|--|
| <i>execute</i>    | Action of the command                                |
| <i>canExecute</i> | Condition, whether a command can be executed or not. |

## 5.68.3 Member Function Documentation

### 5.68.3.1 CanExecute()

```
bool SWE2_Tourplanner.Commands.RelayCommand.CanExecute (
    object parameter )
```

Checks if command can be executed or not.

## Parameters

|                  |                             |
|------------------|-----------------------------|
| <i>parameter</i> | Parameter for the condition |
|------------------|-----------------------------|

## Returns

True, if `_canExecute` is null or if condition is fulfilled, otherwise false

### 5.68.3.2 Execute()

```
virtual void SWE2_Tourplanner.Commands.RelayCommand.Execute (
    object parameter ) [virtual]
```

Executes the Action of the command.

## Parameters

|                  |                          |
|------------------|--------------------------|
| <i>parameter</i> | Parameter for the action |
|------------------|--------------------------|

## 5.68.4 Member Data Documentation

### 5.68.4.1 \_canExecute

```
Predicate<object> SWE2_Tourplanner.Commands.RelayCommand._canExecute [private]
```

Condition, whether the command can be executed or not.

#### 5.68.4.2 `_execute`

`Action<object> SWE2_Tourplanner.Commands.RelayCommand._execute [private]`

Action that gets executed by the command. Can be anonymous.

### 5.68.5 Property Documentation

#### 5.68.5.1 `CanExecuteChanged`

`EventHandler SWE2_Tourplanner.Commands.RelayCommand.CanExecuteChanged [add], [remove]`

EventHandler that manages Requery if CanExecute changes

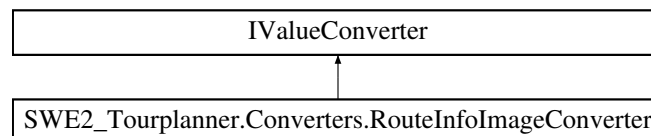
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Commands/RelayCommand.cs

## 5.69 SWE2\_Tourplanner.Converters.RouteInfoImageConverter Class Reference

[RouteInfoImageConverter](#) is used for string - BitmapImage conversion.

Inheritance diagram for SWE2\_Tourplanner.Converters.RouteInfoImageConverter:



### Public Member Functions

- object [Convert](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Converts a path (string) to a BitmapImage*
- object [ConvertBack](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Converts BitmapImage back to Path (string)*

## 5.69.1 Detailed Description

[RouteInfoImageConverter](#) is a used for string - BitmapImage conversion.

## 5.69.2 Member Function Documentation

### 5.69.2.1 Convert()

```
object SWE2_Tourplanner.Converters.RouteInfoImageConverter.Convert (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture )
```

Converts a path (string) to a BitmapImage

#### Parameters

|                   |   |
|-------------------|---|
| <i>value</i>      | The value that needs to be converted    |
| <i>targetType</i> | TargetType for the conversion           |
| <i>parameter</i>  | Additional parameter for the conversion |
| <i>culture</i>    | Culture settings for the conversion     |

#### Returns

Already loaded BitmapImage for Image control. If the path doesn't exist, return empty BitmapImage

### 5.69.2.2 ConvertBack()

```
object SWE2_Tourplanner.Converters.RouteInfoImageConverter.ConvertBack (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture )
```

Converts BitmapImage back to Path (string)

#### Parameters

|                   |   |
|-------------------|---|
| <i>value</i>      | The value that needs to be converted    |
| <i>targetType</i> | TargetType for the conversion           |
| <i>parameter</i>  | Additional parameter for the conversion |
| <i>culture</i>    | Culture settings for the conversion     |

**Returns**

Converted object. In this case Binding.DoNothing is returned because a image can't really be converted back to a path.

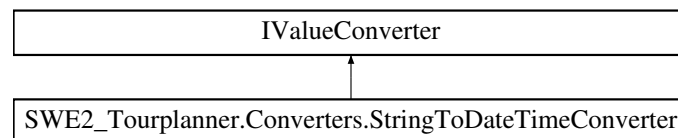
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Converters/RouteInfolImageConverter.cs

## 5.70 SWE2\_Tourplanner.Converters.StringToDateTimeConverter Class Reference

[StringToDateTimeConverter](#) is used for string - DateTime conversion.

Inheritance diagram for SWE2\_Tourplanner.Converters.StringToDateTimeConverter:

**Public Member Functions**

- object [Convert](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Converts DateTime to a string*
- object [ConvertBack](#) (object value, Type targetType, object parameter, CultureInfo culture)  
*Converts a string to a DateTime instance*

**5.70.1 Detailed Description**

[StringToDateTimeConverter](#) is used for string - DateTime conversion.

**5.70.2 Member Function Documentation****5.70.2.1 Convert()**

```

object SWE2_Tourplanner.Converters.StringToDateTimeConverter.Convert (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture )
  
```

Converts DateTime to a string

## Parameters

|                   |   |
|-------------------|---|
| <i>value</i>      | The value that needs to be converted    |
| <i>targetType</i> | TargetType for the conversion           |
| <i>parameter</i>  | Additional parameter for the conversion |
| <i>culture</i>    | Culture settings for the conversion     |

## Returns

Converted DateTime string

### 5.70.2.2 ConvertBack()

```
object SWE2_Tourplanner.Converters.StringToDateTimeConverter.ConvertBack (
    object value,
    Type targetType,
    object parameter,
    CultureInfo culture )
```

Converts a string to a DateTime instance

## Parameters

|                   |   |
|-------------------|---|
| <i>value</i>      | The value that needs to be converted    |
| <i>targetType</i> | TargetType for the conversion           |
| <i>parameter</i>  | Additional parameter for the conversion |
| <i>culture</i>    | Culture settings for the conversion     |

## Returns

Converted object. If string can't be parsed, return new DateTime instance

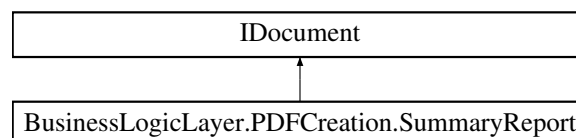
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/Converters/StringToDateTimeConverter.cs

## 5.71 BusinessLogicLayer.PDFCreation.SummaryReport Class Reference

[SummaryReport](#) defines the generation process of Tour Summary Reports as PDF

Inheritance diagram for BusinessLogicLayer.PDFCreation.SummaryReport:



## Public Member Functions

- [SummaryReport](#) (List< [Tour](#) > tours)  
*Default constructor of [SummaryReport](#)*
- void [Compose](#) (IContainer container)  
*Compose defines the base structure of [SummaryReport](#)*
- DocumentMetadata [GetMetadata](#) ()  
*Document Metadata of [SummaryReport](#)*

## Properties

- List< [Tour](#) > [Tours](#) [get]

## Private Member Functions

- void [ComposeHeader](#) (IContainer container)  
*ComposedHeader defines the composition of [SummaryReport](#) headers*
- void [ComposeContent](#) (IContainer container)  
*ComposeContent assembles [ComposeTourStats](#), [ComposeManeuverStats](#) and [ComposeTourLogStats](#) into the main section of [SummaryReport](#)*
- void [ComposeTourStats](#) (IContainer container)  
*ComposeTourStats defines how stats about [Tour](#) entities will be presented in the report*
- void [ComposeManeuverStats](#) (IContainer container)  
*ComposerManeuverStats defines the way how stats about [Maneuver](#) entities will be presented*
- void [ComposeTourLogStats](#) (IContainer container)  
*ComposeTourLogStats defines the way stats about [TourLog](#) entities of the passed [Tours](#) will look like*

### 5.71.1 Detailed Description

[SummaryReport](#) defines the generation process of Tour Summary Reports as PDF

### 5.71.2 Constructor & Destructor Documentation

#### 5.71.2.1 SummaryReport()

```
BusinessLogicLayer.PDFCreation.SummaryReport.SummaryReport (
    List< Tour > tours )
```

Default constructor of [SummaryReport](#)

#### Parameters

|              |   |
|--------------|---|
| <i>tours</i> | Tours that should be included in the summary report |
|--------------|---|

### 5.71.3 Member Function Documentation

#### 5.71.3.1 Compose()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.Compose (
    IContainer container )
```

Compose defines the base structure of [SummaryReport](#)

##### Parameters

|                  |   |
|------------------|---|
| <i>container</i> | Container in which the content will be placed |
|------------------|---|

#### 5.71.3.2 ComposeContent()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.ComposeContent (
    IContainer container ) [private]
```

ComposeContent assembles ComposeTourStats, ComposeManeuverStats and ComposeTourLogStats into the main section of [SummaryReport](#)

##### Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Container in which the main content will be placed |
|------------------|--|

#### 5.71.3.3 ComposeHeader()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.ComposeHeader (
    IContainer container ) [private]
```

ComposedHeader defines the composition of [SummaryReport](#) headers

##### Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Container in which the header will be placed |
|------------------|--|

#### 5.71.3.4 ComposeManeuverStats()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.ComposeManeuverStats (
    IContainer container ) [private]
```

ComposerManeuverStats defines the way how stats about Maneuver entities will be presented

#### Parameters

|                  |   |
|------------------|---|
| <i>container</i> | Container in which stats about Maneuver entities will be placed |
|------------------|---|

### 5.71.3.5 ComposeTourLogStats()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.ComposeTourLogStats (
    IContainer container ) [private]
```

ComposeTourLogStats defines the way stats about TourLog entities of the passed Tours will look like

#### Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Container in which stats about TourLog entities will be placed |
|------------------|--|

### 5.71.3.6 ComposeTourStats()

```
void BusinessLogicLayer.PDFCreation.SummaryReport.ComposeTourStats (
    IContainer container ) [private]
```

ComposeTourStats defines how stats about Tour entities will be presented in the report

#### Parameters

|                  |   |
|------------------|---|
| <i>container</i> | Container in which stats about Tour entities are placed |
|------------------|---|

### 5.71.3.7 GetMetadata()

```
DocumentMetadata BusinessLogicLayer.PDFCreation.SummaryReport.GetMetadata ( )
```

Document Metadata of [SummaryReport](#)

#### Returns

Metadata of the PDF file

## 5.71.4 Property Documentation



#### 5.71.4.1 Tours

```
List<Tour> BusinessLogicLayer.PDFCreation.SummaryReport.Tours [get]
```

Tours included in the summary report

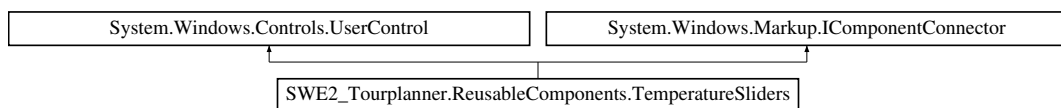
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/PDFCreation/SummaryReport.cs

## 5.72 SWE2\_Tourplanner.ReusableComponents.TemperatureSliders Class Reference

### TemperatureSliders

Inheritance diagram for SWE2\_Tourplanner.ReusableComponents.TemperatureSliders:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

### Private Attributes

- bool **\_contentLoaded**

#### 5.72.1 Detailed Description

### TemperatureSliders

#### 5.72.2 Member Function Documentation

### 5.72.2.1 InitializeComponent()

```
void SWE2_Tourplanner.ReusableComponents.TemperatureSliders.InitializeComponent ( )
```

InitializeComponent

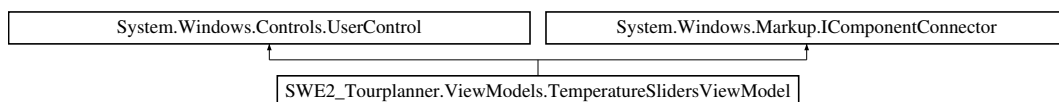
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ReusableComponents/TemperatureSliders.g.i.cs

## 5.73 SWE2\_Tourplanner.ViewModels.TemperatureSlidersViewModel Class Reference

[TemperatureSlidersViewModel](#)

Inheritance diagram for SWE2\_Tourplanner.ViewModels.TemperatureSlidersViewModel:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

### Private Attributes

- bool **\_contentLoaded**

### 5.73.1 Detailed Description

[TemperatureSlidersViewModel](#)

### 5.73.2 Member Function Documentation

### 5.73.2.1 InitializeComponent()

```
void SWE2_Tourplanner.ViewModels.TemperatureSlidersViewModel.InitializeComponent ( )
```

InitializeComponent

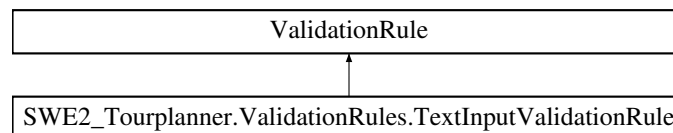
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/ViewModels/TemperatureSlidersViewModel.g.i.cs

## 5.74 SWE2\_Tourplanner.ValidationRules.TextInputValidationRule Class Reference

[TextInputValidationRule](#) is a ValidationRule for the length of text input

Inheritance diagram for SWE2\_Tourplanner.ValidationRules.TextInputValidationRule:



### Public Member Functions

- override ValidationResult [Validate](#) (object value, CultureInfo cultureInfo)  
*Validates whether a string surpasses Min character count and is lower than Max character count.*

### Properties

- int [Min](#) [get, set]
- int [Max](#) [get, set]

### 5.74.1 Detailed Description

[TextInputValidationRule](#) is a ValidationRule for the length of text input

### 5.74.2 Member Function Documentation

#### 5.74.2.1 Validate()

```
override ValidationResult SWE2_Tourplanner.ValidationRules.TextInputValidationRule.Validate (
    object value,
    CultureInfo cultureInfo )
```

Validates whether a string surpasses Min character count and is lower than Max character count.

## Parameters

|                    |                                 |
|--------------------|---------------------------------|
| <i>value</i>       | Text to be validated            |
| <i>cultureInfo</i> | CultureInfo used for conversion |

## Returns

Valid result if text doesn't violate bounds, otherwise invalid result is returned

### 5.74.3 Property Documentation

#### 5.74.3.1 Max

```
int SWE2_Tourplanner.ValidationRules.TextInputValidationRule.Max [get], [set]
```

Maximum character count of text

#### 5.74.3.2 Min

```
int SWE2_Tourplanner.ValidationRules.TextInputValidationRule.Min [get], [set]
```

Minimum character count of text

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ValidationRules/TextInputValidationRule.cs

## 5.75 Common.Entities.Tour Class Reference

The tour is the most important entity of the project. It represents a route.

### Properties

- int **Id** [get, set]  
*Id of the tour.*
- string **Name** [get, set]  
*Name of the tour. Is unique.*
- string **StartLocation** [get, set]  
*Location where the tour is supposed to start.*
- string **EndLocation** [get, set]  
*Location where the tour is supposed to end.*
- string **RouteInfo** [get, set]  
*Path to the route image.*
- double **Distance** [get, set]  
*Overall length of the tour.*
- **ERouteType RouteType** [get, set]  
*Type of the route. Is an important parameter that decides how the route is constructed by MapQuest.*
- string **Description** [get, set]  
*Description of the tour.*
- List< **Maneuver** > **Maneuvers** [get, set]  
*All associated maneuvers of the tour.*
- List< **TourLog** > **TourLogs** [get, set]  
*All associated tour logs of the tour.*

### 5.75.1 Detailed Description

The tour is the most important entity of the project. It represents a route.

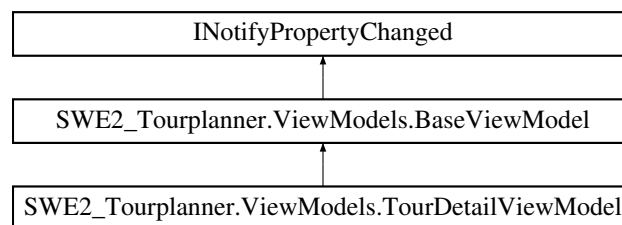
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Entities/Tour.cs

## 5.76 SWE2\_Tourplanner.ViewModels.TourDetailViewModel Class Reference

[TourDetailViewModel](#) provides a Tour and commands for managing its TourLogs

Inheritance diagram for SWE2\_Tourplanner.ViewModels.TourDetailViewModel:



### Public Member Functions

- [TourDetailViewModel](#) ([IDialogService](#) dialogService)

*Default constructor of [TourDetailViewModel](#)*

### Properties

- ObservableCollection< [TourLog](#) > [CurrentTourLogs](#) [get, set]
- [Tour?](#) [SelectedTour](#) [get, set]
- [TourLog](#) [SelectedTourLog](#) [get, set]
- ICommand [AddLogCommand](#) [get]
- ICommand [RemoveLogCommand](#) [get]
- ICommand [EditLogCommand](#) [get]

### Private Member Functions

- void [ShowException](#) (Exception e)

*Helper method for displayin exceptions / errors*

## Private Attributes

- [Tour \\_selectedTour](#)  
*Currently selected Tour*
- [TourLog selectedTourLog](#)  
*Currently selected TourLog*
- [ObservableCollection< TourLog > currentTourLogs](#)  
*TourLogs of the currently selected tour*
- [ITourLogFactory tourLogFactory](#)  
*Object used for creating, updating and deleting tourlogs*
- readonly [IDialogService dialogService](#)  
*Service used for opening TourLog related [Dialogs](#)*

## Additional Inherited Members

### 5.76.1 Detailed Description

[TourDetailViewModel](#) provides a Tour and commands for managing its TourLogs

### 5.76.2 Constructor & Destructor Documentation

#### 5.76.2.1 TourDetailViewModel()

```
SWE2_Tourplanner.ViewModels.TourDetailViewModel.TourDetailViewModel (
    IDialogService dialogService )
```

Default constructor of [TourDetailViewModel](#)

#### Parameters

|                      |  |
|----------------------|--|
| <i>dialogService</i> | Service used for opening TourLog related <a href="#">Dialogs</a> |
|----------------------|--|

### 5.76.3 Member Function Documentation

#### 5.76.3.1 ShowException()

```
void SWE2_Tourplanner.ViewModels.TourDetailViewModel.ShowException (
    Exception e ) [private]
```

Helper method for displayin exceptions / errors

## Parameters

|          |                  |
|----------|------------------|
| <i>e</i> | Thrown exception |
|----------|------------------|

## 5.76.4 Member Data Documentation

### 5.76.4.1 `_selectedTour`

[Tour](#) SWE2\_Tourplanner.ViewModels.TourDetailViewModel.\_selectedTour [private]

Currently selected Tour

### 5.76.4.2 `currentTourLogs`

ObservableCollection<[TourLog](#)> SWE2\_Tourplanner.ViewModels.TourDetailViewModel.currentTourLogs [private]

TourLogs of the currently selected tour

### 5.76.4.3 `dialogService`

readonly [IDialogService](#) SWE2\_Tourplanner.ViewModels.TourDetailViewModel.dialogService [private]

Service used for opening TourLog related [Dialogs](#)

### 5.76.4.4 `selectedTourLog`

[TourLog](#) SWE2\_Tourplanner.ViewModels.TourDetailViewModel.selectedTourLog [private]

Currently selected TourLog

### 5.76.4.5 `tourLogFactory`

[ITourLogFactory](#) SWE2\_Tourplanner.ViewModels.TourDetailViewModel.tourLogFactory [private]

Object used for creating, updating and deleting tourlogs

## 5.76.5 Property Documentation

### 5.76.5.1 AddLogCommand

`ICommand SWE2_Tourplanner.ViewModels.TourDetailViewModel.AddLogCommand [get]`

Command for adding new TourLogs to the currently selected tour

### 5.76.5.2 CurrentTourLogs

`ObservableCollection<TourLog> SWE2_Tourplanner.ViewModels.TourDetailViewModel.CurrentTourLogs [get], [set]`

TourLogs of the currently selected tour

### 5.76.5.3 EditLogCommand

`ICommand SWE2_Tourplanner.ViewModels.TourDetailViewModel.EditLogCommand [get]`

Command for updating currently selected tourlog

### 5.76.5.4 RemoveLogCommand

`ICommand SWE2_Tourplanner.ViewModels.TourDetailViewModel.RemoveLogCommand [get]`

Command for deleting TourLogs from the currently selected tour

### 5.76.5.5 SelectedTour

`Tour? SWE2_Tourplanner.ViewModels.TourDetailViewModel.SelectedTour [get], [set]`

Currently selected tour

### 5.76.5.6 SelectedTourLog

`TourLog SWE2_Tourplanner.ViewModels.TourDetailViewModel.SelectedTourLog [get], [set]`

Currently selected tourlog

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

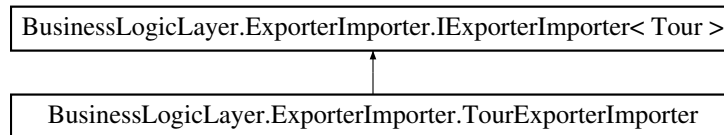
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/TourDetailViewModel.cs



## 5.77 BusinessLogicLayer.ExporterImporter.TourExporterImporter Class Reference

Concrete implementation of [IExporterImporter](#) for Tour entities

Inheritance diagram for BusinessLogicLayer.ExporterImporter.TourExporterImporter:



### Public Member Functions

- [TourExporterImporter](#) ()  
*Default constructor of [TourExporterImporter](#)*
- Task [Export](#) (List< [Tour](#) > tours)  
*Method for exporting tours*
- Task< List< [Tour](#) > > [Import](#) (string filePath)  
*Method for importing tours*

### Private Member Functions

- void [ExportPicture](#) (string originalLocation, string exportLocation)  
*Helper method for exporting pictures*

### Private Attributes

- [ITourPlannerConfig](#) [tourPlannerConfig](#)  
*TourPlannerConfig object, which provides the class with data about ExportsDirectory and PictureDirectory*
- [ILog](#) [logger](#)  
*ILog instance used for logging errors, warnings etc.*

#### 5.77.1 Detailed Description

Concrete implementation of [IExporterImporter](#) for Tour entities

#### 5.77.2 Constructor & Destructor Documentation

##### 5.77.2.1 TourExporterImporter()

`BusinessLogicLayer.ExporterImporter.TourExporterImporter.TourExporterImporter ( )`

Default constructor of [TourExporterImporter](#)

### 5.77.3 Member Function Documentation

#### 5.77.3.1 Export()

```
Task BusinessLogicLayer.ExporterImporter.TourExporterImporter.Export (
    List< Tour > tours )
```

Method for exporting tours

##### Parameters

|              |                      |
|--------------|----------------------|
| <i>tours</i> | Tours to be exported |
|--------------|----------------------|

##### Returns

Task, which exports the passed tours

##### Exceptions

|                                    |   |
|------------------------------------|---|
| <i>BLExporterImporterException</i> | Thrown, when errors (especially file / path related errors) occur during export |
|------------------------------------|---|

#### 5.77.3.2 ExportPicture()

```
void BusinessLogicLayer.ExporterImporter.TourExporterImporter.ExportPicture (
    string originalLocation,
    string exportLocation ) [private]
```

Helper method for exporting pictures

##### Parameters

|                         |                         |
|-------------------------|-------------------------|
| <i>originalLocation</i> | Path to the picture     |
| <i>exportLocation</i>   | New path of the picture |

#### 5.77.3.3 Import()

```
Task<List<Tour> > BusinessLogicLayer.ExporterImporter.TourExporterImporter.Import (
    string filePath )
```

Method for importing tours

## Parameters

|                 |                  |
|-----------------|------------------|
| <i>filePath</i> | Path to the file |
|-----------------|------------------|

## Returns

Task, which saves tours to the datastore and returns them as a list

## Exceptions

|                                    |  |
|------------------------------------|--|
| <i>BLExporterImporterException</i> | Thrown, when DAL or file (format, access, etc.) errors occur during import |
|------------------------------------|--|

## 5.77.4 Member Data Documentation

### 5.77.4.1 logger

```
ILog BusinessLogicLayer.ExporterImporter.TourExporterImporter.logger [private]
```

ILog instance used for logging errors, warnings etc.

### 5.77.4.2 tourPlannerConfig

```
ITourPlannerConfig BusinessLogicLayer.ExporterImporter.TourExporterImporter.tourPlannerConfig [private]
```

TourPlannerConfig object, which provides the class with data about ExportsDirectory and PictureDirectory

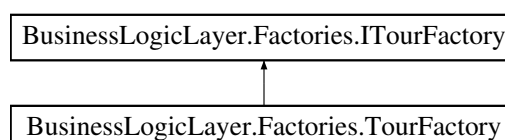
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/ExporterImporter/TourExporterImporter.cs

## 5.78 BusinessLogicLayer.Factories.TourFactory Class Reference

[TourFactory](#) is a concrete implementation of [ITourFactory](#).

Inheritance diagram for BusinessLogicLayer.Factories.TourFactory:



## Public Member Functions

- [TourFactory](#) ()  
*Default constructor of [TourFactory](#)*
- **TourFactory** (IUnitOfWork uow)
- async Task [CreateMapQuestTour](#) ([Tour](#) tour)  
*CreateMapQuestTour is a method for creating new Tours with data from MapQuest*
- async Task [CreateTour](#) ([Tour](#) tour)  
*CreateTour is a method for inserting new Tour entities in the datastore*
- Task [DeleteTour](#) ([Tour](#) tour)  
*DeleteTour is a method for deleting Tour entities in the datastore*
- async Task [UpdateMapQuestTour](#) ([Tour](#) tour)  
*UpdateMapQuestTour is a method for updating a MapQuest Tour*
- Task [UpdateTour](#) ([Tour](#) tour)  
*UpdateTour is a method for updating Tour entities in the datastore*
- List< [Tour](#) > [GetTours](#) ()  
*GetTours is a method for retrieving Tour entities from the datastore*

## Private Member Functions

- async Task [RetrieveMapQuestData](#) ([Tour](#) tour)  
*RetrieveMapQuestData is a helper function for retrieving MapQuest data using the MapQuestClient*

## Private Attributes

- [IMapQuestClient](#) mapQuestClient  
*IMapQuestClient instance used for retrieving data from MapQuest*
- [ILog](#) logger  
*ILog instance used for logging errors, warnings etc.*
- [IUnitOfWork](#) uow  
*IUnitOfWork instance used for accessing the datastore*

### 5.78.1 Detailed Description

[TourFactory](#) is a concrete implementation of [ITourFactory](#).

### 5.78.2 Constructor & Destructor Documentation

#### 5.78.2.1 TourFactory()

```
BusinessLogicLayer.Factories.TourFactory.TourFactory ( )
```

Default constructor of [TourFactory](#)

## 5.78.3 Member Function Documentation

### 5.78.3.1 CreateMapQuestTour()

```
async Task BusinessLogicLayer.Factories.TourFactory.CreateMapQuestTour (
    Tour tour )
```

CreateMapQuestTour is a method for creating new Tours with data from MapQuest

#### Parameters

|             |                    |
|-------------|--------------------|
| <i>tour</i> | Tour to be created |
|-------------|--------------------|

#### Returns

Task, which retrieves data from MapQuest API and creates a Tour

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

### 5.78.3.2 CreateTour()

```
async Task BusinessLogicLayer.Factories.TourFactory.CreateTour (
    Tour tour )
```

CreateTour is a method for inserting new Tour entities in the datastore

#### Parameters

|             |                                     |
|-------------|-------------------------------------|
| <i>tour</i> | Tour that is supposed to be created |
|-------------|-------------------------------------|

#### Returns

Task, which inserts a new Tour entity in the datastore

#### Exceptions

|                           |   |
|---------------------------|---|
| <i>BLFactoryException</i> | Thrown, when the map image doesn't exist or if DAL exceptions occur |
|---------------------------|---|

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

### 5.78.3.3 DeleteTour()

```
Task BusinessLogicLayer.Factories.TourFactory.DeleteTour (
    Tour tour )
```

DeleteTour is a method for deleting Tour entities in the datastore

#### Parameters

|             |                                     |
|-------------|-------------------------------------|
| <i>tour</i> | Tour that is supposed to be deleted |
|-------------|-------------------------------------|

#### Returns

Task, which deletes a Tour in the datastore

#### Exceptions

|                           |  |
|---------------------------|--|
| <i>BLFactoryException</i> | Thrown, when DAL exceptions occur or if there are problems whilst deleting the map image |
|---------------------------|--|

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

### 5.78.3.4 GetTours()

```
List<Tour> BusinessLogicLayer.Factories.TourFactory.GetTours ( )
```

GetTours is a method for retrieving Tour entities from the datastore

#### Returns

All tours of the data store

#### Exceptions

|                           |  |
|---------------------------|--|
| <i>BLFactoryException</i> | Thrown, if there are DAL exceptions or if other errors occur |
|---------------------------|--|

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

### 5.78.3.5 RetrieveMapQuestData()

```
async Task BusinessLogicLayer.Factories.TourFactory.RetrieveMapQuestData (
    Tour tour ) [private]
```

RetrieveMapQuestData is a helper function for retrieving MapQuest data using the MapQuestClient

## Parameters

|             |                               |
|-------------|-------------------------------|
| <i>tour</i> | Tour that needs MapQuest data |
|-------------|-------------------------------|

## Returns

Task, which retrieves data from MapQuest

## Exceptions

|                           |  |
|---------------------------|--|
| <i>BLFactoryException</i> | Thrown, when the MapQuest request was not successful or if MapQuest could not generate a route |
|---------------------------|--|

### 5.78.3.6 UpdateMapQuestTour()

```
async Task BusinessLogicLayer.Factories.TourFactory.UpdateMapQuestTour (
    Tour tour )
```

UpdateMapQuestTour is a method for updating a MapQuest Tour

## Parameters

|             |              |
|-------------|--------------|
| <i>tour</i> | Updated tour |
|-------------|--------------|

## Returns

Task, which retrieves data from MapQuest API and updates Tour entity in datastore

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

### 5.78.3.7 UpdateTour()

```
Task BusinessLogicLayer.Factories.TourFactory.UpdateTour (
    Tour tour )
```

UpdateTour is a method for updating Tour entities in the datastore

## Parameters

|             |                   |
|-------------|-------------------|
| <i>tour</i> | Updated tour data |
|-------------|-------------------|

**Returns**

Task, which updates a Tour entity

**Exceptions**

|                           |   |
|---------------------------|---|
| <i>BLFactoryException</i> | Thrown, when there are DAL exceptions or problems copying / deleting map images |
|---------------------------|---|

Implements [BusinessLogicLayer.Factories.ITourFactory](#).

## 5.78.4 Member Data Documentation

### 5.78.4.1 logger

```
ILog BusinessLogicLayer.Factories.TourFactory.logger [private]
```

ILog instance used for logging errors, warnings etc.

### 5.78.4.2 mapQuestClient

```
IMapQuestClient BusinessLogicLayer.Factories.TourFactory.mapQuestClient [private]
```

IMapQuestClient instance used for retrieving data from MapQuest

### 5.78.4.3 uow

```
IUnitOfWork BusinessLogicLayer.Factories.TourFactory.uow [private]
```

IUnitOfWork instance used for accessing the datastore

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

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Factories/TourFactory.cs

## 5.79 SWE2\_Tourplanner\_Tests.BLTests.TourFactoryTests Class Reference

### Public Member Functions

- void **SetUp** ()
- async Task **CreateTourTest** ()
- async Task **UpdateTourTest** ()
- async Task **DeleteTourTest** ()



## Private Attributes

- Mock< [IUnitOfWork](#) > **mockUOW**
- Mock< [ITourLogRepository](#) > **mockTourLogRepository**
- Mock< [ITourRepository](#) > **mockTourRepository**
- Mock< [IManeuverRepository](#) > **mockManeuverRepository**
- [Tour](#) **testTour**

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/BLTests/TourFactoryTests.cs

## 5.80 Common.Entities.TourLog Class Reference

The tourlog is an entry from a person, who participated in a tour. It's basically a short review of the tour.

### Properties

- int [Id](#) [get, set]  
*Id of the tour log.*
- int [TourId](#) [get, set]  
*Id of the associated tour.*
- DateTime [StartDate](#) [get, set]  
*Start time and date of the tour log.*
- DateTime [EndDate](#) [get, set]  
*End time and date of the tour log.*
- double [Distance](#) [get, set]  
*Distance travelled by user.*
- double [TotalTime](#) [get, set]  
*Total time needed.*
- double [Temperature](#) [get, set]  
*Temperature during the tour.*
- double [Rating](#) [get, set]  
*Rating of the tour.*
- double [AverageSpeed](#) [get, set]  
*Average speed of the user.*
- [EWeather](#) [Weather](#) [get, set]  
*Weather during the tour.*
- [ETravelMethod](#) [TravelMethod](#) [get, set]  
*Travel method of the user.*
- string [Report](#) [get, set]  
*Short comment on the tour.*

### 5.80.1 Detailed Description

The tourlog is an entry from a person, who participated in a tour. It's basically a short review of the tour.

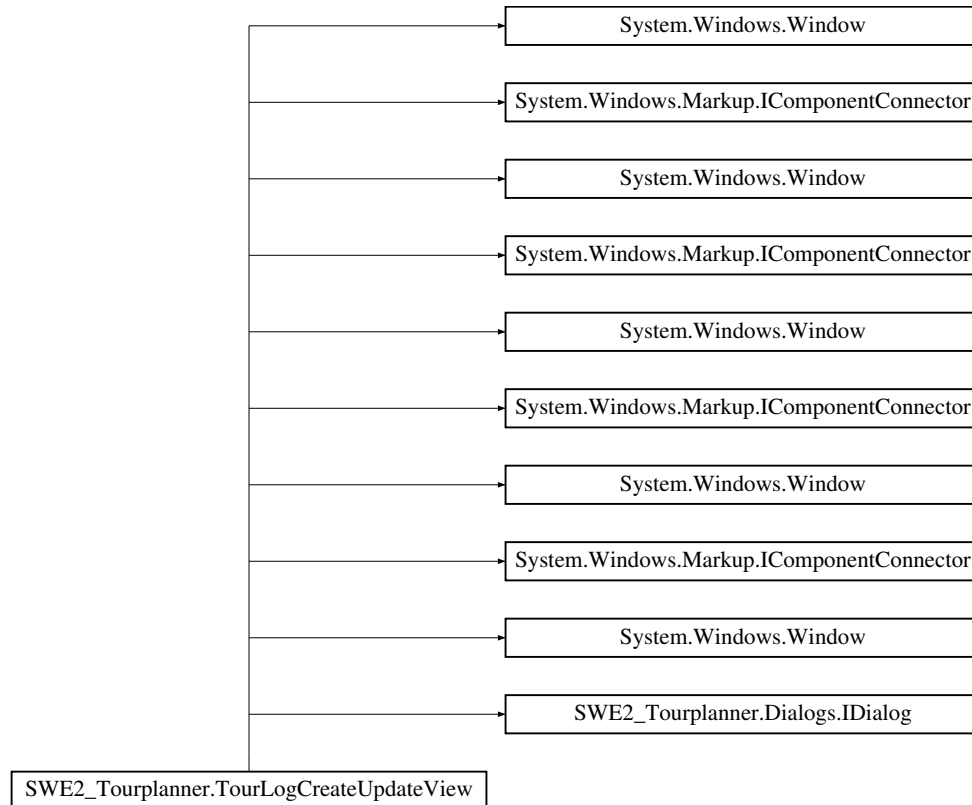
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Entities/TourLog.cs

## 5.81 SWE2\_Tourplanner.TourLogCreateUpdateView Class Reference

### [TourLogCreateUpdateView](#)

Inheritance diagram for SWE2\_Tourplanner.TourLogCreateUpdateView:



### Public Member Functions

- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*
- void [InitializeComponent](#) ()  
*InitializeComponent*

### Private Member Functions

- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)
- void System.Windows.Markup.IComponentConnector. **Connect** (int connectionId, object target)

## Private Attributes

- `bool _contentLoaded`

## Additional Inherited Members

### 5.81.1 Detailed Description

[TourLogCreateUpdateView](#)

Interaction logic for [TourLogCreateUpdateView](#)

### 5.81.2 Member Function Documentation

#### 5.81.2.1 InitializeComponent() [1/4]

```
void SWE2_Tourplanner.TourLogCreateUpdateView.InitializeComponent ( )
```

InitializeComponent

#### 5.81.2.2 InitializeComponent() [2/4]

```
void SWE2_Tourplanner.TourLogCreateUpdateView.InitializeComponent ( )
```

InitializeComponent

#### 5.81.2.3 InitializeComponent() [3/4]

```
void SWE2_Tourplanner.TourLogCreateUpdateView.InitializeComponent ( )
```

InitializeComponent

### 5.81.2.4 InitializeComponent() [4/4]

```
void SWE2_Tourplanner.TourLogCreateUpdateView.InitializeComponent ( )
```

InitializeComponent

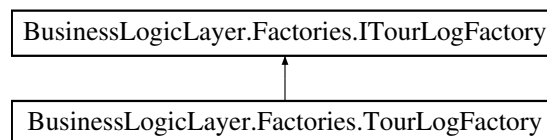
The documentation for this class was generated from the following files:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/TourLogCreateUpdateView.g.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/obj/Debug/net5.0-windows/TourLogCreateUpdateView.g.i.cs
- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/TourLogCreateUpdateView.xaml.cs

## 5.82 BusinessLogicLayer.Factories.TourLogFactory Class Reference

[TourLogFactory](#) is a concrete implementation of [ITourLogFactory](#).

Inheritance diagram for BusinessLogicLayer.Factories.TourLogFactory:



### Public Member Functions

- [TourLogFactory](#) ()  
*Default constructor of [TourLogFactory](#)*
- **TourLogFactory** (IUnitOfWork uow)
- Task [CreateTourLog](#) (TourLog tourLog)  
*CreateTourLog is a method for crating new TourLog entities in the datastore*
- Task [UpdateTourLog](#) (TourLog tourLog)  
*UpdateTourLog is a method for updating TourLog entities in the datastore*
- Task [DeleteTourLog](#) (TourLog tourLog)  
*DeleteTourLog is a method for deleting TourLog entities in the datastore*

### Private Attributes

- ILog [logger](#)  
*ILog instance used for logging errors, warnings etc.*
- IUnitOfWork [uow](#)  
*IUnitOfWork instance used for accessing the datastore*

### 5.82.1 Detailed Description

[TourLogFactory](#) is a concrete implementation of [ITourLogFactory](#).

## 5.82.2 Constructor & Destructor Documentation

### 5.82.2.1 TourLogFactory()

```
BusinessLogicLayer.Factories.TourLogFactory.TourLogFactory ( )
```

Default constructor of [TourLogFactory](#)

## 5.82.3 Member Function Documentation

### 5.82.3.1 CreateTourLog()

```
Task BusinessLogicLayer.Factories.TourLogFactory.CreateTourLog (
    TourLog tourLog )
```

CreateTourLog is a method for crating new TourLog entities in the datastore

#### Parameters

|                |                       |
|----------------|-----------------------|
| <i>tourLog</i> | TourLog to be created |
|----------------|-----------------------|

#### Returns

Task, which creates a new TourLog entity in the datastore

#### Exceptions

|                           |   |
|---------------------------|---|
| <i>BLFactoryException</i> | Thrown, when there are DAL exceptions or other unhandled errors |
|---------------------------|---|

Implements [BusinessLogicLayer.Factories.ITourLogFactory](#).

### 5.82.3.2 DeleteTourLog()

```
Task BusinessLogicLayer.Factories.TourLogFactory.DeleteTourLog (
    TourLog tourLog )
```

DeleteTourLog is a method for deleting TourLog entities in the datastore

**Parameters**

|                |                       |
|----------------|-----------------------|
| <i>tourLog</i> | TourLog to be deleted |
|----------------|-----------------------|

**Returns**

Task, which deletes a TourLog entity in the datastore

**Exceptions**

|                           |   |
|---------------------------|---|
| <i>BLFactoryException</i> | Thrown, when there are DAL exceptions or other unhandled errors |
|---------------------------|---|

Implements [BusinessLogicLayer.Factories.ITourLogFactory](#).

**5.82.3.3 UpdateTourLog()**

```
Task BusinessLogicLayer.Factories.TourLogFactory.UpdateTourLog (
    TourLog tourLog )
```

UpdateTourLog is a method for updating TourLog entities in the datastore

**Parameters**

|                |                        |
|----------------|------------------------|
| <i>tourLog</i> | Updated TourLog entity |
|----------------|------------------------|

**Returns**

Task, which updates a TourLog entity in the database

**Exceptions**

|                           |  |
|---------------------------|--|
| <i>BLFactoryException</i> | Thrown, when there are DAL exceptions or if there are other unhandled errors |
|---------------------------|--|

Implements [BusinessLogicLayer.Factories.ITourLogFactory](#).

**5.82.4 Member Data Documentation****5.82.4.1 logger**

```
ILog BusinessLogicLayer.Factories.TourLogFactory.logger [private]
```

ILog instance used for logging errors, warnings etc.

#### 5.82.4.2 uow

`IUnitOfWork` `BusinessLogicLayer.Factories.TourLogFactory.uow` [private]

IUnitOfWork instance used for accessing the datastore

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

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/Factories/TourLogFactory.cs

## 5.83 SWE2\_Tourplanner\_Tests.BLTests.TourLogFactoryTest Class Reference

### Public Member Functions

- void **SetUp** ()
- async Task **CreateTourLogTest** ()
- async Task **UpdateTourLogTest** ()
- async Task **DeleteTourLogTest** ()

### Private Attributes

- Mock< `IUnitOfWork` > **mockUOW**
- Mock< `ITourLogRepository` > **mockTourLogRepository**
- Mock< `ITourRepository` > **mockTourRepository**
- Mock< `ManeuverRepository` > **mockManeuverRepository**

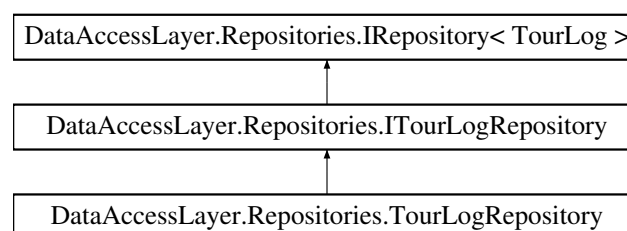
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/BLTests/TourLogFactoryTest.cs

## 5.84 DataAccessLayer.Repositories.TourLogRepository Class Reference

TourLogRepositories are used for querying, inserting, updating and deleting tourlogs. DML commands are not executed in this class. The only way to truly insert, update and delete tourlogs is through the `UnitOfWork` class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

Inheritance diagram for `DataAccessLayer.Repositories.TourLogRepository`:



## Public Member Functions

- [TourLogRepository](#) ()  
*Creates the [TourLogRepository](#) instance.*
- [TourLogRepository](#) ([IDBConnection](#) db, List< [IDBCommand](#) > [commitCommands](#))  
*Creates the [TourLogRepository](#) instance and "connects" it to the [UnitOfWork](#) class*
- void [Delete](#) (int id)  
*Creates a [DeleteTourLogCommand](#) object, if a tourlog with the specified id exists.*
- void [Insert](#) ([TourLog](#) entity)  
*Checks if log data is ok. If so, creates a new [InsertTourLogCommand](#)*
- [TourLog](#) [Read](#) (int id)  
*Function for the retrieval of a log with the specified id*
- List< [TourLog](#) > [ReadAll](#) ()  
*Function for retrieving all tourlogs in the table.*
- void [Update](#) ([TourLog](#) entity)  
*Creates an [UpdateTourLogCommand](#) object if the specified entity (id) exists in the table.*

## Private Member Functions

- [TourLog](#) [ConvertToTourLog](#) (object[] row)  
*Converts object arrays to tourlogs.*
- bool [CheckDBConstraints](#) ([TourLog](#) tourLog)  
*[CheckDBConstraints](#) is used to check whether a tourlog object complies with db constraints or not.*

## Private Attributes

- [IDBConnection](#) db  
*Database connection*
- List< [IDBCommand](#) > [commitCommands](#)  
*Commands to be committed to the database*
- log4net.ILog [logger](#)  
*ILog object used for logging errors etc.*

### 5.84.1 Detailed Description

[TourLogRepositories](#) are used for querying, inserting, updating and deleting tourlogs. DML commands are not executed in this class. The only way to truly insert, update and delete tourlogs is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

### 5.84.2 Constructor & Destructor Documentation



### 5.84.2.1 TourLogRepository() [1/2]

```
DataAccessLayer.Repositories.TourLogRepository.TourLogRepository ( )
```

Creates the [TourLogRepository](#) instance.

### 5.84.2.2 TourLogRepository() [2/2]

```
DataAccessLayer.Repositories.TourLogRepository.TourLogRepository (
    IDbConnection db,
    List< IDbCommand > commitCommands )
```

Creates the [TourLogRepository](#) instance and "connects" it to the [UnitOfWork](#) class

#### Parameters

|                       |  |
|-----------------------|--|
| <i>db</i>             | Database connection                                  |
| <i>commitCommands</i> | Commands for a commit ( <a href="#">UnitOfWork</a> ) |

## 5.84.3 Member Function Documentation

### 5.84.3.1 CheckDBConstraints()

```
bool DataAccessLayer.Repositories.TourLogRepository.CheckDBConstraints (
    TourLog tourLog ) [private]
```

CheckDBConstraints is used to check whether a tourlog object complies with db constraints or not.

#### Parameters

|                |                                       |
|----------------|---------------------------------------|
| <i>tourLog</i> | The tourlog that needs to be checked. |
|----------------|---------------------------------------|

#### Returns

True if constraints are adhered to, false if constraints are not complied with.

### 5.84.3.2 ConvertToTourLog()

```
TourLog DataAccessLayer.Repositories.TourLogRepository.ConvertToTourLog (
    object[] row ) [private]
```

Converts object arrays to tourlogs.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>row</i> | Result of a query |
|------------|-------------------|

**Returns**

Converted tourlog.

**5.84.3.3 Delete()**

```
void DataAccessLayer.Repositories.TourLogRepository.Delete (
    int id )
```

Creates a DeleteTourLogCommand object, if a tourlog with the specified id exists.

**Parameters**

|           |                                 |
|-----------|---------------------------------|
| <i>id</i> | Id of the tourlog to be deleted |
|-----------|---------------------------------|

**Exceptions**

|                                      |  |
|--------------------------------------|--|
| <i>DALRepositoryCommandException</i> | Thrown, when DeleteTourLogCommand can't be created |
|--------------------------------------|--|

**5.84.3.4 Insert()**

```
void DataAccessLayer.Repositories.TourLogRepository.Insert (
    TourLog entity )
```

Checks if log data is ok. If so, creates a new InsertTourLogCommand

**Parameters**

|               |                         |
|---------------|-------------------------|
| <i>entity</i> | TourLog to be inserted. |
|---------------|-------------------------|

**Exceptions**

|                                      |  |
|--------------------------------------|--|
| <i>DALRepositoryCommandException</i> | Thrown, when InsertTourLogCommand can't be created |
|--------------------------------------|--|

**5.84.3.5 Read()**

```
TourLog DataAccessLayer.Repositories.TourLogRepository.Read (
```

```
int id )
```

Function for the retrieval of a log with the specified id

#### Parameters

|           |                      |
|-----------|----------------------|
| <i>id</i> | Id of the wanted log |
|-----------|----------------------|

#### Returns

Log with the specified id (null if id doesn't exist in the table)

### 5.84.3.6 ReadAll()

```
List<TourLog> DataAccessLayer.Repositories.TourLogRepository.ReadAll ( )
```

Function for retrieving all tourlogs in the table.

#### Returns

Collection of all logs in the tourlog table

### 5.84.3.7 Update()

```
void DataAccessLayer.Repositories.TourLogRepository.Update (
    TourLog entity )
```

Creates an UpdateTourLogCommand object if the specified entity (id) exists in the table.

#### Parameters

|               |                      |
|---------------|----------------------|
| <i>entity</i> | New state of the log |
|---------------|----------------------|

#### Exceptions

|                                      |  |
|--------------------------------------|--|
| <i>DALRepositoryCommandException</i> | Thrown, when UpdateTourLogCommand can't be created |
|--------------------------------------|--|

## 5.84.4 Member Data Documentation

#### 5.84.4.1 commitCommands

```
List<IDBCommand> DataAccessLayer.Repositories.TourLogRepository.commitCommands [private]
```

Commands to be committed to the database

#### 5.84.4.2 db

```
IDBConnection DataAccessLayer.Repositories.TourLogRepository.db [private]
```

Database connection

#### 5.84.4.3 logger

```
log4net.ILog DataAccessLayer.Repositories.TourLogRepository.logger [private]
```

ILog object used for logging errors etc.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/TourLogRepository.cs

## 5.85 SWE2\_Tourplanner\_Tests.DALTests.TourLogRepositoryTests Class Reference

### Public Member Functions

- void **Setup** ()
- void **InsertCorrectCallsMock** ()
- void **UpdateCorrectCallsMock** ()
- void **DeleteCorrectCallsMock** ()
- void **ReadCorrectCallsMock** ()
- void **ReadAllCorrectCallsMock** ()

### Private Attributes

- Mock< [IDBConnection](#) > **mockDb**
- List< [IDBCommand](#) > **mockCommandList**
- [ITourLogRepository](#) **tourLogRepository**

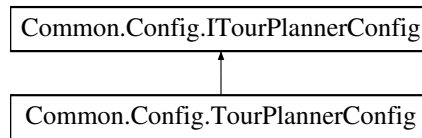
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/DALTests/TourLogRepositoryTests.cs

## 5.86 Common.Config.TourPlannerConfig Class Reference

[TourPlannerConfig](#) is a concrete implementation of the [ITourPlannerConfig](#) interface. It is a singleton and provides the application with vital config data.

Inheritance diagram for Common.Config.TourPlannerConfig:



### Public Member Functions

- void [LoadConfigFromFile](#) (string configPath)  
*LoadConfigFromFile can be used to load different configurations during execution.*

### Static Public Member Functions

- static [ITourPlannerConfig GetTourPlannerConfig](#) ()  
*Static access method for the only instance of [TourPlannerConfig](#).*

### Properties

- string **MapQuestKey** {exportsDirectory}"{pictureDirectory}" [get, private set]
- string **PictureDirectory** [get, private set]
- string **DatabaseConnectionString** [get, private set]
- string **ExportsDirectory** [get, private set]

### Private Member Functions

- [TourPlannerConfig](#) ()  
*Private constructor that is only called once. Automatically loads default config.*
- void [SetUpEnvironment](#) (string exportsDirectory, string pictureDirectory)  
*SetUpEnvironment ensures that exports and picture directory are created if they don't exist yet.*

### Private Attributes

- string [originalConfigPath](#) = "config.json"  
*Path to default configuration file*
- [ILog logger](#)  
*ILog instance used for logging errors, warnings etc.*

### Static Private Attributes

- static [ITourPlannerConfig tourPlannerConfig](#)  
*The one and only [TourPlannerConfig](#) object during execution. Can be accessed through static methods and populated through [LoadConfigFromFile\(\)](#) method.*

### 5.86.1 Detailed Description

[TourPlannerConfig](#) is a concrete implementation of the [ITourPlannerConfig](#) interface. It is a singleton and provides the application with vital config data.

### 5.86.2 Constructor & Destructor Documentation

#### 5.86.2.1 TourPlannerConfig()

```
Common.Config.TourPlannerConfig.TourPlannerConfig ( ) [private]
```

Private constructor that is only called once. Automatically loads default config.

### 5.86.3 Member Function Documentation

#### 5.86.3.1 GetTourPlannerConfig()

```
static ITourPlannerConfig Common.Config.TourPlannerConfig.GetTourPlannerConfig ( ) [static]
```

Static access method for the only instance of [TourPlannerConfig](#).

##### Returns

The only [TourPlannerConfig](#) object during execution

#### 5.86.3.2 LoadConfigFromFile()

```
void Common.Config.TourPlannerConfig.LoadConfigFromFile (
    string configPath )
```

LoadConfigFromFile can be used to load different configurations during execution.

##### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>configPath</i> | Path to the configuration file. |
|-------------------|---------------------------------|

### Exceptions

|                              |  |
|------------------------------|--|
| <i>CommonConfigException</i> | Throw when config file can't be found, when format isn't valid and when config file couldn't be loaded properly. |
|------------------------------|--|

Implements [Common.Config.ITourPlannerConfig](#).

#### 5.86.3.3 SetUpEnvironment()

```
void Common.Config.TourPlannerConfig.SetUpEnvironment (
    string exportsDirectory,
    string pictureDirectory ) [private]
```

SetUpEnvironment ensures that exports and picture directory are created if they don't exist yet.

#### Parameters

|                         |                               |
|-------------------------|-------------------------------|
| <i>exportsDirectory</i> | Path to new exports directory |
| <i>pictureDirectory</i> | Path to new picture directory |

### 5.86.4 Member Data Documentation

#### 5.86.4.1 logger

```
ILog Common.Config.TourPlannerConfig.logger [private]
```

ILog instance used for logging errors, warnings etc.

#### 5.86.4.2 originalConfigPath

```
string Common.Config.TourPlannerConfig.originalConfigPath = "config.json" [private]
```

Path to default configuration file

### 5.86.4.3 tourPlannerConfig

[ITourPlannerConfig](#) Common.Config.TourPlannerConfig.tourPlannerConfig [static], [private]

The one and only [TourPlannerConfig](#) object during execution. Can be accessed through static methods and populated through [LoadConfigFromFile\(\)](#) method.

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

- SWE2-Tourplanner/SWE2-Tourplanner/Common/Config/TourPlannerConfig.cs

## 5.87 SWE2\_Tourplanner\_Tests.CommonTests.TourPlannerConfigTests Class Reference

### Public Member Functions

- void [GetTourPlannerConfigTest](#) ()
- void [LoadConfigFromFileThrowsCommonConfigExceptionTest](#) ()

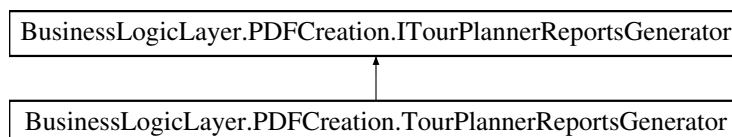
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/CommonTests/TourPlannerConfigTests.cs

## 5.88 BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator Class Reference

[TourPlannerReportsGenerator](#) is a concrete implementation of [ITourPlannerReportsGenerator](#). It generates tour- and summary reports as PDF files.

Inheritance diagram for BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator:



### Public Member Functions

- [TourPlannerReportsGenerator](#) ()  
*Default constructor of [TourPlannerReportsGenerator](#)*
- Task [GenerateSummaryReport](#) (List< [Tour](#) > tours)  
*GenerateSummaryReport creates a summary report for the passed tours.*
- Task [GenerateTourReport](#) ([Tour](#) tour)  
*GenerateTourReport generates a tour report for the passed tour*



## Private Attributes

- [ITourPlannerConfig](#) `config`  
*Object of [ITourPlannerConfig](#) which provides [TourPlannerReportsGenerator](#) with `ExportsDirectory`.*
- [ILog](#) `logger`  
*ILog instance used for logging errors, warnings etc.*

## 5.88.1 Detailed Description

[TourPlannerReportsGenerator](#) is a concrete implementation of [ITourPlannerReportsGenerator](#). It generates tour- and summary reports as PDF files.

## 5.88.2 Constructor & Destructor Documentation

### 5.88.2.1 TourPlannerReportsGenerator()

```
BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator.TourPlannerReportsGenerator ( )
```

Default constructor of [TourPlannerReportsGenerator](#)

## 5.88.3 Member Function Documentation

### 5.88.3.1 GenerateSummaryReport()

```
Task BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator.GenerateSummaryReport (
    List< Tour > tours )
```

`GenerateSummaryReport` creates a summary report for the passed tours.

#### Parameters

|              |  |
|--------------|--|
| <i>tours</i> | Tours used for summary report generation |
|--------------|--|

#### Returns

Task, which creates and automatically opens an summary report

#### Exceptions

|                               |  |
|-------------------------------|--|
| <i>BLPDFCreationException</i> | Thrown, if there are any errors during creation, saving and opening process of summary report generation |
|-------------------------------|--|

Implements [BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator](#).

### 5.88.3.2 GenerateTourReport()

```
Task BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator.GenerateTourReport (
    Tour tour )
```

GenerateTourReport generates a tour report for the passed tour

#### Parameters

|             |                                      |
|-------------|--------------------------------------|
| <i>tour</i> | Tour used for tour report generation |
|-------------|--------------------------------------|

#### Returns

Task, which generates a tour report and opens it automatically

#### Exceptions

|                               |   |
|-------------------------------|---|
| <i>BLPDFCreationException</i> | Thrown, if there are problems during creation, saving and opening process of the pdf file |
|-------------------------------|---|

Implements [BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator](#).

## 5.88.4 Member Data Documentation

### 5.88.4.1 config

```
ITourPlannerConfig BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator.config [private]
```

Object of ITourPlannerConfig which provides [TourPlannerReportsGenerator](#) with ExportsDirectory.

### 5.88.4.2 logger

```
ILog BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator.logger [private]
```

ILog instance used for logging errors, warnings etc.

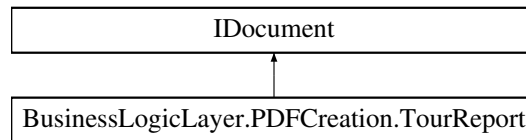
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/PDFCreation/TourPlannerReportGenerator.cs

## 5.89 BusinessLogicLayer.PDFCreation.TourReport Class Reference

[TourReport](#) defines the creation process of tour reports (PDF)

Inheritance diagram for BusinessLogicLayer.PDFCreation.TourReport:



### Public Member Functions

- [TourReport](#) ([Tour](#) tour)  
*Default constructor of [TourReport](#)*
- void [Compose](#) (IContainer container)  
*Compose defines the basic layout of the pdf file*
- DocumentMetadata [GetMetadata](#) ()  
*Metadata of [TourReport](#) pdf files*

### Properties

- [Tour](#) [Tour](#) [get]

### Private Member Functions

- void [ComposeHeader](#) (IContainer container)  
*ComposeHeader defines the structure of the header in the pdf file*
- void [ComposeContent](#) (IContainer container)  
*ComposeContent defines the structure of the content in the pdf file*
- void [ComposeManeuvers](#) (IContainer container)  
*ComposeManeuvers defines how maneuvers will be presented in the document*
- void [ComposeTourLogs](#) (IContainer container)  
*ComposeTourLogs defines the way tourlogs are presented in the document*
- void [ComposeDetails](#) (IContainer container)  
*ComposeDetails is responsible for presenting basic Tour data in the document*
- void [ComposeReports](#) (IContainer container)  
*ComposeReports defines how reports are presented*

### 5.89.1 Detailed Description

[TourReport](#) defines the creation process of tour reports (PDF)

### 5.89.2 Constructor & Destructor Documentation

#### 5.89.2.1 TourReport()

```
BusinessLogicLayer.PDFCreation.TourReport.TourReport (
    Tour tour )
```

Default constructor of [TourReport](#)

**Parameters**

|             |                                      |
|-------------|--------------------------------------|
| <i>tour</i> | Tour used for tour report generation |
|-------------|--------------------------------------|

### 5.89.3 Member Function Documentation

#### 5.89.3.1 Compose()

```
void BusinessLogicLayer.PDFCreation.TourReport.Compose (
    IContainer container )
```

Compose defines the basic layout of the pdf file

**Parameters**

|                  |   |
|------------------|---|
| <i>container</i> | Container, in which the elements will be placed |
|------------------|---|

#### 5.89.3.2 ComposeContent()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeContent (
    IContainer container ) [private]
```

ComposeContent defines the structure of the content in the pdf file

**Parameters**

|                  |  |
|------------------|--|
| <i>container</i> | Content Container, in which further content will be placed |
|------------------|--|

#### 5.89.3.3 ComposeDetails()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeDetails (
    IContainer container ) [private]
```

ComposeDetails is responsible for presenting basic Tour data in the document

**Parameters**

|                  |   |
|------------------|---|
| <i>container</i> | Container, in which Tour data is placed |
|------------------|---|

#### 5.89.3.4 ComposeHeader()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeHeader (  
    IContainer container ) [private]
```

ComposeHeader defines the structure of the header in the pdf file

##### Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Header container, in which further elements are placed |
|------------------|--|

#### 5.89.3.5 ComposeManeuvers()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeManeuvers (  
    IContainer container ) [private]
```

ComposeManeuvers defines how maneuvers will be presented in the document

##### Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Container, in which the maneuvers will be placed |
|------------------|--|

#### 5.89.3.6 ComposeReports()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeReports (  
    IContainer container ) [private]
```

ComposeReports defines how reports are presented

##### Parameters

|                  |   |
|------------------|---|
| <i>container</i> | Container, in which TourLog reports are displayed |
|------------------|---|

#### 5.89.3.7 ComposeTourLogs()

```
void BusinessLogicLayer.PDFCreation.TourReport.ComposeTourLogs (  
    IContainer container ) [private]
```

ComposeTourLogs defines the way tourlogs are presented in the document

## Parameters

|                  |  |
|------------------|--|
| <i>container</i> | Container, in which the tourlog content will be placed |
|------------------|--|

**5.89.3.8 GetMetadata()**

```
DocumentMetadata BusinessLogicLayer.PDFCreation.TourReport.GetMetadata ( )
```

Metadata of [TourReport](#) pdf files

## Returns

Metadata of the document

**5.89.4 Property Documentation****5.89.4.1 Tour**

```
Tour BusinessLogicLayer.PDFCreation.TourReport.Tour [get]
```

Tour used for tour report generation

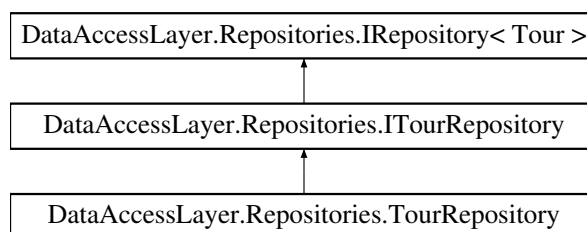
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/BusinessLogicLayer/PDFCreation/TourReport.cs

**5.90 DataAccessLayer.Repositories.TourRepository Class Reference**

TourRepositories are used for querying, inserting, updating and deleting tours. DML commands are not executed in this class. The only way to truly insert, update and delete tours is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

Inheritance diagram for `DataAccessLayer.Repositories.TourRepository`:



## Public Member Functions

- [TourRepository](#) ()  
*Creates the [TourRepository](#) object*
- [TourRepository](#) ([IDBConnection](#) db, List< [IDBCommand](#) > [commitCommands](#))  
*Creates the [TourRepository](#) instance and "connects" it to the wrapping [UnitOfWork](#) class*
- [TourRepository](#) ([IDBConnection](#) db, List< [IDBCommand](#) > [commitCommands](#), [ITourLogRepository](#) [tourLogRepository](#), [IManeuverRepository](#) [maneuverRepository](#))  
*Constructor solely for testing purposes*
- void [Delete](#) (int id)  
*Creates a [DeleteTourCommand](#) object if a tour with the specified id exists.*
- void [Insert](#) ([Tour](#) entity)  
*Checks if the data of the tour is ok. If so, a [InsertTourCommand](#) object gets created.*
- [Tour Read](#) (int id)  
*Retrieves a tour with the specified id.*
- List< [Tour](#) > [ReadAll](#) ()  
*Retrieves all tours in the tour table.*
- void [Update](#) ([Tour](#) entity)  
*Creates [UpdateTourCommand](#) if the specified entity exists.*

## Private Member Functions

- [Tour ConvertToTour](#) (object[] row)  
*Converts object arrays to tours.*
- bool [CheckDBConstraints](#) ([Tour](#) tour)  
*[CheckDBConstraints](#) checks whether db constraints are complied with or not.*

## Private Attributes

- [IDBConnection](#) db  
*Database connection*
- List< [IDBCommand](#) > [commitCommands](#)  
*Commands to be committed to the database*
- [ITourLogRepository](#) [tourLogRepository](#)  
*Repository for retrieving associated tourlogs*
- [IManeuverRepository](#) [maneuverRepository](#)  
*[Repositories](#) for retrieving associated maneuvers*
- [log4net.ILog](#) [logger](#)  
*[ILog](#) object used for logging errors etc.*

### 5.90.1 Detailed Description

TourRepositories are used for querying, inserting, updating and deleting tours. DML commands are not executed in this class. The only way to truly insert, update and delete tours is through the [UnitOfWork](#) class, which ensures that database connections get opened/closed only when really needed. Retrieving data is possible everywhere.

### 5.90.2 Constructor & Destructor Documentation

### 5.90.2.1 TourRepository() [1/3]

```
DataAccessLayer.Repositories.TourRepository.TourRepository ( )
```

Creates the [TourRepository](#) object

### 5.90.2.2 TourRepository() [2/3]

```
DataAccessLayer.Repositories.TourRepository.TourRepository (
    IDbConnection db,
    List< IDbCommand > commitCommands )
```

Creates the [TourRepository](#) instance and "connects" it to the wrapping [UnitOfWork](#) class

#### Parameters

|                       |  |
|-----------------------|--|
| <i>db</i>             | Database connection                                  |
| <i>commitCommands</i> | Commands for a commit ( <a href="#">UnitOfWork</a> ) |

### 5.90.2.3 TourRepository() [3/3]

```
DataAccessLayer.Repositories.TourRepository.TourRepository (
    IDbConnection db,
    List< IDbCommand > commitCommands,
    ITourLogRepository tourLogRepository,
    IManeuverRepository maneuverRepository )
```

Constructor solely for testing purposes

#### Parameters

|                           |   |
|---------------------------|---|
| <i>db</i>                 | Database connection. For tests, a mock can be passed.                 |
| <i>commitCommands</i>     | Commands to be committed.   |
| <i>tourLogRepository</i>  | A repository for tourlogs. A mock can be passed for testing puproses. |
| <i>maneuverRepository</i> | Repository for maneuvers. Mock object can be passed                   |

## 5.90.3 Member Function Documentation

### 5.90.3.1 CheckDBConstraints()

```
bool DataAccessLayer.Repositories.TourRepository.CheckDBConstraints (
    Tour tour ) [private]
```



CheckDBConstraints checks whether db constraints are complied with or not.

**Parameters**

|             |                                    |
|-------------|------------------------------------|
| <i>tour</i> | The tour that needs to be checked. |
|-------------|------------------------------------|

**Returns**

True if constraints are adhered to, false if constraints are not complied with.

**5.90.3.2 ConvertToTour()**

```
Tour DataAccessLayer.Repositories.TourRepository.ConvertToTour (
    object[] row ) [private]
```

Converts object arrays to tours.

**Parameters**

|            |                         |
|------------|-------------------------|
| <i>row</i> | Result (row) of a query |
|------------|-------------------------|

**Returns**

Converted tour.

**5.90.3.3 Delete()**

```
void DataAccessLayer.Repositories.TourRepository.Delete (
    int id )
```

Creates a DeleteTourCommand object if a tour with the specified id exists.

**Parameters**

|           |                              |
|-----------|------------------------------|
| <i>id</i> | Id of the tour to be deleted |
|-----------|------------------------------|

**Exceptions**

|                                      |   |
|--------------------------------------|---|
| <i>DALRepositoryCommandException</i> | Thrown, when DeleteTourCommand can't be created |
|--------------------------------------|---|

**5.90.3.4 Insert()**

```
void DataAccessLayer.Repositories.TourRepository.Insert (
```

```
Tour entity )
```

Checks if the data of the tour is ok. If so, a InsertTourCommand object gets created.

#### Parameters

|               |  |
|---------------|--|
| <i>entity</i> | Tour that is supposed to be inserted into the database |
|---------------|--|

#### Exceptions

|                                      |   |
|--------------------------------------|---|
| <i>DALRepositoryCommandException</i> | Thrown, when InsertTourCommand can't be created |
|--------------------------------------|---|

### 5.90.3.5 Read()

```
Tour DataAccessLayer.Repositories.TourRepository.Read (
    int id )
```

Retrieves a tour with the specified id.

#### Parameters

|           |                        |
|-----------|------------------------|
| <i>id</i> | Id of the wanted tour. |
|-----------|------------------------|

#### Returns

Tour with the specified id (null if id doesn't exist in table). Associated data (maneuvers, tourlogs) are already loaded

### 5.90.3.6 ReadAll()

```
List<Tour> DataAccessLayer.Repositories.TourRepository.ReadAll ( )
```

Retrieves all tours in the tour table.

#### Returns

Collection of all tours. Associated data (maneuvers, tourlogs) are already loaded.

### 5.90.3.7 Update()

```
void DataAccessLayer.Repositories.TourRepository.Update (
    Tour entity )
```

Creates UpdateTourCommand if the specified entity exists.

**Parameters**

|               |                        |
|---------------|------------------------|
| <i>entity</i> | New state of the tour. |
|---------------|------------------------|

**Exceptions**

|                                      |   |
|--------------------------------------|---|
| <i>DALRepositoryCommandException</i> | Thrown, when UpdateTourCommand can't be created |
|--------------------------------------|---|

## 5.90.4 Member Data Documentation

### 5.90.4.1 commitCommands

```
List<IDBCommand> DataAccessLayer.Repositories.TourRepository.commitCommands [private]
```

Commands to be committed to the database

### 5.90.4.2 db

```
IDBConnection DataAccessLayer.Repositories.TourRepository.db [private]
```

Database connection

### 5.90.4.3 logger

```
log4net.ILog DataAccessLayer.Repositories.TourRepository.logger [private]
```

ILog object used for logging errors etc.

### 5.90.4.4 maneuverRepository

```
IManeuverRepository DataAccessLayer.Repositories.TourRepository.maneuverRepository [private]
```

[Repositories](#) for retrieving associated maneuvers

#### 5.90.4.5 tourLogRepository

`ITourLogRepository` `DataAccessLayer.Repositories.TourRepository.tourLogRepository` [private]

Repository for retrieving associated tourlogs

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/Repositories/TourRepository.cs

## 5.91 SWE2\_Tourplanner\_Tests.DALTests.TourRepositoryTests Class Reference

### Public Member Functions

- void **Setup** ()
- void **InsertCorrectCallsMock** ()
- void **UpdateCorrectCallsMock** ()
- void **DeleteCorrectCallsMock** ()
- void **ReadCorrectCallsMock** ()
- void **ReadAllCorrectCallsMock** ()

### Private Attributes

- Mock< `IDBConnection` > **mockDb**
- List< `IDBCommand` > **mockCommandList**
- Mock< `IManeuverRepository` > **maneuverRepository**
- Mock< `ITourLogRepository` > **tourLogRepository**
- `ITourRepository` **tourRepository**

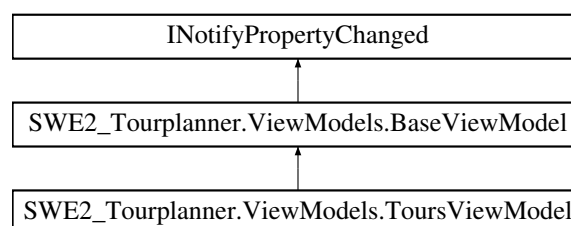
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/DALTests/TourRepositoryTests.cs

## 5.92 SWE2\_Tourplanner.ViewModels.ToursViewModel Class Reference

ToursViewModel provides tours and a way to search / filter tours

Inheritance diagram for SWE2\_Tourplanner.ViewModels.ToursViewModel:



## Public Member Functions

- [ToursViewModel](#) ([ITourFactory](#) tourPlannerFactory, [IDialogService](#) dialogService)  
*Default constructor of [ToursViewModel](#)*

## Properties

- [ObservableCollection](#)< [Tour](#) > [FilteredTours](#) [get, set]
- string [SearchString](#) [get, set]
- [ObservableCollection](#)< [Tour](#) > [Tours](#) [get, set]
- [ICommand](#) [SearchToursCommand](#) [get]

## Private Attributes

- [ITourFactory](#) tourPlannerFactory  
*Object used for retrieving tours*
- [IDialogService](#) dialogService  
*Service used for displaying exceptions / errors*
- string [searchString](#)  
*Search term*
- [ObservableCollection](#)< [Tour](#) > [filteredTours](#)  
*Tours filtered by search term*
- [ObservableCollection](#)< [Tour](#) > [\\_tours](#)  
*All tours found in datastore*

## Additional Inherited Members

### 5.92.1 Detailed Description

ToursViewModel provides tours and a way to search / filter tours

### 5.92.2 Constructor & Destructor Documentation

#### 5.92.2.1 ToursViewModel()

```
SWE2_Tourplanner.ViewModels.ToursViewModel.ToursViewModel (
    ITourFactory tourPlannerFactory,
    IDialogService dialogService )
```

Default constructor of [ToursViewModel](#)

#### Parameters

|                           |   |
|---------------------------|---|
| <i>tourPlannerFactory</i> | Object used for retrieving tours from datastore             |
| <i>dialogService</i>      | Service used for opening dialogs and handling their results |

### 5.92.3 Member Data Documentation

#### 5.92.3.1 \_tours

ObservableCollection<Tour> SWE2\_Tourplanner.ViewModels.ToursViewModel.\_tours [private]

All tours found in datastore

#### 5.92.3.2 dialogService

IDialogService SWE2\_Tourplanner.ViewModels.ToursViewModel.dialogService [private]

Service used for displaying exceptions / errors

#### 5.92.3.3 filteredTours

ObservableCollection<Tour> SWE2\_Tourplanner.ViewModels.ToursViewModel.filteredTours [private]

Tours filtered by search term

#### 5.92.3.4 searchString

string SWE2\_Tourplanner.ViewModels.ToursViewModel.searchString [private]

Search term

#### 5.92.3.5 tourPlannerFactory

ITourFactory SWE2\_Tourplanner.ViewModels.ToursViewModel.tourPlannerFactory [private]

Object used for retrieving tours

### 5.92.4 Property Documentation

#### 5.92.4.1 FilteredTours

```
ObservableCollection<Tour> SWE2_Tourplanner.ViewModels.ToursViewModel.FilteredTours [get], [set]
```

Tours filtered by search term

#### 5.92.4.2 SearchString

```
string SWE2_Tourplanner.ViewModels.ToursViewModel.SearchString [get], [set]
```

Search term

#### 5.92.4.3 SearchToursCommand

```
ICommand SWE2_Tourplanner.ViewModels.ToursViewModel.SearchToursCommand [get]
```

Command for searching / filtering tours

#### 5.92.4.4 Tours

```
ObservableCollection<Tour> SWE2_Tourplanner.ViewModels.ToursViewModel.Tours [get], [set]
```

All tours from the datastore

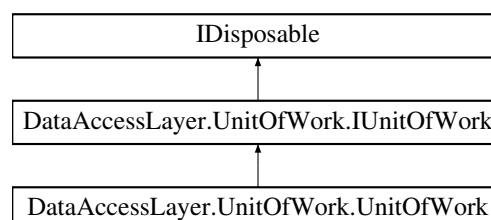
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner/ViewModels/ToursViewModel.cs

## 5.93 DataAccessLayer.UnitOfWork.UnitOfWork Class Reference

Concrete implementation of the [IUnitOfWork](#) interface.

Inheritance diagram for DataAccessLayer.UnitOfWork.UnitOfWork:





## Public Member Functions

- [UnitOfWork](#) ()  
Default constructor for [UnitOfWork](#). Retrieves the database connection from a singleton and properly creates the 3 repositories.
- [UnitOfWork](#) (IDBConnection db, List< [IDBCommand](#) > commitCommands, List< [IDBCommand](#) > rollbackCommands)  
A constructor just for testing purposes.
- int [Commit](#) ()  
Concrete implementation of the [Commit\(\)](#) function of the [IUnitOfWork](#) interface. Takes every issued command for the commit and executes them together.
- void [Dispose](#) ()  
The dispose method is implemented for UnitsOfWork to be used in a using statement.
- int [Rollback](#) ()  
Concrete implementation of the [Rollback\(\)](#) function of the [IUnitOfWork](#) interface. Takes every committed command and executes their Undo() function.

## Properties

- [ITourRepository](#) [TourRepository](#) [get, set]
- [ITourLogRepository](#) [TourLogRepository](#) [get, set]
- [IManeuverRepository](#) [ManeuverRepository](#) [get, set]

## Private Attributes

- List< [IDBCommand](#) > [commitCommands](#)  
*commitCommands is a collection of every issued command for a commit.*
- List< [IDBCommand](#) > [rollbackCommands](#)  
*rollbackCommands is a collection of every committed command.*
- [IDBConnection](#) [db](#)  
*IDbConnection is the database connection.*
- [ITourRepository](#) [tourRepository](#)  
*The provided Tour repository for the user. Insert-, Update- and Delete-TourCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.*
- [ITourLogRepository](#) [tourLogRepository](#)  
*The provided TourLog repository for the user. Insert-, Update- and Delete-TourLogCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.*
- [IManeuverRepository](#) [maneuverRepository](#)  
*The provided Maneuver repository for the user. Insert-, Update- and Delete-ManeuverCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.*
- [log4net.ILog](#) [logger](#)  
*ILog object used for logging errors etc.*

### 5.93.1 Detailed Description

Concrete implementation of the [IUnitOfWork](#) interface.

### 5.93.2 Constructor & Destructor Documentation

### 5.93.2.1 UnitOfWork() [1/2]

```
DataAccessLayer.UnitOfWork.UnitOfWork.UnitOfWork ( )
```

Default constructor for [UnitOfWork](#). Retrieves the database connection from a singleton and properly creates the 3 repositories.

### 5.93.2.2 UnitOfWork() [2/2]

```
DataAccessLayer.UnitOfWork.UnitOfWork.UnitOfWork (
    IDbConnection db,
    List< IDbCommand > commitCommands,
    List< IDbCommand > rollbackCommands )
```

A constructor just for testing purposes.

#### Parameters

|                         |   |
|-------------------------|---|
| <i>db</i>               | The database connection. For testing purposes, a mock can be passed.  |
| <i>commitCommands</i>   | An IDbCommand collection. Can be used to verify if the amount of commands and their types are correct in test-methods.                                |
| <i>rollbackCommands</i> | An IDbCommand collection. Can be used to verify if the <a href="#">Commit()</a> function puts executed commands into the rollbackCommands-Collection. |

## 5.93.3 Member Function Documentation

### 5.93.3.1 Commit()

```
int DataAccessLayer.UnitOfWork.UnitOfWork.Commit ( )
```

Concrete implementation of the [Commit\(\)](#) function of the [IUnitOfWork](#) interface. Takes every issued command for the commit and executes them together.

#### Returns

The amount of rows affected by the commit.

#### Exceptions

|                               |  |
|-------------------------------|--|
| <i>DALUnitOfWorkException</i> | Thrown, when an error occurs during the commit |
|-------------------------------|--|

Implements [DataAccessLayer.UnitOfWork.IUnitOfWork](#).

### 5.93.3.2 Dispose()

```
void DataAccessLayer.UnitOfWork.UnitOfWork.Dispose ( )
```

The dispose method is implemented for UnitsOfWork to be used in a using statement.

### 5.93.3.3 Rollback()

```
int DataAccessLayer.UnitOfWork.UnitOfWork.Rollback ( )
```

Concrete implementation of the [Rollback\(\)](#) function of the [IUnitOfWork](#) interface. Takes every committed command and executes their Undo() function.

#### Returns

The amount of rows affected by the rollback.

#### Exceptions

|  |  |
|--|--|
| <a href="#">DALUnitOfWorkException</a> | Thrown, when an error occurs during the rollback |
|--|--|

Implements [DataAccessLayer.UnitOfWork.IUnitOfWork](#).

## 5.93.4 Member Data Documentation

### 5.93.4.1 commitCommands

```
List<IDBCCommand> DataAccessLayer.UnitOfWork.UnitOfWork.commitCommands [private]
```

commitCommands is a collection of every issued command for a commit.

### 5.93.4.2 db

```
IDBConnection DataAccessLayer.UnitOfWork.UnitOfWork.db [private]
```

IDBConnection is the database connection.

#### 5.93.4.3 logger

```
log4net.ILog DataAccessLayer.UnitOfWork.UnitOfWork.logger [private]
```

ILog object used for logging errors etc.

#### 5.93.4.4 maneuverRepository

```
IManeuverRepository DataAccessLayer.UnitOfWork.UnitOfWork.maneuverRepository [private]
```

The provided Maneuver repository for the user. Insert-, Update- and Delete-ManeuverCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.

#### 5.93.4.5 rollbackCommands

```
List<IDBCCommand> DataAccessLayer.UnitOfWork.UnitOfWork.rollbackCommands [private]
```

rollbackCommands is a collection of every committed command.

#### 5.93.4.6 tourLogRepository

```
ITourLogRepository DataAccessLayer.UnitOfWork.UnitOfWork.tourLogRepository [private]
```

The provided TourLog repository for the user. Insert-, Update- and Delete-TourLogCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.

#### 5.93.4.7 tourRepository

```
ITourRepository DataAccessLayer.UnitOfWork.UnitOfWork.tourRepository [private]
```

The provided Tour repository for the user. Insert-, Update- and Delete-TourCommands produced by Insert(), Update() and Delete() are saved into the commitCommands collection.

### 5.93.5 Property Documentation

### 5.93.5.1 ManeuverRepository

`IManeuverRepository` DataAccessLayer.UnitOfWork.UnitOfWork.ManeuverRepository [get], [set]

A public property for setting and getting the private property maneuverRepository.

### 5.93.5.2 TourLogRepository

`ITourLogRepository` DataAccessLayer.UnitOfWork.UnitOfWork.TourLogRepository [get], [set]

A public property for setting and getting the private property tourLogRepository.

### 5.93.5.3 TourRepository

`ITourRepository` DataAccessLayer.UnitOfWork.UnitOfWork.TourRepository [get], [set]

A public property for setting and getting the private property tourRepository.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/UnitOfWork/UnitOfWork.cs

## 5.94 SWE2\_Tourplanner\_Tests.DALTests.UnitOfWorkTests Class Reference

### Public Member Functions

- void **Setup** ()
- void **CommitCorrectCallsMock** ()
- void **RollbackCorrectCallsMock** ()
- void **CommitAndRollbackCorrectCallsMock** ()

### Private Attributes

- `IUnitOfWork` **uow**
- `Mock< IDbConnection >` **db**
- `List< IDbCommand >` **commitCommands**
- `List< IDbCommand >` **rollbackCommands**
- `List< Mock< IDbCommand > >` **mockCommitCommands**
- `List< Mock< IDbCommand > >` **mockRollbackCommands**

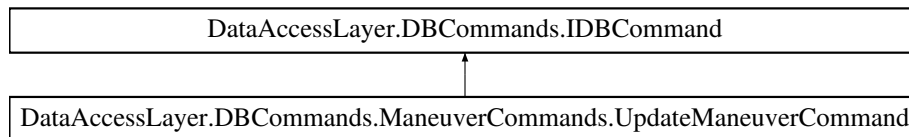
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/DALTests/UnitOfWorkTests.cs

## 5.95 DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand Class Reference

[UpdateManeuverCommand](#) is used for updating maneuvers.

Inheritance diagram for `DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand`:



### Public Member Functions

- [UpdateManeuverCommand](#) (`IDBConnection db`, `Maneuver maneuver`, `Maneuver oldManeuver`)  
*Creates the [UpdateManeuverCommand](#) instance.*
- `int` [Execute](#) ()  
*Updates the data of a specific maneuver.*
- `int` [Undo](#) ()  
*Reverts the maneuver back to its original state.*

### Private Attributes

- `IDBConnection` `db`  
*Connection to the database*
- `Maneuver` `maneuver`  
*New state of the maneuver.*
- `Maneuver` `oldManeuver`  
*Old state of the maneuver.*
- `log4net.ILog` `logger`  
*ILog instance used for logging errors, warnings etc.*

### 5.95.1 Detailed Description

[UpdateManeuverCommand](#) is used for updating maneuvers.

### 5.95.2 Constructor & Destructor Documentation

#### 5.95.2.1 UpdateManeuverCommand()

```

DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.UpdateManeuverCommand (
    IDBConnection db,
    Maneuver maneuver,
    Maneuver oldManeuver )
  
```

Creates the [UpdateManeuverCommand](#) instance.

## Parameters

|                    |   |
|--------------------|---|
| <i>db</i>          | Database connection.                      |
| <i>maneuver</i>    | The new wanted state of the maneuver.     |
| <i>oldManeuver</i> | The old deprecated state of the maneuver. |

### 5.95.3 Member Function Documentation

#### 5.95.3.1 Execute()

```
int DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.Execute ( )
```

Updates the data of a specific maneuver.

## Returns

Amount of rows affected by the update statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

#### 5.95.3.2 Undo()

```
int DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.Undo ( )
```

Reverts the maneuver back to its original state.

## Returns

Amount of rows affected by the rollback to the original maneuver. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.95.4 Member Data Documentation

#### 5.95.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.db [private]
```

Connection to the database

Connection to the database.

#### 5.95.4.2 logger

`log4net.ILog DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.logger` [private]

ILog instance used for logging errors, warnings etc.

#### 5.95.4.3 maneuver

`Maneuver DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.maneuver` [private]

New state of the maneuver.

#### 5.95.4.4 oldManeuver

`Maneuver DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand.oldManeuver` [private]

Old state of the maneuver.

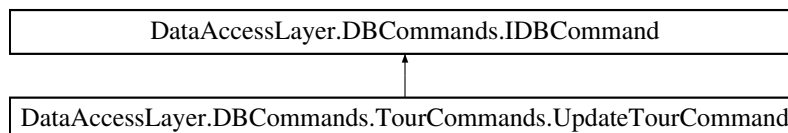
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/ManeuverCommands/UpdateManeuverCommand.cs

## 5.96 DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand Class Reference

`UpdateTourCommand` is used for updating tours.

Inheritance diagram for `DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand`:



### Public Member Functions

- `UpdateTourCommand (IDBConnection db, Tour tour, Tour oldTour)`  
*Creates the `UpdateTourCommand` instance.*
- `int Execute ()`  
*Sets the data of the tour in the database to the new state.*
- `int Undo ()`  
*Reverts the tour to its original state.*



## Private Attributes

- [IDBConnection db](#)  
*Connection to the database.*
- [Tour tour](#)  
*New state of the tour.*
- [Tour oldTour](#)  
*Old state of the tour.*

### 5.96.1 Detailed Description

[UpdateTourCommand](#) is used for updating tours.

### 5.96.2 Constructor & Destructor Documentation

#### 5.96.2.1 UpdateTourCommand()

```
DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.UpdateTourCommand (
    IDBConnection db,
    Tour tour,
    Tour oldTour )
```

Creates the [UpdateTourCommand](#) instance.

##### Parameters

|                |                            |
|----------------|----------------------------|
| <i>db</i>      | Connection to the database |
| <i>tour</i>    | New state of the tour      |
| <i>oldTour</i> | Old state of the tour      |

### 5.96.3 Member Function Documentation

#### 5.96.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.Execute ( )
```

Sets the data of the tour in the database to the new state.

##### Returns

Amount of rows affected by the update-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.96.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.Undo ( )
```

Reverts the tour to its original state.

#### Returns

Amount of rows affected by the rollback to the original state. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

## 5.96.4 Member Data Documentation

### 5.96.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.db [private]
```

Connection to the database.

### 5.96.4.2 oldTour

```
Tour DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.oldTour [private]
```

Old state of the tour.

### 5.96.4.3 tour

```
Tour DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand.tour [private]
```

New state of the tour.

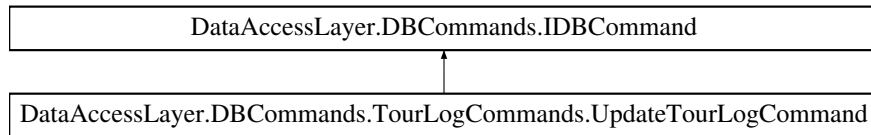
The documentation for this class was generated from the following file:

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourCommands/UpdateTour↔  
Command.cs

## 5.97 DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand Class Reference

[UpdateTourLogCommand](#) is used for updating tourlogs.

Inheritance diagram for `DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand`:



### Public Member Functions

- [UpdateTourLogCommand](#) ([IDBConnection](#) db, [TourLog](#) tourLog, [TourLog](#) oldTourLog)  
*Creates the [UpdateTourLogCommand](#) instance.*
- [int](#) [Execute](#) ()  
*Updates the data of the log in the database according to the new state of the log.*
- [int](#) [Undo](#) ()  
*Reverts the data of the log in the database back to its original state.*

### Private Attributes

- [IDBConnection](#) db  
*Connection to the database.*
- [TourLog](#) tourLog  
*New state of the log.*
- [TourLog](#) oldTourLog  
*Old state of the log.*
- [log4net.ILog](#) logger  
*ILog instance used for logging errors, warnings etc.*

#### 5.97.1 Detailed Description

[UpdateTourLogCommand](#) is used for updating tourlogs.

#### 5.97.2 Constructor & Destructor Documentation

##### 5.97.2.1 UpdateTourLogCommand()

```

DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.UpdateTourLogCommand (
    IDBConnection db,
    TourLog tourLog,
    TourLog oldTourLog )
  
```

Creates the [UpdateTourLogCommand](#) instance.

## Parameters

|                   |                             |
|-------------------|-----------------------------|
| <i>db</i>         | Connection to the database. |
| <i>tourLog</i>    | The new state of the log.   |
| <i>oldTourLog</i> | The old state of the log.   |

### 5.97.3 Member Function Documentation

#### 5.97.3.1 Execute()

```
int DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.Execute ( )
```

Updates the data of the log in the database according to the new state of the log.

## Returns

Amount of rows affected by the update-statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

#### 5.97.3.2 Undo()

```
int DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.Undo ( )
```

Reverts the data of the log in the database back to its original state.

## Returns

Amount of rows affected by the update statement. Expected: 1

Implements [DataAccessLayer.DBCommands.IDBCommand](#).

### 5.97.4 Member Data Documentation

#### 5.97.4.1 db

```
IDBConnection DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.db [private]
```

Connection to the database.

#### 5.97.4.2 logger

`log4net.ILog DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.logger [private]`

ILog instance used for logging errors, warnings etc.

#### 5.97.4.3 oldTourLog

`TourLog DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.oldTourLog [private]`

Old state of the log.

#### 5.97.4.4 tourLog

`TourLog DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand.tourLog [private]`

New state of the log.

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

- SWE2-Tourplanner/SWE2-Tourplanner/DataAccessLayer/DBCommands/TourLogCommands/UpdateTour↵  
LogCommand.cs

## 5.98 SWE2\_Tourplanner\_Tests.UITests.ValidationRulesTests Class Reference

### Public Member Functions

- void **DoubleRangeRuleTest** ()
- void **NoNegativeDistancesRuleTest** ()
- void **TextInputValidationRuleTest** ()

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

- SWE2-Tourplanner/SWE2-Tourplanner/SWE2-Tourplanner-Tests/UITests/ValidationRulesTests.cs



# Index

- [\\_canExecute](#)
    - [SWE2\\_Tourplanner.Commands.RelayCommand,](#)  
[161](#)
  - [\\_execute](#)
    - [SWE2\\_Tourplanner.Commands.RelayCommand,](#)  
[162](#)
  - [\\_selectedTour](#)
    - [SWE2\\_Tourplanner.ViewModels.TourDetailViewModel,](#)  
[175](#)
  - [\\_tours](#)
    - [SWE2\\_Tourplanner.ViewModels.ToursViewModel,](#)  
[215](#)
- [AddEventHandler](#)
  - [XamlGeneratedNamespace.GeneratedInternalTypeHelper,](#)  
[82, 83](#)
- [AddLogCommand](#)
  - [SWE2\\_Tourplanner.ViewModels.TourDetailViewModel,](#)  
[176](#)
- [AddManeuverCommand](#)
  - [SWE2\\_Tourplanner.ViewModels.ManeuversViewModel,](#)  
[147](#)
- [AddTourCommand](#)
  - [SWE2\\_Tourplanner.MainViewModel,](#) [134](#)
- [BLExporterImporterException](#)
  - [BusinessLogicLayer.Exceptions.BLExporterImporterException,](#)  
[28](#)
- [BLFactoryException](#)
  - [BusinessLogicLayer.Exceptions.BLFactoryException,](#)  
[29, 30](#)
- [BLPDFCreationException](#)
  - [BusinessLogicLayer.Exceptions.BLPDFCreationException,](#)  
[31](#)
- [BusinessLogicLayer,](#) [13](#)
- [BusinessLogicLayer.Exceptions,](#) [33](#)
- [BusinessLogicLayer.Exceptions.BLExporterImporterException,](#)  
[27](#)
  - [BLExporterImporterException,](#) [28](#)
- [BusinessLogicLayer.Exceptions.BLFactoryException,](#)  
[29](#)
  - [BLFactoryException,](#) [29, 30](#)
- [BusinessLogicLayer.Exceptions.BLPDFCreationException,](#)  
[30](#)
  - [BLPDFCreationException,](#) [31](#)
- [BusinessLogicLayer.ExporterImporter,](#) [13](#)
- [BusinessLogicLayer.ExporterImporter.IExporterImporter<](#)  
[T >,](#) [96](#)
  - [Export,](#) [97](#)
  - [Import,](#) [97](#)
- [BusinessLogicLayer.ExporterImporter.TourExporterImporter,](#)  
[177](#)
  - [Export,](#) [178](#)
  - [ExportPicture,](#) [178](#)
  - [Import,](#) [178](#)
  - [logger,](#) [179](#)
  - [TourExporterImporter,](#) [177](#)
  - [tourPlannerConfig,](#) [179](#)
- [BusinessLogicLayer.Factories,](#) [13](#)
- [BusinessLogicLayer.Factories.ITourFactory,](#) [119](#)
  - [CreateMapQuestTour,](#) [119](#)
  - [CreateTour,](#) [120](#)
  - [DeleteTour,](#) [120](#)
  - [GetTours,](#) [120](#)
  - [UpdateMapQuestTour,](#) [121](#)
  - [UpdateTour,](#) [121](#)
- [BusinessLogicLayer.Factories.ITourLogFactory,](#) [122](#)
  - [CreateTourLog,](#) [122](#)
  - [DeleteTourLog,](#) [123](#)
  - [UpdateTourLog,](#) [123](#)
- [BusinessLogicLayer.Factories.TourFactory,](#) [179](#)
  - [CreateMapQuestTour,](#) [181](#)
  - [CreateTour,](#) [181](#)
  - [DeleteTour,](#) [181](#)
  - [GetTours,](#) [182](#)
  - [logger,](#) [184](#)
  - [mapQuestClient,](#) [184](#)
  - [RetrieveMapQuestData,](#) [182](#)
  - [TourFactory,](#) [180](#)
  - [uow,](#) [184](#)
  - [UpdateMapQuestTour,](#) [183](#)
  - [UpdateTour,](#) [183](#)
- [BusinessLogicLayer.Factories.TourLogFactory,](#) [188](#)
  - [CreateTourLog,](#) [189](#)
  - [DeleteTourLog,](#) [189](#)
  - [logger,](#) [190](#)
  - [TourLogFactory,](#) [189](#)
  - [uow,](#) [190](#)
  - [UpdateTourLog,](#) [190](#)
- [BusinessLogicLayer.PDFCreation,](#) [14](#)
- [BusinessLogicLayer.PDFCreation.ITourPlannerReportsGenerator,](#)  
[126](#)
  - [BusinessLogicLayer.PDFCreation.SummaryReport,](#) [165](#)
  - [Compose,](#) [167](#)
  - [ComposeContent,](#) [167](#)
  - [ComposeHeader,](#) [167](#)
  - [ComposeManeuverStats,](#) [167](#)
  - [ComposeTourLogStats,](#) [168](#)
  - [ComposeTourStats,](#) [168](#)

- GetMetadata, 168
  - SummaryReport, 166
  - Tours, 168
- BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator,
  - 200
  - config, 202
  - GenerateSummaryReport, 201
  - GenerateTourReport, 202
  - logger, 202
  - TourPlannerReportsGenerator, 201
- BusinessLogicLayer.PDFCreation.TourReport, 203
  - Compose, 204
  - ComposeContent, 204
  - ComposeDetails, 204
  - ComposeHeader, 205
  - ComposeManeuvers, 205
  - ComposeReports, 205
  - ComposeTourLogs, 205
  - GetMetadata, 206
  - Tour, 206
  - TourReport, 203
- CanExecute
  - SWE2\_Tourplanner.Commands.RelayCommand, 161
- CanExecuteChanged
  - SWE2\_Tourplanner.Commands.RelayCommand, 162
- CheckDBConstraints
  - DataAccessLayer.Repositories.ManeuverRepository, 142
  - DataAccessLayer.Repositories.TourLogRepository, 193
  - DataAccessLayer.Repositories.TourRepository, 208
- CIContains
  - Common.Extensions, 75
- client
  - Common.MapQuestClient.MapQuestClient, 152
- Close
  - SWE2\_Tourplanner.Dialogs.IDialog, 93
- CloseCommand
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, 74
- CloseConnection
  - DataAccessLayer.DBConnection.DatabaseConnection, 53
  - DataAccessLayer.DBConnection.IDBConnection, 90
- CloseRequested
  - SWE2\_Tourplanner.Dialogs.IDialogRequestClose, 94
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 37
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 44
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, 75
- SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 104
- SWE2\_Tourplanner.ViewModels.ImportViewModel, 108
- SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 148
- Commit
  - DataAccessLayer.UnitOfWork.IUnitOfWork, 128
  - DataAccessLayer.UnitOfWork.UnitOfWork, 218
- CommitCommand
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 42
- commitCommands
  - DataAccessLayer.Repositories.ManeuverRepository, 145
  - DataAccessLayer.Repositories.TourLogRepository, 195
  - DataAccessLayer.Repositories.TourRepository, 212
  - DataAccessLayer.UnitOfWork.UnitOfWork, 219
- Common, 14
- Common.Config, 14
- Common.Config.ITourPlannerConfig, 124
  - DatabaseConnectionString, 125
  - ExportsDirectory, 125
  - LoadConfigFromFile, 125
  - MapQuestKey, 125
  - PictureDirectory, 126
- Common.Config.TourPlannerConfig, 197
  - GetTourPlannerConfig, 198
  - LoadConfigFromFile, 198
  - logger, 199
  - originalConfigPath, 199
  - SetUpEnvironment, 199
  - TourPlannerConfig, 198
  - tourPlannerConfig, 199
- Common.Entities, 14
- Common.Entities.Maneuver, 138
- Common.Entities.Tour, 172
- Common.Entities.TourLog, 185
- Common.Enums, 15
  - ERouteType, 15
  - ETravelMethod, 15
  - EWeather, 15
- Common.Exceptions, 15
- Common.Exceptions.CommonConfigException, 32
  - CommonConfigException, 32, 33
- Common.Extensions, 75
  - CIContains, 75
- Common.Logging, 16
- Common.Logging.LogHelper, 129
  - GetLogger, 130
  - LogHelper, 130
  - LogHelper, 129
  - LogHelper, 130
- Common.MapQuestClient, 16
- Common.MapQuestClient.IMapQuestClient, 98
  - GetRouteDataFromMapQuest, 99



- RetrieveRouteImage, 99
- Common.MapQuestClient.MapQuestClient, 150
  - client, 152
  - config, 152
  - GetRouteDataFromMapQuest, 151
  - logger, 153
  - MapQuestClient, 151
  - pictureHeight, 153
  - pictureWidth, 153
  - RetrieveRouteImage, 152
  - routeBaseURL, 153
  - staticmapBaseURL, 153
- CommonConfigException
  - Common.Exceptions.CommonConfigException, 32, 33
- Compose
  - BusinessLogicLayer.PDFCreation.SummaryReport, 167
  - BusinessLogicLayer.PDFCreation.TourReport, 204
- ComposeContent
  - BusinessLogicLayer.PDFCreation.SummaryReport, 167
  - BusinessLogicLayer.PDFCreation.TourReport, 204
- ComposeDetails
  - BusinessLogicLayer.PDFCreation.TourReport, 204
- ComposeHeader
  - BusinessLogicLayer.PDFCreation.SummaryReport, 167
  - BusinessLogicLayer.PDFCreation.TourReport, 205
- ComposeManeuvers
  - BusinessLogicLayer.PDFCreation.TourReport, 205
- ComposeManeuverStats
  - BusinessLogicLayer.PDFCreation.SummaryReport, 167
- ComposeReports
  - BusinessLogicLayer.PDFCreation.TourReport, 205
- ComposeTourLogs
  - BusinessLogicLayer.PDFCreation.TourReport, 205
- ComposeTourLogStats
  - BusinessLogicLayer.PDFCreation.SummaryReport, 168
- ComposeTourStats
  - BusinessLogicLayer.PDFCreation.SummaryReport, 168
- ConductCreateUpdateCommand
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 35
- ConductImportCommand
  - SWE2\_Tourplanner.MainViewModel, 134
- config
  - BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator, 202
  - Common.MapQuestClient.MapQuestClient, 152
  - DataAccessLayer.DBConnection.DatabaseConnection, 56
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 107
- ConfigPath
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 103
- configPath
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 103
- ConfirmButtonMessage
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 35
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 42
- ConfirmCommand
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 103
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 148
- ConfirmImportCommand
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 108
- Connection
  - DataAccessLayer.DBConnection.IDbCommand, 89
  - DataAccessLayer.DBConnection.NpgsqlCommand, 157
- Convert
  - SWE2\_Tourplanner.Converters.RouteInfoImageConverter, 163
  - SWE2\_Tourplanner.Converters.StringToDateTimeConverter, 164
- ConvertBack
  - SWE2\_Tourplanner.Converters.RouteInfoImageConverter, 163
  - SWE2\_Tourplanner.Converters.StringToDateTimeConverter, 165
- ConvertToManeuver
  - DataAccessLayer.Repositories.ManeuverRepository, 143
- ConvertToTour
  - DataAccessLayer.Repositories.TourRepository, 210
- ConvertToTourLog
  - DataAccessLayer.Repositories.TourLogRepository, 193
- CreateDelegate
  - XamlGeneratedNamespace.GeneratedInternalTypeHelper, 83
- CreateManeuver
  - XamlGeneratedNamespace.GeneratedInternalTypeHelper, 83
- CreateMapQuestTour
  - BusinessLogicLayer.Factories.ITourFactory, 119
  - BusinessLogicLayer.Factories.TourFactory, 181
- CreateTour
  - BusinessLogicLayer.Factories.ITourFactory, 120
  - BusinessLogicLayer.Factories.TourFactory, 181
- CreateTourLog
  - BusinessLogicLayer.Factories.ITourLogFactory, 122
  - BusinessLogicLayer.Factories.TourLogFactory,

- 189
- CreateUpdateTourLogViewModel
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 189
- 34
- CreateUpdateTourViewModel
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 34
- 41
- CurrentManeuver
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 41
- 148
- currentManeuver
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 148
- 147
- CurrentTourLogs
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 147
- 176
- currentTourLogs
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 176
- 175
- DALDBConnectionException
  - DataAccessLayer.Exceptions.DALDBConnectionException, 45, 46
- 45, 46
- DALParameterException
  - DataAccessLayer.Exceptions.DALParameterException, 47
- 47
- DALRepositoryCommandException
  - DataAccessLayer.Exceptions.DALRepositoryCommandException, 48, 49
- 48, 49
- DALUnitOfWorkException
  - DataAccessLayer.Exceptions.DALUnitOfWorkException, 50
- 50
- DataAccessLayer, 16
- DataAccessLayer.DBCommands, 16
- DataAccessLayer.DBCommands.IDBCommand, 86
  - Execute, 87
  - Undo, 87
- DataAccessLayer.DBCommands.ManeuverCommands, 16
- DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, 57
  - db, 59
  - DeleteManeuverCommand, 58
  - Execute, 58
  - logger, 59
  - maneuver, 59
  - Undo, 58
- DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand, 109
  - db, 110
  - Execute, 110
  - InsertManeuverCommand, 109
  - logger, 110
  - maneuver, 111
  - Undo, 110
- DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 222
  - db, 223
  - Execute, 223
- logger, 223
- maneuver, 224
- Undo, 223
- UpdateManeuverCommand, 222
- DataAccessLayer.DBCommands.TourCommands, 17
  - DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand, 60
    - db, 61
    - DeleteTourCommand, 60
    - Execute, 61
    - tour, 61
    - Undo, 61
  - DataAccessLayer.DBCommands.TourCommands.InsertTourCommand, 111
    - db, 113
    - Execute, 112
    - InsertTourCommand, 112
    - tour, 113
    - Undo, 112
  - DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand, 224
    - db, 226
    - Execute, 225
    - oldTour, 226
    - tour, 226
    - Undo, 225
  - DataAccessLayer.DBCommands.TourLogCommands, 17
    - DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, 62
      - db, 64
      - DeleteTourLogCommand, 63
      - Execute, 63
      - logger, 64
      - tourLog, 64
      - Undo, 63
    - DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand, 113
      - db, 115
      - Execute, 114
      - InsertTourLogCommand, 114
      - logger, 115
      - tourLog, 115
      - Undo, 114
    - DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 227
      - db, 228
      - Execute, 228
      - logger, 228
      - oldTourLog, 229
      - tourLog, 229
      - Undo, 228
      - UpdateTourLogCommand, 227
  - DataAccessLayer.DBConnection, 17
    - DataAccessLayer.DBConnection.DatabaseConnection, 51

- CloseConnection, [53](#)
- config, [56](#)
- DatabaseConnection, [52](#)
- dB, [56](#)
- DeclareParameter, [53](#)
- DefineParameter, [53](#)
- ExecuteStatement, [54](#)
- GetDBConnection, [54](#)
- logger, [56](#)
- npgsqlConnection, [56](#)
- OpenConnection, [55](#)
- QueryDatabase, [55](#)
- DataAccessLayer.DBConnection.IDataReader, [85](#)
  - FieldCount, [86](#)
  - GetValue, [85](#)
  - Read, [86](#)
- DataAccessLayer.DBConnection.IDbCommand, [88](#)
  - Connection, [89](#)
  - ExecuteNonQuery, [88](#)
  - ExecuteReader, [88](#)
  - Parameters, [89](#)
- DataAccessLayer.DBConnection.IDBConnection, [89](#)
  - CloseConnection, [90](#)
  - DeclareParameter, [90](#)
  - DefineParameter, [91](#)
  - ExecuteStatement, [91](#)
  - OpenConnection, [91](#)
  - QueryDatabase, [91](#)
- DataAccessLayer.DBConnection.NpgsqlCommand, [155](#)
  - Connection, [157](#)
  - ExecuteNonQuery, [156](#)
  - ExecuteReader, [156](#)
  - NpgsqlCommand, [155](#)
  - npgsqlCommand, [156](#)
  - Parameters, [157](#)
- DataAccessLayer.DBConnection.NpgsqlDataReader, [157](#)
  - Dispose, [158](#)
  - FieldCount, [159](#)
  - GetValue, [158](#)
  - NpgsqlDataReader, [158](#)
  - npgsqlDataReader, [159](#)
  - Read, [159](#)
- DataAccessLayer.Exceptions, [18](#)
- DataAccessLayer.Exceptions.DALDBConnectionException, [45](#)
  - DALDBConnectionException, [45, 46](#)
- DataAccessLayer.Exceptions.DALParameterException, [46](#)
  - DALParameterException, [47](#)
- DataAccessLayer.Exceptions.DALRepositoryCommandException, [48](#)
  - DALRepositoryCommandException, [48, 49](#)
- DataAccessLayer.Exceptions.DALUnitOfWorkException, [49](#)
  - DALUnitOfWorkException, [50](#)
- DataAccessLayer.Repositories, [18](#)
- DataAccessLayer.Repositories.IManeuverRepository, [98](#)
- DataAccessLayer.Repositories.IRepository< T >, [115](#)
  - Delete, [116](#)
  - Insert, [116](#)
  - Read, [118](#)
  - ReadAll, [118](#)
  - Update, [118](#)
- DataAccessLayer.Repositories.ITourLogRepository, [124](#)
- DataAccessLayer.Repositories.ITourRepository, [127](#)
- DataAccessLayer.Repositories.ManeuverRepository, [141](#)
  - CheckDBConstraints, [142](#)
  - commitCommands, [145](#)
  - ConvertToManeuver, [143](#)
  - db, [145](#)
  - Delete, [143](#)
  - Insert, [143](#)
  - logger, [145](#)
  - ManeuverRepository, [142](#)
  - Read, [144](#)
  - ReadAll, [144](#)
  - Update, [144](#)
- DataAccessLayer.Repositories.TourLogRepository, [191](#)
  - CheckDBConstraints, [193](#)
  - commitCommands, [195](#)
  - ConvertToTourLog, [193](#)
  - db, [196](#)
  - Delete, [194](#)
  - Insert, [194](#)
  - logger, [196](#)
  - Read, [194](#)
  - ReadAll, [195](#)
  - TourLogRepository, [192, 193](#)
  - Update, [195](#)
- DataAccessLayer.Repositories.TourRepository, [206](#)
  - CheckDBConstraints, [208](#)
  - commitCommands, [212](#)
  - ConvertToTour, [210](#)
  - db, [212](#)
  - Delete, [210](#)
  - Insert, [210](#)
  - logger, [212](#)
  - maneuverRepository, [212](#)
  - Read, [211](#)
  - ReadAll, [211](#)
  - tourLogRepository, [212](#)
  - TourRepository, [207, 208](#)
  - Update, [211](#)
- DataAccessLayer.UnitOfWork, [18](#)
- DataAccessLayer.UnitOfWork.IUnitOfWork, [127](#)
  - Commit, [128](#)
  - ManeuverRepository, [128](#)
  - Rollback, [128](#)
  - TourLogRepository, [128](#)
  - TourRepository, [129](#)
- DataAccessLayer.UnitOfWork.UnitOfWork, [216](#)
  - Commit, [218](#)

- commitCommands, [219](#)
- db, [219](#)
- Dispose, [218](#)
- logger, [219](#)
- ManeuverRepository, [220](#)
- maneuverRepository, [220](#)
- Rollback, [219](#)
- rollbackCommands, [220](#)
- TourLogRepository, [221](#)
- tourLogRepository, [220](#)
- TourRepository, [221](#)
- tourRepository, [220](#)
- UnitOfWork, [217](#), [218](#)
- DatabaseConnection
  - DataAccessLayer.DBConnection.DatabaseConnection, [52](#)
- DatabaseConnectionString
  - Common.Config.ITourPlannerConfig, [125](#)
- DataContext
  - SWE2\_Tourplanner.Dialogs.IDialog, [93](#)
- dB
  - DataAccessLayer.DBConnection.DatabaseConnection, [56](#)
- db
  - DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, [59](#)
  - DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand, [110](#)
  - DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, [223](#)
  - DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand, [61](#)
  - DataAccessLayer.DBCommands.TourCommands.InsertTourCommand, [113](#)
  - DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand, [226](#)
  - DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, [64](#)
  - DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand, [115](#)
  - DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, [228](#)
  - DataAccessLayer.Repositories.ManeuverRepository, [145](#)
  - DataAccessLayer.Repositories.TourLogRepository, [196](#)
  - DataAccessLayer.Repositories.TourRepository, [212](#)
  - DataAccessLayer.UnitOfWork.UnitOfWork, [219](#)
- DeclareParameter
  - DataAccessLayer.DBConnection.DatabaseConnection, [53](#)
  - DataAccessLayer.DBConnection.IDBConnection, [90](#)
- DefineParameter
  - DataAccessLayer.DBConnection.DatabaseConnection, [53](#)
  - DataAccessLayer.DBConnection.IDBConnection, [90](#)
- Delete
  - DataAccessLayer.Repositories.IRepository< T >, [116](#)
  - DataAccessLayer.Repositories.ManeuverRepository, [143](#)
  - DataAccessLayer.Repositories.TourLogRepository, [194](#)
  - DataAccessLayer.Repositories.TourRepository, [210](#)
  - DeleteManeuverCommand
    - DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, [58](#)
  - DeleteTour
    - BusinessLogicLayer.Factories.ITourFactory, [120](#)
    - BusinessLogicLayer.Factories.TourFactory, [181](#)
  - DeleteTourCommand
    - DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand, [60](#)
  - DeleteTourLog
    - BusinessLogicLayer.Factories.ITourLogFactory, [123](#)
    - BusinessLogicLayer.Factories.TourLogFactory, [189](#)
  - DeleteTourLogCommand
    - DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, [63](#)
  - Description
    - SWE2\_Tourplanner.Models.CreateUpdateTourViewModel, [42](#)
  - DialogCloseRequestedEventArgs
    - SWE2\_Tourplanner.Dialogs.DialogCloseRequestedEventArgs, [65](#)
  - DialogResult
    - SWE2\_Tourplanner.Dialogs.DialogCloseRequestedEventArgs, [65](#)
  - DialogService
    - SWE2\_Tourplanner.Dialogs.IDialog, [93](#)
    - SWE2\_Tourplanner.Dialogs.DialogService, [66](#)
  - dialogService
    - SWE2\_Tourplanner.MainViewModel, [133](#)
    - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, [175](#)
    - SWE2\_Tourplanner.ViewModels.ToursViewModel, [215](#)
  - Dispose
    - DataAccessLayer.DBConnection.NpgsqlDataReader, [158](#)
    - DataAccessLayer.UnitOfWork.UnitOfWork, [218](#)
  - Distance
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, [35](#)
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, [42](#)
  - EditLogCommand
    - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, [176](#)
  - EditTourCommand

- SWE2\_Tourplanner.MainViewModel, [134](#)
- EndDate
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, [46](#)
  - [35](#)
- EndLocation
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, [54](#)
  - [42](#)
- ERouteType
  - Common.Enums, [15](#)
- ErrorHandlingMessage
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, [74](#)
- errorHandlingMessage
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, [74](#)
- ErrorType
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, [74](#)
- errorType
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, [74](#)
- ErrorViewModel
  - SWE2\_Tourplanner.ViewModels.ErrorViewModel, [73](#)
- ETravelMethod
  - Common.Enums, [15](#)
- EWeather
  - Common.Enums, [15](#)
- Execute
  - DataAccessLayer.DBCommands.IDBCommand, [87](#)
  - DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, [58](#)
  - DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand, [110](#)
  - DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, [223](#)
  - DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand, [61](#)
  - DataAccessLayer.DBCommands.TourCommands.InsertTourCommand, [112](#)
  - DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand, [225](#)
  - DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, [63](#)
  - DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand, [114](#)
  - DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, [228](#)
  - SWE2\_Tourplanner.Commands.RelayCommand, [161](#)
- ExecuteNonQuery
  - DataAccessLayer.DBConnection.IDbCommand, [88](#)
  - DataAccessLayer.DBConnection.NpgsqlCommand, [156](#)
- ExecuteReader
  - DataAccessLayer.DBConnection.IDbCommand, [88](#)
  - DataAccessLayer.DBConnection.NpgsqlCommand, [156](#)
- DataAccessLayer.DBConnection.NpgsqlCommand, [156](#)
- ExecuteStatement
  - DataAccessLayer.DBConnection.DatabaseConnection, [91](#)
  - DataAccessLayer.DBConnection.IDBConnection, [91](#)
- ExitCommand
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, [35](#)
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, [42](#)
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, [103](#)
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, [108](#)
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, [148](#)
- Export
  - BusinessLogicLayer.ExporterImporter.IExporterImporter< T >, [97](#)
  - BusinessLogicLayer.ExporterImporter.TourExporterImporter, [178](#)
- exporterImporter
  - SWE2\_Tourplanner.MainViewModel, [133](#)
- ExportPicture
  - BusinessLogicLayer.ExporterImporter.TourExporterImporter, [178](#)
- ExportsDirectory
  - Common.Config.ITourPlannerConfig, [125](#)
- ExternalResourceControl
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, [78](#)
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, [78](#)
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, [81](#)
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, [80](#)
- FieldCount
  - DataAccessLayer.DBConnection.IDataReader, [86](#)
  - DataAccessLayer.DBConnection.NpgsqlDataReader, [159](#)
- FilteredTours
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, [215](#)
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, [215](#)
- GenerateJSONExportCommand
  - SWE2\_Tourplanner.MainViewModel, [134](#)
- GenerateSummaryReport

- BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator, 201
- GenerateSummaryReportCommand
  - SWE2\_Tourplanner.MainViewModel, 134
- GenerateTourReport
  - BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator, 202
- GenerateTourReportCommand
  - SWE2\_Tourplanner.MainViewModel, 134
- GetDBConnection
  - DataAccessLayer.DBConnection.DatabaseConnection, 54
- GetLogger
  - Common.Logging.LogHelper, 130
- GetLogHelper
  - Common.Logging.LogHelper, 130
- GetMetadata
  - BusinessLogicLayer.PDFCreation.SummaryReport, 168
  - BusinessLogicLayer.PDFCreation.TourReport, 206
- GetPropertyValue
  - XamlGeneratedNamespace.GeneratedInternalTypeHelper, 84
- GetRouteDataFromMapQuest
  - Common.MapQuestClient.IMapQuestClient, 99
  - Common.MapQuestClient.MapQuestClient, 151
- GetTourPlannerConfig
  - Common.Config.TourPlannerConfig, 198
- GetTours
  - BusinessLogicLayer.Factories.ITourFactory, 120
  - BusinessLogicLayer.Factories.TourFactory, 182
- GetValue
  - DataAccessLayer.DBConnection.IDataReader, 85
  - DataAccessLayer.DBConnection.NpgsqlDataReader, 158
- googleFaviconCrawler
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 80
- HeadingMessage
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
- Import
  - BusinessLogicLayer.ExporterImporter.IExporterImporter, 97
  - BusinessLogicLayer.ExporterImporter.TourExporterImporter, 178
- ImportConfigViewModel
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 102
- ImportViewModel
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 107
- InitializeComponent
  - SWE2\_Tourplanner.App, 24
  - SWE2\_Tourplanner.CreateUpdateTourView, 39
- SWE2\_Tourplanner.ErrorView, 71
- SWE2\_Tourplanner.ImportConfigView, 101
- SWE2\_Tourplanner.ImportView, 105, 106
- SWE2\_Tourplanner.MainWindow, 137
- SWE2\_Tourplanner.ManeuverManagementWindow, 140
- SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 78, 79
- SWE2\_Tourplanner.ReusableComponents.TemperatureSliders, 169
- SWE2\_Tourplanner.TourLogCreateUpdateView, 187
- SWE2\_Tourplanner.ViewModels.TemperatureSlidersViewModel, 170
- Insert
  - DataAccessLayer.Repositories.IRepository< T >, 116
  - DataAccessLayer.Repositories.ManeuverRepository, 143
  - DataAccessLayer.Repositories.TourLogRepository, 194
  - DataAccessLayer.Repositories.TourRepository, 210
- InsertManeuverCommand
  - DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuver, 109
- InsertTourCommand
  - DataAccessLayer.DBCommands.TourCommands.InsertTourCommand, 112
- InsertTourLogCommand
  - DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogC, 114
- JsonPaths
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 108
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 107
- Link
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 81
- LinkProperty
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 80
- LinkText
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 81
- LinkTextProperty
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 80
- LoadConfigCommand
  - SWE2\_Tourplanner.MainViewModel, 135
- LoadConfigFromFile
  - Common.Config.ITourPlannerConfig, 125
  - Common.Config.TourPlannerConfig, 198
- logger



- BusinessLogicLayer.ExporterImporter.TourExporterImporter, 179
- BusinessLogicLayer.Factories.TourFactory, 184
- BusinessLogicLayer.Factories.TourLogFactory, 190
- BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator, 202
- Common.Config.TourPlannerConfig, 199
- Common.MapQuestClient.MapQuestClient, 153
- DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, 59
- DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand, 110
- DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 223
- DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, 64
- DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand, 115
- DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 228
- DataAccessLayer.DBConnection.DatabaseConnection, 56
- DataAccessLayer.Repositories.ManeuverRepository, 145
- DataAccessLayer.Repositories.TourLogRepository, 196
- DataAccessLayer.Repositories.TourRepository, 212
- DataAccessLayer.UnitOfWork.UnitOfWork, 219
- LogHelper
  - Common.Logging.LogHelper, 129
- logHelper
  - Common.Logging.LogHelper, 130
- Main
  - SWE2\_Tourplanner.App, 25
- MainViewModel
  - SWE2\_Tourplanner.MainViewModel, 132
- ManageManeuverCommand
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
- maneuver
  - DataAccessLayer.DBCommands.ManeuverCommands.DeleteManeuverCommand, 59
  - DataAccessLayer.DBCommands.ManeuverCommands.InsertManeuverCommand, 111
  - DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 224
- ManeuverRepository
  - DataAccessLayer.Repositories.ManeuverRepository, 142
  - DataAccessLayer.UnitOfWork.IUnitOfWork, 128
  - DataAccessLayer.UnitOfWork.UnitOfWork, 220
- maneuverRepository
  - DataAccessLayer.Repositories.TourRepository, 212
  - DataAccessLayer.UnitOfWork.UnitOfWork, 220
- Maneuvers
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 148
  - maneuvers
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 41
    - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 147
  - ManeuversViewModel
    - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 147
  - maneuversViewModel
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 41
  - ManualTour
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
  - manualTour
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 41
  - MapQuestClient
    - Common.MapQuestClient.MapQuestClient, 151
  - mapQuestClient
    - BusinessLogicLayer.Factories.TourFactory, 184
  - MapQuestKey
    - Common.Config.ITourPlannerConfig, 125
  - Max
    - SWE2\_Tourplanner.ValidationRules.DoubleRangeRule, 69
    - SWE2\_Tourplanner.ValidationRules.TextInputValidationRule, 172
  - Min
    - SWE2\_Tourplanner.ValidationRules.DoubleRangeRule, 69
    - SWE2\_Tourplanner.ValidationRules.TextInputValidationRule, 172
  - name
    - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
  - npqsqlCommand
    - DataAccessLayer.DBConnection.NpgsqlCommand, 155
  - npqsqlCommand
    - DataAccessLayer.DBConnection.NpgsqlCommand, 156
  - npqsqlConnection
    - DataAccessLayer.DBConnection.DatabaseConnection, 56
  - NpgsqlDataReader
    - DataAccessLayer.DBConnection.NpgsqlDataReader, 158
  - npqsqlDataReader
    - DataAccessLayer.DBConnection.NpgsqlDataReader, 159

- oldManeuver
  - DataAccessLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 86
  - 224
- oldTour
  - DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand, 118
  - 226
- oldTourLog
  - DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 144
  - 229
- OnLinkChanged
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 211
  - 79
- OnPropertyChanged
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 118
  - 79
  - SWE2\_Tourplanner.ViewModels.BaseViewModel, 27
- OnStartup
  - SWE2\_Tourplanner.App, 25
- OpenConnection
  - DataAccessLayer.DBConnection.DatabaseConnection, 55
  - DataAccessLayer.DBConnection.IDBConnection, 91
- originalConfigPath
  - Common.Config.TourPlannerConfig, 199
- Owner
  - SWE2\_Tourplanner.Dialogs.IDialog, 93
- owner
  - SWE2\_Tourplanner.Dialogs.DialogService, 68
- Parameters
  - DataAccessLayer.DBConnection.IDbCommand, 89
  - DataAccessLayer.DBConnection.NpgsqlCommand, 157
- pdfGenerator
  - SWE2\_Tourplanner.MainViewModel, 133
- PictureDirectory
  - Common.Config.ITourPlannerConfig, 126
- pictureHeight
  - Common.MapQuestClient.MapQuestClient, 153
- pictureWidth
  - Common.MapQuestClient.MapQuestClient, 153
- PropertyChanged
  - SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 81
  - SWE2\_Tourplanner.ViewModels.BaseViewModel, 27
- QueryDatabase
  - DataAccessLayer.DBConnection.DatabaseConnection, 55
  - DataAccessLayer.DBConnection.IDBConnection, 91
- Rating
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- Read
  - DataAccessLayer.DBConnection.IDataReader, 86
  - DataAccessLayer.DBConnection.NpgsqlDataReader, 159
- ReadAll
  - DataAccessLayer.Repositories.IRepository< T >, 118
  - DataAccessLayer.Repositories.ManeuverRepository, 144
  - DataAccessLayer.Repositories.TourLogRepository, 195
  - DataAccessLayer.Repositories.TourRepository, 211
- Register< TViewModel, TView >
  - SWE2\_Tourplanner.Dialogs.DialogService, 66
  - SWE2\_Tourplanner.Dialogs.IDialogService, 95
- RelayCommand
  - SWE2\_Tourplanner.Commands.RelayCommand, 160
- RemoveLogCommand
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 176
- RemoveManeuverCommand
  - SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 148
- RemoveTourCommand
  - SWE2\_Tourplanner.MainViewModel, 135
- Report
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- RetrieveMapQuestData
  - BusinessLogicLayer.Factories.TourFactory, 182
- RetrieveRouteImage
  - Common.MapQuestClient.IMapQuestClient, 99
  - Common.MapQuestClient.MapQuestClient, 152
- Rollback
  - DataAccessLayer.UnitOfWork.IUnitOfWork, 128
  - DataAccessLayer.UnitOfWork.UnitOfWork, 219
- rollbackCommands
  - DataAccessLayer.UnitOfWork.UnitOfWork, 220
- routeBaseURL
  - Common.MapQuestClient.MapQuestClient, 153
- RouteInfo
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
- RouteType
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43
- RouteTypes
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 43



- 44
- SearchString
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 216
- searchString
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 215
- SearchToursCommand
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 216
- SelectConfigCommand
  - SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 103
- SelectedTour
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 176
- SelectedTourLog
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 176
- selectedTourLog
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 175
- SelectPathsCommand
  - SWE2\_Tourplanner.ViewModels.ImportViewModel, 108
- SelectRouteInfoCommand
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 44
- SetPropertyValue
  - XamlGeneratedNamespace.GeneratedInternalTypeHelper, 84
- SetUpEnvironment
  - Common.Config.TourPlannerConfig, 199
- ShowDialog
  - SWE2\_Tourplanner.Dialogs.IDialog, 93
- ShowDialog< TViewModel >
  - SWE2\_Tourplanner.Dialogs.DialogService, 67
  - SWE2\_Tourplanner.Dialogs.IDialogService, 95
- ShowException
  - SWE2\_Tourplanner.MainViewModel, 132
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 174
- StartDate
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- StartLocation
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 44
- staticmapBaseUrl
  - Common.MapQuestClient.MapQuestClient, 153
- SummaryReport
  - BusinessLogicLayer.PDFCreation.SummaryReport, 166
- SWE2\_Tourplanner, 19
- SWE2\_Tourplanner.App, 23
  - InitializeComponent, 24
  - Main, 25
  - OnStartup, 25
- SWE2\_Tourplanner.Commands, 19
- SWE2\_Tourplanner.Commands.RelayCommand, 160
  - \_canExecute, 161
  - \_execute, 162
  - CanExecute, 161
  - CanExecuteChanged, 162
  - Execute, 161
  - RelayCommand, 160
- SWE2\_Tourplanner.Converters, 19
- SWE2\_Tourplanner.Converters.RouteInfoImageConverter, 162
  - Convert, 163
  - ConvertBack, 163
- SWE2\_Tourplanner.Converters.StringToDateTimeConverter, 164
  - Convert, 164
  - ConvertBack, 165
- SWE2\_Tourplanner.CreateUpdateTourView, 38
  - InitializeComponent, 39
- SWE2\_Tourplanner.Dialogs, 20
- SWE2\_Tourplanner.Dialogs.DialogCloseRequestedEventArgs, 64
  - DialogCloseRequestedEventArgs, 65
  - DialogResult, 65
- SWE2\_Tourplanner.Dialogs.DialogService, 65
  - DialogService, 66
  - Mappings, 68
  - Owner, 68
  - Register< TViewModel, TView >, 66
  - ShowDialog< TViewModel >, 67
- SWE2\_Tourplanner.Dialogs.IDialog, 92
  - Close, 93
  - DataContext, 93
  - DialogResult, 93
  - Owner, 93
  - ShowDialog, 93
- SWE2\_Tourplanner.Dialogs.IDialogRequestClose, 94
  - CloseRequested, 94
- SWE2\_Tourplanner.Dialogs.IDialogService, 95
  - Register< TViewModel, TView >, 95
  - ShowDialog< TViewModel >, 95
- SWE2\_Tourplanner.ErrorView, 70
  - InitializeComponent, 71
- SWE2\_Tourplanner.ImportConfigView, 100
  - InitializeComponent, 101
- SWE2\_Tourplanner.ImportView, 104
  - InitializeComponent, 105, 106
- SWE2\_Tourplanner.MainViewModel, 131
  - AddTourCommand, 134
  - ConductImportCommand, 134
  - dialogService, 133
  - EditTourCommand, 134
  - exporterImporter, 133
  - GenerateJSONExportCommand, 134
  - GenerateSummaryReportCommand, 134
  - GenerateTourReportCommand, 134
  - LoadConfigCommand, 135
  - MainViewModel, 132

- pdfGenerator, 133
- RemoveTourCommand, 135
- ShowException, 132
- TourDetailViewModel, 135
- tourDetailViewModel, 133
- tourPlannerFactory, 133
- ToursViewModel, 135
- toursViewModel, 134
- SWE2\_Tourplanner.MainWindow, 136
  - InitializeComponent, 137
- SWE2\_Tourplanner.ManeuverManagementWindow, 138
  - InitializeComponent, 140
- SWE2\_Tourplanner.ReusableComponents, 20
- SWE2\_Tourplanner.ReusableComponents.ExternalResourceControl, 76
  - ExternalResource\_MouseDown, 78
  - ExternalResourceControl, 78
  - Favicon, 81
  - faviconLocation, 80
  - googleFaviconCrawler, 80
  - InitializeComponent, 78, 79
  - Link, 81
  - LinkProperty, 80
  - LinkText, 81
  - LinkTextProperty, 80
  - OnLinkChanged, 79
  - OnPropertyChanged, 79
  - PropertyChanged, 81
- SWE2\_Tourplanner.ReusableComponents.TemperatureSliders, 169
  - InitializeComponent, 169
- SWE2\_Tourplanner.TourLogCreateUpdateView, 186
  - InitializeComponent, 187
- SWE2\_Tourplanner.ValidationRules, 20
- SWE2\_Tourplanner.ValidationRules.DoubleRangeRule, 68
  - Max, 69
  - Min, 69
  - Validate, 69
- SWE2\_Tourplanner.ValidationRules.ManeuverValidationRule, 149
  - Validate, 149
- SWE2\_Tourplanner.ValidationRules.NoNegativeDistancesRule, 154
  - Validate, 154
- SWE2\_Tourplanner.ValidationRules.TextInputValidationRule, 171
  - Max, 172
  - Min, 172
  - Validate, 171
- SWE2\_Tourplanner.ViewModels, 21
- SWE2\_Tourplanner.ViewModels.BaseViewModel, 26
  - OnPropertyChanged, 27
  - PropertyChanged, 27
- SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 33
  - CloseRequested, 37
  - ConductCreateUpdateCommand, 35
  - ConfirmButtonMessage, 35
  - CreateUpdateTourLogViewModel, 34
  - Distance, 35
  - EndDate, 35
  - ExitCommand, 35
  - HeadingMessage, 36
  - Rating, 36
  - Report, 36
  - StartDate, 36
  - Temperature, 36
  - tourLog, 35
  - TravelMethod, 36
  - TravelMethods, 36
  - Weather, 37
  - WeatherTypes, 37
- SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 40
  - CloseRequested, 44
  - CommitCommand, 42
  - ConfirmButtonMessage, 42
  - CreateUpdateTourViewModel, 41
  - Description, 42
  - Distance, 42
  - EndLocation, 42
  - ExitCommand, 42
  - HeadingMessage, 43
  - ManageManeuverCommand, 43
  - Maneuvers, 43
  - maneuvers, 41
  - maneuversViewModel, 41
  - ManualTour, 43
  - manualTour, 41
  - Name, 43
  - RouteInfo, 43
  - RouteType, 43
  - RouteTypes, 44
  - SelectRouteInfoCommand, 44
  - StartLocation, 44
  - tour, 42
- SWE2\_Tourplanner.ViewModels.ErrorViewModel, 72
  - CloseCommand, 74
  - CloseRequested, 75
  - ErrorHandlingMessage, 74
  - errorHandlingMessage, 74
  - ErrorType, 74
  - errorType, 74
  - ErrorViewModel, 73
- SWE2\_Tourplanner.ViewModels.ImportConfigViewModel, 102
  - CloseRequested, 104
  - ConfigPath, 103
  - configPath, 103
  - ConfirmCommand, 103
  - ExitCommand, 103
  - ImportConfigViewModel, 102
  - SelectConfigCommand, 103
- SWE2\_Tourplanner.ViewModels.ImportViewModel, 106

- CloseRequested, 108
- config, 107
- ConfirmImportCommand, 108
- ExitCommand, 108
- ImportViewModel, 107
- JsonPaths, 108
- jsonPaths, 107
- SelectPathsCommand, 108
- SWE2\_Tourplanner.ViewModels.ManeuversViewModel, 146
  - AddManeuverCommand, 147
  - CloseRequested, 148
  - ConfirmCommand, 148
  - CurrentManeuver, 148
  - currentManeuver, 147
  - ExitCommand, 148
  - Maneuvers, 148
  - maneuvers, 147
  - ManeuversViewModel, 147
  - RemoveManeuverCommand, 148
- SWE2\_Tourplanner.ViewModels.TemperatureSlidersViewModel, 170
  - InitializeComponent, 170
- SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 173
  - \_selectedTour, 175
  - AddLogCommand, 176
  - CurrentTourLogs, 176
  - currentTourLogs, 175
  - dialogService, 175
  - EditLogCommand, 176
  - RemoveLogCommand, 176
  - SelectedTour, 176
  - SelectedTourLog, 176
  - selectedTourLog, 175
  - ShowException, 174
  - TourDetailViewModel, 174
  - tourLogFactory, 175
- SWE2\_Tourplanner.ViewModels.ToursViewModel, 213
  - \_tours, 215
  - dialogService, 215
  - FilteredTours, 215
  - filteredTours, 215
  - SearchString, 216
  - searchString, 215
  - SearchToursCommand, 216
  - tourPlannerFactory, 215
  - Tours, 216
  - ToursViewModel, 214
- SWE2\_Tourplanner\_Tests, 21
  - SWE2\_Tourplanner\_Tests.BLTests, 21
    - SWE2\_Tourplanner\_Tests.BLTests.TourFactoryTests, 184
    - SWE2\_Tourplanner\_Tests.BLTests.TourLogFactoryTest, 191
  - SWE2\_Tourplanner\_Tests.CommonTests, 21
  - SWE2\_Tourplanner\_Tests.CommonTests.ExtensionMethodsTest, 75
  - SWE2\_Tourplanner\_Tests.CommonTests.LogHelperTests, 131
  - SWE2\_Tourplanner\_Tests.CommonTests.TourPlannerConfigTests, 200
  - SWE2\_Tourplanner\_Tests.DALTests, 22
    - SWE2\_Tourplanner\_Tests.DALTests.DBConnectionTests, 57
    - SWE2\_Tourplanner\_Tests.DALTests.ManeuverRepositoryTests, 145
    - SWE2\_Tourplanner\_Tests.DALTests.TourLogRepositoryTests, 196
    - SWE2\_Tourplanner\_Tests.DALTests.TourRepositoryTests, 213
    - SWE2\_Tourplanner\_Tests.DALTests.UnitOfWorkTests, 221
  - SWE2\_Tourplanner\_Tests.UITests, 22
    - SWE2\_Tourplanner\_Tests.UITests.ConverterTests, 33
    - SWE2\_Tourplanner\_Tests.UITests.ValidationRulesTests, 229
- Temperature
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- Tour
  - BusinessLogicLayer.PDFCreation.TourReport, 206
- tour
  - DataAccessLayer.DBCommands.TourCommands.DeleteTourCommand, 61
  - DataAccessLayer.DBCommands.TourCommands.InsertTourCommand, 113
  - DataAccessLayer.DBCommands.TourCommands.UpdateTourCommand, 226
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourViewModel, 42
- TourDetailViewModel
  - SWE2\_Tourplanner.MainViewModel, 135
  - SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 174
- tourDetailViewModel
  - SWE2\_Tourplanner.MainViewModel, 133
- TourExporterImporter
  - BusinessLogicLayer.ExporterImporter.TourExporterImporter, 177
- TourFactory
  - BusinessLogicLayer.Factories.TourFactory, 180
- tourLog
  - DataAccessLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, 64
  - DataAccessLayer.DBCommands.TourLogCommands.InsertTourLogCommand, 115
  - DataAccessLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 229
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 35
- TourLogFactory
  - BusinessLogicLayer.Factories.TourLogFactory, 189
- tourLogFactory

- SWE2\_Tourplanner.ViewModels.TourDetailViewModel, 175
- TourLogRepository
  - DataContextLayer.Repositories.TourLogRepository, 192, 193
  - DataContextLayer.UnitOfWork.IUnitOfWork, 128
  - DataContextLayer.UnitOfWork.UnitOfWork, 221
- tourLogRepository
  - DataContextLayer.Repositories.TourRepository, 212
  - DataContextLayer.UnitOfWork.UnitOfWork, 220
- TourPlannerConfig
  - Common.Config.TourPlannerConfig, 198
- tourPlannerConfig
  - BusinessLogicLayer.ExporterImporter.TourExporterImporter, 179
  - Common.Config.TourPlannerConfig, 199
- tourPlannerFactory
  - SWE2\_Tourplanner.MainViewModel, 133
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 215
- TourPlannerReportsGenerator
  - BusinessLogicLayer.PDFCreation.TourPlannerReportsGenerator, 201
- TourReport
  - BusinessLogicLayer.PDFCreation.TourReport, 203
- TourRepository
  - DataContextLayer.Repositories.TourRepository, 207, 208
  - DataContextLayer.UnitOfWork.IUnitOfWork, 129
  - DataContextLayer.UnitOfWork.UnitOfWork, 221
- tourRepository
  - DataContextLayer.UnitOfWork.UnitOfWork, 220
- Tours
  - BusinessLogicLayer.PDFCreation.SummaryReport, 168
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 216
- ToursViewModel
  - SWE2\_Tourplanner.MainViewModel, 135
  - SWE2\_Tourplanner.ViewModels.ToursViewModel, 214
- toursViewModel
  - SWE2\_Tourplanner.MainViewModel, 134
- TravelMethod
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- TravelMethods
  - SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel, 36
- Undo
  - DataContextLayer.DBCommands.IDBCommand, 87
  - DataContextLayer.DBCommands.ManeuverCommands.DeleteTourCommand, 58
  - DataContextLayer.DBCommands.ManeuverCommands.InsertTourCommand, 110
- DataContextLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 223
- DataContextLayer.DBCommands.TourCommands.DeleteTourCommand, 61
- DataContextLayer.DBCommands.TourCommands.InsertTourCommand, 112
- DataContextLayer.DBCommands.TourCommands.UpdateTourCommand, 225
- DataContextLayer.DBCommands.TourLogCommands.DeleteTourLogCommand, 63
- DataContextLayer.DBCommands.TourLogCommands.InsertTourLogCommand, 114
- DataContextLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 228
- UnitOfWork
  - DataContextLayer.UnitOfWork.UnitOfWork, 217, 218
- uow
  - BusinessLogicLayer.Factories.TourFactory, 184
  - BusinessLogicLayer.Factories.TourLogFactory, 190
- Update
  - DataContextLayer.Repositories.IRepository< T >, 118
  - DataContextLayer.Repositories.ManeuverRepository, 144
  - DataContextLayer.Repositories.TourLogRepository, 195
  - DataContextLayer.Repositories.TourRepository, 211
- UpdateManeuverCommand
  - DataContextLayer.DBCommands.ManeuverCommands.UpdateManeuverCommand, 222
- UpdateMapQuestTour
  - BusinessLogicLayer.Factories.ITourFactory, 121
  - BusinessLogicLayer.Factories.TourFactory, 183
- UpdateTour
  - BusinessLogicLayer.Factories.ITourFactory, 121
  - BusinessLogicLayer.Factories.TourFactory, 183
- UpdateTourCommand
  - DataContextLayer.DBCommands.TourCommands.UpdateTourCommand, 225
- UpdateTourLog
  - BusinessLogicLayer.Factories.ITourLogFactory, 123
  - BusinessLogicLayer.Factories.TourLogFactory, 190
- UpdateTourLogCommand
  - DataContextLayer.DBCommands.TourLogCommands.UpdateTourLogCommand, 227
- Validate
  - SWE2\_Tourplanner.ValidationRules.DoubleRangeRule, 69
  - SWE2\_Tourplanner.ValidationRules.ManeuverValidationRule, 149
  - SWE2\_Tourplanner.ValidationRules.NoNegativeDistancesRule, 154

SWE2\_Tourplanner.ValidationRules.TextInputValidationRule,  
[171](#)

#### Weather

SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel,  
[37](#)

#### WeatherTypes

SWE2\_Tourplanner.ViewModels.CreateUpdateTourLogViewModel,  
[37](#)

XamlGeneratedNamespace, [22](#)

XamlGeneratedNamespace.GeneratedInternalTypeHelper,  
[82](#)

AddEventHandler, [82](#), [83](#)

CreateDelegate, [83](#)

CreateInstance, [83](#)

GetPropertyValue, [84](#)

SetPropertyValue, [84](#)