

Interconnect Backend

Generated by Doxygen 1.13.2

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 Namespace Documentation	13
4.1 BackgroundServices Namespace Reference	13
4.2 BackgroundServices.Impl Namespace Reference	13
4.3 Controllers Namespace Reference	13
4.4 ControllersTests Namespace Reference	14
4.5 Database Namespace Reference	14
4.6 Database.Migrations Namespace Reference	14
4.7 Hubs Namespace Reference	14
4.8 HubsTests Namespace Reference	15
4.9 Interconnect Namespace Reference	15
4.10 Interconnect.Middlewares Namespace Reference	15
4.11 InterconnectTests Namespace Reference	15
4.12 Mappers Namespace Reference	15
4.13 Models Namespace Reference	16
4.14 Models.Config Namespace Reference	16
4.15 Models.Database Namespace Reference	16
4.16 Models.DTO Namespace Reference	17
4.17 Models.Enums Namespace Reference	17
4.17.1 Enumeration Type Documentation	17
4.17.1.1 ConnectionStatus	17
4.17.1.2 EntityType	17
4.17.1.3 VirtualMachineState	18
4.18 Models.Requests Namespace Reference	19
4.19 Models.Responses Namespace Reference	19
4.20 NativeLibrary Namespace Reference	19
4.21 NativeLibrary.Interop Namespace Reference	20
4.22 NativeLibrary.Structs Namespace Reference	20
4.23 NativeLibrary.Utils Namespace Reference	20
4.24 NativeLibrary.Utils.Impl Namespace Reference	20
4.25 NativeLibrary.Wrappers Namespace Reference	20
4.26 NativeLibrary.Wrappers.Impl Namespace Reference	20
4.27 Repositories Namespace Reference	21
4.28 Repositories.Impl Namespace Reference	21
4.29 RepositoriesTests Namespace Reference	21

4.30 Services Namespace Reference	22
4.31 Services.Impl Namespace Reference	22
4.32 Services.Utils Namespace Reference	23
4.33 ServicesTests Namespace Reference	23
4.34 ServicesTests.Utils Namespace Reference	23
4.35 TestUtils Namespace Reference	23
5 Class Documentation	25
5.1 Database.Migrations.AppDbContextModelSnapshot Class Reference	25
5.2 Services.Utils.BaseDefinitionBuilder< TDefinition > Class Template Reference	25
5.3 Models.BaseEntity Class Reference	26
5.3.1 Detailed Description	26
5.4 Models.Responses.BaseResponse< TResponse, TSelf > Class Template Reference	27
5.4.1 Detailed Description	27
5.4.2 Member Function Documentation	28
5.4.2.1 WithEmptySuccess()	28
5.4.2.2 WithSuccess()	28
5.5 Database.Migrations.BootableDiskModel Class Reference	29
5.5.1 Detailed Description	29
5.6 Models.Database.BootableDiskModel Class Reference	29
5.6.1 Detailed Description	29
5.7 Models.DTO.BootableDiskModelDTO Class Reference	30
5.8 Services.Impl.BootableDiskProviderService Class Reference	30
5.8.1 Member Function Documentation	30
5.8.1.1 GetAvailableBootableDiskModels()	30
5.8.1.2 GetBootableDiskPathById()	30
5.9 ServicesTests.BootableDiskProviderServiceTests Class Reference	31
5.10 Repositories.Impl.BootableDiskRepository Class Reference	31
5.10.1 Member Function Documentation	31
5.10.1.1 GetById()	31
5.10.1.2 GetOnlyWithNotNullablePath()	32
5.11 RepositoriesTests.BootableDiskRepositoryTests Class Reference	32
5.12 Models.Responses.BootableDisksResponse Class Reference	32
5.13 Models.Responses.CapturedPacketResponse Class Reference	33
5.14 Database.Migrations.ChangesToConnection Class Reference	34
5.14.1 Detailed Description	34
5.15 Database.Migrations.ChangeTypeOfVMEntityStrings Class Reference	34
5.15.1 Detailed Description	35
5.16 Models.Requests.ConnectEntitiesRequest Class Reference	35
5.17 Models.ConnectionInfo Class Reference	35
5.17.1 Detailed Description	36
5.18 Models.Responses.ConnectionInfoResponse Class Reference	36

5.19 Hubs.ConnectionStatusHub Class Reference	37
5.19.1 Detailed Description	37
5.19.2 Member Function Documentation	37
5.19.2.1 Ping()	37
5.20 HubsTests.ConnectionStatusHubTests Class Reference	37
5.21 Models.Requests.CreateVirtualMachineEntityRequest Class Reference	38
5.22 Models.Requests.CreateVirtualNetworkNodeEntityRequest Class Reference	38
5.23 Database.Migrations.DeviceDefinition Class Reference	38
5.23.1 Detailed Description	39
5.24 Services.Impl.EntitiesConnectorService Class Reference	39
5.24.1 Member Function Documentation	39
5.24.1.1 ConnectTwoEntities()	39
5.24.1.2 ConnectTwoVirtualMachines()	40
5.24.1.3 ConnectVirtualMachineToInternet()	40
5.24.1.4 ConnectVirtualMachineToVirtualNetworkNode()	41
5.25 Services.Impl.EntitiesDisconnectorService Class Reference	41
5.25.1 Member Function Documentation	42
5.25.1.1 DisconnectEntities()	42
5.25.1.2 DisconnectVirtualMachineFromInternet()	42
5.25.1.3 DisconnectVirtualMachineFromVirtualMachine()	42
5.25.1.4 DisconnectVirtualMachineFromVirtualNetworkNode()	43
5.26 Controllers.EntityController Class Reference	43
5.26.1 Detailed Description	44
5.26.2 Member Function Documentation	44
5.26.2.1 CreateInternetEntity()	44
5.26.2.2 CreateVirtualMachineEntity()	44
5.26.2.3 CreateVirtualNetworkNodeEntity()	45
5.26.2.4 UpdateEntityPosition()	45
5.27 Interconnect.Middlewares.ExceptionHandlerMiddleware Class Reference	45
5.27.1 Detailed Description	46
5.27.2 Member Function Documentation	46
5.27.2.1 InvokeAsync()	46
5.28 InterconnectTests.ExceptionHandlerMiddlewareTests Class Reference	46
5.29 Models.Responses.ExceptionResponse Class Reference	46
5.30 Controllers.HypervisorConnectionController Class Reference	47
5.30.1 Detailed Description	47
5.30.2 Member Function Documentation	48
5.30.2.1 ConnectionInfo()	48
5.31 ControllersTests.HypervisorConnectionControllerTests Class Reference	48
5.32 Services.Impl.HypervisorConnectionService Class Reference	48
5.32.1 Member Function Documentation	49
5.32.1.1 GetConnectionInfo()	49

5.32.1.2 GetConnectionStatus()	49
5.32.1.3 InitializeConnection()	49
5.33 ServicesTests.HypervisorConnectionServiceTests Class Reference	50
5.34 Services.IBootableDiskProviderService Interface Reference	50
5.34.1 Detailed Description	50
5.34.2 Member Function Documentation	50
5.34.2.1 GetAvailableBootableDiskModels()	50
5.34.2.2 GetBootableDiskPathById()	51
5.35 Repositories.IBootableDiskRepository Interface Reference	51
5.35.1 Detailed Description	51
5.35.2 Member Function Documentation	51
5.35.2.1 GetById()	51
5.35.2.2 GetOnlyWithNotNullablePath()	52
5.36 Services.IEntitiesConnectorService Interface Reference	52
5.36.1 Detailed Description	53
5.36.2 Member Function Documentation	53
5.36.2.1 ConnectTwoEntities()	53
5.36.2.2 ConnectTwoVirtualMachines()	53
5.36.2.3 ConnectVirtualMachineToInternet()	53
5.36.2.4 ConnectVirtualMachineToVirtualNetworkNode()	54
5.37 Services.IEntitiesDisconnectorService Interface Reference	54
5.37.1 Detailed Description	55
5.37.2 Member Function Documentation	55
5.37.2.1 DisconnectEntities()	55
5.37.2.2 DisconnectVirtualMachineFromInternet()	55
5.37.2.3 DisconnectVirtualMachineFromVirtualMachine()	56
5.37.2.4 DisconnectVirtualMachineFromVirtualNetworkNode()	56
5.38 Services.IHypervisorConnectionService Interface Reference	56
5.38.1 Detailed Description	57
5.38.2 Member Function Documentation	57
5.38.2.1 GetConnectionInfo()	57
5.38.2.2 GetConnectionStatus()	57
5.38.2.3 InitializeConnection()	57
5.39 Services.IInfoService Interface Reference	58
5.39.1 Detailed Description	58
5.39.2 Member Function Documentation	58
5.39.2.1 GetSystemInfo()	58
5.40 Repositories.IInternetEntityRepository Interface Reference	59
5.40.1 Detailed Description	59
5.40.2 Member Function Documentation	59
5.40.2.1 Create()	59
5.40.2.2 GetAll()	60

5.40.2.3 GetById()	60
5.40.2.4 UpdatePosition()	60
5.41 Services.IInternetEntityService Interface Reference	61
5.41.1 Detailed Description	61
5.41.2 Member Function Documentation	61
5.41.2.1 CreateInternet()	61
5.41.2.2 GetInternetEntities()	61
5.41.2.3 UpdateInternetEntityPosition()	61
5.42 NativeLibrary.Utils.INativeList< T > Interface Template Reference	62
5.43 Controllers.InfoController Class Reference	62
5.43.1 Detailed Description	63
5.43.2 Member Function Documentation	63
5.43.2.1 GetSystemInfo()	63
5.44 ControllersTests.InfoControllerTests Class Reference	63
5.45 Models.InformationModel Class Reference	63
5.45.1 Detailed Description	64
5.46 Services.Impl.InfoService Class Reference	64
5.46.1 Member Function Documentation	64
5.46.1.1 GetSystemInfo()	64
5.47 ServicesTests.InfoServiceTests Class Reference	65
5.48 Database.Migrations.InitialCreate Class Reference	65
5.48.1 Detailed Description	65
5.49 Services.Impl.InitializationService Class Reference	65
5.50 ServicesTests.InitializationServiceTests Class Reference	66
5.51 Models.Config.InterconnectConfig Class Reference	66
5.52 Database.InterconnectDbContext Class Reference	66
5.52.1 Detailed Description	67
5.53 Models.Responses.InternetEntitiesResponse Class Reference	67
5.54 Database.Migrations.InternetEntity Class Reference	68
5.54.1 Detailed Description	68
5.55 Models.Database.InternetEntityModel Class Reference	68
5.55.1 Detailed Description	69
5.56 Models.DTO.InternetEntityModelDTO Class Reference	69
5.57 Repositories.Impl.InternetEntityRepository Class Reference	70
5.57.1 Member Function Documentation	70
5.57.1.1 Create()	70
5.57.1.2 GetAll()	70
5.57.1.3 GetById()	71
5.57.1.4 UpdatePosition()	71
5.58 RepositoriesTests.InternetEntityRepositoryTests Class Reference	71
5.59 Services.Impl.InternetEntityService Class Reference	72
5.59.1 Member Function Documentation	72

5.59.1.1 CreateInternet()	72
5.59.1.2 GetInternetEntities()	72
5.59.1.3 UpdateInternetEntityPosition()	72
5.60 Repositories.IPacketSnifferRepository Interface Reference	73
5.60.1 Detailed Description	73
5.60.2 Member Function Documentation	73
5.60.2.1 AddSniffer()	73
5.60.2.2 GetOpenedSniffers()	74
5.60.2.3 RemoveSniffer()	74
5.61 Services.IPacketSnifferService Interface Reference	74
5.61.1 Detailed Description	75
5.61.2 Member Function Documentation	75
5.61.2.1 GetCapturedPacket()	75
5.61.2.2 GetOpenedSniffers()	75
5.61.2.3 ListenForPacket()	75
5.61.2.4 StartListeningForBridge()	76
5.62 NativeLibrary.Wrappers.IPacketSnifferWrapper Interface Reference	76
5.62.1 Detailed Description	77
5.62.2 Member Function Documentation	77
5.62.2.1 GetNumberOfPackets()	77
5.62.2.2 GetPacket()	77
5.62.2.3 ListenForPacket()	77
5.62.2.4 OpenSnifferHandler()	77
5.63 NativeLibrary.Wrappers.IVirtualizationWrapper Interface Reference	78
5.63.1 Detailed Description	79
5.63.2 Member Function Documentation	79
5.63.2.1 AttachDeviceToVirtualMachine()	79
5.63.2.2 CloseStream()	79
5.63.2.3 CreateVirtualMachine()	79
5.63.2.4 CreateVirtualNetwork()	80
5.63.2.5 DestroyNetwork()	80
5.63.2.6 DetachDeviceFromVirtualMachine()	80
5.63.2.7 GetConnectionInfo()	81
5.63.2.8 GetDataFromStream()	81
5.63.2.9 GetListOfVirtualMachines()	81
5.63.2.10 GetVirtualMachineInfo()	81
5.63.2.11 InitializeConnection()	82
5.63.2.12 IsConnectionAlive()	82
5.63.2.13 OpenVirtualMachineConsole()	82
5.63.2.14 SendDataToStream()	82
5.63.2.15 UpdateVmDevice()	83
5.64 Repositories.IVirtualMachineConsoleDataRepository Interface Reference	83

5.64.1 Detailed Description	84
5.64.2 Member Function Documentation	84
5.64.2.1 AddDataToConsole()	84
5.64.2.2 GetData()	84
5.65 Services.IVirtualMachineConsoleService Interface Reference	84
5.65.1 Detailed Description	85
5.65.2 Member Function Documentation	85
5.65.2.1 CloseStream()	85
5.65.2.2 GetBytesFromConsole()	85
5.65.2.3 GetInitialConsoleData()	85
5.65.2.4 GetStreams()	86
5.65.2.5 OpenVirtualMachineConsole()	86
5.65.2.6 SendBytesToVirtualMachineConsoleByUuid()	86
5.66 Repositories.IVirtualMachineConsoleStreamRepository Interface Reference	87
5.66.1 Detailed Description	87
5.66.2 Member Function Documentation	87
5.66.2.1 Add()	87
5.66.2.2 GetAllStreams()	88
5.66.2.3 GetByUuid()	88
5.66.2.4 Remove()	88
5.67 Repositories.IVirtualMachineEntityNetworkInterfaceRepository Interface Reference	89
5.67.1 Detailed Description	89
5.67.2 Member Function Documentation	89
5.67.2.1 Create()	89
5.67.2.2 GetByIds()	90
5.67.2.3 GetByNetworkName()	91
5.67.2.4 GetByVirtualMachineId()	91
5.67.2.5 Remove()	91
5.67.2.6 Update()	92
5.68 Repositories.IVirtualMachineEntityRepository Interface Reference	92
5.68.1 Detailed Description	93
5.68.2 Member Function Documentation	93
5.68.2.1 Add()	93
5.68.2.2 GetAll()	93
5.68.2.3 GetById()	93
5.68.2.4 Update()	93
5.69 Services.IVirtualMachineEntityService Interface Reference	94
5.69.1 Detailed Description	94
5.69.2 Member Function Documentation	95
5.69.2.1 CreateEntity()	95
5.69.2.2 GetById()	96
5.69.2.3 GetEntities()	96

5.69.2.4 GetMacAddresses()	97
5.69.2.5 UpdateEntityPosition()	97
5.69.2.6 UpdateEntityVmUUID()	97
5.70 Services.IVirtualMachineManagerService Interface Reference	98
5.70.1 Detailed Description	98
5.70.2 Member Function Documentation	98
5.70.2.1 AttachVirtualNetworkInterfaceToVirtualMachine()	98
5.70.2.2 CreateVirtualMachine()	98
5.70.2.3 GetListOfVirtualMachines()	99
5.71 Repositories.IVirtualNetworkConnectionRepository Interface Reference	99
5.71.1 Detailed Description	100
5.71.2 Member Function Documentation	100
5.71.2.1 Create()	100
5.71.2.2 GetAll()	100
5.71.2.3 GetById()	100
5.71.2.4 GetUsingEntityId()	101
5.71.2.5 Remove()	101
5.72 Services.IVirtualNetworkNodeConnector Interface Reference	102
5.72.1 Detailed Description	102
5.72.2 Member Function Documentation	102
5.72.2.1 ConnectTwoVirtualNetworkNodes()	102
5.72.2.2 ConnectVirtualNetworkNodeToInternet()	103
5.72.2.3 DisconnectTwoVirtualNetworkNodes()	103
5.72.2.4 DisconnectVirtualNetworkNodeFromInternet()	103
5.73 Repositories.IVirtualNetworkNodeEntityRepository Interface Reference	104
5.73.1 Detailed Description	104
5.73.2 Member Function Documentation	104
5.73.2.1 Create()	104
5.73.2.2 CreateInvisible()	105
5.73.2.3 GetAll()	105
5.73.2.4 GetById()	105
5.73.2.5 GetVisible()	106
5.73.2.6 Remove()	106
5.73.2.7 UpdateEntityPosition()	106
5.73.2.8 UpdateNetwork()	107
5.74 Repositories.IVirtualNetworkRepository Interface Reference	107
5.74.1 Detailed Description	108
5.74.2 Member Function Documentation	108
5.74.2.1 Create()	108
5.74.2.2 GetAll()	108
5.74.2.3 GetByBridgeName()	108
5.74.2.4 GetById()	109

5.74.2.5 GetByUuid()	109
5.74.2.6 GetByUuidWithVirtualNetworkNodes()	109
5.74.2.7 Remove()	110
5.75 Services.IVirtualNetworkService Interface Reference	110
5.75.1 Detailed Description	111
5.75.2 Member Function Documentation	111
5.75.2.1 AttachNetworkInterfaceToVirtualMachine()	111
5.75.2.2 CreateInternetVirtualNetwork()	112
5.75.2.3 CreateNodeVirtualNetwork()	112
5.75.2.4 CreateVirtualNetworkNode()	112
5.75.2.5 GetAllVirtualNetworks()	112
5.75.2.6 GetVirtualNetworkConnections()	113
5.75.2.7 GetVirtualNetworkUsingBridgeName()	113
5.75.2.8 GetVisibleVirtualNetworkNodeEntities()	113
5.75.2.9 UpdateNetworkForVirtualMachineNetworkInterface()	113
5.75.2.10 UpdateVirtualNetworkNodeEntityPosition()	114
5.76 Database.Migrations.ModificationToVirtualNetwork Class Reference	114
5.76.1 Detailed Description	115
5.77 NativeLibrary.Structs.NativeConnectionInfo Struct Reference	115
5.78 NativeLibrary.Structs.NativeExecutionInfo Struct Reference	115
5.79 NativeLibrary.Utils.Impl.NativeList< T > Class Template Reference	115
5.80 NativeLibrary.Structs.NativePacket Struct Reference	116
5.81 NativeLibrary.Structs.NativeStreamData Struct Reference	116
5.82 NativeLibrary.Structs.NativeStreamInfo Struct Reference	116
5.83 NativeLibrary.Structs.NativeVirtualMachineInfo Struct Reference	117
5.84 Models.Packet Class Reference	117
5.84.1 Detailed Description	117
5.85 Models.PacketSniffer Class Reference	118
5.85.1 Detailed Description	118
5.86 BackgroundServices.Impl.PacketSnifferBackgroundService Class Reference	118
5.86.1 Detailed Description	119
5.86.2 Constructor & Destructor Documentation	119
5.86.2.1 PacketSnifferBackgroundService()	119
5.86.3 Member Function Documentation	119
5.86.3.1 ExecuteAsync()	119
5.87 Hubs.PacketSnifferHub Class Reference	119
5.87.1 Detailed Description	120
5.88 Repositories.Impl.PacketSnifferRepository Class Reference	120
5.88.1 Member Function Documentation	120
5.88.1.1 AddSniffer()	120
5.88.1.2 GetOpenedSniffers()	121
5.88.1.3 RemoveSniffer()	121

5.89 Services.Impl.PacketSnifferService Class Reference	121
5.89.1 Member Function Documentation	122
5.89.1.1 GetCapturedPacket()	122
5.89.1.2 GetOpenedSniffers()	122
5.89.1.3 ListenForPacket()	122
5.89.1.4 StartListeningForBridge()	123
5.90 NativeLibrary.Wrappers.Impl.PacketSnifferWrapper Class Reference	124
5.90.1 Member Function Documentation	124
5.90.1.1 GetNumberOfPackets()	124
5.90.1.2 GetPacket()	125
5.90.1.3 OpenSnifferHandler()	125
5.91 Interconnect.Program Class Reference	125
5.91.1 Detailed Description	125
5.91.2 Member Function Documentation	125
5.91.2.1 Main()	125
5.92 Database.Migrations.RemoveOsType Class Reference	126
5.92.1 Detailed Description	126
5.93 Database.Migrations.RenameSwitchToNode Class Reference	126
5.93.1 Detailed Description	127
5.94 Database.Migrations.RenameUuidToVmUuid Class Reference	127
5.94.1 Detailed Description	127
5.95 Models.Requests.SendDataToConsoleRequest Class Reference	127
5.96 Models.StreamData Class Reference	127
5.96.1 Detailed Description	128
5.97 Models.StreamInfo Class Reference	128
5.97.1 Detailed Description	128
5.98 Models.Responses.StringResponse Class Reference	128
5.99 Models.Responses.SystemInfoResponse Class Reference	129
5.100 Models.TerminalData Class Reference	129
5.100.1 Detailed Description	130
5.101 Database.Migrations.UpdateBootableDiskModelPath Class Reference	130
5.101.1 Detailed Description	130
5.102 Models.Requests.UpdateEntityPositionRequest Class Reference	130
5.103 Database.Migrations.UpdateOperatingSystemType Class Reference	131
5.103.1 Detailed Description	131
5.104 NativeLibrary.Wrappers.Impl.VirtualizationWrapper Class Reference	131
5.104.1 Member Function Documentation	132
5.104.1.1 AttachDeviceToVirtualMachine()	132
5.104.1.2 CloseStream()	132
5.104.1.3 CreateVirtualMachine()	133
5.104.1.4 CreateVirtualNetwork()	133
5.104.1.5 DestroyNetwork()	133

5.104.1.6 DetachDeviceFromVirtualMachine()	133
5.104.1.7 GetConnectionInfo()	134
5.104.1.8 GetDataFromStream()	134
5.104.1.9 GetListOfVirtualMachines()	134
5.104.1.10 GetVirtualMachineInfo()	134
5.104.1.11 InitializeConnection()	135
5.104.1.12 IsConnectionAlive()	135
5.104.1.13 OpenVirtualMachineConsole()	135
5.104.1.14 SendDataToStream()	135
5.104.1.15 UpdateVmDevice()	136
5.105 BackgroundServices.Impl.VirtualMachineConsoleBackgroundService Class Reference	136
5.105.1 Detailed Description	137
5.105.2 Constructor & Destructor Documentation	137
5.105.2.1 VirtualMachineConsoleBackgroundService()	137
5.105.3 Member Function Documentation	137
5.105.3.1 ExecuteAsync()	137
5.106 Repositories.Impl.VirtualMachineConsoleDataRepository Class Reference	138
5.106.1 Member Function Documentation	138
5.106.1.1 AddDataToConsole()	138
5.106.1.2 GetData()	138
5.107 RepositoriesTests.VirtualMachineConsoleDataRepositoryTests Class Reference	139
5.108 Hubs.VirtualMachineConsoleHub Class Reference	139
5.108.1 Detailed Description	139
5.108.2 Member Function Documentation	139
5.108.2.1 GetInitialDataForConsole()	139
5.108.2.2 JoinConsoleGroup()	140
5.108.2.3 LeaveConsoleGroup()	140
5.108.2.4 SendDataToConsole()	140
5.109 HubsTests.VirtualMachineConsoleHubTests Class Reference	141
5.110 Services.Impl.VirtualMachineConsoleService Class Reference	141
5.110.1 Member Function Documentation	141
5.110.1.1 CloseStream()	141
5.110.1.2 GetBytesFromConsole()	142
5.110.1.3 GetInitialConsoleData()	142
5.110.1.4 GetStreams()	142
5.110.1.5 OpenVirtualMachineConsole()	142
5.110.1.6 SendBytesToVirtualMachineConsoleByUuid()	143
5.111 ServicesTests.VirtualMachineConsoleServiceTests Class Reference	143
5.112 Repositories.Impl.VirtualMachineConsoleStreamRepository Class Reference	144
5.112.1 Member Function Documentation	144
5.112.1.1 Add()	144
5.112.1.2 GetAllStreams()	144

5.112.1.3 GetByUuid()	144
5.112.1.4 Remove()	145
5.113 RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests Class Reference	145
5.114 Controllers.VirtualMachineController Class Reference	146
5.114.1 Detailed Description	146
5.114.2 Member Function Documentation	146
5.114.2.1 GetAvailableBootableDisks()	146
5.115 Models.VirtualMachineCreateDefinition Class Reference	146
5.115.1 Detailed Description	147
5.115.2 Member Function Documentation	147
5.115.2.1 GetVirtualMachineNameWithPrefix()	147
5.116 Services.Utils.VirtualMachineDefinitionBuilder Class Reference	147
5.117 ServicesTests.Utils.VirtualMachineDefinitionBuilderTests Class Reference	148
5.118 Models.Responses.VirtualMachineEntitiesResponse Class Reference	149
5.119 Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO Class Reference	149
5.120 Models.DTO.VirtualMachineEntityDTO Class Reference	150
5.121 Database.Migrations.VirtualMachineEntityInterfaces Class Reference	150
5.121.1 Detailed Description	151
5.122 Models.DTO.VirtualMachineEntityMacAddressDTO Class Reference	151
5.123 Models.Database.VirtualMachineEntityModel Class Reference	151
5.123.1 Detailed Description	151
5.124 Models.Database.VirtualMachineEntityNetworkInterfaceModel Class Reference	152
5.124.1 Detailed Description	152
5.125 Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository Class Reference	152
5.125.1 Member Function Documentation	153
5.125.1.1 Create()	153
5.125.1.2 GetByIds()	153
5.125.1.3 GetByNetworkName()	154
5.125.1.4 GetByVirtualMachineId()	154
5.125.1.5 Remove()	154
5.125.1.6 Update()	154
5.126 Repositories.Impl.VirtualMachineEntityRepository Class Reference	155
5.126.1 Member Function Documentation	155
5.126.1.1 Add()	155
5.126.1.2 GetAll()	156
5.126.1.3 GetById()	156
5.126.1.4 Update()	156
5.127 RepositoriesTests.VirtualMachineEntityRepositoryTests Class Reference	157
5.128 Models.Responses.VirtualMachineEntityResponse Class Reference	157
5.129 Services.Impl.VirtualMachineEntityService Class Reference	158
5.129.1 Member Function Documentation	158
5.129.1.1 CreateEntity()	158

5.129.1.2 GetById()	159
5.129.1.3 GetEntities()	159
5.129.1.4 GetMacAddresses()	159
5.129.1.5 UpdateEntityPosition()	160
5.129.1.6 UpdateEntityVmUUID()	160
5.130 ServicesTests.VirtualMachineEntityServiceTests Class Reference	161
5.131 Models.VirtualMachineInfo Class Reference	161
5.131.1 Detailed Description	161
5.132 Models.Responses.VirtualMachineInfoResponse Class Reference	161
5.133 Services.Impl.VirtualMachineManagerService Class Reference	162
5.133.1 Member Function Documentation	162
5.133.1.1 AttachVirtualNetworkInterfaceToVirtualMachine()	162
5.133.1.2 CreateVirtualMachine()	163
5.133.1.3 GetListOfVirtualMachines()	163
5.134 ServicesTests.VirtualMachineManagerServiceTests Class Reference	164
5.135 Models.Responses.VirtualMachinesInfoResponse Class Reference	164
5.136 Database.Migrations.VirtualMachineType Class Reference	165
5.136.1 Detailed Description	165
5.137 Database.Migrations.VirtualNetwork Class Reference	165
5.137.1 Detailed Description	165
5.138 Models.DTO.VirtualNetworkConnectionDTO Class Reference	166
5.139 Repositories.Impl.VirtualNetworkConnectionRepository Class Reference	166
5.139.1 Member Function Documentation	166
5.139.1.1 Create()	166
5.139.1.2 GetAll()	167
5.139.1.3 GetById()	167
5.139.1.4 GetUsingEntityId()	167
5.139.1.5 Remove()	168
5.140 RepositoriesTests.VirtualNetworkConnectionRepositoryTests Class Reference	168
5.141 Models.Responses.VirtualNetworkConnectionsResponse Class Reference	169
5.142 Controllers.VirtualNetworkController Class Reference	169
5.142.1 Detailed Description	170
5.142.2 Member Function Documentation	170
5.142.2.1 ConnectEntities()	170
5.142.2.2 DisconnectEntities()	170
5.142.2.3 GetAllConnections()	171
5.143 Models.VirtualNetworkCreateDefinition Class Reference	171
5.143.1 Detailed Description	171
5.144 Services.Utils.VirtualNetworkCreateDefinitionBuilder Class Reference	172
5.145 ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests Class Reference	173
5.146 Database.Migrations.VirtualNetworkEntities Class Reference	173
5.146.1 Detailed Description	173

5.147 Models.Database.VirtualNetworkEntityConnectionModel Class Reference	173
5.147.1 Detailed Description	174
5.148 Models.Requests.VirtualNetworkEntityConnectionRequest Class Reference	174
5.149 Models.VirtualNetworkInterfaceCreateDefinition Class Reference	174
5.149.1 Detailed Description	175
5.150 Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder Class Reference	175
5.151 ServicesTests.Utills.VirtualNetworkInterfaceCreateDefinitionBuilderTests Class Reference	176
5.152 Database.Migrations.VirtualNetworkIpAddress Class Reference	176
5.152.1 Detailed Description	176
5.153 Models.Database.VirtualNetworkModel Class Reference	176
5.153.1 Detailed Description	177
5.154 Services.Impl.VirtualNetworkNodeConnector Class Reference	177
5.154.1 Member Function Documentation	178
5.154.1.1 ConnectTwoVirtualNetworkNodes()	178
5.154.1.2 ConnectVirtualNetworkNodeToInternet()	178
5.154.1.3 DisconnectTwoVirtualNetworkNodes()	178
5.154.1.4 DisconnectVirtualNetworkNodeFromInternet()	179
5.155 Models.DTO.VirtualNetworkNodeEntityDTO Class Reference	179
5.156 Models.Database.VirtualNetworkNodeEntityModel Class Reference	180
5.156.1 Detailed Description	180
5.157 Repositories.Impl.VirtualNetworkNodeEntityRepository Class Reference	181
5.157.1 Member Function Documentation	181
5.157.1.1 Create()	181
5.157.1.2 CreateInvisible()	182
5.157.1.3 GetAll()	182
5.157.1.4 GetById()	182
5.157.1.5 GetVisible()	183
5.157.1.6 Remove()	183
5.157.1.7 UpdateEntityPosition()	183
5.157.1.8 UpdateNetwork()	183
5.158 RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests Class Reference	184
5.159 Models.Responses.VirtualNetworkNodesEntitiesResponse Class Reference	184
5.160 Repositories.Impl.VirtualNetworkRepository Class Reference	185
5.160.1 Member Function Documentation	186
5.160.1.1 Create()	186
5.160.1.2 GetAll()	186
5.160.1.3 GetByBridgeName()	186
5.160.1.4 GetById()	186
5.160.1.5 GetByUuid()	187
5.160.1.6 GetByUuidWithVirtualNetworkNodes()	187
5.160.1.7 Remove()	187
5.161 Services.Impl.VirtualNetworkService Class Reference	188

5.161.1 Member Function Documentation	189
5.161.1.1 AttachNetworkInterfaceToVirtualMachine()	189
5.161.1.2 CreateInternetVirtualNetwork()	189
5.161.1.3 CreateNodeVirtualNetwork()	189
5.161.1.4 CreateVirtualNetworkNode()	189
5.161.1.5 GetAllVirtualNetworks()	190
5.161.1.6 GetVirtualNetworkConnections()	190
5.161.1.7 GetVirtualNetworkUsingBridgeName()	190
5.161.1.8 GetVisibleVirtualNetworkNodeEntities()	191
5.161.1.9 UpdateNetworkForVirtualMachineNetworkInterface()	191
5.161.1.10 UpdateVirtualNetworkNodeEntityPosition()	191
5.162 Database.Migrations.VirtualNetworkVarChar Class Reference	192
5.162.1 Detailed Description	192
5.163 Database.Migrations.VirtualSwitch Class Reference	192
5.163.1 Detailed Description	192
Index	193

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

BackgroundServices	13
BackgroundServices.Impl	13
Controllers	13
ControllersTests	14
Database	14
Database.Migrations	14
Hubs	14
HubsTests	15
Interconnect	15
Interconnect.Middlewares	15
InterconnectTests	15
Mappers	15
Models	16
Models.Config	16
Models.Database	16
Models.DTO	17
Models.Enums	17
Models.Requests	19
Models.Responses	19
NativeLibrary	19
NativeLibrary.Interop	20
NativeLibrary.Structs	20
NativeLibrary.Utils	20
NativeLibrary.Utils.Impl	20
NativeLibrary.Wrappers	20
NativeLibrary.Wrappers.Impl	20
Repositories	21
Repositories.Impl	21
RepositoriesTests	21
Services	22
Services.Impl	22
Services.Utils	23
ServicesTests	23
ServicesTests.Utils	23
TestUtils	23

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

BackgroundService	
BackgroundServices.Impl.PacketSnifferBackgroundService	118
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService	136
Services.Utls.BaseDefinitionBuilder< TDefinition >	25
Services.Utls.VirtualMachineDefinitionBuilder	147
Services.Utls.VirtualNetworkCreateDefinitionBuilder	172
Services.Utls.VirtualNetworkInterfaceCreateDefinitionBuilder	175
Models.BaseEntity	26
Models.DTO.InternetEntityModelDTO	69
Models.DTO.VirtualMachineEntityDTO	150
Models.DTO.VirtualNetworkNodeEntityDTO	179
Models.Responses.BaseResponse< TResponse, TSelf >	27
Models.Responses.BootableDisksResponse	32
Models.Responses.CapturedPacketResponse	33
Models.Responses.ConnectionInfoResponse	36
Models.Responses.ExceptionResponse	46
Models.Responses.InternetEntitiesResponse	67
Models.Responses.StringResponse	128
Models.Responses.VirtualMachineEntitiesResponse	149
Models.Responses.VirtualMachineEntityResponse	157
Models.Responses.VirtualMachineInfoResponse	161
Models.Responses.VirtualMachinesInfoResponse	164
Models.Responses.VirtualNetworkConnectionsResponse	169
Models.Responses.VirtualNetworkNodesEntitiesResponse	184
Models.Database.BootableDiskModel	29
Models.DTO.BootableDiskModelDTO	30
ServicesTests.BootableDiskProviderServiceTests	31
RepositoriesTests.BootableDiskRepositoryTests	32
Models.Requests.ConnectEntitiesRequest	35
Models.ConnectionInfo	35
HubsTests.ConnectionStatusHubTests	37
ControllerBase	
Controllers.EntityController	43
Controllers.HypervisorConnectionController	47

Controllers.InfoController	62
Controllers.VirtualMachineController	146
Controllers.VirtualNetworkController	169
Models.Requests.CreateVirtualMachineEntityRequest	38
Models.Requests.CreateVirtualNetworkNodeEntityRequest	38
DbContext	
Database.InterconnectDbContext	66
Interconnect.Middlewares.ExceptionHandlerMiddleware	45
InterconnectTests.ExceptionHandlerMiddlewareTests	46
Hub	
Hubs.ConnectionStatusHub	37
Hubs.PacketSnifferHub	119
Hubs.VirtualMachineConsoleHub	139
ControllersTests.HypervisorConnectionControllerTests	48
ServicesTests.HypervisorConnectionServiceTests	50
Services.IBootableDiskProviderService	50
Services.Impl.BootableDiskProviderService	30
Repositories.IBootableDiskRepository	51
Repositories.Impl.BootableDiskRepository	31
Services.IEntitiesConnectorService	52
Services.Impl.EntitiesConnectorService	39
Services.IEntitiesDisconnectorService	54
Services.Impl.EntitiesDisconnectorService	41
IHostedService	
Services.IHypervisorConnectionService	56
Services.Impl.HypervisorConnectionService	48
Services.IInfoService	58
Services.Impl.InfoService	64
Repositories.IInternetEntityRepository	59
Repositories.Impl.InternetEntityRepository	70
Services.IInternetEntityService	61
Services.Impl.InternetEntityService	72
ControllersTests.InfoControllerTests	63
Models.InformationModel	63
ServicesTests.InfoServiceTests	65
ServicesTests.InitializationServiceTests	66
Models.Config.InterconnectConfig	66
Models.Database.InternetEntityModel	68
RepositoriesTests.InternetEntityRepositoryTests	71
Repositories.IPacketSnifferRepository	73
Repositories.Impl.PacketSnifferRepository	120
Services.IPacketSnifferService	74
Services.Impl.PacketSnifferService	121
NativeLibrary.Wrappers.IPacketSnifferWrapper	76
NativeLibrary.Wrappers.Impl.PacketSnifferWrapper	124
IReadOnlyList	
NativeLibrary.Utils.INativeList< T >	62
NativeLibrary.Utils.Impl.NativeList< T >	115
NativeLibrary.Wrappers.IVirtualizationWrapper	78
NativeLibrary.Wrappers.Impl.VirtualizationWrapper	131
Repositories.IVirtualMachineConsoleDataRepository	83
Repositories.Impl.VirtualMachineConsoleDataRepository	138
Services.IVirtualMachineConsoleService	84
Services.Impl.VirtualMachineConsoleService	141

Repositories.IVirtualMachineConsoleStreamRepository	87
Repositories.Impl.VirtualMachineConsoleStreamRepository	144
Repositories.IVirtualMachineEntityNetworkInterfaceRepository	89
Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository	152
Repositories.IVirtualMachineEntityRepository	92
Repositories.Impl.VirtualMachineEntityRepository	155
Services.IVirtualMachineEntityService	94
Services.Impl.VirtualMachineEntityService	158
Services.IVirtualMachineManagerService	98
Services.Impl.VirtualMachineManagerService	162
Repositories.IVirtualNetworkConnectionRepository	99
Repositories.Impl.VirtualNetworkConnectionRepository	166
Services.IVirtualNetworkNodeConnector	102
Services.Impl.VirtualNetworkNodeConnector	177
Repositories.IVirtualNetworkNodeEntityRepository	104
Repositories.Impl.VirtualNetworkNodeEntityRepository	181
Repositories.IVirtualNetworkRepository	107
Repositories.Impl.VirtualNetworkRepository	185
Services.IVirtualNetworkService	110
Services.Impl.VirtualNetworkService	188
Migration	
Database.Migrations.BootableDiskModel	29
Database.Migrations.ChangeTypeOfVMEntityStrings	34
Database.Migrations.ChangesToConnection	34
Database.Migrations.DeviceDefinition	38
Database.Migrations.InitialCreate	65
Database.Migrations.InternetEntity	68
Database.Migrations.ModificationToVirtualNetwork	114
Database.Migrations.RemoveOsType	126
Database.Migrations.RenameSwitchToNode	126
Database.Migrations.RenameUuidToVmUuid	127
Database.Migrations.UpdateBootableDiskModelPath	130
Database.Migrations.UpdateOperatingSystemType	131
Database.Migrations.VirtualMachineEntityInterfaces	150
Database.Migrations.VirtualMachineType	165
Database.Migrations.VirtualNetwork	165
Database.Migrations.VirtualNetworkEntities	173
Database.Migrations.VirtualNetworkIpAddress	176
Database.Migrations.VirtualNetworkVarChar	192
Database.Migrations.VirtualSwitch	192
ModelSnapshot	
Database.Migrations.AppDbContextModelSnapshot	25
NativeLibrary.Structs.NativeConnectionInfo	115
NativeLibrary.Structs.NativeExecutionInfo	115
NativeLibrary.Structs.NativePacket	116
NativeLibrary.Structs.NativeStreamData	116
NativeLibrary.Structs.NativeStreamInfo	116
NativeLibrary.Structs.NativeVirtualMachineInfo	117
Models.Packet	117
Models.PacketSniffer	118
Interconnect.Program	125
Models.Requests.SendDataToConsoleRequest	127
Models.StreamData	127
Models.StreamInfo	128
Models.Responses.SystemInfoResponse	129

Models.TerminalData	129
Models.Requests.UpdateEntityPositionRequest	130
RepositoriesTests.VirtualMachineConsoleDataRepositoryTests	139
HubsTests.VirtualMachineConsoleHubTests	141
ServicesTests.VirtualMachineConsoleServiceTests	143
RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests	145
Models.VirtualMachineCreateDefinition	146
ServicesTests.Utils.VirtualMachineDefinitionBuilderTests	148
Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO	149
Models.DTO.VirtualMachineEntityMacAddressDTO	151
Models.Database.VirtualMachineEntityModel	151
Models.Database.VirtualMachineEntityNetworkInterfaceModel	152
RepositoriesTests.VirtualMachineEntityRepositoryTests	157
ServicesTests.VirtualMachineEntityServiceTests	161
Models.VirtualMachineInfo	161
ServicesTests.VirtualMachineManagerServiceTests	164
Models.DTO.VirtualNetworkConnectionDTO	166
RepositoriesTests.VirtualNetworkConnectionRepositoryTests	168
Models.VirtualNetworkCreateDefinition	171
ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests	173
Models.Database.VirtualNetworkEntityConnectionModel	173
Models.Requests.VirtualNetworkEntityConnectionRequest	174
Models.VirtualNetworkInterfaceCreateDefinition	174
ServicesTests.Utils.VirtualNetworkInterfaceCreateDefinitionBuilderTests	176
Models.Database.VirtualNetworkModel	176
Models.Database.VirtualNetworkNodeEntityModel	180
RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests	184

Chapter 3

Class Index

3.1 Class List

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

Database.Migrations.AppDbContextModelSnapshot	25
Services.Utills.BaseDefinitionBuilder< TDefinition >	25
Models.BaseEntity	
Base class representing an entity with a position on the board	26
Models.Responses.BaseResponse< TResponse, TSelf >	
Base class for API responses	27
Database.Migrations.BootableDiskModel	
29	
Models.Database.BootableDiskModel	
Database model representing a bootable disk	29
Models.DTO.BootableDiskModelDTO	30
Services.Impl.BootableDiskProviderService	30
ServicesTests.BootableDiskProviderServiceTests	31
Repositories.Impl.BootableDiskRepository	31
RepositoriesTests.BootableDiskRepositoryTests	32
Models.Responses.BootableDisksResponse	32
Models.Responses.CapturedPacketResponse	33
Database.Migrations.ChangesToConnection	
34	
Database.Migrations.ChangeTypeOfVMEntityStrings	
34	
Models.Requests.ConnectEntitiesRequest	35
Models.ConnectionInfo	
Information about the connection to the virtualization hypervisor	35
Models.Responses.ConnectionInfoResponse	36
Hubs.ConnectionStatusHub	
SignalR hub for checking connection status	37
HubsTests.ConnectionStatusHubTests	37
Models.Requests.CreateVirtualMachineEntityRequest	38
Models.Requests.CreateVirtualNetworkNodeEntityRequest	38
Database.Migrations.DeviceDefinition	
38	
Services.Impl.EntitiesConnectorService	39
Services.Impl.EntitiesDisconnectorService	41
Controllers.EntityController	
Controller managing entities in the system	43

Interconnect.Middlewares.ExceptionHandlerMiddleware	
Middleware for global exception handling in the application	45
InterconnectTests.ExceptionHandlerMiddlewareTests	46
Models.Responses.ExceptionResponse	46
Controllers.HypervisorConnectionController	
Controller managing hypervisor connection	47
ControllersTests.HypervisorConnectionControllerTests	48
Services.Impl.HypervisorConnectionService	48
ServicesTests.HypervisorConnectionServiceTests	50
Services.IBootableDiskProviderService	
Service providing information about available bootable disks	50
Repositories.IBootableDiskRepository	
Repository managing bootable disks in the database	51
Services.IEntitiesConnectorService	
Service responsible for connecting network entities	52
Services.IEntitiesDisconnectorService	
Service responsible for disconnecting network entities	54
Services.IHypervisorConnectionService	
Service managing connection to the virtualization hypervisor	56
Services.IInfoService	
Service providing system information	58
Repositories.IInternetEntityRepository	
Repository managing Internet entities in the database	59
Services.IInternetEntityService	
Service managing Internet entities	61
NativeLibrary.Utils.INativeList< T >	62
Controllers.InfoController	
Controller providing system information	62
ControllersTests.InfoControllerTests	63
Models.InformationModel	
Model containing operating system information	63
Services.Impl.InfoService	64
ServicesTests.InfoServiceTests	65
Database.Migrations.InitialCreate	65
Services.Impl.InitializationService	65
ServicesTests.InitializationServiceTests	66
Models.Config.InterconnectConfig	66
Database.InterconnectDbContext	
Database context for the Interconnect application	66
Models.Responses.InternetEntitiesResponse	67
Database.Migrations.InternetEntity	68
Models.Database.InternetEntityModel	
Database model representing an Internet entity connected to a virtual network	68
Models.DTO.InternetEntityModelDTO	69
Repositories.Impl.InternetEntityRepository	70
RepositoriesTests.InternetEntityRepositoryTests	71
Services.Impl.InternetEntityService	72
Repositories.IPacketSnifferRepository	
Repository managing packet analyzers in memory	73
Services.IPacketSnifferService	
Service for capturing network packets	74
NativeLibrary.Wrappers.IPacketSnifferWrapper	
Interface wrapper for native packet analyzer library	76
NativeLibrary.Wrappers.IVirtualizationWrapper	
Interface wrapper for native virtualization library	78

Repositories.IVirtualMachineConsoleDataRepository	
Repository managing virtual machine console data in memory	83
Services.IVirtualMachineConsoleService	
Service managing virtual machine console	84
Repositories.IVirtualMachineConsoleStreamRepository	
Repository managing virtual machine console streams in memory	87
Repositories.IVirtualMachineEntityNetworkInterfaceRepository	
Repository managing virtual machine network interfaces in the database	89
Repositories.IVirtualMachineEntityRepository	
Repository managing virtual machine entities in the database	92
Services.IVirtualMachineEntityService	
Service managing virtual machine entities	94
Services.IVirtualMachineManagerService	
Service managing virtual machines in the hypervisor	98
Repositories.IVirtualNetworkConnectionRepository	
Repository managing entity connections in the virtual network	99
Services.IVirtualNetworkNodeConnector	
Service responsible for connecting virtual network nodes	102
Repositories.IVirtualNetworkNodeEntityRepository	
Repository managing virtual network nodes in the database	104
Repositories.IVirtualNetworkRepository	
Repository managing virtual networks in the database	107
Services.IVirtualNetworkService	
Service managing virtual networks	110
Database.Migrations.ModificationToVirtualNetwork	
114	
NativeLibrary.Structs.NativeConnectionInfo	115
NativeLibrary.Structs.NativeExecutionInfo	115
NativeLibrary.Utils.Impl.NativeList< T >	115
NativeLibrary.Structs.NativePacket	116
NativeLibrary.Structs.NativeStreamData	116
NativeLibrary.Structs.NativeStreamInfo	116
NativeLibrary.Structs.NativeVirtualMachineInfo	117
Models.Packet	
Represents a captured network packet	117
Models.PacketSniffer	
Represents a packet sniffer for a specific network bridge	118
BackgroundServices.Impl.PacketSnifferBackgroundService	
Background service for managing network packet sniffing and broadcasting	118
Hubs.PacketSnifferHub	
SignalR hub for transmitting captured network packets	119
Repositories.Impl.PacketSnifferRepository	120
Services.Impl.PacketSnifferService	121
NativeLibrary.Wrappers.Impl.PacketSnifferWrapper	124
Interconnect.Program	
Main class of the Interconnect application	125
Database.Migrations.RemoveOsType	
126	
Database.Migrations.RenameSwitchToNode	
126	
Database.Migrations.RenameUuidToVmUuid	
127	
Models.Requests.SendDataToConsoleRequest	127
Models.StreamData	
Represents data read from a stream	127
Models.StreamInfo	
Information about a virtual machine console stream	128
Models.Responses.StringResponse	128

Models.Responses.SystemInfoResponse	129
Models.TerminalData	
Virtual machine terminal data	129
Database.Migrations.UpdateBootableDiskModelPath	
130	
Models.Requests.UpdateEntityPositionRequest	130
Database.Migrations.UpdateOperatingSystemType	
131	
NativeLibrary.Wrappers.Impl.VirtualizationWrapper	131
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService	
Background service for managing virtual machine console data streaming	136
Repositories.Impl.VirtualMachineConsoleDataRepository	138
RepositoriesTests.VirtualMachineConsoleDataRepositoryTests	139
Hubs.VirtualMachineConsoleHub	
SignalR hub for communicating with virtual machine console	139
HubsTests.VirtualMachineConsoleHubTests	141
Services.Impl.VirtualMachineConsoleService	141
ServicesTests.VirtualMachineConsoleServiceTests	143
Repositories.Impl.VirtualMachineConsoleStreamRepository	144
RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests	145
Controllers.VirtualMachineController	
Controller managing virtual machines	146
Models.VirtualMachineCreateDefinition	
Definition for creating a new virtual machine	146
Services.Util.VirtualMachineDefinitionBuilder	147
ServicesTests.Util.VirtualMachineDefinitionBuilderTests	148
Models.Responses.VirtualMachineEntitiesResponse	149
Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO	149
Models.DTO.VirtualMachineEntityDTO	150
Database.Migrations.VirtualMachineEntityInterfaces	
150	
Models.DTO.VirtualMachineEntityMacAddressDTO	151
Models.Database.VirtualMachineEntityModel	
Database model representing a virtual machine entity	151
Models.Database.VirtualMachineEntityNetworkInterfaceModel	
Database model representing a virtual machine network interface	152
Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository	152
Repositories.Impl.VirtualMachineEntityRepository	155
RepositoriesTests.VirtualMachineEntityRepositoryTests	157
Models.Responses.VirtualMachineEntityResponse	157
Services.Impl.VirtualMachineEntityService	158
ServicesTests.VirtualMachineEntityServiceTests	161
Models.VirtualMachineInfo	
Information about a virtual machine	161
Models.Responses.VirtualMachineInfoResponse	161
Services.Impl.VirtualMachineManagerService	162
ServicesTests.VirtualMachineManagerServiceTests	164
Models.Responses.VirtualMachinesInfoResponse	164
Database.Migrations.VirtualMachineType	
165	
Database.Migrations.VirtualNetwork	
165	
Models.DTO.VirtualNetworkConnectionDTO	166
Repositories.Impl.VirtualNetworkConnectionRepository	166
RepositoriesTests.VirtualNetworkConnectionRepositoryTests	168
Models.Responses.VirtualNetworkConnectionsResponse	169
Controllers.VirtualNetworkController	
Controller managing virtual networks and connections	169

Models.VirtualNetworkCreateDefinition	
Definition for creating a new virtual network	171
Services.Utills.VirtualNetworkCreateDefinitionBuilder	172
ServicesTests.Utills.VirtualNetworkDefinitionBuilderTests	173
Database.Migrations.VirtualNetworkEntities	
173	
Models.Database.VirtualNetworkEntityConnectionModel	
Database model representing a connection between entities in a virtual network	173
Models.Requests.VirtualNetworkEntityConnectionRequest	174
Models.VirtualNetworkInterfaceCreateDefinition	
Definition for creating a network interface for a virtual machine	174
Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder	175
ServicesTests.Utills.VirtualNetworkInterfaceCreateDefinitionBuilderTests	176
Database.Migrations.VirtualNetworkIpAddress	
176	
Models.Database.VirtualNetworkModel	
Database model representing a virtual network	176
Services.Impl.VirtualNetworkNodeConnector	177
Models.DTO.VirtualNetworkNodeEntityDTO	179
Models.Database.VirtualNetworkNodeEntityModel	
Database model representing a virtual network node (switch/router)	180
Repositories.Impl.VirtualNetworkNodeEntityRepository	181
RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests	184
Models.Responses.VirtualNetworkNodesEntitiesResponse	184
Repositories.Impl.VirtualNetworkRepository	185
Services.Impl.VirtualNetworkService	188
Database.Migrations.VirtualNetworkVarChar	
192	
Database.Migrations.VirtualSwitch	
192	

Chapter 4

Namespace Documentation

4.1 BackgroundServices Namespace Reference

Classes

- class **BackgroundServicesInitializer**
Initializer for background services.

4.2 BackgroundServices.Impl Namespace Reference

Classes

- class **PacketAnalyzer**
- class [PacketSnifferBackgroundService](#)
Background service for managing network packet sniffing and broadcasting.
- class [VirtualMachineConsoleBackgroundService](#)
Background service for managing virtual machine console data streaming.

4.3 Controllers Namespace Reference

Classes

- class [EntityController](#)
Controller managing entities in the system.
- class [HypervisorConnectionController](#)
Controller managing hypervisor connection.
- class [InfoController](#)
Controller providing system information.
- class [VirtualMachineController](#)
Controller managing virtual machines.
- class [VirtualNetworkController](#)
Controller managing virtual networks and connections.

4.4 ControllersTests Namespace Reference

Classes

- class [HypervisorConnectionControllerTests](#)
- class [InfoControllerTests](#)

4.5 Database Namespace Reference

Classes

- class [InterconnectDbContext](#)
Database context for the [Interconnect](#) application.

4.6 Database.Migrations Namespace Reference

Classes

- class [AppDbContextModelSnapshot](#)
- class [BootableDiskModel](#)
- class [ChangesToConnection](#)
- class [ChangeTypeOfVmEntityStrings](#)
- class [DeviceDefinition](#)
- class [InitialCreate](#)
- class [InternetEntity](#)
- class [ModificationToVirtualNetwork](#)
- class [RemoveOsType](#)
- class [RenameSwitchToNode](#)
- class [RenameUuidToVmUuid](#)
- class [UpdateBootableDiskModelPath](#)
- class [UpdateOperatingSystemType](#)
- class [VirtualMachineEntityInterfaces](#)
- class [VirtualMachineType](#)
- class [VirtualNetwork](#)
- class [VirtualNetworkEntities](#)
- class [VirtualNetworkIpAddress](#)
- class [VirtualNetworkVarChar](#)
- class [VirtualSwitch](#)

4.7 Hubs Namespace Reference

Classes

- class [ConnectionStatusHub](#)
SignalR hub for checking connection status.
- class [PacketSnifferHub](#)
SignalR hub for transmitting captured network packets.
- class [VirtualMachineConsoleHub](#)
SignalR hub for communicating with virtual machine console.

4.8 HubsTests Namespace Reference

Classes

- class [ConnectionStatusHubTests](#)
- class [VirtualMachineConsoleHubTests](#)

4.9 Interconnect Namespace Reference

Classes

- class [Program](#)
Main class of the [Interconnect](#) application.

4.10 Interconnect.Middlewares Namespace Reference

Classes

- class [ExceptionHandlerMiddleware](#)
Middleware for global exception handling in the application.
- class **ExceptionHandlerMiddlewareExtensions**
Extensions for registering exception handling middleware.

4.11 InterconnectTests Namespace Reference

Classes

- class [ExceptionHandlerMiddlewareTests](#)

4.12 Mappers Namespace Reference

Classes

- class **BootableDiskModelMapper**
Mapper for converting bootable disk models.
- class **InternetEntityMapper**
Mapper for converting Internet entity models.
- class **NativeConnectionInfoMapper**
- class **NativeStreamDataMapper**
- class **NativeVirtualMachineInfoMapper**
- class **VirtualMachineEntityMapper**
Mapper for converting virtual machine entity models.
- class **VirtualNetworkEntityConnectionMapper**
- class **VirtualNetworkNodeEntityMapper**

4.13 Models Namespace Reference

Classes

- class [BaseEntity](#)
Base class representing an entity with a position on the board.
- class [ConnectionInfo](#)
Information about the connection to the virtualization hypervisor.
- class [InformationModel](#)
Model containing operating system information.
- class [Packet](#)
Represents a captured network packet.
- class [PacketSniffer](#)
Represents a packet sniffer for a specific network bridge.
- class [StreamData](#)
Represents data read from a stream.
- class [StreamInfo](#)
Information about a virtual machine console stream.
- class [TerminalData](#)
Virtual machine terminal data.
- class [VirtualMachineCreateDefinition](#)
Definition for creating a new virtual machine.
- class [VirtualMachineInfo](#)
Information about a virtual machine.
- class [VirtualNetworkCreateDefinition](#)
Definition for creating a new virtual network.
- class [VirtualNetworkInterfaceCreateDefinition](#)
Definition for creating a network interface for a virtual machine.

4.14 Models.Config Namespace Reference

Classes

- class [InterconnectConfig](#)

4.15 Models.Database Namespace Reference

Classes

- class [BootableDiskModel](#)
Database model representing a bootable disk.
- class [InternetEntityModel](#)
Database model representing an Internet entity connected to a virtual network.
- class [VirtualMachineEntityModel](#)
Database model representing a virtual machine entity.
- class [VirtualMachineEntityNetworkInterfaceModel](#)
Database model representing a virtual machine network interface.
- class [VirtualNetworkEntityConnectionModel](#)
Database model representing a connection between entities in a virtual network.
- class [VirtualNetworkModel](#)
Database model representing a virtual network.
- class [VirtualNetworkNodeEntityModel](#)
Database model representing a virtual network node (switch/router).

4.16 Models.DTO Namespace Reference

Classes

- class [BootableDiskModelDTO](#)
- class [InternetEntityModelDTO](#)
- class [VirtualMachineEntitiesWithMacAddressesDTO](#)
- class [VirtualMachineEntityDTO](#)
- class [VirtualMachineEntityMacAddressDTO](#)
- class [VirtualNetworkConnectionDTO](#)
- class [VirtualNetworkNodeEntityDTO](#)

4.17 Models.Enums Namespace Reference

Enumerations

- enum [ConnectionStatus](#) { [ALIVE](#) = 1 , [DEAD](#) = 0 }
Connection status with the hypervisor.
- enum [DataLinkLayerPacketType](#) { [Ipv4](#) = 0 , [Arp](#) = 1 , [Unknown](#) = 3 }
- enum [EntityType](#) { [VirtualMachine](#) = 1 , [Network](#) = 2 , [VirtualNetworkNode](#) = 3 , [Internet](#) = 4 }
Type of entity in a virtual network.
- enum [VirtualMachineEntityType](#) { [Host](#) = 1 , [Router](#) = 2 , [Server](#) = 3 }
- enum [VirtualMachineState](#) { [Unknown](#) = 0 , [Booted](#) = 1 }
State of the virtual machine.

4.17.1 Enumeration Type Documentation

4.17.1.1 ConnectionStatus

enum [Models.Enums.ConnectionStatus](#)

Connection status with the hypervisor.

Enumerator

ALIVE	Connection is active.
DEAD	Connection is inactive.

4.17.1.2 EntityType

enum [Models.Enums.EntityType](#)

Type of entity in a virtual network.

Enumerator

VirtualMachine	Virtual machine entity.
Network	Network - used only in frontend for connecting two entities.
VirtualNetworkNode	Virtual network node (switch/router).
Internet	Internet entity.

4.17.1.3 VirtualMachineState

enum `Models.Enums.VirtualMachineState`

State of the virtual machine.

Enumerator

Unknown	Unknown state.
Booted	Virtual machine is running.

4.18 Models.Requests Namespace Reference

Classes

- class [ConnectEntitiesRequest](#)
- class [CreateVirtualMachineEntityRequest](#)
- class [CreateVirtualNetworkNodeEntityRequest](#)
- class [SendDataToConsoleRequest](#)
- class [UpdateEntityPositionRequest](#)
- class [VirtualNetworkEntityConnectionRequest](#)

4.19 Models.Responses Namespace Reference

Classes

- class [BaseResponse< TResponse, TSelf >](#)
Base class for API responses.
- class [BootableDisksResponse](#)
- class [CapturedPacketResponse](#)
- class [ConnectionInfoResponse](#)
- class [ExceptionResponse](#)
- class [InternetEntitiesResponse](#)
- class [StringResponse](#)
- class [SystemInfoResponse](#)
- class [VirtualMachineEntitiesResponse](#)
- class [VirtualMachineEntityResponse](#)
- class [VirtualMachineInfoResponse](#)
- class [VirtualMachinesInfoResponse](#)
- class [VirtualNetworkConnectionsResponse](#)
- class [VirtualNetworkNodesEntitiesResponse](#)

4.20 NativeLibrary Namespace Reference

Classes

- class **Constants**
- class **ExecutionInfoAnalyzer**
Analyzer for native operation execution information.
- class **LibraryInitializer**
Initializer for native virtualization library.

4.21 NativeLibrary.Interop Namespace Reference

Classes

- class [InteropPacketSniffer](#)
- class [InteropVirtualization](#)

4.22 NativeLibrary.Structs Namespace Reference

Classes

- struct [NativeConnectionInfo](#)
- struct [NativeExecutionInfo](#)
- struct [NativePacket](#)
- struct [NativeStreamData](#)
- struct [NativeStreamInfo](#)
- struct [NativeVirtualMachineInfo](#)

4.23 NativeLibrary.Utils Namespace Reference

Classes

- interface [INativeList< T >](#)

4.24 NativeLibrary.Utils.Impl Namespace Reference

Classes

- class [NativeList< T >](#)

4.25 NativeLibrary.Wrappers Namespace Reference

Classes

- interface [IPacketSnifferWrapper](#)
Interface wrapper for native packet analyzer library.
- interface [IVirtualizationWrapper](#)
Interface wrapper for native virtualization library.

4.26 NativeLibrary.Wrappers.Impl Namespace Reference

Classes

- class [PacketSnifferWrapper](#)
- class [VirtualizationWrapper](#)

4.27 Repositories Namespace Reference

Classes

- interface [IBootableDiskRepository](#)
Repository managing bootable disks in the database.
- interface [IInternetEntityRepository](#)
Repository managing Internet entities in the database.
- interface [IPacketSnifferRepository](#)
Repository managing packet analyzers in memory.
- interface [IVirtualMachineConsoleDataRepository](#)
Repository managing virtual machine console data in memory.
- interface [IVirtualMachineConsoleStreamRepository](#)
Repository managing virtual machine console streams in memory.
- interface [IVirtualMachineEntityNetworkInterfaceRepository](#)
Repository managing virtual machine network interfaces in the database.
- interface [IVirtualMachineEntityRepository](#)
Repository managing virtual machine entities in the database.
- interface [IVirtualNetworkConnectionRepository](#)
Repository managing entity connections in the virtual network.
- interface [IVirtualNetworkNodeEntityRepository](#)
Repository managing virtual network nodes in the database.
- interface [IVirtualNetworkRepository](#)
Repository managing virtual networks in the database.
- class **RepositoriesInitializer**
Initializer for application repositories.

4.28 RepositoriesImpl Namespace Reference

Classes

- class [BootableDiskRepository](#)
- class [InternetEntityRepository](#)
- class [PacketSnifferRepository](#)
- class [VirtualMachineConsoleDataRepository](#)
- class [VirtualMachineConsoleStreamRepository](#)
- class [VirtualMachineEntityNetworkInterfaceRepository](#)
- class [VirtualMachineEntityRepository](#)
- class [VirtualNetworkConnectionRepository](#)
- class [VirtualNetworkNodeEntityRepository](#)
- class [VirtualNetworkRepository](#)

4.29 RepositoriesTests Namespace Reference

Classes

- class [BootableDiskRepositoryTests](#)
- class [InternetEntityRepositoryTests](#)
- class [VirtualMachineConsoleDataRepositoryTests](#)
- class [VirtualMachineConsoleStreamRepositoryTests](#)
- class [VirtualMachineEntityRepositoryTests](#)
- class [VirtualNetworkConnectionRepositoryTests](#)
- class [VirtualNetworkNodeEntityRepositoryTests](#)

4.30 Services Namespace Reference

Classes

- interface [IBootableDiskProviderService](#)
Service providing information about available bootable disks.
- interface [IEntitiesConnectorService](#)
Service responsible for connecting network entities.
- interface [IEntitiesDisconnectService](#)
Service responsible for disconnecting network entities.
- interface [IHypervisorConnectionService](#)
Service managing connection to the virtualization hypervisor.
- interface [IInfoService](#)
Service providing system information.
- interface [IInitializationService](#)
Service responsible for application initialization during startup.
- interface [IInternetEntityService](#)
Service managing Internet entities.
- interface [IPacketSnifferService](#)
Service for capturing network packets.
- interface [IVirtualMachineConsoleService](#)
Service managing virtual machine console.
- interface [IVirtualMachineEntityService](#)
Service managing virtual machine entities.
- interface [IVirtualMachineManagerService](#)
Service managing virtual machines in the hypervisor.
- interface [IVirtualNetworkNodeConnector](#)
Service responsible for connecting virtual network nodes.
- interface [IVirtualNetworkService](#)
Service managing virtual networks.
- class [ServicesInitializer](#)
Initializer for application services.

4.31 Services.Impl Namespace Reference

Classes

- class [BootableDiskProviderService](#)
- class [EntitiesConnectorService](#)
- class [EntitiesDisconnectService](#)
- class [HypervisorConnectionService](#)
- class [InfoService](#)
- class [InitializationService](#)
- class [InternetEntityService](#)
- class [PacketSnifferService](#)
- class [VirtualMachineConsoleService](#)
- class [VirtualMachineEntityService](#)
- class [VirtualMachineManagerService](#)
- class [VirtualNetworkNodeConnector](#)
- class [VirtualNetworkService](#)

4.32 Services.Utills Namespace Reference

Classes

- class [BaseDefinitionBuilder< TDefinition >](#)
- class **EntitiesUtills**
- class **MacAddressGenerator**
- class [VirtualMachineDefinitionBuilder](#)
- class [VirtualNetworkCreateDefinitionBuilder](#)
- class **VirtualNetworkDefinitionUtills**
- class [VirtualNetworkInterfaceCreateDefinitionBuilder](#)
- class **VirtualNetworkUtills**

4.33 ServicesTests Namespace Reference

Classes

- class [BootableDiskProviderServiceTests](#)
- class [HypervisorConnectionServiceTests](#)
- class [InfoServiceTests](#)
- class [InitializationServiceTests](#)
- class [VirtualMachineConsoleServiceTests](#)
- class [VirtualMachineEntityServiceTests](#)
- class [VirtualMachineManagerServiceTests](#)

4.34 ServicesTests.Utills Namespace Reference

Classes

- class [VirtualMachineDefinitionBuilderTests](#)
- class [VirtualNetworkDefinitionBuilderTests](#)
- class [VirtualNetworkInterfaceCreateDefinitionBuilderTests](#)

4.35 TestUtills Namespace Reference

Classes

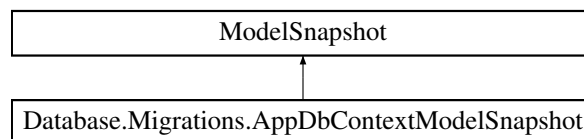
- class **ResponseExtractor**
- class **TestMocks**

Chapter 5

Class Documentation

5.1 Database.Migrations.AppDbContextModelSnapshot Class Reference

Inheritance diagram for Database.Migrations.AppDbContextModelSnapshot:



Protected Member Functions

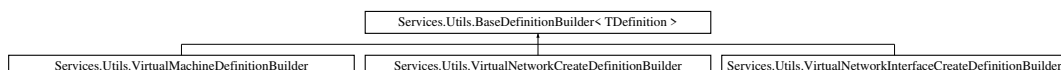
- override void **BuildModel** (ModelBuilder modelBuilder)

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

- Database/Migrations/AppDbContextModelSnapshot.cs

5.2 Services.Utls.BaseDefinitionBuilder< TDefinition > Class Template Reference

Inheritance diagram for Services.Utls.BaseDefinitionBuilder< TDefinition >:



Public Member Functions

- BaseDefinitionBuilder< TDefinition > **SetFromCreateDefinition** (TDefinition definition)
- string **Build** ()

Protected Member Functions

- delegate void **BuildingBlock** (XmlWriter writer)
- void **ChecksEverythingIsProvided** (params object?[] args)
- StringWriter **CreateXmlWriter** ()

Protected Attributes

- **XmlWriter**

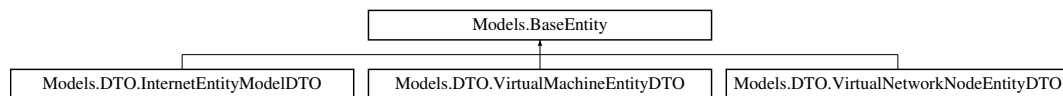
The documentation for this class was generated from the following file:

- Services/Utils/BaseDefinitionBuilder.cs

5.3 Models.BaseEntity Class Reference

Base class representing an entity with a position on the board.

Inheritance diagram for Models.BaseEntity:



Properties

- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

5.3.1 Detailed Description

Base class representing an entity with a position on the board.

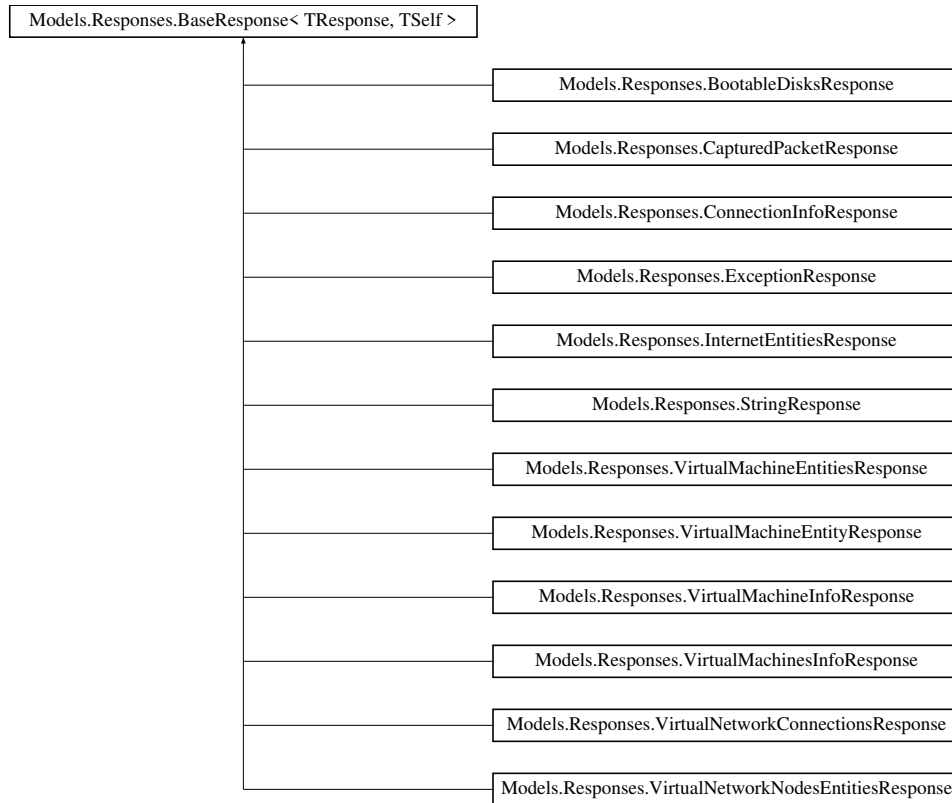
The documentation for this class was generated from the following file:

- Models/BaseEntity.cs

5.4 Models.Responses.BaseResponse< TResponse, TSelf > Class Template Reference

Base class for API responses.

Inheritance diagram for Models.Responses.BaseResponse< TResponse, TSelf >:



Static Public Member Functions

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

5.4.1 Detailed Description

Base class for API responses.

Template Parameters

<i>TResponse</i>	Type of response data.
<i>TSelf</i>	Type of the inheriting class.

Type Constraints

TSelf : BaseResponse

TSelf : TResponse

TSelf : TSelf

TSelf : new()

5.4.2 Member Function Documentation

5.4.2.1 WithEmptySuccess()

```
static TSelf Models.Responses.BaseResponse< TResponse, TSelf >.WithEmptySuccess () [inline],  
[static]
```

Creates an empty successful response.

Returns

Empty successful response.

5.4.2.2 WithSuccess()

```
static TSelf Models.Responses.BaseResponse< TResponse, TSelf >.WithSuccess (  
    TResponse data) [inline], [static]
```

Creates a successful response with data.

Parameters

<i>data</i>	Data to return.
-------------	-----------------

Returns

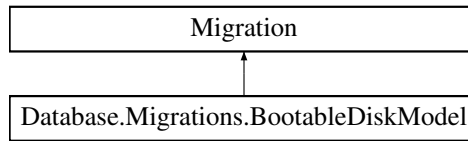
Successful response.

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

- Models/Responses/BaseResponse.cs

5.5 Database.Migrations.BootableDiskModel Class Reference

Inheritance diagram for Database.Migrations.BootableDiskModel:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.5.1 Detailed Description

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

- Database/Migrations/20250728193116_BootableDiskModel.cs
- Database/Migrations/20250728193116_BootableDiskModel.Designer.cs

5.6 Models.Database.BootableDiskModel Class Reference

[Database](#) model representing a bootable disk.

Properties

- required int **Id** [get, set]
Disk identifier.
- required string **Name** [get, set]
Disk name.
- string? **Version** [get, set]
Operating system version on the disk.
- string? **Path** [get, set]
Path to the disk file.

5.6.1 Detailed Description

[Database](#) model representing a bootable disk.

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

- Models/Database/BootableDiskModel.cs

5.7 Models.DTO.BootableDiskModelDTO Class Reference

Properties

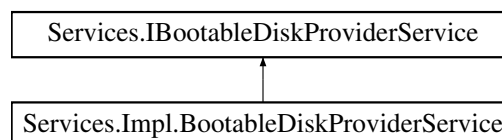
- required int **Id** [get, set]
- required string **Name** [get, set]
- string? **Version** [get, set]

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

- Models/DTO/BootableDiskModelDTO.cs

5.8 Services.Impl.BootableDiskProviderService Class Reference

Inheritance diagram for Services.Impl.BootableDiskProviderService:



Public Member Functions

- **BootableDiskProviderService** ([IBootableDiskRepository](#) repository)
- async Task< List< [BootableDiskModelDTO](#) > > [GetAvailableBootableDiskModels](#) ()
Retrieves a list of available bootable disk models.
- async Task< string?> [GetBootableDiskPathById](#) (int id)
Retrieves the path to a bootable disk by its identifier.

5.8.1 Member Function Documentation

5.8.1.1 GetAvailableBootableDiskModels()

```
async Task< List< BootableDiskModelDTO > > Services.Impl.BootableDiskProviderService.GetAvailableBootableDiskModels () [inline]
```

Retrieves a list of available bootable disk models.

Returns

List of bootable disks.

Implements [Services.IBootableDiskProviderService](#).

5.8.1.2 GetBootableDiskPathById()

```
async Task< string?> Services.Impl.BootableDiskProviderService.GetBootableDiskPathById (int id) [inline]
```

Retrieves the path to a bootable disk by its identifier.

Parameters

<i>id</i>	Disk identifier.
-----------	------------------

Returns

Path to the disk or null if not found.

Implements [Services.IBootableDiskProviderService](#).

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

- Services/Impl/BootableDiskProviderService.cs

5.9 ServicesTests.BootableDiskProviderServiceTests Class Reference

Public Member Functions

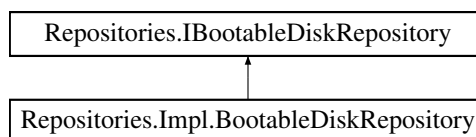
- void **SetUp** ()
- void **TearDown** ()
- async Task **GetAvailableBootableDiskModels_WhenInvoked_ShouldGetAllBootableDisksWithPath** ()
- async Task **GetBootableDiskPathById_WhenInvokedWithCorrectId_ShouldReturnBootableDiskPath** ()

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

- ServicesTests/BootableDiskProviderServiceTests.cs

5.10 Repositories.Impl.BootableDiskRepository Class Reference

Inheritance diagram for Repositories.Impl.BootableDiskRepository:



Public Member Functions

- **BootableDiskRepository** ([InterconnectDbContext](#) context)
- async Task< List< [BootableDiskModel](#) > > **GetOnlyWithNotNullablePath** ()
Retrieves a list of bootable disks with non-null paths.
- async Task< [BootableDiskModel](#) > **GetById** (int id)
Retrieves a bootable disk by identifier.

5.10.1 Member Function Documentation

5.10.1.1 GetById()

```
async Task< BootableDiskModel > Repositories.Impl.BootableDiskRepository.GetById (
    int id) [inline]
```

Retrieves a bootable disk by identifier.

Parameters

<i>id</i>	Disk identifier.
-----------	------------------

Returns

Bootable disk.

Implements [Repositories.IBootableDiskRepository](#).

5.10.1.2 GetOnlyWithNotNullablePath()

```
async Task< List< BootableDiskModel > > Repositories.Impl.BootableDiskRepository.GetOnlyWithNotNullablePath () [inline]
```

Retrieves a list of bootable disks with non-null paths.

Returns

List of disks with paths.

Implements [Repositories.IBootableDiskRepository](#).

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

- Repositories/Impl/BootableDiskRepository.cs

5.11 RepositoriesTests.BootableDiskRepositoryTests Class Reference**Public Member Functions**

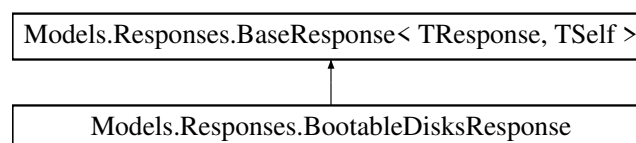
- void **SetUp** ()
- void **TearDown** ()
- async Task **GetOnlyWithNotNullablePath_WhenInvoked_ShouldReturnBootableDisksWithNotNullablePaths** ()
- async Task **GetById_WhenInvoked_ShouldGetBootableDiskUsingProvidedId** ()

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

- RepositoriesTests/BootableDiskRepositoryTests.cs

5.12 Models.Responses.BootableDisksResponse Class Reference

Inheritance diagram for Models.Responses.BootableDisksResponse:



Additional Inherited Members

Static Public Member Functions inherited from

[Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

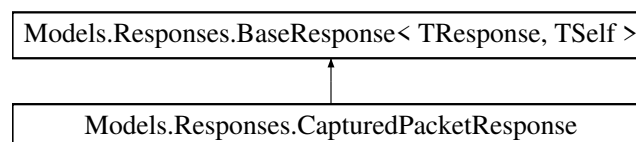
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/BootableDisksResponse.cs

5.13 Models.Responses.CapturedPacketResponse Class Reference

Inheritance diagram for Models.Responses.CapturedPacketResponse:



Additional Inherited Members

Static Public Member Functions inherited from

[Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

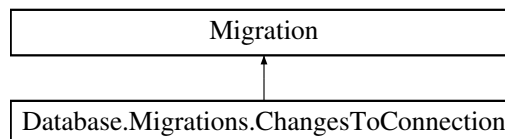
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/CapturedPacketResponse.cs

5.14 Database.Migrations.ChangesToConnection Class Reference

Inheritance diagram for Database.Migrations.ChangesToConnection:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

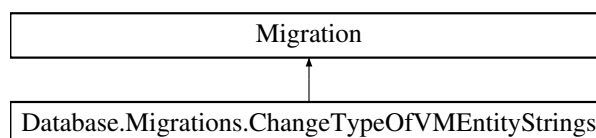
5.14.1 Detailed Description

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

- Database/Migrations/20250907104504_ChangesToConnection.cs
- Database/Migrations/20250907104504_ChangesToConnection.Designer.cs

5.15 Database.Migrations.ChangeTypeOfVMEntityStrings Class Reference

Inheritance diagram for Database.Migrations.ChangeTypeOfVMEntityStrings:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.15.1 Detailed Description

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

- Database/Migrations/20250726223025_ChangeTypeOfVMEntityStrings.cs
- Database/Migrations/20250726223025_ChangeTypeOfVMEntityStrings.Designer.cs

5.16 Models.Requests.ConnectEntitiesRequest Class Reference

Properties

- required int **SourceEntityId** [get, set]
- [EntityType](#) **SourceEntityType** [get, set]
- required int **DestinationEntityId** [get, set]
- [EntityType](#) **DestinationEntityType** [get, set]

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

- Models/Requests/ConnectEntitiesRequest.cs

5.17 Models.ConnectionInfo Class Reference

Information about the connection to the virtualization hypervisor.

Properties

- uint **CpuCount** [get, set]
Number of CPUs.
- uint **CpuFreq** [get, set]
CPU frequency in MHz.
- long **TotalMemory** [get, set]
Total system memory in bytes.
- required string **ConnectionUrl** [get, set]
Connection URL to the hypervisor.
- required string **DriverType** [get, set]
Virtualization driver type.
- required string **LibVersion** [get, set]
Libvirt library version.
- required string **DriverVersion** [get, set]
Virtualization driver version.

5.17.1 Detailed Description

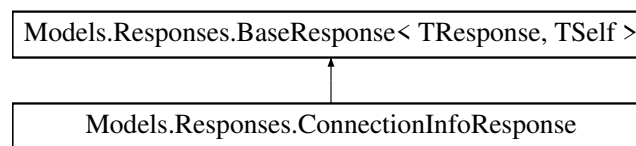
Information about the connection to the virtualization hypervisor.

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

- Models/ConnectionInfo.cs

5.18 Models.Responses.ConnectionInfoResponse Class Reference

Inheritance diagram for Models.Responses.ConnectionInfoResponse:



Additional Inherited Members

Static Public Member Functions inherited from

[Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

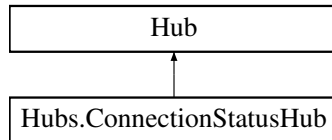
The documentation for this class was generated from the following file:

- Models/Responses/ConnectionInfoResponse.cs

5.19 Hubs.ConnectionStatusHub Class Reference

SignalR hub for checking connection status.

Inheritance diagram for Hubs.ConnectionStatusHub:



Public Member Functions

- [StringResponse Ping \(\)](#)
Checks connection with the server.

5.19.1 Detailed Description

SignalR hub for checking connection status.

5.19.2 Member Function Documentation

5.19.2.1 Ping()

```
StringResponse Hubs.ConnectionStatusHub.Ping () [inline]
```

Checks connection with the server.

Returns

Response confirming connection.

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

- Hubs/ConnectionStatusHub.cs

5.20 HubsTests.ConnectionStatusHubTests Class Reference

Public Member Functions

- void **Setup** ()
- void **Teardown** ()
- void **Ping_WhenInvoked_ShouldReturnPong** ()

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

- HubsTests/ConnectionStatusHubTests.cs

5.21 Models.Requests.CreateVirtualMachineEntityRequest Class Reference

Properties

- required string **Name** [get, set]
- required uint **Memory** [get, set]
- required uint **VirtualCPUs** [get, set]
- required int **BootableDiskId** [get, set]
- required VirtualMachineEntityType **Type** [get, set]

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

- Models/Requests/CreateVirtualMachineEntityRequest.cs

5.22 Models.Requests.CreateVirtualNetworkNodeEntityRequest Class Reference

Properties

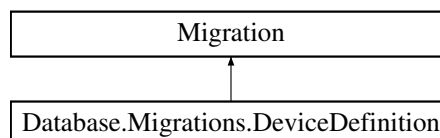
- required string **Name** [get, set]

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

- Models/Requests/CreateVirtualNetworkNodeEntityRequest.cs

5.23 Database.Migrations.DeviceDefinition Class Reference

Inheritance diagram for Database.Migrations.DeviceDefinition:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

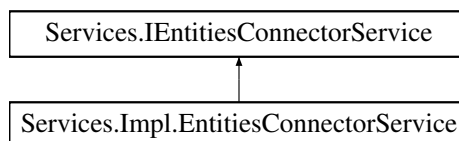
5.23.1 Detailed Description

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

- Database/Migrations/20250914152455_DeviceDefinition.cs
- Database/Migrations/20250914152455_DeviceDefinition.Designer.cs

5.24 Services.Impl.EntitiesConnectorService Class Reference

Inheritance diagram for Services.Impl.EntitiesConnectorService:



Public Member Functions

- **EntitiesConnectorService** ([IVirtualMachineEntityRepository](#) vmEntityRepository, [IVirtualNetworkConnectionRepository](#) connectionRepository, [IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeRepository, [IInternetEntityRepository](#) internetRepository, [IVirtualNetworkService](#) networkService, [IVirtualNetworkNodeConnector](#) virtualNetworkNodeConnector)
- `async Task< VirtualNetworkConnectionDTO > ConnectTwoEntities` (int sourceEntityId, [EntityType](#) sourceEntityType, int destinationEntityId, [EntityType](#) destinationEntityType)
Connects two entities of different types.
- `async Task< VirtualNetworkConnectionDTO > ConnectTwoVirtualMachines` (int sourceEntityId, int destinationEntityId)
Connects two virtual machines.
- `async Task< VirtualNetworkConnectionDTO > ConnectVirtualMachineToVirtualNetworkNode` (int sourceEntityId, int destinationEntityId)
Connects a virtual machine to a virtual network node.
- `async Task< VirtualNetworkConnectionDTO > ConnectVirtualMachineToInternet` (int sourceEntityId, int destinationEntityId)
Connects a virtual machine to the Internet.

5.24.1 Member Function Documentation

5.24.1.1 ConnectTwoEntities()

```

async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesConnectorService.ConnectTwoEntities (
    int sourceEntityId,
    EntityType sourceEntityType,
    int destinationEntityId,
    EntityType destinationEntityType) [inline]
  
```

Connects two entities of different types.

Parameters

<i>sourceEntityId</i>	Source entity identifier.
<i>sourceEntityType</i>	Source entity type.
<i>destinationEntityId</i>	Destination entity identifier.
<i>destinationEntityType</i>	Destination entity type.

Returns

Created connection data.

Implements [Services.IEntitiesConnectorService](#).

5.24.1.2 ConnectTwoVirtualMachines()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesConnectorService.ConnectTwo↔
VirtualMachines (
    int sourceEntityId,
    int destinationEntityId) [inline]
```

Connects two virtual machines.

Parameters

<i>sourceEntityId</i>	First virtual machine identifier.
<i>destination↔ EntityId</i>	Second virtual machine identifier.

Returns

Created connection data.

Implements [Services.IEntitiesConnectorService](#).

5.24.1.3 ConnectVirtualMachineToInternet()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesConnectorService.Connect↔
VirtualMachineToInternet (
    int sourceEntityId,
    int destinationEntityId) [inline]
```

Connects a virtual machine to the Internet.

Parameters

<i>sourceEntityId</i>	Virtual machine identifier.
<i>destination↔ EntityId</i>	Internet entity identifier.

Returns

Created connection data.

Implements [Services.IEntitiesConnectorService](#).

5.24.1.4 ConnectVirtualMachineToVirtualNetworkNode()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesConnectorService.Connect↔
VirtualMachineToVirtualNetworkNode (
    int sourceEntityId,
    int destinationEntityId) [inline]
```

Connects a virtual machine to a virtual network node.

Parameters

<i>sourceEntityId</i>	Virtual machine identifier.
<i>destination↔EntityId</i>	Network node identifier.

Returns

Created connection data.

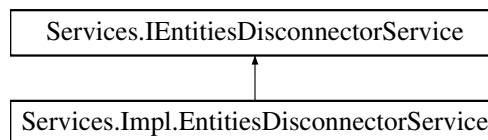
Implements [Services.IEntitiesConnectorService](#).

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

- Services/Impl/EntitiesConnectorService.cs

5.25 Services.Impl.EntitiesDisconnectService Class Reference

Inheritance diagram for Services.Impl.EntitiesDisconnectService:



Public Member Functions

- **EntitiesDisconnectService** ([IVirtualizationWrapper](#) wrapper, [IVirtualMachineEntityRepository](#) vmEntity↔Repository, [IVirtualNetworkConnectionRepository](#) connectionRepository, [IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeRepository, [IVirtualNetworkRepository](#) networkRepository, [IVirtualNetworkNodeConnector](#) virtualNetworkNodeConnector, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterface↔Repository)
- async Task< [VirtualNetworkConnectionDTO](#) > [DisconnectEntities](#) (int connectionId)
Disconnects entities based on connection identifier.
- async Task [DisconnectVirtualMachineFromVirtualNetworkNode](#) (int connectionId, int sourceEntityId, int destinationEntityId)
Disconnects a virtual machine from a virtual network node.
- async Task [DisconnectVirtualMachineFromVirtualMachine](#) (int connectionId, int sourceEntityId, int destinationEntityId)
Disconnects two virtual machines.
- async Task [DisconnectVirtualMachineFromInternet](#) (int connectionId, int virtualMachineEntityId, int internet↔EntityId)
Disconnects a virtual machine from the Internet.

5.25.1 Member Function Documentation

5.25.1.1 DisconnectEntities()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesDisconnectorService.DisconnectEntities (
    int connectionId) [inline]
```

Disconnects entities based on connection identifier.

Parameters

<i>connectionId</i>	Connection identifier to remove.
---------------------	----------------------------------

Returns

Removed connection data.

Implements [Services.IEntitiesDisconnectorService](#).

5.25.1.2 DisconnectVirtualMachineFromInternet()

```
async Task Services.Impl.EntitiesDisconnectorService.DisconnectVirtualMachineFromInternet (
    int connectionId,
    int virtualMachineEntityId,
    int internetEntityId) [inline]
```

Disconnects a virtual machine from the Internet.

Parameters

<i>connectionId</i>	Connection identifier.
<i>virtualMachineEntityId</i>	Virtual machine identifier.
<i>internetEntityId</i>	Internet entity identifier.

Implements [Services.IEntitiesDisconnectorService](#).

5.25.1.3 DisconnectVirtualMachineFromVirtualMachine()

```
async Task Services.Impl.EntitiesDisconnectorService.DisconnectVirtualMachineFromVirtualMachine (
    int connectionId,
    int sourceEntityId,
    int destinationEntityId) [inline]
```

Disconnects two virtual machines.

Parameters

<i>connectionId</i>	Connection identifier.
<i>sourceEntityId</i>	First virtual machine identifier.
<i>destination↔EntityId</i>	Second virtual machine identifier.

Implements [Services.IEntitiesDisconnectorService](#).

5.25.1.4 DisconnectVirtualMachineFromVirtualNetworkNode()

```
async Task Services.Impl.EntitiesDisconnectorService.DisconnectVirtualMachineFromVirtual↔
NetworkNode (
    int connectionId,
    int sourceEntityId,
    int destinationEntityId) [inline]
```

Disconnects a virtual machine from a virtual network node.

Parameters

<i>connectionId</i>	Connection identifier.
<i>sourceEntityId</i>	Virtual machine identifier.
<i>destination↔EntityId</i>	Network node identifier.

Implements [Services.IEntitiesDisconnectorService](#).

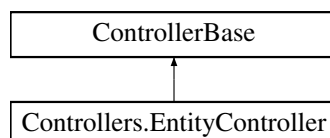
The documentation for this class was generated from the following file:

- Services/Impl/EntitiesDisconnectorService.cs

5.26 Controllers.EntityController Class Reference

Controller managing entities in the system.

Inheritance diagram for Controllers.EntityController:



Public Member Functions

- **EntityController** ([IVirtualMachineEntityService](#) vmEntityService, [IVirtualNetworkService](#) virtualNetworkService, [IInternetEntityService](#) internetEntityService)
- `async Task< ActionResult< VirtualMachineEntityResponse > > CreateVirtualMachineEntity (CreateVirtualMachineEntityRequest req)`
Creates a new virtual machine entity.
- `async Task< ActionResult< VirtualNetworkNodesEntitiesResponse > > CreateVirtualNetworkNodeEntity (CreateVirtualNetworkNodeEntityRequest req)`
Creates a new virtual network node.
- `async Task< ActionResult< InternetEntitiesResponse > > CreateInternetEntity ()`
Creates a new Internet entity.
- `async Task< ActionResult< StringResponse > > UpdateEntityPosition (UpdateEntityPositionRequest req)`
Updates entity position on the board.
- `async Task< ActionResult< VirtualMachineEntitiesResponse > > GetAllVirtualMachineEntities ()`
- `async Task< ActionResult< VirtualNetworkNodesEntitiesResponse > > GetAllVirtualNetworkNodeEntities ()`
- `async Task< ActionResult< InternetEntitiesResponse > > GetAllInternetEntities ()`

5.26.1 Detailed Description

Controller managing entities in the system.

5.26.2 Member Function Documentation

5.26.2.1 CreateInternetEntity()

```
async Task< ActionResult< InternetEntitiesResponse > > Controllers.EntityController.CreateInternetEntity () [inline]
```

Creates a new Internet entity.

Returns

Created Internet entity.

5.26.2.2 CreateVirtualMachineEntity()

```
async Task< ActionResult< VirtualMachineEntityResponse > > Controllers.EntityController.CreateVirtualMachineEntity (
    CreateVirtualMachineEntityRequest req) [inline]
```

Creates a new virtual machine entity.

Parameters

<i>req</i>	Data for creating virtual machine.
------------	------------------------------------

Returns

Created virtual machine entity.

5.26.2.3 CreateVirtualNetworkNodeEntity()

```
async Task< ActionResult< VirtualNetworkNodesEntitiesResponse > > Controllers.EntityController.CreateVirtualNetworkNodeEntity (
    CreateVirtualNetworkNodeEntityRequest req) [inline]
```

Creates a new virtual network node.

Parameters

<i>req</i>	Data for creating node.
------------	-------------------------

Returns

Created network node.

5.26.2.4 UpdateEntityPosition()

```
async Task< ActionResult< StringResponse > > Controllers.EntityController.UpdateEntityPosition (
    UpdateEntityPositionRequest req) [inline]
```

Updates entity position on the board.

Parameters

<i>req</i>	Data for updating position.
------------	-----------------------------

Returns

Operation confirmation.

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

- Controllers/EntityController.cs

5.27 Interconnect.Middlewares.ExceptionHandlerMiddleware Class Reference

Middleware for global exception handling in the application.

Public Member Functions

- **ExceptionHandlerMiddleware** (RequestDelegate next)
- async Task **InvokeAsync** (HttpContext context)
Processes HTTP request and catches exceptions.

5.27.1 Detailed Description

Middleware for global exception handling in the application.

5.27.2 Member Function Documentation

5.27.2.1 InvokeAsync()

```
async Task Interconnect.Middlewares.ExceptionHandlerMiddleware.InvokeAsync (
    HttpContext context) [inline]
```

Processes HTTP request and catches exceptions.

Parameters

<i>context</i>	HTTP context.
----------------	---------------

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

- Interconnect/Middlewares/ExceptionHandlerMiddleware.cs

5.28 InterconnectTests.ExceptionHandlerMiddlewareTests Class Reference

Public Member Functions

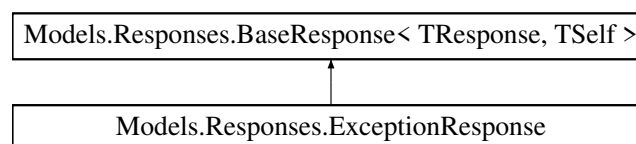
- async Task **InvokeAsync_NoException_CallsNextMiddleware** ()
- async Task **InvokeAsync_WhenExceptionThrown_ReturnsErrorResponse** ()

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

- InterconnectTests/ExceptionHandlerMiddlewareTests.cs

5.29 Models.Responses.ExceptionResponse Class Reference

Inheritance diagram for Models.Responses.ExceptionResponse:



Additional Inherited Members**Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)**

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

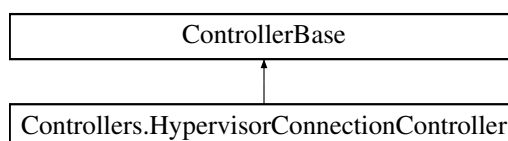
The documentation for this class was generated from the following file:

- Models/Responses/ExceptionResponse.cs

5.30 Controllers.HypervisorConnectionController Class Reference

Controller managing hypervisor connection.

Inheritance diagram for Controllers.HypervisorConnectionController:

**Public Member Functions**

- **HypervisorConnectionController** (IHypervisorConnectionService hypervisorConnectionService)
- ActionResult< [ConnectionInfoResponse](#) > [ConnectionInfo](#) ()
Retrieves hypervisor connection information.

5.30.1 Detailed Description

Controller managing hypervisor connection.

5.30.2 Member Function Documentation

5.30.2.1 ConnectionInfo()

```
ActionResult< ConnectionInfoResponse > Controllers.HypervisorConnectionController.ConnectionInfo () [inline]
```

Retrieves hypervisor connection information.

Returns

Connection information.

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

- Controllers/HypervisorConnectionController.cs

5.31 ControllersTests.HypervisorConnectionControllerTests Class Reference

Public Member Functions

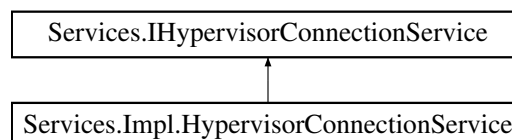
- void **Setup** ()
- void **ConnectionInfo_WhenInvoked_ShouldGetConnectionInfo** ()

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

- ControllersTests/HypervisorConnectionControllerTests.cs

5.32 Services.Impl.HypervisorConnectionService Class Reference

Inheritance diagram for Services.Impl.HypervisorConnectionService:



Public Member Functions

- **HypervisorConnectionService** ([IVirtualizationWrapper](#) vmManager)
- void [InitializeConnection](#) (string? connectionUrl)
Initializes connection to the hypervisor.
- [ConnectionStatus](#) [GetConnectionStatus](#) ()
Retrieves the current connection status with the hypervisor.
- [ConnectionInfo](#) [GetConnectionInfo](#) ()
Retrieves connection and hypervisor information.

5.32.1 Member Function Documentation

5.32.1.1 GetConnectionInfo()

```
ConnectionInfo Services.Impl.HypervisorConnectionService.GetConnectionInfo () [inline]
```

Retrieves connection and hypervisor information.

Returns

Connection information.

Implements [Services.IHypervisorConnectionService](#).

5.32.1.2 GetConnectionStatus()

```
ConnectionStatus Services.Impl.HypervisorConnectionService.GetConnectionStatus () [inline]
```

Retrieves the current connection status with the hypervisor.

Returns

Connection status.

Implements [Services.IHypervisorConnectionService](#).

5.32.1.3 InitializeConnection()

```
void Services.Impl.HypervisorConnectionService.InitializeConnection (  
    string? connectionUrl) [inline]
```

Initializes connection to the hypervisor.

Parameters

<i>connectionUrl</i>	Connection URL or null for default.
----------------------	-------------------------------------

Implements [Services.IHypervisorConnectionService](#).

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

- Services/Impl/HypervisorConnectionService.cs

5.33 ServicesTests.HypervisorConnectionServiceTests Class Reference

Public Member Functions

- void **SetUp** ()
- void **InitializeConnection_WhenInvokedWithoutConnectionUrl_ShouldConnectUsingNativeLibrary↔
WithDefaultConnectionUrl** ()
- void **InitializeConnection_WhenInvoked_ShouldConnectUsingNativeLibrary** ()
- void **GetConnectionInfo_WhenInvoked_ShouldGetConnectionInfo** ()
- void **GetConnectionStatus_WhenConnectionStatusIsAlive_ShouldReturnConnectionIsAlive** ()
- void **GetConnectionStatus_WhenConnectionStatusIsAlive_ShouldReturnConnectionIsDead** ()

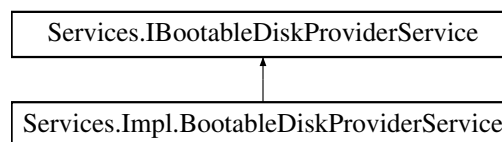
The documentation for this class was generated from the following file:

- ServicesTests/HypervisorConnectionServiceTests.cs

5.34 Services.IBootableDiskProviderService Interface Reference

Service providing information about available bootable disks.

Inheritance diagram for Services.IBootableDiskProviderService:



Public Member Functions

- Task< List< [BootableDiskModelDTO](#) > > [GetAvailableBootableDiskModels](#) ()
Retrieves a list of available bootable disk models.
- Task< string?> [GetBootableDiskPathById](#) (int id)
Retrieves the path to a bootable disk by its identifier.

5.34.1 Detailed Description

Service providing information about available bootable disks.

5.34.2 Member Function Documentation

5.34.2.1 GetAvailableBootableDiskModels()

```
Task< List< BootableDiskModelDTO > > Services.IBootableDiskProviderService.GetAvailable↔
BootableDiskModels ()
```

Retrieves a list of available bootable disk models.

Returns

List of bootable disks.

Implemented in [Services.Impl.BootableDiskProviderService](#).

5.34.2.2 GetBootableDiskPathById()

```
Task< string?> Services.IBootableDiskProviderService.GetBootableDiskPathById (
    int id)
```

Retrieves the path to a bootable disk by its identifier.

Parameters

<i>id</i>	Disk identifier.
-----------	------------------

Returns

Path to the disk or null if not found.

Implemented in [Services.Impl.BootableDiskProviderService](#).

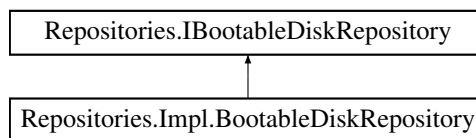
The documentation for this interface was generated from the following file:

- Services/IBootableDiskProviderService.cs

5.35 Repositories.IBootableDiskRepository Interface Reference

Repository managing bootable disks in the database.

Inheritance diagram for Repositories.IBootableDiskRepository:



Public Member Functions

- Task< List< [BootableDiskModel](#) > > [GetOnlyWithNotNullablePath](#) ()
Retrieves a list of bootable disks with non-null paths.
- Task< [BootableDiskModel](#) > [GetById](#) (int id)
Retrieves a bootable disk by identifier.

5.35.1 Detailed Description

Repository managing bootable disks in the database.

5.35.2 Member Function Documentation

5.35.2.1 GetById()

```
Task< BootableDiskModel > Repositories.IBootableDiskRepository.GetById (
    int id)
```

Retrieves a bootable disk by identifier.

Parameters

<i>id</i>	Disk identifier.
-----------	------------------

Returns

Bootable disk.

Implemented in [Repositories.Impl.BootableDiskRepository](#).

5.35.2.2 GetOnlyWithNotNullablePath()

```
Task< List< BootableDiskModel > > Repositories.IBootableDiskRepository.GetOnlyWithNotNullablePath ()
```

Retrieves a list of bootable disks with non-null paths.

Returns

List of disks with paths.

Implemented in [Repositories.Impl.BootableDiskRepository](#).

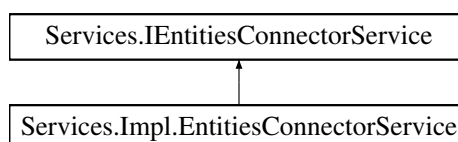
The documentation for this interface was generated from the following file:

- Repositories/IBootableDiskRepository.cs

5.36 Services.IEntitiesConnectorService Interface Reference

Service responsible for connecting network entities.

Inheritance diagram for Services.IEntitiesConnectorService:



Public Member Functions

- Task< [VirtualNetworkConnectionDTO](#) > [ConnectTwoEntities](#) (int sourceEntityId, [EntityType](#) sourceEntityType, int destinationEntityId, [EntityType](#) destinationEntityType)
Connects two entities of different types.
- Task< [VirtualNetworkConnectionDTO](#) > [ConnectTwoVirtualMachines](#) (int sourceEntityId, int destinationEntityId)
Connects two virtual machines.
- Task< [VirtualNetworkConnectionDTO](#) > [ConnectVirtualMachineToVirtualNetworkNode](#) (int sourceEntityId, int destinationEntityId)
Connects a virtual machine to a virtual network node.
- Task< [VirtualNetworkConnectionDTO](#) > [ConnectVirtualMachineToInternet](#) (int sourceEntityId, int destinationEntityId)
Connects a virtual machine to the Internet.

5.36.1 Detailed Description

Service responsible for connecting network entities.

5.36.2 Member Function Documentation

5.36.2.1 ConnectTwoEntities()

```
Task< VirtualNetworkConnectionDTO > Services.IEntitiesConnectorService.ConnectTwoEntities (
    int sourceEntityId,
    EntityType sourceEntityType,
    int destinationEntityId,
    EntityType destinationEntityType)
```

Connects two entities of different types.

Parameters

<i>sourceEntityId</i>	Source entity identifier.
<i>sourceEntityType</i>	Source entity type.
<i>destinationEntityId</i>	Destination entity identifier.
<i>destinationEntityType</i>	Destination entity type.

Returns

Created connection data.

Implemented in [Services.Impl.EntitiesConnectorService](#).

5.36.2.2 ConnectTwoVirtualMachines()

```
Task< VirtualNetworkConnectionDTO > Services.IEntitiesConnectorService.ConnectTwoVirtual↔
Machines (
    int sourceEntityId,
    int destinationEntityId)
```

Connects two virtual machines.

Parameters

<i>sourceEntityId</i>	First virtual machine identifier.
<i>destination↔ EntityId</i>	Second virtual machine identifier.

Returns

Created connection data.

Implemented in [Services.Impl.EntitiesConnectorService](#).

5.36.2.3 ConnectVirtualMachineToInternet()

```
Task< VirtualNetworkConnectionDTO > Services.IEntitiesConnectorService.ConnectVirtualMachine↔
ToInternet (
    int sourceEntityId,
    int destinationEntityId)
```

Connects a virtual machine to the Internet.

Parameters

<i>sourceEntityId</i>	Virtual machine identifier.
<i>destinationEntityId</i>	Internet entity identifier.

Returns

Created connection data.

Implemented in [Services.Impl.EntitiesConnectorService](#).

5.36.2.4 ConnectVirtualMachineToVirtualNetworkNode()

```
Task< VirtualNetworkConnectionDTO > Services.IEntitiesConnectorService.ConnectVirtualMachine↔
ToVirtualNetworkNode (
    int sourceEntityId,
    int destinationEntityId)
```

Connects a virtual machine to a virtual network node.

Parameters

<i>sourceEntityId</i>	Virtual machine identifier.
<i>destinationEntityId</i>	Network node identifier.

Returns

Created connection data.

Implemented in [Services.Impl.EntitiesConnectorService](#).

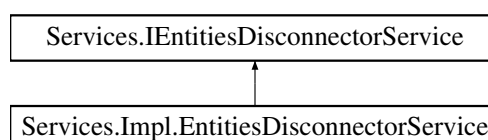
The documentation for this interface was generated from the following file:

- Services/IEntitiesConnectorService.cs

5.37 Services.IEntitiesDisconnectorService Interface Reference

Service responsible for disconnecting network entities.

Inheritance diagram for Services.IEntitiesDisconnectorService:



Public Member Functions

- Task< [VirtualNetworkConnectionDTO](#) > [DisconnectEntities](#) (int connectionId)
Disconnects entities based on connection identifier.
- Task [DisconnectVirtualMachineFromVirtualNetworkNode](#) (int connectionId, int sourceEntityId, int destinationEntityId)
Disconnects a virtual machine from a virtual network node.
- Task [DisconnectVirtualMachineFromVirtualMachine](#) (int connectionId, int sourceEntityId, int destinationEntityId)
Disconnects two virtual machines.
- Task [DisconnectVirtualMachineFromInternet](#) (int connectionId, int virtualMachineEntityId, int internetEntityId)
Disconnects a virtual machine from the Internet.

5.37.1 Detailed Description

Service responsible for disconnecting network entities.

5.37.2 Member Function Documentation

5.37.2.1 DisconnectEntities()

```
Task< VirtualNetworkConnectionDTO > Services.IEntitiesDisconnectorService.DisconnectEntities (
    int connectionId)
```

Disconnects entities based on connection identifier.

Parameters

<i>connectionId</i>	Connection identifier to remove.
---------------------	----------------------------------

Returns

Removed connection data.

Implemented in [Services.Impl.EntitiesDisconnectorService](#).

5.37.2.2 DisconnectVirtualMachineFromInternet()

```
Task Services.IEntitiesDisconnectorService.DisconnectVirtualMachineFromInternet (
    int connectionId,
    int virtualMachineEntityId,
    int internetEntityId)
```

Disconnects a virtual machine from the Internet.

Parameters

<i>connectionId</i>	Connection identifier.
<i>virtualMachineEntityId</i>	Virtual machine identifier.
<i>internetEntityId</i>	Internet entity identifier.

Implemented in [Services.Impl.EntitiesDisconnectorService](#).

5.37.2.3 DisconnectVirtualMachineFromVirtualMachine()

```
Task Services.IEntitiesDisconnectorService.DisconnectVirtualMachineFromVirtualMachine (
    int connectionId,
    int sourceEntityId,
    int destinationEntityId)
```

Disconnects two virtual machines.

Parameters

<i>connectionId</i>	Connection identifier.
<i>sourceEntityId</i>	First virtual machine identifier.
<i>destinationEntityId</i>	Second virtual machine identifier.

Implemented in [Services.Impl.EntitiesDisconnectorService](#).

5.37.2.4 DisconnectVirtualMachineFromVirtualNetworkNode()

```
Task Services.IEntitiesDisconnectorService.DisconnectVirtualMachineFromVirtualNetworkNode (
    int connectionId,
    int sourceEntityId,
    int destinationEntityId)
```

Disconnects a virtual machine from a virtual network node.

Parameters

<i>connectionId</i>	Connection identifier.
<i>sourceEntityId</i>	Virtual machine identifier.
<i>destinationEntityId</i>	Network node identifier.

Implemented in [Services.Impl.EntitiesDisconnectorService](#).

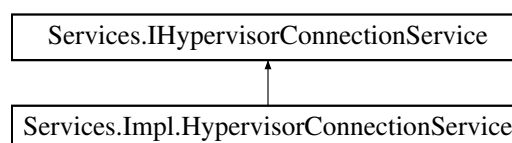
The documentation for this interface was generated from the following file:

- Services/IEntitiesDisconnectorService.cs

5.38 Services.IHypervisorConnectionService Interface Reference

Service managing connection to the virtualization hypervisor.

Inheritance diagram for Services.IHypervisorConnectionService:



Public Member Functions

- void [InitializeConnection](#) (string? connectionUrl)
Initializes connection to the hypervisor.
- [ConnectionStatus](#) [GetConnectionStatus](#) ()
Retrieves the current connection status with the hypervisor.
- [ConnectionInfo](#) [GetConnectionInfo](#) ()
Retrieves connection and hypervisor information.

5.38.1 Detailed Description

Service managing connection to the virtualization hypervisor.

5.38.2 Member Function Documentation

5.38.2.1 GetConnectionInfo()

[ConnectionInfo](#) `Services.IHypervisorConnectionService.GetConnectionInfo ()`

Retrieves connection and hypervisor information.

Returns

Connection information.

Implemented in [Services.Impl.HypervisorConnectionService](#).

5.38.2.2 GetConnectionStatus()

[ConnectionStatus](#) `Services.IHypervisorConnectionService.GetConnectionStatus ()`

Retrieves the current connection status with the hypervisor.

Returns

Connection status.

Implemented in [Services.Impl.HypervisorConnectionService](#).

5.38.2.3 InitializeConnection()

`void Services.IHypervisorConnectionService.InitializeConnection (`
 `string? connectionUrl)`

Initializes connection to the hypervisor.

Parameters

<code>connectionUrl</code>	Connection URL or null for default.
----------------------------	-------------------------------------

Implemented in [Services.Impl.HypervisorConnectionService](#).

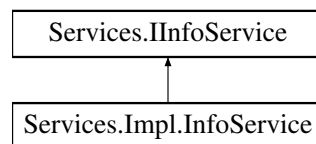
The documentation for this interface was generated from the following file:

- `Services/IHypervisorConnectionService.cs`

5.39 Services.IInfoService Interface Reference

Service providing system information.

Inheritance diagram for `Services.IInfoService`:



Public Member Functions

- [InformationModel](#) `GetSystemInfo ()`
Retrieves operating system information.

5.39.1 Detailed Description

Service providing system information.

5.39.2 Member Function Documentation

5.39.2.1 GetSystemInfo()

`InformationModel` `Services.IInfoService.GetSystemInfo ()`

Retrieves operating system information.

Returns

System information.

Implemented in [Services.Impl.InfoService](#).

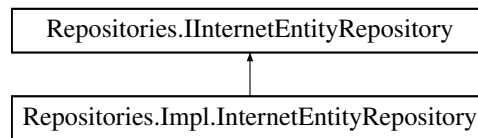
The documentation for this interface was generated from the following file:

- `Services/IInfoService.cs`

5.40 Repositories.IInternetEntityRepository Interface Reference

Repository managing Internet entities in the database.

Inheritance diagram for Repositories.IInternetEntityRepository:



Public Member Functions

- Task< [InternetEntityModel](#) > [Create](#) ([VirtualNetworkModel](#) virtualNetwork)
Creates a new Internet entity.
- Task< List< [InternetEntityModel](#) > > [GetAll](#) ()
Retrieves all Internet entities.
- Task< [InternetEntityModel](#) > [GetById](#) (int id)
Retrieves an Internet entity by identifier.
- Task< [InternetEntityModel](#) > [UpdatePosition](#) (int id, int x, int y)
Updates the position of an Internet entity.

5.40.1 Detailed Description

Repository managing Internet entities in the database.

5.40.2 Member Function Documentation

5.40.2.1 Create()

```
Task< InternetEntityModel > Repositories.IInternetEntityRepository.Create (
    VirtualNetworkModel virtualNetwork)
```

Creates a new Internet entity.

Parameters

<i>virtualNetwork</i>	Virtual network to connect.
-----------------------	-----------------------------

Returns

Created Internet entity.

Implemented in [Repositories.Impl.InternetEntityRepository](#).

5.40.2.2 GetAll()

```
Task< List< InternetEntityModel > > Repositories.IInternetEntityRepository.GetAll ()
```

Retrieves all Internet entities.

Returns

List of Internet entities.

Implemented in [Repositories.Impl.InternetEntityRepository](#).

5.40.2.3 GetByld()

```
Task< InternetEntityModel > Repositories.IInternetEntityRepository.GetById (  
    int id)
```

Retrieves an Internet entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Internet entity.

Implemented in [Repositories.Impl.InternetEntityRepository](#).

5.40.2.4 UpdatePosition()

```
Task< InternetEntityModel > Repositories.IInternetEntityRepository.UpdatePosition (  
    int id,  
    int x,  
    int y)
```

Updates the position of an Internet entity.

Parameters

<i>id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated entity.

Implemented in [Repositories.Impl.InternetEntityRepository](#).

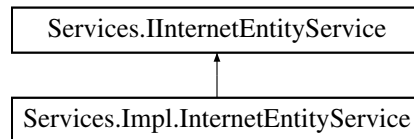
The documentation for this interface was generated from the following file:

- [Repositories/IInternetEntityRepository.cs](#)

5.41 Services.IInternetEntityService Interface Reference

Service managing Internet entities.

Inheritance diagram for Services.IInternetEntityService:



Public Member Functions

- Task< [InternetEntityModelDTO](#) > [CreateInternet](#) ()
Creates a new Internet entity.
- Task< List< [InternetEntityModelDTO](#) > > [GetInternetEntities](#) ()
Retrieves a list of all Internet entities.
- Task< [InternetEntityModelDTO](#) > [UpdateInternetEntityPosition](#) (int entityId, int x, int y)
Updates the position of an Internet entity on the board.

5.41.1 Detailed Description

Service managing Internet entities.

5.41.2 Member Function Documentation

5.41.2.1 CreateInternet()

```
Task< InternetEntityModelDTO > Services.IInternetEntityService.CreateInternet ()
```

Creates a new Internet entity.

Returns

Created Internet entity.

Implemented in [Services.Impl.InternetEntityService](#).

5.41.2.2 GetInternetEntities()

```
Task< List< InternetEntityModelDTO > > Services.IInternetEntityService.GetInternetEntities ()
```

Retrieves a list of all Internet entities.

Returns

List of Internet entities.

Implemented in [Services.Impl.InternetEntityService](#).

5.41.2.3 UpdateInternetEntityPosition()

```
Task< InternetEntityModelDTO > Services.IInternetEntityService.UpdateInternetEntityPosition (
    int entityId,
    int x,
    int y)
```

Updates the position of an Internet entity on the board.

Parameters

<i>entity↔ Id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated Internet entity.

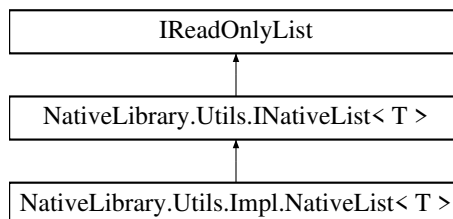
Implemented in [Services.Impl.InternetEntityService](#).

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

- Services/IInternetEntityService.cs

5.42 NativeLibrary.Utls.INativeList< T > Interface Template Reference

Inheritance diagram for NativeLibrary.Utls.INativeList< T >:



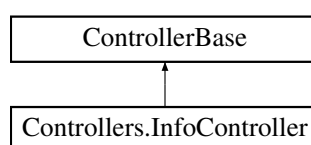
The documentation for this interface was generated from the following file:

- NativeLibrary/Utls/INativeList.cs

5.43 Controllers.InfoController Class Reference

Controller providing system information.

Inheritance diagram for Controllers.InfoController:



Public Member Functions

- **InfoController** (*IInfoService* infoService)
- `ActionResult< SystemInfoResponse > GetSystemInfo ()`
Retrieves operating system information.

5.43.1 Detailed Description

Controller providing system information.

5.43.2 Member Function Documentation

5.43.2.1 GetSystemInfo()

```
ActionResult< SystemInfoResponse > Controllers.InfoController.GetSystemInfo () [inline]
```

Retrieves operating system information.

Returns

System information.

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

- Controllers/InfoController.cs

5.44 ControllersTests.InfoControllerTests Class Reference

Public Member Functions

- void **Setup** ()
- void **GetSystemInfo_Invoke_ShouldReturnSystemInfo** ()

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

- ControllersTests/InfoControllerTests.cs

5.45 Models.InformationModel Class Reference

Model containing operating system information.

Properties

- required string **OsDescription** [get, set]
Operating system description.
- required string **OsArch** [get, set]
Operating system architecture.

5.45.1 Detailed Description

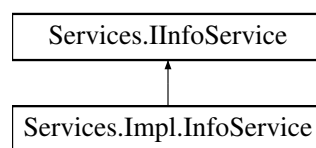
Model containing operating system information.

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

- Models/InformationModel.cs

5.46 Services.Impl.InfoService Class Reference

Inheritance diagram for Services.Impl.InfoService:



Public Member Functions

- [InformationModel](#) [GetSystemInfo](#) ()
Retrieves operating system information.

5.46.1 Member Function Documentation

5.46.1.1 GetSystemInfo()

```
InformationModel Services.Impl.InfoService.GetSystemInfo () [inline]
```

Retrieves operating system information.

Returns

System information.

Implements [Services.IInfoService](#).

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

- Services/Impl/InfoService.cs

5.47 ServicesTests.InfoServiceTests Class Reference

Public Member Functions

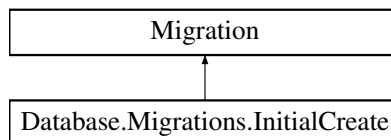
- void **GetOperatingSystem_Invoke_ShouldReturnOperatingSystem** ()

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

- ServicesTests/InfoServiceTests.cs

5.48 Database.Migrations.InitialCreate Class Reference

Inheritance diagram for Database.Migrations.InitialCreate:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.48.1 Detailed Description

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

- Database/Migrations/20250726213527_InitialCreate.cs
- Database/Migrations/20250726213527_InitialCreate.Designer.cs

5.49 Services.Impl.InitializationService Class Reference

Public Member Functions

- **InitializationService** (IHypervisorConnectionService connService, IOptions< InterconnectConfig > config)
- void **Initialize** ()
- Task **StartAsync** (CancellationTokens cancellationTokens)
- Task **StopAsync** (CancellationTokens cancellationTokens)

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

- Services/Impl/InitializationService.cs

5.50 ServicesTests.InitializationServiceTests Class Reference

Public Member Functions

- void **SetUp** ()
- void **Initialize_WhenInvoked_ShouldConnectToHypervisor** ()

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

- ServicesTests/InitializationServiceTests.cs

5.51 Models.Config.InterconnectConfig Class Reference

Properties

- required string **VmPrefix** [get, set]
- required string **HypervisorUrl** [get, set]
- required string **DatabaseConnectionUrl** [get, set]
- required int **MaxConsoleDataHistory** [get, set]

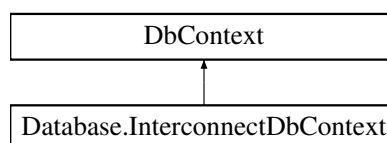
The documentation for this class was generated from the following file:

- Models/Config/InterconnectConfig.cs

5.52 Database.InterconnectDbContext Class Reference

[Database](#) context for the [Interconnect](#) application.

Inheritance diagram for Database.InterconnectDbContext:



Public Member Functions

- **InterconnectDbContext** (DbContextOptions< [InterconnectDbContext](#) > options)

Protected Member Functions

- override void **OnConfiguring** (DbContextOptionsBuilder optionsBuilder)

Properties

- `DbSet< VirtualMachineEntityModel > VirtualMachineEntityModels` [get, set]
Collection of virtual machine entities.
- `DbSet< BootableDiskModel > BootableDiskModels` [get, set]
Collection of bootable disks.
- `DbSet< VirtualNetworkEntityConnectionModel > VirtualNetworkEntityConnectionModels` [get, set]
Collection of entity connections in virtual network.
- `DbSet< VirtualNetworkNodeEntityModel > VirtualNetworkNodeEntityModels` [get, set]
Collection of virtual network nodes.
- `DbSet< InternetEntityModel > InternetEntityModels` [get, set]
Collection of Internet entities.
- `DbSet< VirtualNetworkModel > VirtualNetworkModels` [get, set]
Collection of virtual networks.
- `DbSet< VirtualMachineEntityNetworkInterfaceModel > VirtualMachineEntityNetworkInterfaceModels` [get, set]
Collection of virtual machine network interfaces.

5.52.1 Detailed Description

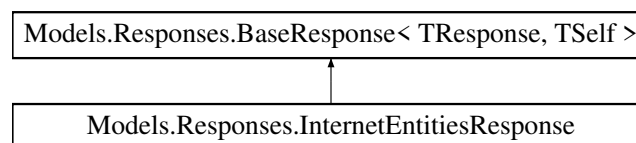
[Database](#) context for the [Interconnect](#) application.

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

- Database/InterconnectDbContext.cs

5.53 Models.Responses.InternetEntitiesResponse Class Reference

Inheritance diagram for Models.Responses.InternetEntitiesResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static `TSelf WithSuccess (TResponse data)`
Creates a successful response with data.
- static `TSelf WithEmptySuccess ()`
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

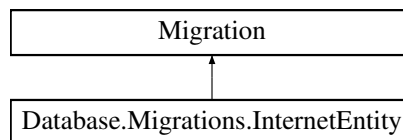
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/InternetEntitiesResponse.cs

5.54 Database.Migrations.InternetEntity Class Reference

Inheritance diagram for Database.Migrations.InternetEntity:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.54.1 Detailed Description

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

- Database/Migrations/20250907212022_InternetEntity.cs
- Database/Migrations/20250907212022_InternetEntity.Designer.cs

5.55 Models.Database.InternetEntityModel Class Reference

[Database](#) model representing an Internet entity connected to a virtual network.

Properties

- int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position.
- required int **Y** [get, set]
Y coordinate of the entity position.
- required [VirtualNetworkModel](#) **VirtualNetwork** [get, set]
Virtual network to which the entity is connected.

5.55.1 Detailed Description

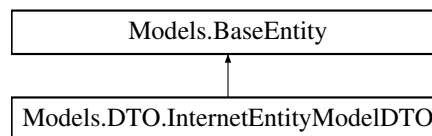
[Database](#) model representing an Internet entity connected to a virtual network.

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

- Models/Database/InternetEntityModel.cs

5.56 Models.DTO.InternetEntityModelIDTO Class Reference

Inheritance diagram for Models.DTO.InternetEntityModelIDTO:



Properties

- required string **IpAddress** [get, set]

Properties inherited from [Models.BaseEntity](#)

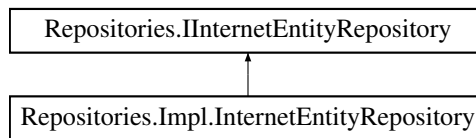
- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

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

- Models/DTO/InternetEntityModelIDTO.cs

5.57 Repositories.Impl.InternetEntityRepository Class Reference

Inheritance diagram for Repositories.Impl.InternetEntityRepository:



Public Member Functions

- **InternetEntityRepository** ([InterconnectDbContext](#) context)
- `async Task< InternetEntityModel > Create` ([VirtualNetworkModel](#) virtualNetwork)
Creates a new Internet entity.
- `async Task< List< InternetEntityModel > > GetAll` ()
Retrieves all Internet entities.
- `async Task< InternetEntityModel > GetById` (int id)
Retrieves an Internet entity by identifier.
- `async Task< InternetEntityModel > UpdatePosition` (int id, int x, int y)
Updates the position of an Internet entity.

5.57.1 Member Function Documentation

5.57.1.1 Create()

```

async Task< InternetEntityModel > Repositories.Impl.InternetEntityRepository.Create (
    VirtualNetworkModel virtualNetwork) [inline]
  
```

Creates a new Internet entity.

Parameters

<i>virtualNetwork</i>	Virtual network to connect.
-----------------------	-----------------------------

Returns

Created Internet entity.

Implements [Repositories.IInternetEntityRepository](#).

5.57.1.2 GetAll()

```

async Task< List< InternetEntityModel > > Repositories.Impl.InternetEntityRepository.GetAll
() [inline]
  
```

Retrieves all Internet entities.

Returns

List of Internet entities.

Implements [Repositories.IInternetEntityRepository](#).

5.57.1.3 GetById()

```
async Task< InternetEntityModel > Repositories.Impl.InternetEntityRepository.GetById (
    int id) [inline]
```

Retrieves an Internet entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Internet entity.

Implements [Repositories.IInternetEntityRepository](#).

5.57.1.4 UpdatePosition()

```
async Task< InternetEntityModel > Repositories.Impl.InternetEntityRepository.UpdatePosition (
    int id,
    int x,
    int y) [inline]
```

Updates the position of an Internet entity.

Parameters

<i>id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated entity.

Implements [Repositories.IInternetEntityRepository](#).

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

- Repositories/Impl/InternetEntityRepository.cs

5.58 RepositoriesTests.InternetEntityRepositoryTests Class Reference

Public Member Functions

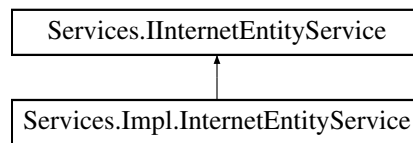
- void **SetUp** ()
- void **TearDown** ()
- async Task **Create_WhenInvoked_ShouldCreateNewInternetEntity** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllInternetEntities** ()

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

- RepositoriesTests/InternetEntityRepositoryTests.cs

5.59 Services.Impl.InternetEntityService Class Reference

Inheritance diagram for Services.Impl.InternetEntityService:



Public Member Functions

- **InternetEntityService** ([IInternetEntityRepository](#) internetRepository, [IVirtualNetworkService](#) virtualNetworkService)
- `async Task< InternetEntityModelDTO > CreateInternet ()`
Creates a new Internet entity.
- `async Task< List< InternetEntityModelDTO > > GetInternetEntities ()`
Retrieves a list of all Internet entities.
- `async Task< InternetEntityModelDTO > UpdateInternetEntityPosition (int entityId, int x, int y)`
Updates the position of an Internet entity on the board.

5.59.1 Member Function Documentation

5.59.1.1 CreateInternet()

```
async Task< InternetEntityModelDTO > Services.Impl.InternetEntityService.CreateInternet ()
[inline]
```

Creates a new Internet entity.

Returns

Created Internet entity.

Implements [Services.IInternetEntityService](#).

5.59.1.2 GetInternetEntities()

```
async Task< List< InternetEntityModelDTO > > Services.Impl.InternetEntityService.GetInternet↔
Entities () [inline]
```

Retrieves a list of all Internet entities.

Returns

List of Internet entities.

Implements [Services.IInternetEntityService](#).

5.59.1.3 UpdateInternetEntityPosition()

```
async Task< InternetEntityModelDTO > Services.Impl.InternetEntityService.UpdateInternet↔
EntityPosition (
    int entityId,
    int x,
    int y) [inline]
```

Updates the position of an Internet entity on the board.

Parameters

<i>entity↔ Id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated Internet entity.

Implements [Services.IInternetEntityService](#).

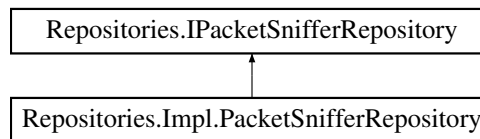
The documentation for this class was generated from the following file:

- Services/Impl/InternetEntityService.cs

5.60 Repositories.IPacketSnifferRepository Interface Reference

Repository managing packet analyzers in memory.

Inheritance diagram for Repositories.IPacketSnifferRepository:



Public Member Functions

- void [AddSniffer](#) ([PacketSniffer](#) sniffer)
Adds a packet analyzer to the repository.
- void [RemoveSniffer](#) (string bridgeName)
Removes a packet analyzer for a specified bridge.
- List< [PacketSniffer](#) > [GetOpenedSniffers](#) ()
Retrieves a list of opened packet analyzers.

5.60.1 Detailed Description

Repository managing packet analyzers in memory.

5.60.2 Member Function Documentation

5.60.2.1 AddSniffer()

```
void Repositories.IPacketSnifferRepository.AddSniffer (
    PacketSniffer sniffer)
```

Adds a packet analyzer to the repository.

Parameters

<i>sniffer</i>	Packet analyzer.
----------------	----------------------------------

Implemented in [Repositories.Impl.PacketSnifferRepository](#).

5.60.2.2 GetOpenedSniffers()

```
List< PacketSniffer > Repositories.IPacketSnifferRepository.GetOpenedSniffers ()
```

Retrieves a list of opened packet analyzers.

Returns

List of opened analyzers.

Implemented in [Repositories.Impl.PacketSnifferRepository](#).

5.60.2.3 RemoveSniffer()

```
void Repositories.IPacketSnifferRepository.RemoveSniffer (
    string bridgeName)
```

Removes a packet analyzer for a specified bridge.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

Implemented in [Repositories.Impl.PacketSnifferRepository](#).

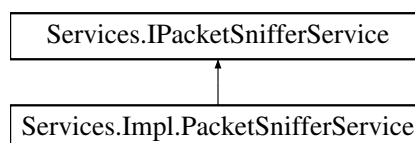
The documentation for this interface was generated from the following file:

- [Repositories/IPacketSnifferRepository.cs](#)

5.61 Services.IPacketSnifferService Interface Reference

Service for capturing network packets.

Inheritance diagram for Services.IPacketSnifferService:



Public Member Functions

- void [StartListeningForBridge](#) (string interfaceName)
Starts listening for packets on a specified network bridge.
- List< [PacketSniffer](#) > [GetOpenedSniffers](#) ()
Retrieves a list of active packet analyzers.
- bool [ListenForPacket](#) ([PacketSniffer](#) sniffer)
Listens for packets using the specified analyzer.
- [Packet?](#) [GetCapturedPacket](#) ()
Retrieves the last captured packet.

5.61.1 Detailed Description

Service for capturing network packets.

5.61.2 Member Function Documentation

5.61.2.1 GetCapturedPacket()

```
Packet? Services.IPacketSnifferService.GetCapturedPacket ()
```

Retrieves the last captured packet.

Returns

Captured packet or null if no packet available.

Implemented in [Services.Impl.PacketSnifferService](#).

5.61.2.2 GetOpenedSniffers()

```
List< PacketSniffer > Services.IPacketSnifferService.GetOpenedSniffers ()
```

Retrieves a list of active packet analyzers.

Returns

List of opened packet sniffers.

Implemented in [Services.Impl.PacketSnifferService](#).

5.61.2.3 ListenForPacket()

```
bool Services.IPacketSnifferService.ListenForPacket (  
    PacketSniffer sniffer)
```

Listens for packets using the specified analyzer.

Parameters

<i>sniffer</i>	Packet analyzer.
----------------	----------------------------------

Returns

True if packet was captured successfully, false otherwise.

Implemented in [Services.Impl.PacketSnifferService](#).

5.61.2.4 StartListeningForBridge()

```
void Services.IPacketSnifferService.StartListeningForBridge (
    string interfaceName)
```

Starts listening for packets on a specified network bridge.

Parameters

<i>interfaceName</i>	Network interface name.
----------------------	-------------------------

Implemented in [Services.Impl.PacketSnifferService](#).

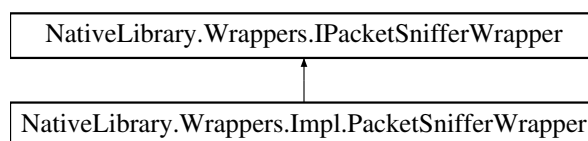
The documentation for this interface was generated from the following file:

- Services/IPacketSnifferService.cs

5.62 NativeLibrary.Wrappers.IPacketSnifferWrapper Interface Reference

Interface wrapper for native packet analyzer library.

Inheritance diagram for NativeLibrary.Wrappers.IPacketSnifferWrapper:



Public Member Functions

- IntPtr [OpenSnifferHandler](#) (string bridgeName)
Opens a packet analyzer handler for the specified bridge.
- bool [ListenForPacket](#) (IntPtr handler, string bridgeName)
Listens for a network packet.
- [NativePacket GetPacket](#) ()
Gets the last captured packet.
- int [GetNumberOfPackets](#) ()
Gets the number of captured packets.

5.62.1 Detailed Description

Interface wrapper for native packet analyzer library.

5.62.2 Member Function Documentation

5.62.2.1 GetNumberOfPackets()

```
int NativeLibrary.Wrappers.IPacketSnifferWrapper.GetNumberOfPackets ()
```

Gets the number of captured packets.

Returns

Number of packets.

Implemented in [NativeLibrary.Wrappers.Impl.PacketSnifferWrapper](#).

5.62.2.2 GetPacket()

```
NativePacket NativeLibrary.Wrappers.IPacketSnifferWrapper.GetPacket ()
```

Gets the last captured packet.

Returns

Captured packet.

Implemented in [NativeLibrary.Wrappers.Impl.PacketSnifferWrapper](#).

5.62.2.3 ListenForPacket()

```
bool NativeLibrary.Wrappers.IPacketSnifferWrapper.ListenForPacket (  
    IntPtr handler,  
    string bridgeName)
```

Listens for a network packet.

Parameters

<i>handler</i>	Pointer to the analyzer handler.
<i>bridgeName</i>	Name of the network bridge.

Returns

True if a packet was captured.

5.62.2.4 OpenSnifferHandler()

```
IntPtr NativeLibrary.Wrappers.IPacketSnifferWrapper.OpenSnifferHandler (  
    string bridgeName)
```

Opens a packet analyzer handler for the specified bridge.

Parameters

<i>bridgeName</i>	Name of the network bridge.
-------------------	-----------------------------

Returns

Pointer to the analyzer handler.

Implemented in [NativeLibrary.Wrappers.Impl.PacketSnifferWrapper](#).

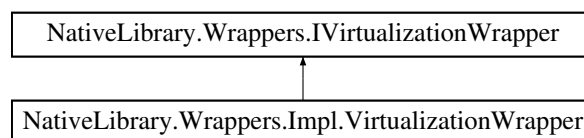
The documentation for this interface was generated from the following file:

- NativeLibrary/Wrappers/IPacketSnifferWrapper.cs

5.63 NativeLibrary.Wrappers.IVirtualizationWrapper Interface Reference

Interface wrapper for native virtualization library.

Inheritance diagram for NativeLibrary.Wrappers.IVirtualizationWrapper:



Public Member Functions

- void [InitializeConnection](#) (string? connectionUrl)
Initializes connection with the hypervisor.
- [NativeConnectionInfo GetConnectionInfo](#) ()
Gets connection information from the hypervisor.
- void [CreateVirtualMachine](#) (string xmlDefinition)
Creates a virtual machine based on XML definition.
- [NativeVirtualMachineInfo GetVirtualMachineInfo](#) (string name)
Gets information about a virtual machine.
- INativeList< [NativeVirtualMachineInfo](#) > [GetListOfVirtualMachines](#) ()
Gets a list of all virtual machines.
- bool [IsConnectionAlive](#) ()
Checks if the connection with the hypervisor is active.
- IntPtr [OpenVirtualMachineConsole](#) (Guid uuid)
Opens a console stream for a virtual machine.
- [NativeStreamData GetDataFromStream](#) (IntPtr stream)
Gets data from the console stream.
- void [SendDataToStream](#) (IntPtr stream, string data)
Sends data to the console stream.
- void [CloseStream](#) (IntPtr stream)
Closes the console stream.

- IntPtr [CreateVirtualNetwork](#) (string networkDefinition)
Creates a virtual network based on definition.
- void [AttachDeviceToVirtualMachine](#) (Guid uuid, string deviceDefinition)
Attaches a device to a virtual machine.
- void [DetachDeviceFromVirtualMachine](#) (Guid uuid, string deviceDefinition)
Detaches a device from a virtual machine.
- void [UpdateVmDevice](#) (Guid uuid, string deviceDefinition)
Updates a device on a virtual machine.
- void [DestroyNetwork](#) (string name)
Destroys a virtual network.

5.63.1 Detailed Description

Interface wrapper for native virtualization library.

5.63.2 Member Function Documentation

5.63.2.1 AttachDeviceToVirtualMachine()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.AttachDeviceToVirtualMachine (
    Guid uuid,
    string deviceDefinition)
```

Attaches a device to a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.2 CloseStream()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.CloseStream (
    IntPtr stream)
```

Closes the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
---------------	------------------------

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.3 CreateVirtualMachine()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.CreateVirtualMachine (
    string xmlDefinition)
```

Creates a virtual machine based on XML definition.

Parameters

<i>xmlDefinition</i>	Machine definition in XML format.
----------------------	-----------------------------------

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.4 CreateVirtualNetwork()

```
IntPtr NativeLibrary.Wrappers.IVirtualizationWrapper.CreateVirtualNetwork (
    string networkDefinition)
```

Creates a virtual network based on definition.

Parameters

<i>networkDefinition</i>	Network definition in XML format.
--------------------------	-----------------------------------

Returns

Pointer to the created network.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.5 DestroyNetwork()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.DestroyNetwork (
    string name)
```

Destroys a virtual network.

Parameters

<i>name</i>	Name of the network to destroy.
-------------	---------------------------------

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.6 DetachDeviceFromVirtualMachine()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.DetachDeviceFromVirtualMachine (
    Guid uuid,
    string deviceDefinition)
```

Detaches a device from a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.7 GetConnectionInfo()

```
NativeConnectionInfo NativeLibrary.Wrappers.IVirtualizationWrapper.GetConnectionInfo ()
```

Gets connection information from the hypervisor.

Returns

Connection information.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.8 GetDataFromStream()

```
NativeStreamData NativeLibrary.Wrappers.IVirtualizationWrapper.GetDataFromStream (
    IntPtr stream)
```

Gets data from the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
---------------	------------------------

Returns

Data from the stream.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.9 GetListOfVirtualMachines()

```
INativeList< NativeVirtualMachineInfo > NativeLibrary.Wrappers.IVirtualizationWrapper.Get↔
ListOfVirtualMachines ()
```

Gets a list of all virtual machines.

Returns

List of virtual machines.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.10 GetVirtualMachineInfo()

```
NativeVirtualMachineInfo NativeLibrary.Wrappers.IVirtualizationWrapper.GetVirtualMachineInfo (
    string name)
```

Gets information about a virtual machine.

Parameters

<i>name</i>	Name of the virtual machine.
-------------	------------------------------

Returns

Virtual machine information.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.11 InitializeConnection()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.InitializeConnection (
    string? connectionUrl)
```

Initializes connection with the hypervisor.

Parameters

<i>connectionUrl</i>	Connection URL or null for default.
----------------------	-------------------------------------

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.12 IsConnectionAlive()

```
bool NativeLibrary.Wrappers.IVirtualizationWrapper.IsConnectionAlive ()
```

Checks if the connection with the hypervisor is active.

Returns

True if connection is active.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.13 OpenVirtualMachineConsole()

```
IntPtr NativeLibrary.Wrappers.IVirtualizationWrapper.OpenVirtualMachineConsole (
    Guid uuid)
```

Opens a console stream for a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
-------------	------------------------------

Returns

Pointer to the console stream.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.14 SendDataToStream()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.SendDataToStream (
    IntPtr stream,
    string data)
```

Sends data to the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
<i>data</i>	Data to send.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.63.2.15 UpdateVmDevice()

```
void NativeLibrary.Wrappers.IVirtualizationWrapper.UpdateVmDevice (
    Guid uuid,
    string deviceDefinition)
```

Updates a device on a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implemented in [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

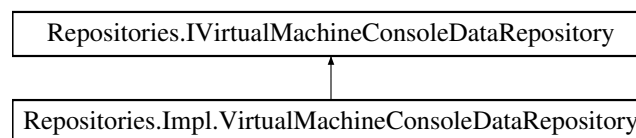
The documentation for this interface was generated from the following file:

- NativeLibrary/Wrappers/IVirtualizationWrapper.cs

5.64 Repositories.IVirtualMachineConsoleDataRepository Interface Reference

Repository managing virtual machine console data in memory.

Inheritance diagram for Repositories.IVirtualMachineConsoleDataRepository:

**Public Member Functions**

- void [AddDataToConsole](#) (Guid uuid, byte[] data)
Adds data to a virtual machine console.
- List< byte > [GetData](#) (Guid uuid)
Retrieves data from a virtual machine console.

5.64.1 Detailed Description

Repository managing virtual machine console data in memory.

5.64.2 Member Function Documentation

5.64.2.1 AddDataToConsole()

```
void Repositories.IVirtualMachineConsoleDataRepository.AddDataToConsole (
    Guid uuid,
    byte[] data)
```

Adds data to a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
<i>data</i>	Data to add.

Implemented in [Repositories.Impl.VirtualMachineConsoleDataRepository](#).

5.64.2.2 GetData()

```
List< byte > Repositories.IVirtualMachineConsoleDataRepository.GetData (
    Guid uuid)
```

Retrieves data from a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

List of bytes from console.

Implemented in [Repositories.Impl.VirtualMachineConsoleDataRepository](#).

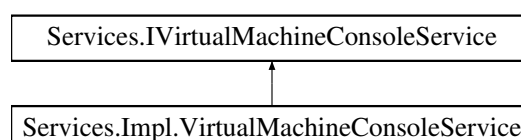
The documentation for this interface was generated from the following file:

- Repositories/IVirtualMachineConsoleDataRepository.cs

5.65 Services.IVirtualMachineConsoleService Interface Reference

Service managing virtual machine console.

Inheritance diagram for Services.IVirtualMachineConsoleService:



Public Member Functions

- `byte[] GetInitialConsoleData` (Guid uuid)
Retrieves initial console data from a virtual machine.
- `StreamInfo OpenVirtualMachineConsole` (Guid uuid)
Opens a virtual machine console stream.
- `void SendBytesToVirtualMachineConsoleByUuid` (Guid uuid, string bytes)
Sends bytes to a virtual machine console.
- `StreamData GetBytesFromConsole` (StreamInfo stream)
Retrieves data from a console stream.
- `List< StreamInfo > GetStreams` ()
Retrieves a list of all opened streams.
- `void CloseStream` (StreamInfo stream)
Closes a console stream.

5.65.1 Detailed Description

Service managing virtual machine console.

5.65.2 Member Function Documentation**5.65.2.1 CloseStream()**

```
void Services.IVirtualMachineConsoleService.CloseStream (
    StreamInfo stream)
```

Closes a console stream.

Parameters

<i>stream</i>	Stream information to close.
---------------	------------------------------

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

5.65.2.2 GetBytesFromConsole()

```
StreamData Services.IVirtualMachineConsoleService.GetBytesFromConsole (
    StreamInfo stream)
```

Retrieves data from a console stream.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Returns

Stream data.

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

5.65.2.3 GetInitialConsoleData()

```
byte[] Services.IVirtualMachineConsoleService.GetInitialConsoleData (
    Guid uuid)
```

Retrieves initial console data from a virtual machine.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Initial console data.

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

5.65.2.4 GetStreams()

```
List< StreamInfo > Services.IVirtualMachineConsoleService.GetStreams ()
```

Retrieves a list of all opened streams.

Returns

List of opened streams.

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

5.65.2.5 OpenVirtualMachineConsole()

```
StreamInfo Services.IVirtualMachineConsoleService.OpenVirtualMachineConsole (  
    Guid uuid)
```

Opens a virtual machine console stream.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Opened stream information.

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

5.65.2.6 SendBytesToVirtualMachineConsoleByUuid()

```
void Services.IVirtualMachineConsoleService.SendBytesToVirtualMachineConsoleByUuid (  
    Guid uuid,  
    string bytes)
```

Sends bytes to a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
<i>bytes</i>	Bytes to send in text format.

Implemented in [Services.Impl.VirtualMachineConsoleService](#).

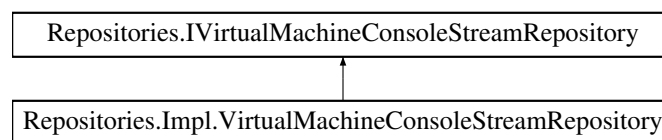
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineConsoleService.cs

5.66 Repositories.IVirtualMachineConsoleStreamRepository Interface Reference

Repository managing virtual machine console streams in memory.

Inheritance diagram for Repositories.IVirtualMachineConsoleStreamRepository:



Public Member Functions

- void [Add](#) ([StreamInfo](#) stream)
Adds a console stream to the repository.
- void [Remove](#) ([StreamInfo](#) stream)
Removes a console stream from the repository.
- [StreamInfo?](#) [GetByUuid](#) (Guid uuid)
Retrieves a stream by virtual machine UUID.
- List< [StreamInfo](#) > [GetAllStreams](#) ()
Retrieves all opened streams.

5.66.1 Detailed Description

Repository managing virtual machine console streams in memory.

5.66.2 Member Function Documentation

5.66.2.1 Add()

```
void Repositories.IVirtualMachineConsoleStreamRepository.Add (
    StreamInfo stream)
```

Adds a console stream to the repository.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Implemented in [Repositories.Impl.VirtualMachineConsoleStreamRepository](#).

5.66.2.2 GetAllStreams()

```
List< StreamInfo > Repositories.IVirtualMachineConsoleStreamRepository.GetAllStreams ()
```

Retrieves all opened streams.

Returns

List of streams.

Implemented in [Repositories.Impl.VirtualMachineConsoleStreamRepository](#).

5.66.2.3 GetByUuid()

```
StreamInfo? Repositories.IVirtualMachineConsoleStreamRepository.GetByUuid (  
    Guid uuid)
```

Retrieves a stream by virtual machine UUID.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Stream information or null.

Implemented in [Repositories.Impl.VirtualMachineConsoleStreamRepository](#).

5.66.2.4 Remove()

```
void Repositories.IVirtualMachineConsoleStreamRepository.Remove (  
    StreamInfo stream)
```

Removes a console stream from the repository.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Implemented in [Repositories.Impl.VirtualMachineConsoleStreamRepository](#).

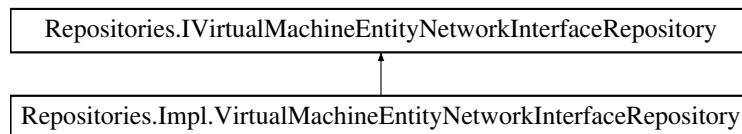
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualMachineConsoleStreamRepository.cs](#)

5.67 Repositories.IVirtualMachineEntityNetworkInterfaceRepository Interface Reference

Repository managing virtual machine network interfaces in the database.

Inheritance diagram for Repositories.IVirtualMachineEntityNetworkInterfaceRepository:



Public Member Functions

- Task [Create](#) ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Creates a new network interface.
- Task [Update](#) ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Updates a network interface.
- Task [Remove](#) ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Removes a network interface.
- Task< [VirtualMachineEntityNetworkInterfaceModel](#)?> [GetByIds](#) (int virtualMachineId, int connectionId)
Retrieves a network interface by virtual machine and connection identifiers.
- Task< List< [VirtualMachineEntityNetworkInterfaceModel](#) > > [GetByVirtualMachineId](#) (int id)
Retrieves all network interfaces for a virtual machine.
- Task< List< [VirtualMachineEntityNetworkInterfaceModel](#) > > [GetByNetworkName](#) (int virtualMachineId, string networkName)
Retrieves network interfaces connected to a specific network.

5.67.1 Detailed Description

Repository managing virtual machine network interfaces in the database.

5.67.2 Member Function Documentation

5.67.2.1 Create()

```
Task Repositories.IVirtualMachineEntityNetworkInterfaceRepository.Create (
    VirtualMachineEntityNetworkInterfaceModel networkInterface)
```

Creates a new network interface.

Parameters

<i>networkInterface</i>	Interface model to create.
-------------------------	----------------------------

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

5.67.2.2 GetByIds()

```
Task< VirtualMachineEntityNetworkInterfaceModel?> Repositories.IVirtualMachineEntityNetwork↵  
InterfaceRepository.GetByIds (   
    int virtualMachineId,   
    int connectionId)
```

Retrieves a network interface by virtual machine and connection identifiers.

Parameters

<i>virtualMachineId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.

Returns

Network interface or null.

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

5.67.2.3 GetByNetworkName()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.IVirtualMachineEntityNetworkInterfaceRepository.GetByNetworkName (
    int virtualMachineId,
    string networkName)
```

Retrieves network interfaces connected to a specific network.

Parameters

<i>virtualMachineId</i>	Virtual machine identifier.
<i>networkName</i>	Network name.

Returns

List of network interfaces.

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

5.67.2.4 GetByVirtualMachineId()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.IVirtualMachineEntityNetworkInterfaceRepository.GetByVirtualMachineId (
    int id)
```

Retrieves all network interfaces for a virtual machine.

Parameters

<i>id</i>	Virtual machine identifier.
-----------	-----------------------------

Returns

List of network interfaces.

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

5.67.2.5 Remove()

```
Task Repositories.IVirtualMachineEntityNetworkInterfaceRepository.Remove (
    VirtualMachineEntityNetworkInterfaceModel networkInterface)
```

Removes a network interface.

Parameters

<code>networkInterface</code>	Interface model to remove.
-------------------------------	----------------------------

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

5.67.2.6 Update()

```
Task Repositories.IVirtualMachineEntityNetworkInterfaceRepository.Update (
    VirtualMachineEntityNetworkInterfaceModel networkInterface)
```

Updates a network interface.

Parameters

<code>networkInterface</code>	Interface model to update.
-------------------------------	----------------------------

Implemented in [Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository](#).

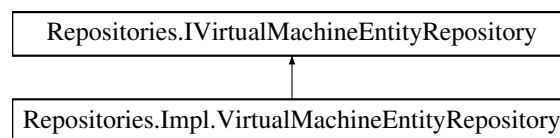
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualMachineEntityNetworkInterfaceRepository.cs](#)

5.68 Repositories.IVirtualMachineEntityRepository Interface Reference

Repository managing virtual machine entities in the database.

Inheritance diagram for Repositories.IVirtualMachineEntityRepository:



Public Member Functions

- Task [Add](#) ([VirtualMachineEntityModel](#) model)
Adds a new virtual machine entity.
- Task< List< [VirtualMachineEntityModel](#) > > [GetAll](#) ()
Retrieves all virtual machine entities.
- Task< [VirtualMachineEntityModel](#) > [GetById](#) (int id)
Retrieves a virtual machine entity by identifier.
- Task [Update](#) ([VirtualMachineEntityModel](#) model)
Updates a virtual machine entity.

5.68.1 Detailed Description

Repository managing virtual machine entities in the database.

5.68.2 Member Function Documentation

5.68.2.1 Add()

```
Task Repositories.IVirtualMachineEntityRepository.Add (  
    VirtualMachineEntityModel model)
```

Adds a new virtual machine entity.

Parameters

<i>model</i>	Entity model to add.
--------------	----------------------

Implemented in [Repositories.Impl.VirtualMachineEntityRepository](#).

5.68.2.2 GetAll()

```
Task< List< VirtualMachineEntityModel > > Repositories.IVirtualMachineEntityRepository.GetAll  
( )
```

Retrieves all virtual machine entities.

Returns

List of entities.

Implemented in [Repositories.Impl.VirtualMachineEntityRepository](#).

5.68.2.3 GetById()

```
Task< VirtualMachineEntityModel > Repositories.IVirtualMachineEntityRepository.GetById (  
    int id)
```

Retrieves a virtual machine entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Virtual machine entity.

Implemented in [Repositories.Impl.VirtualMachineEntityRepository](#).

5.68.2.4 Update()

```
Task Repositories.IVirtualMachineEntityRepository.Update (  
    VirtualMachineEntityModel model)
```

Updates a virtual machine entity.

Parameters

<i>model</i>	Entity model to update.
--------------	-------------------------

Implemented in [Repositories.Impl.VirtualMachineEntityRepository](#).

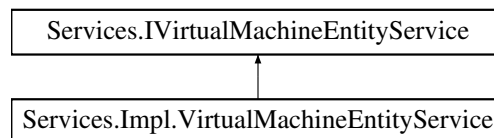
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualMachineEntityRepository.cs](#)

5.69 Services.IVirtualMachineEntityService Interface Reference

Service managing virtual machine entities.

Inheritance diagram for Services.IVirtualMachineEntityService:



Public Member Functions

- Task< List< [VirtualMachineEntityDTO](#) > > [GetEntities](#) ()
Retrieves a list of all virtual machine entities.
- Task< List< [VirtualMachineEntityMacAddressDTO](#) > > [GetMacAddresses](#) ()
Retrieves a list of virtual machine MAC addresses.
- Task< [VirtualMachineEntityDTO](#) > [CreateEntity](#) (string name, int bootableDiskId, uint memory, uint virtualCpus, VirtualMachineEntityType type, int x, int y)
Creates a new virtual machine entity.
- Task< [VirtualMachineEntityDTO](#) > [UpdateEntityPosition](#) (int id, int x, int y)
Updates the position of a virtual machine entity on the board.
- Task< [VirtualMachineEntityDTO](#) > [GetById](#) (int id)
Retrieves a virtual machine entity by identifier.
- Task< [VirtualMachineEntityDTO](#) > [UpdateEntityVmUUID](#) (int id, string uuid)
Updates the virtual machine UUID for an entity.

5.69.1 Detailed Description

Service managing virtual machine entities.

5.69.2 Member Function Documentation

5.69.2.1 CreateEntity()

```
Task< VirtualMachineEntityDTO > Services.IVirtualMachineEntityService.CreateEntity (
    string name,
    int bootableDiskId,
    uint memory,
    uint virtualCpus,
    VirtualMachineEntityType type,
    int x,
    int y)
```

Creates a new virtual machine entity.

Parameters

<i>name</i>	Virtual machine name.
<i>bootable</i> ↔ <i>DiskId</i>	Bootable disk identifier.
<i>memory</i>	Amount of RAM in MB.
<i>virtualCpus</i>	Number of virtual CPUs.
<i>type</i>	Virtual machine entity type.
<i>x</i>	X coordinate position.
<i>y</i>	Y coordinate position.

Returns

Created virtual machine entity.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

5.69.2.2 GetById()

```
Task< VirtualMachineEntityDTO > Services.IVirtualMachineEntityService.GetById (
    int id)
```

Retrieves a virtual machine entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Virtual machine entity.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

5.69.2.3 GetEntities()

```
Task< List< VirtualMachineEntityDTO > > Services.IVirtualMachineEntityService.GetEntities ()
```

Retrieves a list of all virtual machine entities.

Returns

List of virtual machine entities.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

5.69.2.4 GetMacAddresses()

```
Task< List< VirtualMachineEntityMacAddressDTO > > Services.IVirtualMachineEntityService.GetMacAddresses ()
```

Retrieves a list of virtual machine MAC addresses.

Returns

List of MAC addresses.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

5.69.2.5 UpdateEntityPosition()

```
Task< VirtualMachineEntityDTO > Services.IVirtualMachineEntityService.UpdateEntityPosition (
    int id,
    int x,
    int y)
```

Updates the position of a virtual machine entity on the board.

Parameters

<i>id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated entity.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

5.69.2.6 UpdateEntityVmUUID()

```
Task< VirtualMachineEntityDTO > Services.IVirtualMachineEntityService.UpdateEntityVmUUID (
    int id,
    string uuid)
```

Updates the virtual machine UUID for an entity.

Parameters

<i>id</i>	Entity identifier.
<i>uuid</i>	New virtual machine UUID.

Returns

Updated entity.

Implemented in [Services.Impl.VirtualMachineEntityService](#).

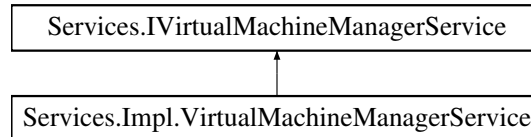
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineEntityService.cs

5.70 Services.IVirtualMachineManagerService Interface Reference

Service managing virtual machines in the hypervisor.

Inheritance diagram for Services.IVirtualMachineManagerService:



Public Member Functions

- [VirtualMachineInfo CreateVirtualMachine](#) ([VirtualMachineCreateDefinition](#) definition)
Creates a new virtual machine in the hypervisor.
- List< [VirtualMachineInfo](#) > [GetListOfVirtualMachines](#) ()
Retrieves a list of all virtual machines from the hypervisor.
- void [AttachVirtualNetworkInterfaceToVirtualMachine](#) (string name, string networkName, string macAddress)
Attaches a network interface to a virtual machine.

5.70.1 Detailed Description

Service managing virtual machines in the hypervisor.

5.70.2 Member Function Documentation

5.70.2.1 AttachVirtualNetworkInterfaceToVirtualMachine()

```
void Services.IVirtualMachineManagerService.AttachVirtualNetworkInterfaceToVirtualMachine (
    string name,
    string networkName,
    string macAddress)
```

Attaches a network interface to a virtual machine.

Parameters

<i>name</i>	Virtual machine name.
<i>networkName</i>	Virtual network name.
<i>macAddress</i>	Interface MAC address.

Implemented in [Services.Impl.VirtualMachineManagerService](#).

5.70.2.2 CreateVirtualMachine()

```
VirtualMachineInfo Services.IVirtualMachineManagerService.CreateVirtualMachine (
    VirtualMachineCreateDefinition definition)
```

Creates a new virtual machine in the hypervisor.

Parameters

<i>definition</i>	Virtual machine definition to create.
-------------------	---------------------------------------

Returns

Information about the created virtual machine.

Implemented in [Services.Impl.VirtualMachineManagerService](#).

5.70.2.3 GetListOfVirtualMachines()

```
List< VirtualMachineInfo > Services.IVirtualMachineManagerService.GetListOfVirtualMachines ()
```

Retrieves a list of all virtual machines from the hypervisor.

Returns

List of virtual machines.

Implemented in [Services.Impl.VirtualMachineManagerService](#).

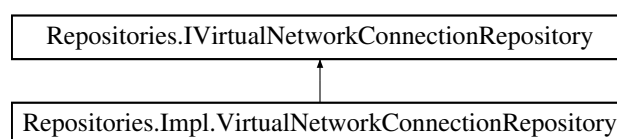
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineManagerService.cs

5.71 Repositories.IVirtualNetworkConnectionRepository Interface Reference

Repository managing entity connections in the virtual network.

Inheritance diagram for Repositories.IVirtualNetworkConnectionRepository:



Public Member Functions

- Task< List< [VirtualNetworkEntityConnectionModel](#) > > [GetUsingEntityId](#) (int id, [EntityType](#) type)
Retrieves connections associated with a specific entity.
- Task< [VirtualNetworkEntityConnectionModel](#) > [GetById](#) (int id)
Retrieves a connection by identifier.
- Task< [VirtualNetworkEntityConnectionModel](#) > [Create](#) (int firstEntityId, [EntityType](#) firstEntityType, int secondEntityId, [EntityType](#) secondEntityType)
Creates a new connection between two entities.
- Task< List< [VirtualNetworkEntityConnectionModel](#) > > [GetAll](#) ()
Retrieves all connections.
- Task [Remove](#) (int id)
Removes a connection.

5.71.1 Detailed Description

Repository managing entity connections in the virtual network.

5.71.2 Member Function Documentation

5.71.2.1 Create()

```
Task< VirtualNetworkEntityConnectionModel > Repositories.IVirtualNetworkConnectionRepository.Create (
    int firstEntityId,
    EntityType firstEntityType,
    int secondEntityId,
    EntityType secondEntityType)
```

Creates a new connection between two entities.

Parameters

<i>firstEntityId</i>	First entity identifier.
<i>firstEntityType</i>	First entity type.
<i>secondEntityId</i>	Second entity identifier.
<i>secondEntityType</i>	Second entity type.

Returns

Created connection.

Implemented in [Repositories.Impl.VirtualNetworkConnectionRepository](#).

5.71.2.2 GetAll()

```
Task< List< VirtualNetworkEntityConnectionModel > > Repositories.IVirtualNetworkConnectionRepository.GetAll ()
```

Retrieves all connections.

Returns

List of connections.

Implemented in [Repositories.Impl.VirtualNetworkConnectionRepository](#).

5.71.2.3 GetById()

```
Task< VirtualNetworkEntityConnectionModel > Repositories.IVirtualNetworkConnectionRepository.GetById (
    int id)
```

Retrieves a connection by identifier.

Parameters

<i>id</i>	Connection identifier.
-----------	------------------------

Returns

Connection.

Implemented in [Repositories.Impl.VirtualNetworkConnectionRepository](#).

5.71.2.4 GetUsingEntityId()

```
Task< List< VirtualNetworkEntityConnectionModel > > Repositories.IVirtualNetworkConnectionRepository.GetUsingEntityId (
    int id,
    EntityType type)
```

Retrieves connections associated with a specific entity.

Parameters

<i>id</i>	Entity identifier.
<i>type</i>	Entity type.

Returns

List of connections.

Implemented in [Repositories.Impl.VirtualNetworkConnectionRepository](#).

5.71.2.5 Remove()

```
Task Repositories.IVirtualNetworkConnectionRepository.Remove (
    int id)
```

Removes a connection.

Parameters

<i>id</i>	Connection identifier.
-----------	------------------------

Implemented in [Repositories.Impl.VirtualNetworkConnectionRepository](#).

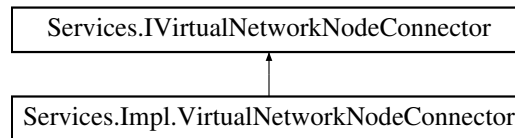
The documentation for this interface was generated from the following file:

- Repositories/IVirtualNetworkConnectionRepository.cs

5.72 Services.IVirtualNetworkNodeConnector Interface Reference

Service responsible for connecting virtual network nodes.

Inheritance diagram for Services.IVirtualNetworkNodeConnector:



Public Member Functions

- Task< [VirtualNetworkConnectionDTO](#) > [ConnectTwoVirtualNetworkNodes](#) (int sourceVirtualNetworkNode, int destinationVirtualNetworkNode)
Connects two virtual network nodes.
- Task [DisconnectTwoVirtualNetworkNodes](#) (int connectionId, int firstVirtualNetworkNodeId, int secondVirtualNetworkNode)
Disconnects two virtual network nodes.
- Task< [VirtualNetworkConnectionDTO](#) > [ConnectVirtualNetworkNodeToInternet](#) (int virtualNetworkNodeId, int internetId)
Connects a virtual network node to the Internet.
- Task [DisconnectVirtualNetworkNodeFromInternet](#) (int connectionId, int virtualNetworkNodeId)
Disconnects a virtual network node from the Internet.

5.72.1 Detailed Description

Service responsible for connecting virtual network nodes.

5.72.2 Member Function Documentation

5.72.2.1 ConnectTwoVirtualNetworkNodes()

```
Task< VirtualNetworkConnectionDTO > Services.IVirtualNetworkNodeConnector.ConnectTwoVirtualNetworkNodes (
    int sourceVirtualNetworkNode,
    int destinationVirtualNetworkNode)
```

Connects two virtual network nodes.

Parameters

<i>sourceVirtualNetworkNode</i>	First node identifier.
<i>destinationVirtualNetworkNode</i>	Second node identifier.

Returns

Created connection data.

Implemented in [Services.Impl.VirtualNetworkNodeConnector](#).

5.72.2.2 ConnectVirtualNetworkNodeToInternet()

```
Task< VirtualNetworkConnectionDTO > Services.IVirtualNetworkNodeConnector.ConnectVirtual↵  
NetworkNodeToInternet (   
    int virtualNetworkNodeId,   
    int internetId)
```

Connects a virtual network node to the Internet.

Parameters

<i>virtualNetwork↵ NodeId</i>	Network node identifier.
<i>internetId</i>	Internet entity identifier.

Returns

Created connection data.

Implemented in [Services.Impl.VirtualNetworkNodeConnector](#).

5.72.2.3 DisconnectTwoVirtualNetworkNodes()

```
Task Services.IVirtualNetworkNodeConnector.DisconnectTwoVirtualNetworkNodes (   
    int connectionId,   
    int firstVirtualNetworkNodeId,   
    int secondVirtualNetworkNode)
```

Disconnects two virtual network nodes.

Parameters

<i>connectionId</i>	Connection identifier.
<i>firstVirtualNetworkNodeId</i>	First node identifier.
<i>secondVirtualNetworkNode</i>	Second node identifier.

Implemented in [Services.Impl.VirtualNetworkNodeConnector](#).

5.72.2.4 DisconnectVirtualNetworkNodeFromInternet()

```
Task Services.IVirtualNetworkNodeConnector.DisconnectVirtualNetworkNodeFromInternet (   
    int connectionId,   
    int virtualNetworkNodeId)
```

Disconnects a virtual network node from the Internet.

Parameters

<i>connectionId</i>	Connection identifier.
<i>virtualNetwork↵ NodeId</i>	Network node identifier.

Implemented in [Services.Impl.VirtualNetworkNodeConnector](#).

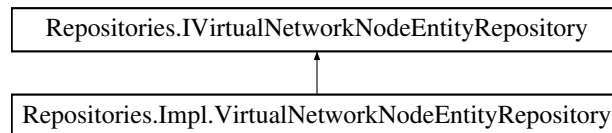
The documentation for this interface was generated from the following file:

- Services/IVirtualNetworkNodeConnector.cs

5.73 Repositories.IVirtualNetworkNodeEntityRepository Interface Reference

Repository managing virtual network nodes in the database.

Inheritance diagram for Repositories.IVirtualNetworkNodeEntityRepository:



Public Member Functions

- Task< [VirtualNetworkNodeEntityModel](#) > [Create](#) (string name, [VirtualNetworkModel](#) virtualNetwork)
Creates a new visible virtual network node.
- Task< [VirtualNetworkNodeEntityModel](#) > [CreateInvisible](#) ([VirtualNetworkModel](#) virtualNetwork)
Creates a new invisible virtual network node.
- Task< List< [VirtualNetworkNodeEntityModel](#) > > [GetAll](#) ()
Retrieves all virtual network nodes.
- Task< List< [VirtualNetworkNodeEntityModel](#) > > [GetVisible](#) ()
Retrieves visible virtual network nodes.
- Task< [VirtualNetworkNodeEntityModel](#) > [GetById](#) (int id)
Retrieves a virtual network node by identifier.
- Task< [VirtualNetworkNodeEntityModel](#) > [UpdateEntityPosition](#) (int id, int x, int y)
Updates the position of a virtual network node.
- Task< [VirtualNetworkNodeEntityModel](#) > [UpdateNetwork](#) (int id, [VirtualNetworkModel](#) virtualNetwork)
Updates the virtual network for a node.
- Task [Remove](#) (int id)
Removes a virtual network node.

5.73.1 Detailed Description

Repository managing virtual network nodes in the database.

5.73.2 Member Function Documentation

5.73.2.1 Create()

```
Task< VirtualNetworkNodeEntityModel > Repositories.IVirtualNetworkNodeEntityRepository.Create(
(
    string name,
    VirtualNetworkModel virtualNetwork)
```

Creates a new visible virtual network node.

Parameters

<i>name</i>	Node name.
<i>virtualNetwork</i>	Virtual network.

Returns

Created node.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.2 CreateInvisible()

```
Task< VirtualNetworkNodeEntityModel > Repositories.IVirtualNetworkNodeEntityRepository.CreateInvisible (
    VirtualNetworkModel virtualNetwork)
```

Creates a new invisible virtual network node.

Parameters

<i>virtualNetwork</i>	Virtual network.
-----------------------	------------------

Returns

Created node.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.3 GetAll()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.IVirtualNetworkNodeEntityRepository.GetAll ()
```

Retrieves all virtual network nodes.

Returns

List of nodes.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.4 GetById()

```
Task< VirtualNetworkNodeEntityModel > Repositories.IVirtualNetworkNodeEntityRepository.GetById (
    int id)
```

Retrieves a virtual network node by identifier.

Parameters

<i>id</i>	Node identifier.
-----------	------------------

Returns

Network node.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.5 GetVisible()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.IVirtualNetworkNodeEntityRepository.↵  
GetVisible ()
```

Retrieves visible virtual network nodes.

Returns

List of visible nodes.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.6 Remove()

```
Task Repositories.IVirtualNetworkNodeEntityRepository.Remove (  
    int id)
```

Removes a virtual network node.

Parameters

<i>id</i>	Node identifier.
-----------	------------------

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.7 UpdateEntityPosition()

```
Task< VirtualNetworkNodeEntityModel > Repositories.IVirtualNetworkNodeEntityRepository.↵  
UpdateEntityPosition (  
    int id,  
    int x,  
    int y)
```

Updates the position of a virtual network node.

Parameters

<i>id</i>	Node identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated node.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

5.73.2.8 UpdateNetwork()

```
Task< VirtualNetworkNodeEntityModel > Repositories.IVirtualNetworkNodeEntityRepository.UpdateNetwork (
    int id,
    VirtualNetworkModel virtualNetwork)
```

Updates the virtual network for a node.

Parameters

<i>id</i>	Node identifier.
<i>virtualNetwork</i>	New virtual network.

Returns

Updated node.

Implemented in [Repositories.Impl.VirtualNetworkNodeEntityRepository](#).

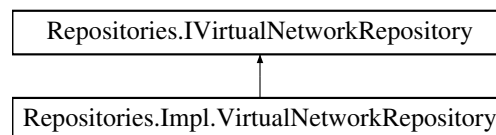
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualNetworkNodeEntityRepository.cs](#)

5.74 Repositories.IVirtualNetworkRepository Interface Reference

Repository managing virtual networks in the database.

Inheritance diagram for Repositories.IVirtualNetworkRepository:



Public Member Functions

- Task< [VirtualNetworkModel](#) > [Create](#) (string bridgeName, Guid uuid, string? ipAddress=null)
Creates a new virtual network.
- Task< [VirtualNetworkModel](#) > [GetById](#) (int id)
Retrieves a virtual network by identifier.
- Task< [VirtualNetworkModel](#) > [GetByUuid](#) (Guid uuid)
Retrieves a virtual network by UUID.
- Task< [VirtualNetworkModel](#) > [GetByUuidWithVirtualNetworkNodes](#) (Guid uuid)
Retrieves a virtual network with nodes by UUID.
- Task [Remove](#) (int id)
Removes a virtual network.
- Task< List< [VirtualNetworkModel](#) > > [GetAll](#) ()
Retrieves all virtual networks.
- Task< [VirtualNetworkModel](#) > [GetByBridgeName](#) (string bridgeName)
Retrieves a virtual network by bridge name.

5.74.1 Detailed Description

Repository managing virtual networks in the database.

5.74.2 Member Function Documentation

5.74.2.1 Create()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.Create (  
    string bridgeName,  
    Guid uuid,  
    string? ipAddress = null)
```

Creates a new virtual network.

Parameters

<i>bridgeName</i>	Network bridge name.
<i>uuid</i>	Network UUID.
<i>ipAddress</i>	Network IP address (optional).

Returns

Created virtual network.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.2 GetAll()

```
Task< List< VirtualNetworkModel > > Repositories.IVirtualNetworkRepository.GetAll ()
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.3 GetByBridgeName()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.GetByBridgeName (  
    string bridgeName)
```

Retrieves a virtual network by bridge name.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

Returns

Virtual network.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.4 GetById()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.GetById (
    int id)
```

Retrieves a virtual network by identifier.

Parameters

<i>id</i>	Network identifier.
-----------	---------------------

Returns

Virtual network.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.5 GetByUuid()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.GetByUuid (
    Guid uuid)
```

Retrieves a virtual network by UUID.

Parameters

<i>uuid</i>	Network UUID.
-------------	---------------

Returns

Virtual network.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.6 GetByUuidWithVirtualNetworkNodes()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.GetByUuidWithVirtualNetworkNodes (
    Guid uuid)
```

Retrieves a virtual network with nodes by UUID.

Parameters

<i>uuid</i>	Network UUID.
-------------	---------------

Returns

Virtual network with nodes.

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

5.74.2.7 Remove()

```
Task Repositories.IVirtualNetworkRepository.Remove (
    int id)
```

Removes a virtual network.

Parameters

<i>id</i>	Network identifier.
-----------	---------------------

Implemented in [Repositories.Impl.VirtualNetworkRepository](#).

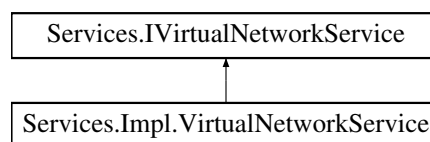
The documentation for this interface was generated from the following file:

- Repositories/IVirtualNetworkRepository.cs

5.75 Services.IVirtualNetworkService Interface Reference

Service managing virtual networks.

Inheritance diagram for Services.IVirtualNetworkService:



Public Member Functions

- Task< List< [VirtualNetworkConnectionDTO](#) > > [GetVirtualNetworkConnections](#) ()
Retrieves a list of all connections in the virtual network.
- Task< [VirtualNetworkNodeEntityDTO](#) > [CreateVirtualNetworkNode](#) (string? name)
Creates a new virtual network node.
- Task< List< [VirtualNetworkNodeEntityDTO](#) > > [GetVisibleVirtualNetworkNodeEntities](#) ()
Retrieves a list of visible virtual network nodes.
- Task< [VirtualNetworkNodeEntityDTO](#) > [UpdateVirtualNetworkNodeEntityPosition](#) (int entityId, int x, int y)
Updates the position of a virtual network node.
- Task [AttachNetworkInterfaceToVirtualMachine](#) (int vmId, int connectionId, [VirtualNetworkInterfaceCreateDefinition](#) interfaceDefinition)
Attaches a network interface to a virtual machine.
- Task [UpdateNetworkForVirtualMachineNetworkInterface](#) (int vmId, int connectionId, string networkName)
Updates the network for a virtual machine network interface.
- Task< [VirtualNetworkModel](#) > [CreateInternetVirtualNetwork](#) ()
Creates a virtual network for Internet connection.
- Task< [VirtualNetworkModel](#) > [CreateNodeVirtualNetwork](#) ()
Creates a virtual network for a network node.
- Task< List< [VirtualNetworkModel](#) > > [GetAllVirtualNetworks](#) ()
Retrieves all virtual networks.
- Task< [VirtualNetworkModel](#) > [GetVirtualNetworkUsingBridgeName](#) (string bridgeName)
Retrieves a virtual network based on bridge name.

5.75.1 Detailed Description

Service managing virtual networks.

5.75.2 Member Function Documentation

5.75.2.1 [AttachNetworkInterfaceToVirtualMachine\(\)](#)

```
Task Services.IVirtualNetworkService.AttachNetworkInterfaceToVirtualMachine (
    int vmId,
    int connectionId,
    VirtualNetworkInterfaceCreateDefinition interfaceDefinition)
```

Attaches a network interface to a virtual machine.

Parameters

<i>vmId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.
<i>interfaceDefinition</i>	Network interface definition.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.2 CreateInternetVirtualNetwork()

```
Task< VirtualNetworkModel > Services.IVirtualNetworkService.CreateInternetVirtualNetwork ()
```

Creates a virtual network for Internet connection.

Returns

Created virtual network.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.3 CreateNodeVirtualNetwork()

```
Task< VirtualNetworkModel > Services.IVirtualNetworkService.CreateNodeVirtualNetwork ()
```

Creates a virtual network for a network node.

Returns

Created virtual network.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.4 CreateVirtualNetworkNode()

```
Task< VirtualNetworkNodeEntityDTO > Services.IVirtualNetworkService.CreateVirtualNetworkNode (
    string? name)
```

Creates a new virtual network node.

Parameters

<i>name</i>	Node name.
-------------	------------

Returns

Created network node.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.5 GetAllVirtualNetworks()

```
Task< List< VirtualNetworkModel > > Services.IVirtualNetworkService.GetAllVirtualNetworks ()
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.6 GetVirtualNetworkConnections()

```
Task< List< VirtualNetworkConnectionDTO > > Services.IVirtualNetworkService.GetVirtualNetworkConnections ()
```

Retrieves a list of all connections in the virtual network.

Returns

List of connections.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.7 GetVirtualNetworkUsingBridgeName()

```
Task< VirtualNetworkModel > Services.IVirtualNetworkService.GetVirtualNetworkUsingBridgeName (
    string bridgeName)
```

Retrieves a virtual network based on bridge name.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

Returns

Virtual network.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.8 GetVisibleVirtualNetworkNodeEntities()

```
Task< List< VirtualNetworkNodeEntityDTO > > Services.IVirtualNetworkService.GetVisibleVirtualNetworkNodeEntities ()
```

Retrieves a list of visible virtual network nodes.

Returns

List of visible nodes.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.9 UpdateNetworkForVirtualMachineNetworkInterface()

```
Task Services.IVirtualNetworkService.UpdateNetworkForVirtualMachineNetworkInterface (
    int vmId,
    int connectionId,
    string networkName)
```

Updates the network for a virtual machine network interface.

Parameters

<i>vmId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.
<i>networkName</i>	New network name.

Implemented in [Services.Impl.VirtualNetworkService](#).

5.75.2.10 UpdateVirtualNetworkNodeEntityPosition()

```
Task< VirtualNetworkNodeEntityDTO > Services.IVirtualNetworkService.UpdateVirtualNetworkNode↵
EntityPosition (
    int entityId,
    int x,
    int y)
```

Updates the position of a virtual network node.

Parameters

<i>entity↵ Id</i>	Node identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated node.

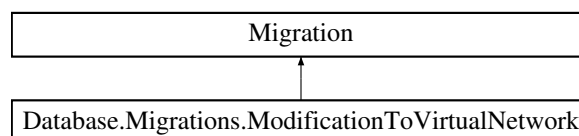
Implemented in [Services.Impl.VirtualNetworkService](#).

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

- Services/IVirtualNetworkService.cs

5.76 Database.Migrations.ModificationToVirtualNetwork Class Reference

Inheritance diagram for Database.Migrations.ModificationToVirtualNetwork:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.76.1 Detailed Description

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

- Database/Migrations/20250906194200_ModificationToVirtualNetwork.cs
- Database/Migrations/20250906194200_ModificationToVirtualNetwork.Designer.cs

5.77 NativeLibrary.Structs.NativeConnectionInfo Struct Reference

Public Attributes

- uint **CpuCount**
- uint **CpuFreq**
- long **TotalMemory**
- string **ConnectionUrl**
- string **DriverType**
- NativeVersion **LibVersion**
- NativeVersion **DriverVersion**

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

- NativeLibrary/Structs/NativeConnectionInfo.cs

5.78 NativeLibrary.Structs.NativeExecutionInfo Struct Reference

Public Attributes

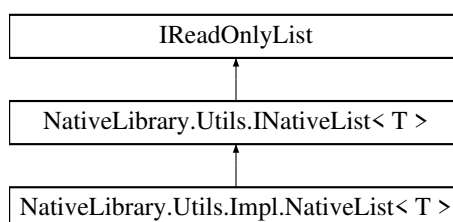
- bool **ErrorOccurred**
- string **Msg**

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

- NativeLibrary/Structs/NativeExecutionInfo.cs

5.79 NativeLibrary.Utils.Impl.NativeList< T > Class Template Reference

Inheritance diagram for NativeLibrary.Utils.Impl.NativeList< T >:



Public Member Functions

- **NativeList** (nint data, int size)
- IEnumerator< T > **GetEnumerator** ()

Properties

- T **this[int index]** [get]
- int **Count** [get]

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

- NativeLibrary/Utils/Impl/NativeList.cs

5.80 NativeLibrary.Structs.NativePacket Struct Reference

Public Attributes

- string **InterfaceName**
- IntPtr **Content**
- int **ContentLength**
- uint **TimestampMicroseconds**

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

- NativeLibrary/Structs/NativePacket.cs

5.81 NativeLibrary.Structs.NativeStreamData Struct Reference

Public Attributes

- byte[] **Buffer**
- bool **IsStreamBroken**

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

- NativeLibrary/Structs/NativeStreamData.cs

5.82 NativeLibrary.Structs.NativeStreamInfo Struct Reference

Public Attributes

- string **Uuid**

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

- NativeLibrary/Structs/NativeStreamInfo.cs

5.83 NativeLibrary.Structs.NativeVirtualMachineInfo Struct Reference

Public Attributes

- string **Uuid**
- string **Name**
- ulong **UsedMemory**
- byte **State**

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

- NativeLibrary/Structs/NativeVirtualMachineInfo.cs

5.84 Models.Packet Class Reference

Represents a captured network packet.

Properties

- required int **Id** [get, set]
Packet identifier.
- required DataLinkLayerPacketType **DataLinkLayerPacketType** [get, set]
Data link layer packet type.
- required string **SourceMacAddress** [get, set]
Source MAC address.
- required string **DestinationMacAddress** [get, set]
Destination MAC address.
- required string **Content** [get, set]
Packet content in hexadecimal format.
- required ulong **TimestampMicroseconds** [get, set]
Packet capture timestamp in microseconds.
- int? **IpVersion** [get, set]
IP protocol version (4 or 6), if applicable.
- string? **SourceIpAddress** [get, set]
Source IP address, if applicable.
- string? **DestinationIpAddress** [get, set]
Destination IP address, if applicable.

5.84.1 Detailed Description

Represents a captured network packet.

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

- Models/Packet.cs

5.85 Models.PacketSniffer Class Reference

Represents a packet sniffer for a specific network bridge.

Properties

- required string **BridgeName** [get, set]
Name of the network bridge being monitored.
- required IntPtr **Handler** [get, set]
Pointer to the native packet sniffer handler.

5.85.1 Detailed Description

Represents a packet sniffer for a specific network bridge.

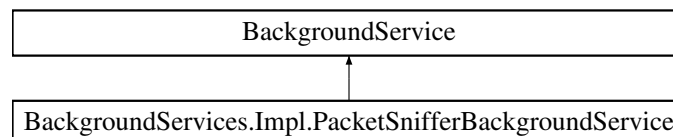
The documentation for this class was generated from the following file:

- Models/PacketSniffer.cs

5.86 BackgroundServices.Impl.PacketSnifferBackgroundService Class Reference

Background service for managing network packet sniffing and broadcasting.

Inheritance diagram for BackgroundServices.Impl.PacketSnifferBackgroundService:



Public Member Functions

- [PacketSnifferBackgroundService](#) (ILogger< [PacketSnifferBackgroundService](#) > logger, IHubContext< [PacketSnifferHub](#) > hubContext, [IPacketSnifferService](#) packetSnifferService, [IVirtualNetworkService](#) virtualNetworkService)
Initializes a new instance of the [PacketSnifferBackgroundService](#).

Protected Member Functions

- override async Task [ExecuteAsync](#) (CancellationToken stoppingToken)
Executes the background service to listen for and broadcast captured packets.

5.86.1 Detailed Description

Background service for managing network packet sniffing and broadcasting.

5.86.2 Constructor & Destructor Documentation

5.86.2.1 PacketSnifferBackgroundService()

```
BackgroundServices.Impl.PacketSnifferBackgroundService.PacketSnifferBackgroundService (
    ILogger< PacketSnifferBackgroundService > logger,
    IHubContext< PacketSnifferHub > hubContext,
    IPacketSnifferService packetSnifferService,
    IVirtualNetworkService virtualNetworkService) [inline]
```

Initializes a new instance of the [PacketSnifferBackgroundService](#).

Parameters

<i>logger</i>	Logger instance.
<i>hubContext</i>	SignalR hub context for packet transmission.
<i>packetSnifferService</i>	Packet sniffer service.
<i>virtualNetworkService</i>	Virtual network service.

5.86.3 Member Function Documentation

5.86.3.1 ExecuteAsync()

```
override async Task BackgroundServices.Impl.PacketSnifferBackgroundService.ExecuteAsync (
    CancellationToken stoppingToken) [inline], [protected]
```

Executes the background service to listen for and broadcast captured packets.

Parameters

<i>stoppingToken</i>	Cancellation token for stopping the service.
----------------------	--

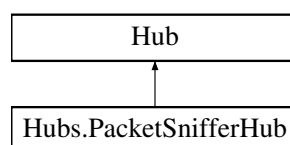
The documentation for this class was generated from the following file:

- BackgroundServices/Impl/PacketSnifferBackgroundService.cs

5.87 Hubs.PacketSnifferHub Class Reference

SignalR hub for transmitting captured network packets.

Inheritance diagram for Hubs.PacketSnifferHub:



Public Member Functions

- async Task **JoinDefaultGroup** ()
Adds client to default packet analyzer group.

Static Public Attributes

- static readonly string **DefaultGroupName** = "default"
Default group name for packet analyzer.

5.87.1 Detailed Description

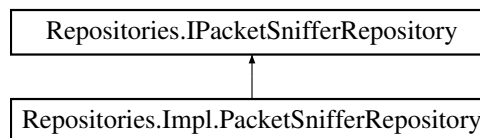
SignalR hub for transmitting captured network packets.

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

- Hubs/PacketSnifferHub.cs

5.88 Repositories.Impl.PacketSnifferRepository Class Reference

Inheritance diagram for Repositories.Impl.PacketSnifferRepository:



Public Member Functions

- void **AddSniffer** ([PacketSniffer](#) sniffer)
Adds a packet analyzer to the repository.
- void **RemoveSniffer** (string bridgeName)
Removes a packet analyzer for a specified bridge.
- List< [PacketSniffer](#) > **GetOpenedSniffers** ()
Retrieves a list of opened packet analyzers.

5.88.1 Member Function Documentation

5.88.1.1 AddSniffer()

```
void Repositories.Impl.PacketSnifferRepository.AddSniffer (
    PacketSniffer sniffer) [inline]
```

Adds a packet analyzer to the repository.

Parameters

<i>sniffer</i>	Packet analyzer.
----------------	------------------

Implements [Repositories.IPacketSnifferRepository](#).

5.88.1.2 GetOpenedSniffers()

```
List< PacketSniffer > Repositories.Impl.PacketSnifferRepository.GetOpenedSniffers () [inline]
```

Retrieves a list of opened packet analyzers.

Returns

List of opened analyzers.

Implements [Repositories.IPacketSnifferRepository](#).

5.88.1.3 RemoveSniffer()

```
void Repositories.Impl.PacketSnifferRepository.RemoveSniffer (  
    string bridgeName) [inline]
```

Removes a packet analyzer for a specified bridge.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

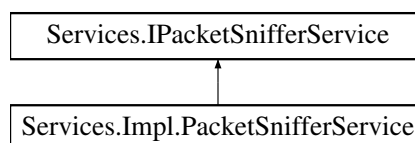
Implements [Repositories.IPacketSnifferRepository](#).

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

- Repositories/Impl/PacketSnifferRepository.cs

5.89 Services.Impl.PacketSnifferService Class Reference

Inheritance diagram for Services.Impl.PacketSnifferService:



Public Member Functions

- **PacketSnifferService** ([IPacketSnifferWrapper](#) snifferWrapper, [IPacketSnifferRepository](#) packetSnifferRepository)
- void **StartListeningForBridge** (string bridgeName)
Starts listening for packets on a specified network bridge.
- List< [PacketSniffer](#) > **GetOpenedSniffers** ()
Retrieves a list of active packet analyzers.
- bool **ListenForPacket** ([PacketSniffer](#) sniffer)
Listens for packets using the specified analyzer.
- [Packet?](#) **GetCapturedPacket** ()
Retrieves the last captured packet.

5.89.1 Member Function Documentation

5.89.1.1 GetCapturedPacket()

```
Packet? Services.Impl.PacketSnifferService.GetCapturedPacket () [inline]
```

Retrieves the last captured packet.

Returns

Captured packet or null if no packet available.

Implements [Services.IPacketSnifferService](#).

5.89.1.2 GetOpenedSniffers()

```
List< PacketSniffer > Services.Impl.PacketSnifferService.GetOpenedSniffers () [inline]
```

Retrieves a list of active packet analyzers.

Returns

List of opened packet sniffers.

Implements [Services.IPacketSnifferService](#).

5.89.1.3 ListenForPacket()

```
bool Services.Impl.PacketSnifferService.ListenForPacket (
    PacketSniffer sniffer) [inline]
```

Listens for packets using the specified analyzer.

Parameters

<i>sniffer</i>	Packet analyzer.
----------------	----------------------------------

Returns

True if packet was captured successfully, false otherwise.

Implements [Services.IPacketSnifferService](#).

5.89.1.4 StartListeningForBridge()

```
void Services::Impl::PacketSnifferService::StartListeningForBridge (  
    string interfaceName) [inline]
```

Starts listening for packets on a specified network bridge.

Parameters

<i>interfaceName</i>	Network interface name.
----------------------	-------------------------

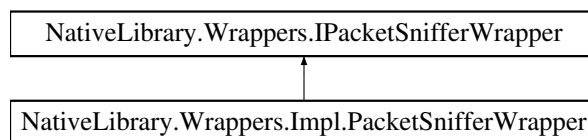
Implements [Services.IPacketSnifferService](#).

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

- Services/Impl/PacketSnifferService.cs

5.90 NativeLibrary.Wrappers.Impl.PacketSnifferWrapper Class Reference

Inheritance diagram for NativeLibrary.Wrappers.Impl.PacketSnifferWrapper:



Public Member Functions

- int [GetNumberOfPackets](#) ()
Gets the number of captured packets.
- [NativePacket](#) [GetPacket](#) ()
Gets the last captured packet.
- IntPtr [OpenSnifferHandler](#) (string bridgeName)
Opens a packet analyzer handler for the specified bridge.
- bool [ListenForPacket](#) (IntPtr handler, string bridgeName)

Public Member Functions inherited from [NativeLibrary.Wrappers.IPacketSnifferWrapper](#)

- bool [ListenForPacket](#) (IntPtr handler, string bridgeName)
Listens for a network packet.

5.90.1 Member Function Documentation

5.90.1.1 GetNumberOfPackets()

```
int NativeLibrary.Wrappers.Impl.PacketSnifferWrapper.GetNumberOfPackets () [inline]
```

Gets the number of captured packets.

Returns

Number of packets.

Implements [NativeLibrary.Wrappers.IPacketSnifferWrapper](#).

5.90.1.2 GetPacket()

```
NativePacket NativeLibrary.Wrappers.Impl.PacketSnifferWrapper.GetPacket () [inline]
```

Gets the last captured packet.

Returns

Captured packet.

Implements [NativeLibrary.Wrappers.IPacketSnifferWrapper](#).

5.90.1.3 OpenSnifferHandler()

```
IntPtr NativeLibrary.Wrappers.Impl.PacketSnifferWrapper.OpenSnifferHandler (
    string bridgeName) [inline]
```

Opens a packet analyzer handler for the specified bridge.

Parameters

<i>bridgeName</i>	Name of the network bridge.
-------------------	-----------------------------

Returns

Pointer to the analyzer handler.

Implements [NativeLibrary.Wrappers.IPacketSnifferWrapper](#).

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

- [NativeLibrary/Wrappers/Impl/PacketSnifferWrapper.cs](#)

5.91 Interconnect.Program Class Reference

Main class of the [Interconnect](#) application.

Static Public Member Functions

- static void [Main](#) (string[] args)
Application entry point.

5.91.1 Detailed Description

Main class of the [Interconnect](#) application.

5.91.2 Member Function Documentation

5.91.2.1 Main()

```
static void Interconnect.Program.Main (
    string[] args) [inline], [static]
```

Application entry point.

Parameters

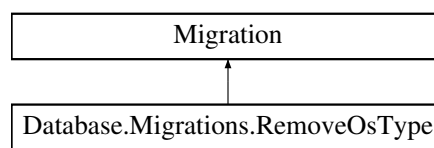
<i>args</i>	Command line arguments.
-------------	-------------------------

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

- Interconnect/Program.cs

5.92 Database.Migrations.RemoveOsType Class Reference

Inheritance diagram for Database.Migrations.RemoveOsType:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

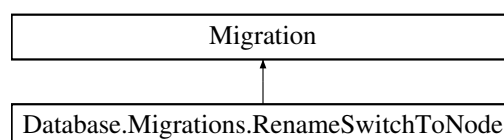
5.92.1 Detailed Description

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

- Database/Migrations/20250927183203_RemoveOsType.cs
- Database/Migrations/20250927183203_RemoveOsType.Designer.cs

5.93 Database.Migrations.RenameSwitchToNode Class Reference

Inheritance diagram for Database.Migrations.RenameSwitchToNode:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

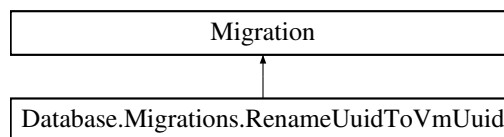
5.93.1 Detailed Description

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

- Database/Migrations/20251019212536_RenameSwitchToNode.cs
- Database/Migrations/20251019212536_RenameSwitchToNode.Designer.cs

5.94 Database.Migrations.RenameUuidToVmUuid Class Reference

Inheritance diagram for Database.Migrations.RenameUuidToVmUuid:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.94.1 Detailed Description

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

- Database/Migrations/20250727102040_RenameUuidToVmUuid.cs
- Database/Migrations/20250727102040_RenameUuidToVmUuid.Designer.cs

5.95 Models.Requests.SendDataToConsoleRequest Class Reference

Properties

- required string **Uuid** [get, set]
- required string **Data** [get, set]

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

- Models/Requests/SendDataToConsoleRequest.cs

5.96 Models.StreamData Class Reference

Represents data read from a stream.

Properties

- required byte[] **Data** [get, set]
Data read from the stream.
- bool **IsStreamBroken** [get, set]
Indicates whether the stream has been broken.

5.96.1 Detailed Description

Represents data read from a stream.

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

- Models/StreamData.cs

5.97 Models.StreamInfo Class Reference

Information about a virtual machine console stream.

Properties

- Guid **Uuid** [get, set]
Unique UUID identifier of the virtual machine.
- IntPtr **Stream** [get, set]
Pointer to the native console stream.

5.97.1 Detailed Description

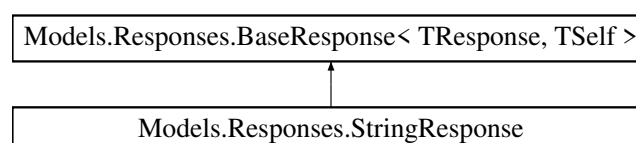
Information about a virtual machine console stream.

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

- Models/StreamInfo.cs

5.98 Models.Responses.StringResponse Class Reference

Inheritance diagram for Models.Responses.StringResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/StringResponse.cs

5.99 Models.Responses.SystemInfoResponse Class Reference

Properties

- required string **Os** [get, set]
- required string **Arch** [get, set]

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

- Models/Responses/SystemInfoResponse.cs

5.100 Models.TerminalData Class Reference

Virtual machine terminal data.

Properties

- required string **Uuid** [get, set]
UUID of the virtual machine.
- required string **Data** [get, set]
Terminal data in text format.

5.100.1 Detailed Description

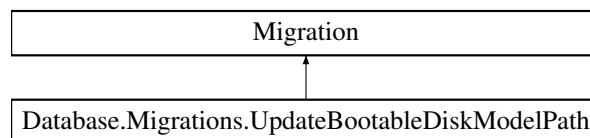
Virtual machine terminal data.

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

- Models/TerminalData.cs

5.101 Database.Migrations.UpdateBootableDiskModelPath Class Reference

Inheritance diagram for Database.Migrations.UpdateBootableDiskModelPath:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.101.1 Detailed Description

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

- Database/Migrations/20250728193215_UpdateBootableDiskModelPath.cs
- Database/Migrations/20250728193215_UpdateBootableDiskModelPath.Designer.cs

5.102 Models.Requests.UpdateEntityPositionRequest Class Reference

Properties

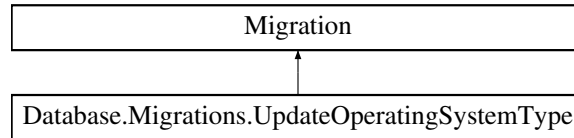
- required int **Id** [get, set]
- required [EntityType](#) **Type** [get, set]
- required int **X** [get, set]
- required int **Y** [get, set]

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

- Models/Requests/UpdateEntityPositionRequest.cs

5.103 Database.Migrations.UpdateOperatingSystemType Class Reference

Inheritance diagram for Database.Migrations.UpdateOperatingSystemType:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

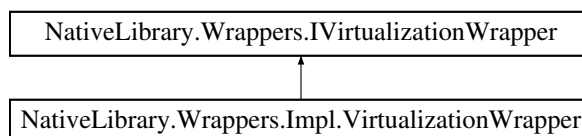
5.103.1 Detailed Description

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

- Database/Migrations/20250728194949_UpdateOperatingSystemType.cs
- Database/Migrations/20250728194949_UpdateOperatingSystemType.Designer.cs

5.104 NativeLibrary.Wrappers.Impl.VirtualizationWrapper Class Reference

Inheritance diagram for NativeLibrary.Wrappers.Impl.VirtualizationWrapper:



Public Member Functions

- void [InitializeConnection](#) (string? connectionUrl)
Initializes connection with the hypervisor.
- [NativeConnectionInfo GetConnectionInfo](#) ()
Gets connection information from the hypervisor.
- void [CreateVirtualMachine](#) (string xmlDefinition)
Creates a virtual machine based on XML definition.
- [NativeVirtualMachineInfo GetVirtualMachineInfo](#) (string name)
Gets information about a virtual machine.
- INativeList< [NativeVirtualMachineInfo](#) > [GetListOfVirtualMachines](#) ()

- Gets a list of all virtual machines.*
- bool [IsConnectionAlive](#) ()
Checks if the connection with the hypervisor is active.
- IntPtr [OpenVirtualMachineConsole](#) (Guid uuid)
Opens a console stream for a virtual machine.
- void [SendDataToStream](#) (IntPtr stream, string data)
Sends data to the console stream.
- [NativeStreamData GetDataFromStream](#) (IntPtr stream)
Gets data from the console stream.
- void [CloseStream](#) (IntPtr stream)
Closes the console stream.
- IntPtr [CreateVirtualNetwork](#) (string networkDefinition)
Creates a virtual network based on definition.
- void [AttachDeviceToVirtualMachine](#) (Guid uuid, string deviceDefinition)
Attaches a device to a virtual machine.
- void [DetachDeviceFromVirtualMachine](#) (Guid uuid, string deviceDefinition)
Detaches a device from a virtual machine.
- void [UpdateVmDevice](#) (Guid uuid, string deviceDefinition)
Updates a device on a virtual machine.
- void [DestroyNetwork](#) (string name)
Destroys a virtual network.

5.104.1 Member Function Documentation

5.104.1.1 AttachDeviceToVirtualMachine()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.AttachDeviceToVirtualMachine (
    Guid uuid,
    string deviceDefinition) [inline]
```

Attaches a device to a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.2 CloseStream()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.CloseStream (
    IntPtr stream) [inline]
```

Closes the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
---------------	------------------------

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.3 CreateVirtualMachine()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.CreateVirtualMachine (
    string xmlDefinition) [inline]
```

Creates a virtual machine based on XML definition.

Parameters

<i>xmlDefinition</i>	Machine definition in XML format.
----------------------	-----------------------------------

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.4 CreateVirtualNetwork()

```
IntPtr NativeLibrary.Wrappers.Impl.VirtualizationWrapper.CreateVirtualNetwork (
    string networkDefinition) [inline]
```

Creates a virtual network based on definition.

Parameters

<i>networkDefinition</i>	Network definition in XML format.
--------------------------	-----------------------------------

Returns

Pointer to the created network.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.5 DestroyNetwork()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.DestroyNetwork (
    string name) [inline]
```

Destroys a virtual network.

Parameters

<i>name</i>	Name of the network to destroy.
-------------	---------------------------------

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.6 DetachDeviceFromVirtualMachine()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.DetachDeviceFromVirtualMachine (
    Guid uuid,
    string deviceDefinition) [inline]
```

Detaches a device from a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.7 GetConnectionInfo()

```
NativeConnectionInfo NativeLibrary.Wrappers.Impl.VirtualizationWrapper.GetConnectionInfo ()
[inline]
```

Gets connection information from the hypervisor.

Returns

Connection information.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.8 GetDataFromStream()

```
NativeStreamData NativeLibrary.Wrappers.Impl.VirtualizationWrapper.GetDataFromStream (
    IntPtr stream) [inline]
```

Gets data from the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
---------------	------------------------

Returns

Data from the stream.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.9 GetListOfVirtualMachines()

```
INativeList< NativeVirtualMachineInfo > NativeLibrary.Wrappers.Impl.VirtualizationWrapper.↵
GetListOfVirtualMachines () [inline]
```

Gets a list of all virtual machines.

Returns

List of virtual machines.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.10 GetVirtualMachineInfo()

```
NativeVirtualMachineInfo NativeLibrary.Wrappers.Impl.VirtualizationWrapper.GetVirtualMachine↵
Info (
    string name) [inline]
```

Gets information about a virtual machine.

Parameters

<i>name</i>	Name of the virtual machine.
-------------	------------------------------

Returns

Virtual machine information.

Implements [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.104.1.11 InitializeConnection()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.InitializeConnection (
    string? connectionUrl) [inline]
```

Initializes connection with the hypervisor.

Parameters

<i>connectionUrl</i>	Connection URL or null for default.
----------------------	-------------------------------------

Implements [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.104.1.12 IsConnectionAlive()

```
bool NativeLibrary.Wrappers.Impl.VirtualizationWrapper.IsConnectionAlive () [inline]
```

Checks if the connection with the hypervisor is active.

Returns

True if connection is active.

Implements [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.104.1.13 OpenVirtualMachineConsole()

```
IntPtr NativeLibrary.Wrappers.Impl.VirtualizationWrapper.OpenVirtualMachineConsole (
    Guid uuid) [inline]
```

Opens a console stream for a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
-------------	------------------------------

Returns

Pointer to the console stream.

Implements [NativeLibrary.Wrappers.Impl.VirtualizationWrapper](#).

5.104.1.14 SendDataToStream()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.SendDataToStream (
    IntPtr stream,
    string data) [inline]
```

Sends data to the console stream.

Parameters

<i>stream</i>	Pointer to the stream.
<i>data</i>	Data to send.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

5.104.1.15 UpdateVmDevice()

```
void NativeLibrary.Wrappers.Impl.VirtualizationWrapper.UpdateVmDevice (
    Guid uuid,
    string deviceDefinition) [inline]
```

Updates a device on a virtual machine.

Parameters

<i>uuid</i>	UUID of the virtual machine.
<i>deviceDefinition</i>	Device definition in XML format.

Implements [NativeLibrary.Wrappers.IVirtualizationWrapper](#).

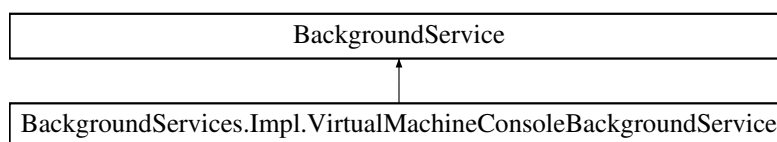
The documentation for this class was generated from the following file:

- [NativeLibrary/Wrappers/Impl/VirtualizationWrapper.cs](#)

5.105 BackgroundServices.Impl.VirtualMachineConsoleBackgroundService Class Reference ↩

Background service for managing virtual machine console data streaming.

Inheritance diagram for BackgroundServices.Impl.VirtualMachineConsoleBackgroundService:



Public Member Functions

- [VirtualMachineConsoleBackgroundService](#) (ILogger< [VirtualMachineConsoleBackgroundService](#) > logger, [IVirtualMachineConsoleService](#) vmConsoleService, IHubContext< [VirtualMachineConsoleHub](#) > hub↩ Context)

Initializes a new instance of the [VirtualMachineConsoleBackgroundService](#).

Protected Member Functions

- override async Task [ExecuteAsync](#) (Cancellation token stoppingToken)
Executes the background service to listen for and stream console data.

5.105.1 Detailed Description

Background service for managing virtual machine console data streaming.

5.105.2 Constructor & Destructor Documentation

5.105.2.1 VirtualMachineConsoleBackgroundService()

BackgroundServices.Impl.VirtualMachineConsoleBackgroundService.VirtualMachineConsoleBackgroundService (

```
ILogger< VirtualMachineConsoleBackgroundService > logger,  
IVirtualMachineConsoleService vmConsoleService,  
IHubContext< VirtualMachineConsoleHub > hubContext) [inline]
```

Initializes a new instance of the [VirtualMachineConsoleBackgroundService](#).

Parameters

<i>logger</i>	Logger instance.
<i>vmConsoleService</i>	Virtual machine console service.
<i>hubContext</i>	SignalR hub context for console communication.

5.105.3 Member Function Documentation

5.105.3.1 ExecuteAsync()

```
override async Task BackgroundServices.Impl.VirtualMachineConsoleBackgroundService.ExecuteAsync (
    CancellationToken stoppingToken) [inline], [protected]
```

Executes the background service to listen for and stream console data.

Parameters

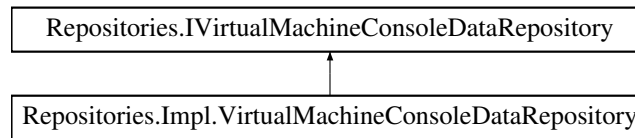
<i>stoppingToken</i>	Cancellation token for stopping the service.
----------------------	--

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

- BackgroundServices/Impl/VirtualMachineConsoleBackgroundService.cs

5.106 Repositories.Impl.VirtualMachineConsoleDataRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualMachineConsoleDataRepository:



Public Member Functions

- **VirtualMachineConsoleDataRepository** (IOptions< [InterconnectConfig](#) > config)
- void [AddDataToConsole](#) (Guid uuid, byte[] data)
Adds data to a virtual machine console.
- List< byte > [GetData](#) (Guid uuid)
Retrieves data from a virtual machine console.

5.106.1 Member Function Documentation

5.106.1.1 AddDataToConsole()

```
void Repositories.Impl.VirtualMachineConsoleDataRepository.AddDataToConsole (
    Guid uuid,
    byte[] data) [inline]
```

Adds data to a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
<i>data</i>	Data to add.

Implements [Repositories.IVirtualMachineConsoleDataRepository](#).

5.106.1.2 GetData()

```
List< byte > Repositories.Impl.VirtualMachineConsoleDataRepository.GetData (
    Guid uuid) [inline]
```

Retrieves data from a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

List of bytes from console.

Implements [Repositories.IVirtualMachineConsoleDataRepository](#).

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

- Repositories/Impl/VirtualMachineConsoleDataRepository.cs

5.107 RepositoriesTests.VirtualMachineConsoleDataRepositoryTests Class Reference

Public Member Functions

- void **Setup** ()
- void **AddDataToConsole_WhenInvoked_ShouldAddDataToConsole** ()
- void **AddDataToConsole_ExceedDataLength_ShouldTrimData** ()

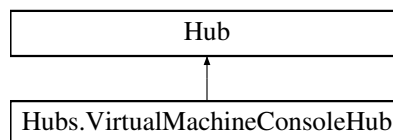
The documentation for this class was generated from the following file:

- RepositoriesTests/VirtualMachineConsoleDataRepositoryTests.cs

5.108 Hubs.VirtualMachineConsoleHub Class Reference

SignalR hub for communicating with virtual machine console.

Inheritance diagram for Hubs.VirtualMachineConsoleHub:



Public Member Functions

- **VirtualMachineConsoleHub** ([IVirtualMachineConsoleService](#) virtualMachineConsole)
- async Task [JoinConsoleGroup](#) (string uuid)
Adds client to console group for a specific virtual machine.
- async Task [LeaveConsoleGroup](#) (string uuid)
Removes client from console group for a specific virtual machine.
- TerminalDataResponse [GetInitialDataForConsole](#) (string uuid)
Retrieves initial console data for a virtual machine.
- void [SendDataToConsole](#) ([SendDataToConsoleRequest](#) request)
Sends data to a virtual machine console.

5.108.1 Detailed Description

SignalR hub for communicating with virtual machine console.

5.108.2 Member Function Documentation

5.108.2.1 GetInitialDataForConsole()

```
TerminalDataResponse Hubs.VirtualMachineConsoleHub.GetInitialDataForConsole (
    string uuid) [inline]
```

Retrieves initial console data for a virtual machine.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Initial console data.

5.108.2.2 JoinConsoleGroup()

```
async Task Hubs.VirtualMachineConsoleHub.JoinConsoleGroup (  
    string uuid) [inline]
```

Adds client to console group for a specific virtual machine.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

5.108.2.3 LeaveConsoleGroup()

```
async Task Hubs.VirtualMachineConsoleHub.LeaveConsoleGroup (  
    string uuid) [inline]
```

Removes client from console group for a specific virtual machine.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

5.108.2.4 SendDataToConsole()

```
void Hubs.VirtualMachineConsoleHub.SendDataToConsole (  
    SendDataToConsoleRequest request) [inline]
```

Sends data to a virtual machine console.

Parameters

<i>request</i>	Data to send.
----------------	---------------

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

- Hubs/VirtualMachineConsoleHub.cs

5.109 HubsTests.VirtualMachineConsoleHubTests Class Reference

Public Member Functions

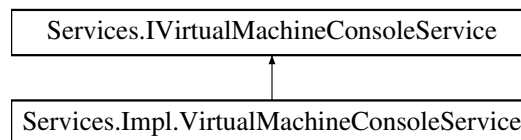
- void **Setup** ()
- void **Teardown** ()
- void **GetInitialDataForConsole_WhenInvoked_ShouldGetInitialDataForConsole** ()
- void **SendDataToConsole_WhenInvoked_ShouldSendDataToConsole** ()

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

- HubsTests/VirtualMachineConsoleHubTests.cs

5.110 Services.Impl.VirtualMachineConsoleService Class Reference

Inheritance diagram for Services.Impl.VirtualMachineConsoleService:



Public Member Functions

- **VirtualMachineConsoleService** ([IVirtualizationWrapper](#) virtualizationWrapper, [IVirtualMachineConsoleDataRepository](#) consoleDataRepository, [IVirtualMachineConsoleStreamRepository](#) consoleStreamRepository)
- byte[] [GetInitialConsoleData](#) (Guid uuid)
Retrieves initial console data from a virtual machine.
- [StreamData](#) [GetBytesFromConsole](#) ([StreamInfo](#) stream)
Retrieves data from a console stream.
- [StreamInfo](#) [OpenVirtualMachineConsole](#) (Guid uuid)
Opens a virtual machine console stream.
- void [SendBytesToVirtualMachineConsoleByUuid](#) (Guid uuid, string data)
Sends bytes to a virtual machine console.
- List< [StreamInfo](#) > [GetStreams](#) ()
Retrieves a list of all opened streams.
- void [CloseStream](#) ([StreamInfo](#) stream)
Closes a console stream.

5.110.1 Member Function Documentation

5.110.1.1 CloseStream()

```
void Services.Impl.VirtualMachineConsoleService.CloseStream (
    StreamInfo stream) [inline]
```

Closes a console stream.

Parameters

<i>stream</i>	Stream information to close.
---------------	------------------------------

Implements [Services.IVirtualMachineConsoleService](#).

5.110.1.2 GetBytesFromConsole()

```
StreamData Services.Impl.VirtualMachineConsoleService.GetBytesFromConsole (  
    StreamInfo stream) [inline]
```

Retrieves data from a console stream.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Returns

Stream data.

Implements [Services.IVirtualMachineConsoleService](#).

5.110.1.3 GetInitialConsoleData()

```
byte[] Services.Impl.VirtualMachineConsoleService.GetInitialConsoleData (  
    Guid uuid) [inline]
```

Retrieves initial console data from a virtual machine.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Initial console data.

Implements [Services.IVirtualMachineConsoleService](#).

5.110.1.4 GetStreams()

```
List< StreamInfo > Services.Impl.VirtualMachineConsoleService.GetStreams () [inline]
```

Retrieves a list of all opened streams.

Returns

List of opened streams.

Implements [Services.IVirtualMachineConsoleService](#).

5.110.1.5 OpenVirtualMachineConsole()

```
StreamInfo Services.Impl.VirtualMachineConsoleService.OpenVirtualMachineConsole (  
    Guid uuid) [inline]
```

Opens a virtual machine console stream.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Opened stream information.

Implements [Services.IVirtualMachineConsoleService](#).

5.110.1.6 SendBytesToVirtualMachineConsoleByUuid()

```
void Services.Impl.VirtualMachineConsoleService.SendBytesToVirtualMachineConsoleByUuid (
    Guid uuid,
    string bytes) [inline]
```

Sends bytes to a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
<i>bytes</i>	Bytes to send in text format.

Implements [Services.IVirtualMachineConsoleService](#).

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

- Services/Impl/VirtualMachineConsoleService.cs

5.111 ServicesTests.VirtualMachineConsoleServiceTests Class Reference

Public Member Functions

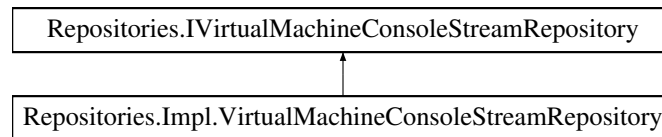
- void **Setup** ()
- void **GetInitialConsoleData_WhenInvoked_ShouldReturnInitialConsoleData** ()
- void **GetBytesFromConsole_WhenInvoked_ShouldGetBytesFromConsole** ()
- void **GetBytesFromConsole_WhenInvoked_ShouldSaveBytesToRepository** ()
- void **OpenVirtualMachineConsole_WhenInvoked_ShouldOpenVirtualMachineConsole** ()
- void **OpenVirtualMachineConsole_WhenInvokedAndVmCreated_ShouldAddStreamToRepository** ()
- void **SendBytesToVirtualMachineConsoleByUuid_WhenInvoked_ShouldSendDataToStream** ()
- void **SendBytesToVirtualMachineConsoleByUuid_WhenInvokedWithNotOpenedConsole_ShouldOpenConsole** ()
- void **GetStreams_WhenInvoked_ShouldReturnStreams** ()
- void **CloseStream_WhenInvoked_ShouldRemoveStreamFromRepository** ()
- void **CloseStream_WhenInvoked_ShouldCloseStream** ()

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

- ServicesTests/VirtualMachineConsoleServiceTests.cs

5.112 Repositories.Impl.VirtualMachineConsoleStreamRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualMachineConsoleStreamRepository:



Public Member Functions

- void [Add](#) ([StreamInfo](#) stream)
Adds a console stream to the repository.
- List< [StreamInfo](#) > [GetAllStreams](#) ()
Retrieves all opened streams.
- [StreamInfo?](#) [GetByUuid](#) (Guid uuid)
Retrieves a stream by virtual machine UUID.
- void [Remove](#) ([StreamInfo](#) stream)
Removes a console stream from the repository.

5.112.1 Member Function Documentation

5.112.1.1 Add()

```
void Repositories.Impl.VirtualMachineConsoleStreamRepository.Add (
    StreamInfo stream) [inline]
```

Adds a console stream to the repository.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Implements [Repositories.IVirtualMachineConsoleStreamRepository](#).

5.112.1.2 GetAllStreams()

```
List< StreamInfo > Repositories.Impl.VirtualMachineConsoleStreamRepository.GetAllStreams ()
[inline]
```

Retrieves all opened streams.

Returns

List of streams.

Implements [Repositories.IVirtualMachineConsoleStreamRepository](#).

5.112.1.3 GetByUuid()

```
StreamInfo? Repositories.Impl.VirtualMachineConsoleStreamRepository.GetByUuid (
    Guid uuid) [inline]
```

Retrieves a stream by virtual machine UUID.

Parameters

<i>uuid</i>	Virtual machine UUID.
-------------	-----------------------

Returns

Stream information or null.

Implements [Repositories.IVirtualMachineConsoleStreamRepository](#).

5.112.1.4 Remove()

```
void Repositories.Impl.VirtualMachineConsoleStreamRepository.Remove (
    StreamInfo stream) [inline]
```

Removes a console stream from the repository.

Parameters

<i>stream</i>	Stream information.
---------------	---------------------

Implements [Repositories.IVirtualMachineConsoleStreamRepository](#).

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

- Repositories/Impl/VirtualMachineConsoleStreamRepository.cs

5.113 RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests Class Reference

Public Member Functions

- void **Setup** ()
- void **Add_WhenInvoked_ShouldAddNewStream** ()
- void **GetAllStreams_WhenInvoked_ShouldGetAllStreams** ()
- void **Remove_WhenInvoked_ShouldRemoveStream** ()

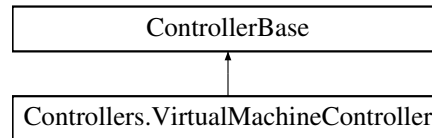
The documentation for this class was generated from the following file:

- RepositoriesTests/VirtualMachineConsoleStreamRepositoryTests.cs

5.114 Controllers.VirtualMachineController Class Reference

Controller managing virtual machines.

Inheritance diagram for Controllers.VirtualMachineController:



Public Member Functions

- **VirtualMachineController** ([IBootableDiskProviderService](#) bootableDiskProviderService)
- `async Task< ActionResult< BootableDisksResponse > > GetAvailableBootableDisks ()`
Retrieves a list of available bootable disks.

5.114.1 Detailed Description

Controller managing virtual machines.

5.114.2 Member Function Documentation

5.114.2.1 GetAvailableBootableDisks()

```

async Task< ActionResult< BootableDisksResponse > > Controllers.VirtualMachineController.↔
GetAvailableBootableDisks () [inline]
  
```

Retrieves a list of available bootable disks.

Returns

List of bootable disks.

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

- Controllers/VirtualMachineController.cs

5.115 Models.VirtualMachineCreateDefinition Class Reference

Definition for creating a new virtual machine.

Public Member Functions

- `string GetVirtualMachineNameWithPrefix (string prefix)`
Generates a virtual machine name with a prefix.

Properties

- required string **Name** [get, set]
Virtual machine name.
- required uint **Memory** [get, set]
Amount of RAM memory in MB.
- required uint **VirtualCpus** [get, set]
Number of virtual CPUs.
- required string **BootableDiskPath** [get, set]
Path to the bootable disk.

5.115.1 Detailed Description

Definition for creating a new virtual machine.

5.115.2 Member Function Documentation

5.115.2.1 GetVirtualMachineNameWithPrefix()

```
string Models.VirtualMachineCreateDefinition.GetVirtualMachineNameWithPrefix (
    string prefix) [inline]
```

Generates a virtual machine name with a prefix.

Parameters

<i>prefix</i>	Prefix to add before the name.
---------------	--------------------------------

Returns

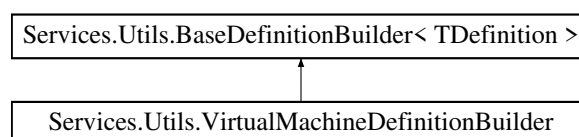
Virtual machine name with prefix.

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

- Models/VirtualMachineCreateDefinition.cs

5.116 Services.Utills.VirtualMachineDefinitionBuilder Class Reference

Inheritance diagram for Services.Utills.VirtualMachineDefinitionBuilder:



Public Member Functions

- [VirtualMachineDefinitionBuilder](#) **SetPrefix** (string prefix)
- override [VirtualMachineDefinitionBuilder](#) **SetFromCreateDefinition** ([VirtualMachineCreateDefinition](#) definition)
- override string **Build** ()

Public Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition >](#) **SetFromCreateDefinition** (TDefinition definition)
- string **Build** ()

Additional Inherited Members

Protected Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- delegate void **BuildingBlock** (XmlWriter writer)
- void **ChecksEverythingIsProvided** (params object?[] args)
- StringWriter **CreateXmlWriter** ()

Protected Attributes inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- **XmlWriter**

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

- Services/Utls/VirtualMachineDefinitionBuilder.cs

5.117 ServicesTests.Utls.VirtualMachineDefinitionBuilderTests Class Reference

Public Member Functions

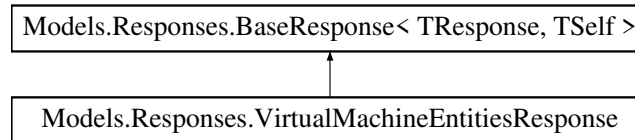
- void **OnSetup** ()
- void **Build_WhenNothingsIsProvided_ShouldThrowException** ()
- void **SetFromCreateDefinition_WhenPrefixIsNotProvided_ShouldThrowException** ()
- void **Build_WhenBootableDiskPathProvided_ShouldIncludeBootableDiskPathInXml** ()
- void **Build_WhenMemoryProvided_ShouldIncludeMemoryInXml** ()
- void **Build_WhenNameProvided_ShouldIncludeNameInXml** ()
- void **Build_WhenVirtualCpusProvided_ShouldIncludeVirtualCpusInXml** ()

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

- ServicesTests/Utls/VirtualMachineDefinitionBuilderTests.cs

5.118 Models.Responses.VirtualMachineEntitiesResponse Class Reference

Inheritance diagram for Models.Responses.VirtualMachineEntitiesResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualMachineEntitiesResponse.cs

5.119 Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO Class Reference

Properties

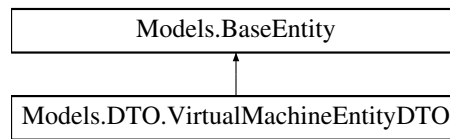
- required List< [VirtualMachineEntityDTO](#) > **VirtualMachineEntities** [get, set]
- required List< [VirtualMachineEntityMacAddressDTO](#) > **MacAddressEntities** [get, set]

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

- Models/DTO/VirtualMachineEntitiesWithMacAddressesDTO.cs

5.120 Models.DTO.VirtualMachineEntityDTO Class Reference

Inheritance diagram for Models.DTO.VirtualMachineEntityDTO:



Properties

- Guid? **VmUuid** [get, set]
- required string **Name** [get, set]
- VirtualMachineEntityType **Type** = VirtualMachineEntityType.Host [get, set]
- [VirtualMachineState](#) **State** = VirtualMachineState.Unknown [get, set]

Properties inherited from [Models.BaseEntity](#)

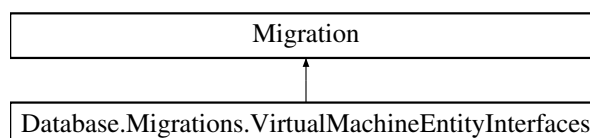
- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

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

- Models/DTO/VirtualMachineEntityDTO.cs

5.121 Database.Migrations.VirtualMachineEntityInterfaces Class Reference

Inheritance diagram for Database.Migrations.VirtualMachineEntityInterfaces:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.121.1 Detailed Description

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

- Database/Migrations/20260118124006_VirtualMachineEntityInterfaces.cs
- Database/Migrations/20260118124006_VirtualMachineEntityInterfaces.Designer.cs

5.122 Models.DTO.VirtualMachineEntityMacAddressDTO Class Reference

Properties

- required int **VirtualMachineEntityId** [get, set]
- List< string > **MacAddresses** = [] [get, set]

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

- Models/DTO/VirtualMachineEntityMacAddressDTO.cs

5.123 Models.Database.VirtualMachineEntityModel Class Reference

[Database](#) model representing a virtual machine entity.

Properties

- int **Id** [get, set]
Entity identifier.
- Guid? **VmUuid** [get, set]
Virtual machine UUID in the hypervisor.
- required string **Name** [get, set]
Virtual machine name.
- VirtualMachineEntityType **Type** = VirtualMachineEntityType.Host [get, set]
Virtual machine entity type.
- required int **X** [get, set]
X coordinate of the entity position.
- required int **Y** [get, set]
Y coordinate of the entity position.

5.123.1 Detailed Description

[Database](#) model representing a virtual machine entity.

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

- Models/Database/VirtualMachineEntityModel.cs

5.124 Models.Database.VirtualMachineEntityNetworkInterfaceModel Class Reference

[Database](#) model representing a virtual machine network interface.

Properties

- int **Id** [get, set]
Interface identifier.
- required string **Definition** [get, set]
Network interface definition in XML format.
- int **VirtualMachineEntityId** [get, set]
Virtual machine entity identifier.
- [VirtualMachineEntityModel?](#) **VirtualMachineEntity** [get, set]
Virtual machine entity.
- int **VirtualNetworkEntityConnectionId** [get, set]
Network connection identifier.
- [VirtualNetworkEntityConnectionModel?](#) **VirtualNetworkEntityConnection** [get, set]
Network connection.

5.124.1 Detailed Description

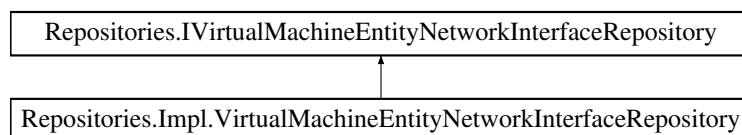
[Database](#) model representing a virtual machine network interface.

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

- Models/Database/VirtualMachineEntityNetworkInterfaceModel.cs

5.125 Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository:



Public Member Functions

- **VirtualMachineEntityNetworkInterfaceRepository** ([InterconnectDbContext](#) context)
- Task **Create** ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Creates a new network interface.
- Task **Update** ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Updates a network interface.
- Task **Remove** ([VirtualMachineEntityNetworkInterfaceModel](#) networkInterface)
Removes a network interface.
- Task< [VirtualMachineEntityNetworkInterfaceModel?](#)> **GetByIds** (int virtualMachineId, int connectionId)
Retrieves a network interface by virtual machine and connection identifiers.
- Task< List< [VirtualMachineEntityNetworkInterfaceModel](#) > > **GetByNetworkName** (int virtualMachineId, string networkName)
Retrieves network interfaces connected to a specific network.
- Task< List< [VirtualMachineEntityNetworkInterfaceModel](#) > > **GetByVirtualMachineId** (int id)
Retrieves all network interfaces for a virtual machine.

5.125.1 Member Function Documentation

5.125.1.1 Create()

```
Task Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.Create (
    VirtualMachineEntityNetworkInterfaceModel networkInterface) [inline]
```

Creates a new network interface.

Parameters

<i>networkInterface</i>	Interface model to create.
-------------------------	----------------------------

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.125.1.2 GetByIds()

```
Task< VirtualMachineEntityNetworkInterfaceModel?> Repositories.Impl.VirtualMachineEntity↔
NetworkInterfaceRepository.GetByIds (
    int virtualMachineId,
    int connectionId)
```

Retrieves a network interface by virtual machine and connection identifiers.

Parameters

<i>virtual↔ MachineId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.

Returns

Network interface or null.

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.125.1.3 GetByNetworkName()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.GetByNetworkName (
    int virtualMachineId,
    string networkName)
```

Retrieves network interfaces connected to a specific network.

Parameters

<i>virtualMachineId</i>	Virtual machine identifier.
<i>networkName</i>	Network name.

Returns

List of network interfaces.

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.125.1.4 GetByVirtualMachineId()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.GetByVirtualMachineId (
    int id)
```

Retrieves all network interfaces for a virtual machine.

Parameters

<i>id</i>	Virtual machine identifier.
-----------	-----------------------------

Returns

List of network interfaces.

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.125.1.5 Remove()

```
Task Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.Remove (
    VirtualMachineEntityNetworkInterfaceModel networkInterface) [inline]
```

Removes a network interface.

Parameters

<i>networkInterface</i>	Interface model to remove.
-------------------------	----------------------------

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.125.1.6 Update()

```
Task Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.Update (
    VirtualMachineEntityNetworkInterfaceModel networkInterface) [inline]
```

Updates a network interface.

Parameters

<i>networkInterface</i>	Interface model to update.
-------------------------	----------------------------

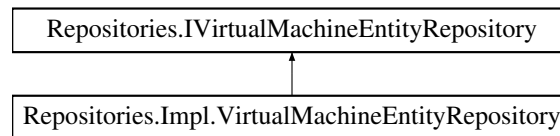
Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

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

- Repositories/Impl/VirtualMachineEntityNetworkInterfaceRepository.cs

5.126 Repositories.Impl.VirtualMachineEntityRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualMachineEntityRepository:



Public Member Functions

- **VirtualMachineEntityRepository** ([InterconnectDbContext](#) context)
- async Task **Add** ([VirtualMachineEntityModel](#) model)
Adds a new virtual machine entity.
- async Task< List< [VirtualMachineEntityModel](#) > > **GetAll** ()
Retrieves all virtual machine entities.
- async Task< [VirtualMachineEntityModel](#) > **GetById** (int id)
Retrieves a virtual machine entity by identifier.
- async Task **Update** ([VirtualMachineEntityModel](#) model)
Updates a virtual machine entity.

5.126.1 Member Function Documentation

5.126.1.1 Add()

```
async Task Repositories.Impl.VirtualMachineEntityRepository.Add (
    VirtualMachineEntityModel model) [inline]
```

Adds a new virtual machine entity.

Parameters

<i>model</i>	Entity model to add.
--------------	----------------------

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.126.1.2 GetAll()

```
async Task< List< VirtualMachineEntityModel > > Repositories.Impl.VirtualMachineEntityRepository.GetAll () [inline]
```

Retrieves all virtual machine entities.

Returns

List of entities.

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.126.1.3 GetById()

```
async Task< VirtualMachineEntityModel > Repositories.Impl.VirtualMachineEntityRepository.GetById (
    int id) [inline]
```

Retrieves a virtual machine entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Virtual machine entity.

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.126.1.4 Update()

```
async Task Repositories.Impl.VirtualMachineEntityRepository.Update (
    VirtualMachineEntityModel model) [inline]
```

Updates a virtual machine entity.

Parameters

<i>model</i>	Entity model to update.
--------------	-------------------------

Implements [Repositories.IVirtualMachineEntityRepository](#).

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

- `Repositories/Impl/VirtualMachineEntityRepository.cs`

5.127 RepositoriesTests.VirtualMachineEntityRepositoryTests Class Reference

Public Member Functions

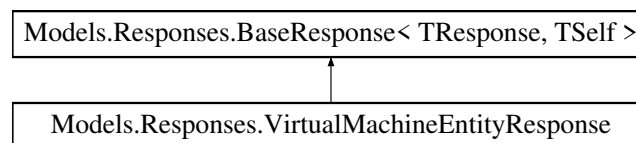
- void **Setup** ()
- void **Teardown** ()
- async Task **Add_WhenInvoked_ShouldAddEntity** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllEntities** ()
- async Task **GetById_WhenInvoked_ShouldGetEntityById** ()
- async Task **Update_WhenInvoked_ShouldUpdateEntity** ()

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

- RepositoriesTests/VirtualMachineEntityRepositoryTests.cs

5.128 Models.Responses.VirtualMachineEntityResponse Class Reference

Inheritance diagram for Models.Responses.VirtualMachineEntityResponse:



Additional Inherited Members

Static Public Member Functions inherited from

[Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

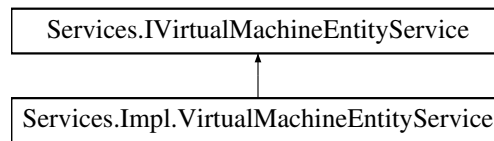
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualMachineEntityResponse.cs

5.129 Services.Impl.VirtualMachineEntityService Class Reference

Inheritance diagram for Services.Impl.VirtualMachineEntityService:



Public Member Functions

- **VirtualMachineEntityService** ([IVirtualMachineEntityRepository](#) repository, [IBootableDiskProviderService](#) bootableDiskProviderService, [IVirtualMachineManagerService](#) vmManagerService, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterfaceRepository)
- `async Task< VirtualMachineEntityDTO > CreateEntity` (string name, int bootableDiskId, uint memory, uint virtualCpus, VirtualMachineEntityType type, int x, int y)
Creates a new virtual machine entity.
- `async Task< List< VirtualMachineEntityDTO > > GetEntities` ()
Retrieves a list of all virtual machine entities.
- `async Task< List< VirtualMachineEntityMacAddressDTO > > GetMacAddresses` ()
Retrieves a list of virtual machine MAC addresses.
- `async Task< VirtualMachineEntityDTO > GetById` (int id)
Retrieves a virtual machine entity by identifier.
- `async Task< VirtualMachineEntityDTO > UpdateEntityPosition` (int id, int x, int y)
Updates the position of a virtual machine entity on the board.
- `async Task< VirtualMachineEntityDTO > UpdateEntityVmUUID` (int id, string uuid)
Updates the virtual machine UUID for an entity.

5.129.1 Member Function Documentation

5.129.1.1 CreateEntity()

```

async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.CreateEntity (
    string name,
    int bootableDiskId,
    uint memory,
    uint virtualCpus,
    VirtualMachineEntityType type,
    int x,
    int y) [inline]
  
```

Creates a new virtual machine entity.

Parameters

<i>name</i>	Virtual machine name.
<i>bootableDiskId</i>	Bootable disk identifier.
<i>memory</i>	Amount of RAM in MB.
<i>virtualCpus</i>	Number of virtual CPUs.

<i>type</i>	Virtual machine entity type.
<i>x</i>	X coordinate position.
<i>y</i>	Y coordinate position.

Returns

Created virtual machine entity.

Implements [Services.IVirtualMachineEntityService](#).

5.129.1.2 GetById()

```
async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.GetById (
    int id) [inline]
```

Retrieves a virtual machine entity by identifier.

Parameters

<i>id</i>	Entity identifier.
-----------	--------------------

Returns

Virtual machine entity.

Implements [Services.IVirtualMachineEntityService](#).

5.129.1.3 GetEntities()

```
async Task< List< VirtualMachineEntityDTO > > Services.Impl.VirtualMachineEntityService.Get↵
Entities () [inline]
```

Retrieves a list of all virtual machine entities.

Returns

List of virtual machine entities.

Implements [Services.IVirtualMachineEntityService](#).

5.129.1.4 GetMacAddresses()

```
async Task< List< VirtualMachineEntityMacAddressDTO > > Services.Impl.VirtualMachineEntity↵
Service.GetMacAddresses () [inline]
```

Retrieves a list of virtual machine MAC addresses.

Returns

List of MAC addresses.

Implements [Services.IVirtualMachineEntityService](#).

5.129.1.5 UpdateEntityPosition()

```
async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.UpdateEntity↵  
Position (   
    int id,   
    int x,   
    int y) [inline]
```

Updates the position of a virtual machine entity on the board.

Parameters

<i>id</i>	Entity identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated entity.

Implements [Services.IVirtualMachineEntityService](#).

5.129.1.6 UpdateEntityVmUUID()

```
async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.UpdateEntity↵  
VmUUID (   
    int id,   
    string uuid) [inline]
```

Updates the virtual machine UUID for an entity.

Parameters

<i>id</i>	Entity identifier.
<i>uuid</i>	New virtual machine UUID.

Returns

Updated entity.

Implements [Services.IVirtualMachineEntityService](#).

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

- Services/Impl/VirtualMachineEntityService.cs

5.130 ServicesTests.VirtualMachineEntityServiceTests Class Reference

Public Member Functions

- void **SetUp** ()
- async Task **CreateEntity_WhenInvoked_ShouldReturnEntity** ()
- async Task **CreateEntity_WhenInvoked_ShouldCreateEntityInRepository** ()
- async Task **GetEntities_WhenEntitiesExist_ShouldReturnListOfEntities** ()
- async Task **GetEntityById_WhenEntityExist_ShouldReturnEntity** ()
- async Task **UpdateEntityPosition_WhenEntityExist_UpdatesPosition** ()
- async Task **UpdateEntityVmUUID_WhenInvoked_ShouldUpdateEntityVmUUID** ()

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

- ServicesTests/VirtualMachineEntityServiceTests.cs

5.131 Models.VirtualMachineInfo Class Reference

Information about a virtual machine.

Properties

- required string **Uuid** [get, set]
Virtual machine UUID.
- required string **Name** [get, set]
Virtual machine name.
- required ulong **UsedMemory** [get, set]
Used memory in bytes.
- required byte **State** [get, set]
Virtual machine state.

5.131.1 Detailed Description

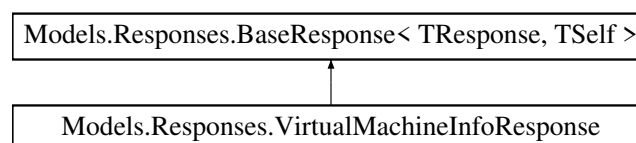
Information about a virtual machine.

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

- Models/VirtualMachineInfo.cs

5.132 Models.Responses.VirtualMachineInfoResponse Class Reference

Inheritance diagram for Models.Responses.VirtualMachineInfoResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

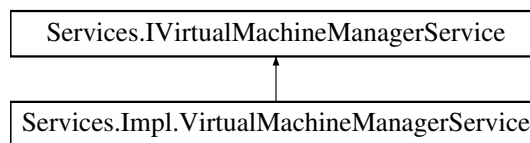
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualMachineInfoResponse.cs

5.133 Services.Impl.VirtualMachineManagerService Class Reference

Inheritance diagram for Services.Impl.VirtualMachineManagerService:



Public Member Functions

- **VirtualMachineManagerService** (IVirtualizationWrapper vmManager, IOptions< [InterconnectConfig](#) > config, [IVirtualMachineConsoleService](#) vmConsoleService)
- [VirtualMachineInfo](#) **CreateVirtualMachine** ([VirtualMachineCreateDefinition](#) definition)
Creates a new virtual machine in the hypervisor.
- List< [VirtualMachineInfo](#) > **GetListOfVirtualMachines** ()
Retrieves a list of all virtual machines from the hypervisor.
- void **AttachVirtualNetworkInterfaceToVirtualMachine** (string name, string networkName, string macAddress)
Attaches a network interface to a virtual machine.

5.133.1 Member Function Documentation

5.133.1.1 AttachVirtualNetworkInterfaceToVirtualMachine()

```

void Services.Impl.VirtualMachineManagerService.AttachVirtualNetworkInterfaceToVirtualMachine
(
    string name,
    string networkName,
    string macAddress) [inline]
  
```

Attaches a network interface to a virtual machine.

Parameters

<i>name</i>	Virtual machine name.
<i>networkName</i>	Virtual network name.
<i>macAddress</i>	Interface MAC address.

Implements [Services.IVirtualMachineManagerService](#).

5.133.1.2 CreateVirtualMachine()

```
VirtualMachineInfo Services.Impl.VirtualMachineManagerService.CreateVirtualMachine (  
    VirtualMachineCreateDefinition definition) [inline]
```

Creates a new virtual machine in the hypervisor.

Parameters

<i>definition</i>	Virtual machine definition to create.
-------------------	---------------------------------------

Returns

Information about the created virtual machine.

Implements [Services.IVirtualMachineManagerService](#).

5.133.1.3 GetListOfVirtualMachines()

```
List< VirtualMachineInfo > Services.Impl.VirtualMachineManagerService.GetListOfVirtualMachines  
( ) [inline]
```

Retrieves a list of all virtual machines from the hypervisor.

Returns

List of virtual machines.

Implements [Services.IVirtualMachineManagerService](#).

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

- Services/Impl/VirtualMachineManagerService.cs

5.134 ServicesTests.VirtualMachineManagerServiceTests Class Reference

Public Member Functions

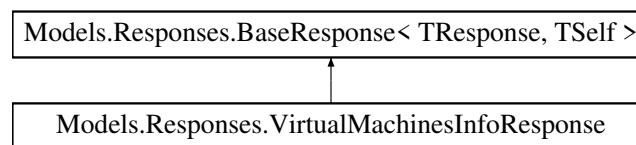
- void **SetUp** ()
- void **CreateVirtualMachine_WhenInvoked_ShouldCreateVirtualMachineUsingNativeLibrary** ()
- void **CreateVirtualMachine_AfterVirtualMachineCreated_ShouldOpenVirtualMachineConsole** ()
- void **GetListOfVirtualMachines_WhenInvoked_ShouldGetListOfVirtualMachinesUsingNativeLibrary** ()
- void **AttachVirtualNetworkInterfaceToVirtualMachine_WhenInvoked_ShouldAttachVirtualNetworkInterfaceToVirtualMachine** ()

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

- ServicesTests/VirtualMachineManagerServiceTests.cs

5.135 Models.Responses.VirtualMachinesInfoResponse Class Reference

Inheritance diagram for Models.Responses.VirtualMachinesInfoResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

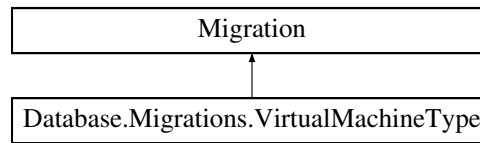
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualMachinesInfoResponse.cs

5.136 Database.Migrations.VirtualMachineType Class Reference

Inheritance diagram for Database.Migrations.VirtualMachineType:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

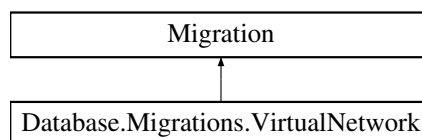
5.136.1 Detailed Description

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

- Database/Migrations/20260117231943_VirtualMachineType.cs
- Database/Migrations/20260117231943_VirtualMachineType.Designer.cs

5.137 Database.Migrations.VirtualNetwork Class Reference

Inheritance diagram for Database.Migrations.VirtualNetwork:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.137.1 Detailed Description

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

- Database/Migrations/20250914132927_VirtualNetwork.cs
- Database/Migrations/20250914132927_VirtualNetwork.Designer.cs

5.138 Models.DTO.VirtualNetworkConnectionDTO Class Reference

Properties

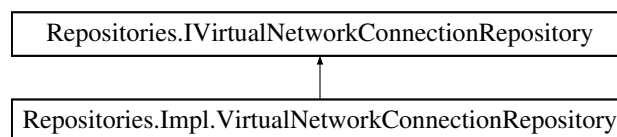
- required int **Id** [get, set]
- required int **SourceEntityId** [get, set]
- required [EntityType](#) **SourceEntityType** [get, set]
- required int **DestinationEntityId** [get, set]
- required [EntityType](#) **DestinationEntityType** [get, set]

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

- Models/DTO/VirtualNetworkConnectionDTO.cs

5.139 Repositories.Impl.VirtualNetworkConnectionRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualNetworkConnectionRepository:



Public Member Functions

- **VirtualNetworkConnectionRepository** ([InterconnectDbContext](#) context)
- async Task< [VirtualNetworkEntityConnectionModel](#) > **Create** (int firstEntityId, [EntityType](#) firstEntityType, int secondEntityId, [EntityType](#) secondEntityType)
Creates a new connection between two entities.
- Task< List< [VirtualNetworkEntityConnectionModel](#) > > **GetAll** ()
Retrieves all connections.
- async Task< [VirtualNetworkEntityConnectionModel](#) > **GetById** (int id)
Retrieves a connection by identifier.
- async Task< List< [VirtualNetworkEntityConnectionModel](#) > > **GetUsingEntityId** (int id, [EntityType](#) type)
Retrieves connections associated with a specific entity.
- async Task **Remove** (int id)
Removes a connection.

5.139.1 Member Function Documentation

5.139.1.1 Create()

```

async Task< VirtualNetworkEntityConnectionModel > Repositories.Impl.VirtualNetworkConnectionRepository.Create (
    int firstEntityId,
    EntityType firstEntityType,
    int secondEntityId,
    EntityType secondEntityType) [inline]
  
```

Creates a new connection between two entities.

Parameters

<i>firstEntityId</i>	First entity identifier.
<i>firstEntityType</i>	First entity type.
<i>secondEntityId</i>	Second entity identifier.
<i>secondEntityType</i>	Second entity type.

Returns

Created connection.

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

5.139.1.2 GetAll()

```
Task< List< VirtualNetworkEntityConnectionModel > > Repositories.Impl.VirtualNetworkConnection↵  
Repository.GetAll () [inline]
```

Retrieves all connections.

Returns

List of connections.

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

5.139.1.3 GetById()

```
async Task< VirtualNetworkEntityConnectionModel > Repositories.Impl.VirtualNetworkConnection↵  
Repository.GetById (   
    int id) [inline]
```

Retrieves a connection by identifier.

Parameters

<i>id</i>	Connection identifier.
-----------	------------------------

Returns

Connection.

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

5.139.1.4 GetUsingEntityId()

```
async Task< List< VirtualNetworkEntityConnectionModel > > Repositories.Impl.VirtualNetwork↵  
ConnectionRepository.GetUsingEntityId (   
    int id,   
    EntityType type) [inline]
```

Retrieves connections associated with a specific entity.

Parameters

<i>id</i>	Entity identifier.
<i>type</i>	Entity type.

Returns

List of connections.

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

5.139.1.5 Remove()

```
async Task Repositories.Impl.VirtualNetworkConnectionRepository.Remove (
    int id) [inline]
```

Removes a connection.

Parameters

<i>id</i>	Connection identifier.
-----------	------------------------

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

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

- Repositories/Impl/VirtualNetworkConnectionRepository.cs

5.140 RepositoriesTests.VirtualNetworkConnectionRepositoryTests

Class Reference

Public Member Functions

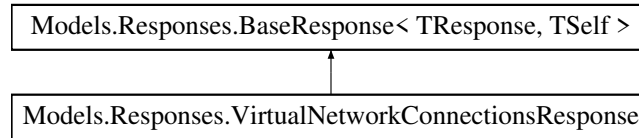
- void **SetUp** ()
- void **TearDown** ()
- async Task **Create_WhenInvoked_ShouldCreateVirtualNetworkConnectionEntity** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllVirtualNetworkConnectionEntities** ()
- async Task **GetUsingEntityId_WhenInvokedWithSourceEntity_ShouldGetVirtualNetworkConnection↵Entity** ()
- async Task **GetUsingEntityId_WhenInvokedWithDestinationEntity_ShouldGetVirtualNetwork↵ConnectionEntity** ()

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

- RepositoriesTests/VirtualNetworkConnectionRepositoryTests.cs

5.141 Models.Responses.VirtualNetworkConnectionsResponse Class Reference

Inheritance diagram for Models.Responses.VirtualNetworkConnectionsResponse:



Additional Inherited Members

Static Public Member Functions inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

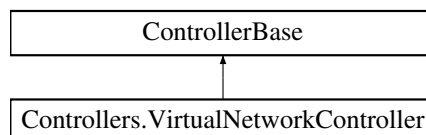
The documentation for this class was generated from the following file:

- Models/Responses/VirtualNetworkConnectionsResponse.cs

5.142 Controllers.VirtualNetworkController Class Reference

Controller managing virtual networks and connections.

Inheritance diagram for Controllers.VirtualNetworkController:



Public Member Functions

- **VirtualNetworkController** ([IVirtualNetworkService](#) virtualNetworkService, [IEntitiesConnectorService](#) entitiesConnectorService, [IEntitiesDisconnectService](#) entitiesDisconnectService)
- async Task< [VirtualNetworkConnectionsResponse](#) > [ConnectEntities](#) ([ConnectEntitiesRequest](#) req)
Connects two entities in the virtual network.
- async Task< [StringResponse](#) > [DisconnectEntities](#) ([VirtualNetworkEntityConnectionRequest](#) req)
Disconnects entities in the virtual network.
- async Task< [VirtualNetworkConnectionsResponse](#) > [GetAllConnections](#) ()
Retrieves all connections in the virtual network.

5.142.1 Detailed Description

Controller managing virtual networks and connections.

5.142.2 Member Function Documentation

5.142.2.1 ConnectEntities()

```
async Task< VirtualNetworkConnectionsResponse > Controllers.VirtualNetworkController.ConnectEntities (
    ConnectEntitiesRequest req) [inline]
```

Connects two entities in the virtual network.

Parameters

<i>req</i>	Connection data.
------------	------------------

Returns

Created connection.

5.142.2.2 DisconnectEntities()

```
async Task< StringResponse > Controllers.VirtualNetworkController.DisconnectEntities (
    VirtualNetworkEntityConnectionRequest req) [inline]
```

Disconnects entities in the virtual network.

Parameters

<i>req</i>	Connection data to remove.
------------	----------------------------

Returns

Operation confirmation.

5.142.2.3 GetAllConnections()

```
async Task< VirtualNetworkConnectionsResponse > Controllers.VirtualNetworkController.GetAllConnections () [inline]
```

Retrieves all connections in the virtual network.

Returns

List of connections.

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

- Controllers/VirtualNetworkController.cs

5.143 Models.VirtualNetworkCreateDefinition Class Reference

Definition for creating a new virtual network.

Properties

- required string **NetworkName** [get, set]
Virtual network name.
- required string **BridgeName** [get, set]
Network bridge name.
- bool? **ForwardNat** [get, set]
Whether to enable NAT for the network.
- string? **MacAddress** [get, set]
Network interface MAC address.
- string? **IpAddress** [get, set]
Network IP address.
- string? **NetMask** [get, set]
Subnet mask.
- string? **DhcpStartRange** [get, set]
DHCP range start address.
- string? **DhcpEndRange** [get, set]
DHCP range end address.

5.143.1 Detailed Description

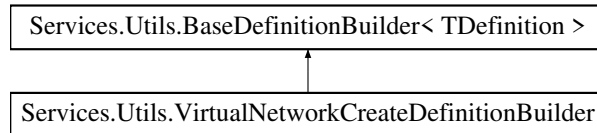
Definition for creating a new virtual network.

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

- Models/VirtualNetworkCreateDefinition.cs

5.144 Services.Utls.VirtualNetworkCreateDefinitionBuilder Class Reference

Inheritance diagram for Services.Utls.VirtualNetworkCreateDefinitionBuilder:



Public Member Functions

- override [VirtualNetworkCreateDefinitionBuilder](#) **SetFromCreateDefinition** ([VirtualNetworkCreateDefinition](#) definition)
- override string **Build** ()

Public Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition >](#) **SetFromCreateDefinition** (TDefinition definition)
- string **Build** ()

Additional Inherited Members

Protected Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- delegate void **BuildingBlock** (XmlWriter writer)
- void **ChecksEverythingIsProvided** (params object?[] args)
- StringWriter **CreateXmlWriter** ()

Protected Attributes inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- XmlWriter

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

- Services/Utls/VirtualNetworkCreateDefinitionBuilder.cs

5.145 ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests Class Reference

Public Member Functions

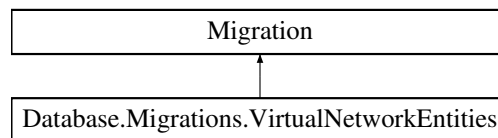
- void **OnSetup** ()
- void **Build_WhenNothingsProvided_ShouldThrowException** ()
- void **Build_WhenInvoked_ShouldIncludeNetworkName** ()
- void **Build_WhenInvoked_ShouldIncludeMacAddress** ()
- void **Build_WhenInvoked_ShouldIncludeIpAddress** ()
- void **Build_WhenInvoked_ShouldIncludeNetmask** ()
- void **Build_WhenInvoked_ShouldIncludeDhcpRange** ()
- void **Build_WhenInvoked_ShouldIncludeBridgeName** ()
- void **Build_WhenInvokedWithDhcpDisabled_ShouldNotIncludeDhcpRange** ()

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

- ServicesTests/Utils/VirtualNetworkDefinitionBuilderTests.cs

5.146 Database.Migrations.VirtualNetworkEntities Class Reference

Inheritance diagram for Database.Migrations.VirtualNetworkEntities:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.146.1 Detailed Description

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

- Database/Migrations/20250906140923_VirtualNetworkEntities.cs
- Database/Migrations/20250906140923_VirtualNetworkEntities.Designer.cs

5.147 Models.Database.VirtualNetworkEntityConnectionModel Class Reference

[Database](#) model representing a connection between entities in a virtual network.

Properties

- int **Id** [get, set]
Connection identifier.
- required int **SourceEntityId** [get, set]
Source entity identifier.
- required [EntityType](#) **SourceEntityType** [get, set]
Source entity type.
- required int **DestinationEntityId** [get, set]
Destination entity identifier.
- required [EntityType](#) **DestinationEntityType** [get, set]
Destination entity type.

5.147.1 Detailed Description

[Database](#) model representing a connection between entities in a virtual network.

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

- Models/Database/VirtualNetworkEntityConnectionModel.cs

5.148 Models.Requests.VirtualNetworkEntityConnectionRequest Class Reference

Properties

- required int **Id** [get, set]

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

- Models/Requests/VirtualNetworkEntityConnectionRequest.cs

5.149 Models.VirtualNetworkInterfaceCreateDefinition Class Reference

Definition for creating a network interface for a virtual machine.

Properties

- required string **MacAddress** [get, set]
Network interface MAC address.
- required string **NetworkName** [get, set]
Virtual network name to connect to.

5.149.1 Detailed Description

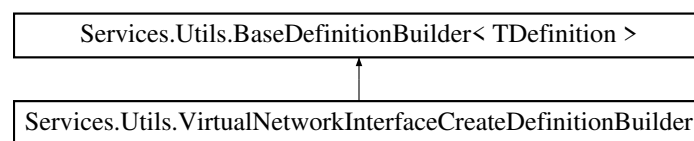
Definition for creating a network interface for a virtual machine.

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

- Models/VirtualNetworkInterfaceCreateDefinition.cs

5.150 Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder Class Reference

Inheritance diagram for Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder:



Public Member Functions

- override [VirtualNetworkInterfaceCreateDefinitionBuilder](#) **SetFromCreateDefinition** ([VirtualNetworkInterfaceCreateDefinition](#) definition)
- [VirtualNetworkInterfaceCreateDefinitionBuilder](#) **SetFromXml** (string xml)
- [VirtualNetworkInterfaceCreateDefinitionBuilder](#) **SetNetworkName** (string networkName)
- override string **Build** ()

Public Member Functions inherited from [Services.Utills.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition >](#) **SetFromCreateDefinition** (TDefinition definition)
- string **Build** ()

Additional Inherited Members

Protected Member Functions inherited from [Services.Utills.BaseDefinitionBuilder< TDefinition >](#)

- delegate void **BuildingBlock** (XmlWriter writer)
- void **ChecksEverythingIsProvided** (params object?[] args)
- StringWriter **CreateXmlWriter** ()

Protected Attributes inherited from [Services.Utills.BaseDefinitionBuilder< TDefinition >](#)

- XmlWriter

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

- Services/Utills/VirtualNetworkInterfaceDefinitionBuilder.cs

5.151 ServicesTests.Utils.VirtualNetworkInterfaceCreateDefinitionBuilderTests Class Reference [↔](#)

Public Member Functions

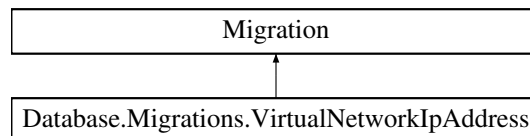
- void **OnSetup** ()
- void **Build_WhenNothingIsProvided_ShouldThrowException** ()
- void **Build_WhenInvoked_ShouldContainNetworkName** ()
- void **Build_WhenInvoked_ShouldContainMacAddress** ()

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

- ServicesTests/Utils/VirtualNetworkInterfaceCreateDefinitionBuilderTests.cs

5.152 Database.Migrations.VirtualNetworkIpAddress Class Reference

Inheritance diagram for Database.Migrations.VirtualNetworkIpAddress:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.152.1 Detailed Description

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

- Database/Migrations/20250927161009_VirtualNetworkIpAddress.cs
- Database/Migrations/20250927161009_VirtualNetworkIpAddress.Designer.cs

5.153 Models.Database.VirtualNetworkModel Class Reference

[Database](#) model representing a virtual network.

Properties

- **int Id** [get, set]
Network identifier.
- **required string BridgeName** [get, set]
Network bridge name.
- **required Guid Uuid** [get, set]
Unique network UUID identifier.
- **string? IpAddress** [get, set]
Network IP address.
- **List< [VirtualNetworkNodeEntityModel](#) > VirtualNetworkNodes** = [] [get, set]
List of virtual network nodes.
- **List< [InternetEntityModel](#) > Internets** = [] [get, set]
List of Internet entities connected to the network.

5.153.1 Detailed Description

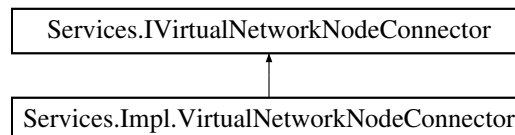
[Database](#) model representing a virtual network.

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

- Models/Database/VirtualNetworkModel.cs

5.154 Services.Impl.VirtualNetworkNodeConnector Class Reference

Inheritance diagram for Services.Impl.VirtualNetworkNodeConnector:



Public Member Functions

- **VirtualNetworkNodeConnector** ([IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeRepository, [IVirtualNetworkConnectionRepository](#) connectionRepository, [IVirtualNetworkRepository](#) networkRepository, [IInternetEntityRepository](#) internetEntityRepository, [IVirtualNetworkService](#) virtualNetworkService, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterfaceRepository)
- **async Task< [VirtualNetworkConnectionDTO](#) > ConnectTwoVirtualNetworkNodes** (int sourceVirtualNetworkNode, int destinationVirtualNetworkNode)
Connects two virtual network nodes.
- **async Task DisconnectTwoVirtualNetworkNodes** (int connectionId, int firstVirtualNetworkNodeId, int secondVirtualNetworkNode)
Disconnects two virtual network nodes.
- **async Task< [VirtualNetworkConnectionDTO](#) > ConnectVirtualNetworkNodeToInternet** (int virtualNetworkNodeId, int internetId)
Connects a virtual network node to the Internet.
- **async Task DisconnectVirtualNetworkNodeFromInternet** (int connectionId, int virtualNetworkNodeId)
Disconnects a virtual network node from the Internet.

5.154.1 Member Function Documentation

5.154.1.1 ConnectTwoVirtualNetworkNodes()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.VirtualNetworkNodeConnector.Connect↔  
TwoVirtualNetworkNodes (  
    int sourceVirtualNetworkNode,  
    int destinationVirtualNetworkNode) [inline]
```

Connects two virtual network nodes.

Parameters

<i>sourceVirtualNetworkNode</i>	First node identifier.
<i>destinationVirtualNetworkNode</i>	Second node identifier.

Returns

Created connection data.

Implements [Services.IVirtualNetworkNodeConnector](#).

5.154.1.2 ConnectVirtualNetworkNodeToInternet()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.VirtualNetworkNodeConnector.Connect↔  
VirtualNetworkNodeToInternet (  
    int virtualNetworkNodeId,  
    int internetId) [inline]
```

Connects a virtual network node to the Internet.

Parameters

<i>virtualNetwork↔ NodeId</i>	Network node identifier.
<i>internetId</i>	Internet entity identifier.

Returns

Created connection data.

Implements [Services.IVirtualNetworkNodeConnector](#).

5.154.1.3 DisconnectTwoVirtualNetworkNodes()

```
async Task Services.Impl.VirtualNetworkNodeConnector.DisconnectTwoVirtualNetworkNodes (  
    int connectionId,  
    int firstVirtualNetworkNodeId,  
    int secondVirtualNetworkNode) [inline]
```

Disconnects two virtual network nodes.

Parameters

<i>connectionId</i>	Connection identifier.
<i>firstVirtualNetworkNodeId</i>	First node identifier.
<i>secondVirtualNetworkNode</i>	Second node identifier.

Implements [Services.IVirtualNetworkNodeConnector](#).

5.154.1.4 DisconnectVirtualNetworkNodeFromInternet()

```
async Task Services.Impl.VirtualNetworkNodeConnector.DisconnectVirtualNetworkNodeFromInternet
(
    int connectionId,
    int virtualNetworkNodeId) [inline]
```

Disconnects a virtual network node from the Internet.

Parameters

<i>connectionId</i>	Connection identifier.
<i>virtualNetworkNodeId</i>	Network node identifier.

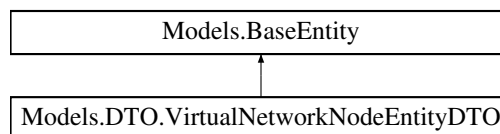
Implements [Services.IVirtualNetworkNodeConnector](#).

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

- Services/Impl/VirtualNetworkNodeConnector.cs

5.155 Models.DTO.VirtualNetworkNodeEntityDTO Class Reference

Inheritance diagram for Models.DTO.VirtualNetworkNodeEntityDTO:



Properties

- string? **Name** [get, set]
- Guid **Uuid** [get, set]

Properties inherited from [Models.BaseEntity](#)

- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

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

- Models/DTO/VirtualNetworkNodeEntityDTO.cs

5.156 Models.Database.VirtualNetworkNodeEntityModel Class Reference

[Database](#) model representing a virtual network node (switch/router).

Properties

- int **Id** [get, set]
Node identifier.
- string? **Name** [get, set]
Network node name.
- required bool **Visible** [get, set]
Determines whether the node is visible on the board.
- int **X** [get, set]
X coordinate of the node position.
- int **Y** [get, set]
Y coordinate of the node position.
- required [VirtualNetworkModel](#) **VirtualNetwork** [get, set]
Virtual network to which the node belongs.

5.156.1 Detailed Description

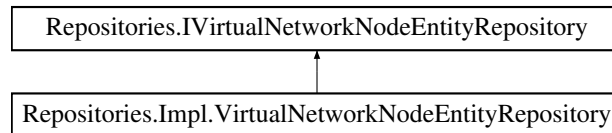
[Database](#) model representing a virtual network node (switch/router).

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

- Models/Database/VirtualNetworkNodeEntityModel.cs

5.157 Repositories.Impl.VirtualNetworkNodeEntityRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualNetworkNodeEntityRepository:



Public Member Functions

- **VirtualNetworkNodeEntityRepository** ([InterconnectDbContext](#) context)
- async Task< [VirtualNetworkNodeEntityModel](#) > [Create](#) (string name, [VirtualNetworkModel](#) virtualNetwork)
Creates a new visible virtual network node.
- async Task< [VirtualNetworkNodeEntityModel](#) > [CreateInvisible](#) ([VirtualNetworkModel](#) virtualNetwork)
Creates a new invisible virtual network node.
- Task< List< [VirtualNetworkNodeEntityModel](#) > > [GetAll](#) ()
Retrieves all virtual network nodes.
- Task< [VirtualNetworkNodeEntityModel](#) > [GetById](#) (int id)
Retrieves a virtual network node by identifier.
- Task< List< [VirtualNetworkNodeEntityModel](#) > > [GetVisible](#) ()
Retrieves visible virtual network nodes.
- async Task [Remove](#) (int id)
Removes a virtual network node.
- async Task< [VirtualNetworkNodeEntityModel](#) > [UpdateEntityPosition](#) (int id, int x, int y)
Updates the position of a virtual network node.
- async Task< [VirtualNetworkNodeEntityModel](#) > [UpdateNetwork](#) (int id, [VirtualNetworkModel](#) virtualNetwork)
Updates the virtual network for a node.

5.157.1 Member Function Documentation

5.157.1.1 Create()

```

async Task< VirtualNetworkNodeEntityModel > Repositories.Impl.VirtualNetworkNodeEntityRepository.Create (
    string name,
    VirtualNetworkModel virtualNetwork) [inline]
  
```

Creates a new visible virtual network node.

Parameters

<i>name</i>	Node name.
<i>virtualNetwork</i>	Virtual network.

Returns

Created node.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.2 CreateInvisible()

```
async Task< VirtualNetworkNodeEntityModel > Repositories.Impl.VirtualNetworkNodeEntityRepository.CreateInvisible (
    VirtualNetworkModel virtualNetwork) [inline]
```

Creates a new invisible virtual network node.

Parameters

<i>virtualNetwork</i>	Virtual network.
-----------------------	------------------

Returns

Created node.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.3 GetAll()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.Impl.VirtualNetworkNodeEntityRepository.GetAll () [inline]
```

Retrieves all virtual network nodes.

Returns

List of nodes.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.4 GetById()

```
Task< VirtualNetworkNodeEntityModel > Repositories.Impl.VirtualNetworkNodeEntityRepository.GetById (
    int id) [inline]
```

Retrieves a virtual network node by identifier.

Parameters

<i>id</i>	Node identifier.
-----------	------------------

Returns

Network node.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.5 GetVisible()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.Impl.VirtualNetworkNodeEntityRepository.GetVisible () [inline]
```

Retrieves visible virtual network nodes.

Returns

List of visible nodes.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.6 Remove()

```
async Task Repositories.Impl.VirtualNetworkNodeEntityRepository.Remove (
    int id) [inline]
```

Removes a virtual network node.

Parameters

<i>id</i>	Node identifier.
-----------	------------------

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.7 UpdateEntityPosition()

```
async Task< VirtualNetworkNodeEntityModel > Repositories.Impl.VirtualNetworkNodeEntityRepository.UpdateEntityPosition (
    int id,
    int x,
    int y) [inline]
```

Updates the position of a virtual network node.

Parameters

<i>id</i>	Node identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated node.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.157.1.8 UpdateNetwork()

```
async Task< VirtualNetworkNodeEntityModel > Repositories.Impl.VirtualNetworkNodeEntityRepository.UpdateNetwork (
    int id,
    VirtualNetworkModel virtualNetwork) [inline]
```

Updates the virtual network for a node.

Parameters

<i>id</i>	Node identifier.
<i>virtualNetwork</i>	New virtual network.

Returns

Updated node.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

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

- `Repositories/Impl/VirtualNetworkNodeEntityRepository.cs`

5.158 RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests Class Reference

Public Member Functions

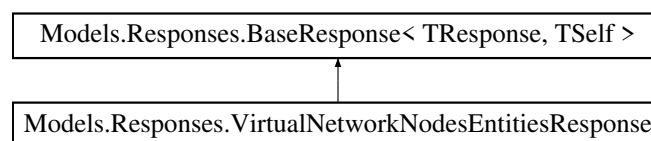
- void **SetUp** ()
- void **Teardown** ()
- async Task **Create_WhenInvoked_ShouldCreateVisibleVirtualNetworkNode** ()
- async Task **Create_WhenInvoked_ShouldCreateInvisibleVirtualNetworkNode** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllVirtualNetworkNodes** ()
- async Task **GetById_WhenInvokedWithCorrectId_ShouldGetVirtualNetworkNodeById** ()
- async Task **UpdateEntityPosition_WhenInvoked_ShouldUpdateEntityPosition** ()

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

- `RepositoriesTests/VirtualNetworkNodeEntityRepositoryTests.cs`

5.159 Models.Responses.VirtualNetworkNodesEntitiesResponse Class Reference

Inheritance diagram for Models.Responses.VirtualNetworkNodesEntitiesResponse:



Additional Inherited Members

Static Public Member Functions inherited from

[Models.Responses.BaseResponse< TResponse, TSelf >](#)

- static TSelf [WithSuccess](#) (TResponse data)
Creates a successful response with data.
- static TSelf [WithEmptySuccess](#) ()
Creates an empty successful response.

Properties inherited from [Models.Responses.BaseResponse< TResponse, TSelf >](#)

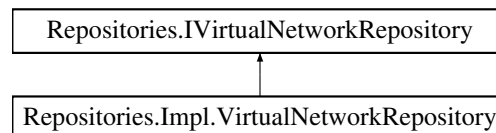
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualNetworkNodesEntitiesResponse.cs

5.160 Repositories.Impl.VirtualNetworkRepository Class Reference

Inheritance diagram for Repositories.Impl.VirtualNetworkRepository:



Public Member Functions

- **VirtualNetworkRepository** ([InterconnectDbContext](#) context)
- async Task< [VirtualNetworkModel](#) > [Create](#) (string bridgeName, Guid uuid, string? ipAddress=null)
Creates a new virtual network.
- Task< List< [VirtualNetworkModel](#) > > [GetAll](#) ()
Retrieves all virtual networks.
- Task< [VirtualNetworkModel](#) > [GetByBridgeName](#) (string bridgeName)
Retrieves a virtual network by bridge name.
- Task< [VirtualNetworkModel](#) > [GetById](#) (int id)
Retrieves a virtual network by identifier.
- Task< [VirtualNetworkModel](#) > [GetByUuid](#) (Guid uuid)
Retrieves a virtual network by UUID.
- Task< [VirtualNetworkModel](#) > [GetByUuidWithVirtualNetworkNodes](#) (Guid uuid)
Retrieves a virtual network with nodes by UUID.
- async Task [Remove](#) (int id)
Removes a virtual network.

5.160.1 Member Function Documentation

5.160.1.1 Create()

```
async Task< VirtualNetworkModel > Repositories.Impl.VirtualNetworkRepository.Create (
    string bridgeName,
    Guid uuid,
    string? ipAddress = null) [inline]
```

Creates a new virtual network.

Parameters

<i>bridgeName</i>	Network bridge name.
<i>uuid</i>	Network UUID.
<i>ipAddress</i>	Network IP address (optional).

Returns

Created virtual network.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.2 GetAll()

```
Task< List< VirtualNetworkModel > > Repositories.Impl.VirtualNetworkRepository.GetAll ()
[inline]
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.3 GetByBridgeName()

```
Task< VirtualNetworkModel > Repositories.Impl.VirtualNetworkRepository.GetByBridgeName (
    string bridgeName) [inline]
```

Retrieves a virtual network by bridge name.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

Returns

Virtual network.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.4 GetById()

```
Task< VirtualNetworkModel > Repositories.Impl.VirtualNetworkRepository.GetById (
    int id) [inline]
```

Retrieves a virtual network by identifier.

Parameters

<i>id</i>	Network identifier.
-----------	---------------------

Returns

Virtual network.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.5 GetByUuid()

```
Task< VirtualNetworkModel > Repositories.Impl.VirtualNetworkRepository.GetByUuid (  
    Guid uuid) [inline]
```

Retrieves a virtual network by UUID.

Parameters

<i>uuid</i>	Network UUID.
-------------	---------------

Returns

Virtual network.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.6 GetByUuidWithVirtualNetworkNodes()

```
Task< VirtualNetworkModel > Repositories.Impl.VirtualNetworkRepository.GetByUuidWithVirtual↔  
NetworkNodes (  
    Guid uuid) [inline]
```

Retrieves a virtual network with nodes by UUID.

Parameters

<i>uuid</i>	Network UUID.
-------------	---------------

Returns

Virtual network with nodes.

Implements [Repositories.IVirtualNetworkRepository](#).

5.160.1.7 Remove()

```
async Task Repositories.Impl.VirtualNetworkRepository.Remove (  
    int id) [inline]
```

Removes a virtual network.

Parameters

<i>id</i>	Network identifier.
-----------	---------------------

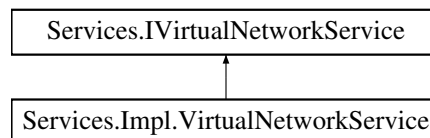
Implements [Repositories.IVirtualNetworkRepository](#).

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

- `Repositories/Impl/VirtualNetworkRepository.cs`

5.161 Services.Impl.VirtualNetworkService Class Reference

Inheritance diagram for `Services.Impl.VirtualNetworkService`:



Public Member Functions

- **VirtualNetworkService** ([IVirtualizationWrapper](#) wrapper, [IVirtualMachineEntityRepository](#) vmEntityRepository, [IVirtualNetworkConnectionRepository](#) connectionRepository, [IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeRepository, [IVirtualNetworkRepository](#) networkRepository, [IPacketSnifferService](#) packetSnifferService, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterfaceRepository)
- `async Task< List< VirtualNetworkConnectionDTO > > GetVirtualNetworkConnections ()`
Retrieves a list of all connections in the virtual network.
- `async Task< VirtualNetworkNodeEntityDTO > CreateVirtualNetworkNode (string? name)`
Creates a new virtual network node.
- `async Task< List< VirtualNetworkNodeEntityDTO > > GetVisibleVirtualNetworkNodeEntities ()`
Retrieves a list of visible virtual network nodes.
- `async Task< VirtualNetworkNodeEntityDTO > UpdateVirtualNetworkNodeEntityPosition (int entityId, int x, int y)`
Updates the position of a virtual network node.
- `async Task AttachNetworkInterfaceToVirtualMachine (int vmId, int connectionId, VirtualNetworkInterfaceCreateDefinition interfaceDefinition)`
Attaches a network interface to a virtual machine.
- `async Task UpdateNetworkForVirtualMachineNetworkInterface (int vmId, int connectionId, string networkName)`
Updates the network for a virtual machine network interface.
- `async Task< VirtualNetworkModel > CreateNodeVirtualNetwork ()`
Creates a virtual network for a network node.
- `async Task< VirtualNetworkModel > CreateInternetVirtualNetwork ()`
Creates a virtual network for Internet connection.
- `Task< List< VirtualNetworkModel > > GetAllVirtualNetworks ()`
Retrieves all virtual networks.
- `Task< VirtualNetworkModel > GetVirtualNetworkUsingBridgeName (string bridgeName)`
Retrieves a virtual network based on bridge name.

5.161.1 Member Function Documentation

5.161.1.1 AttachNetworkInterfaceToVirtualMachine()

```
async Task Services.Impl.VirtualNetworkService.AttachNetworkInterfaceToVirtualMachine (
    int vmId,
    int connectionId,
    VirtualNetworkInterfaceCreateDefinition interfaceDefinition) [inline]
```

Attaches a network interface to a virtual machine.

Parameters

<i>vmId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.
<i>interfaceDefinition</i>	Network interface definition.

Implements [Services.IVirtualNetworkService](#).

5.161.1.2 CreateInternetVirtualNetwork()

```
async Task< VirtualNetworkModel > Services.Impl.VirtualNetworkService.CreateInternetVirtual↔
Network () [inline]
```

Creates a virtual network for Internet connection.

Returns

Created virtual network.

Implements [Services.IVirtualNetworkService](#).

5.161.1.3 CreateNodeVirtualNetwork()

```
async Task< VirtualNetworkModel > Services.Impl.VirtualNetworkService.CreateNodeVirtualNetwork
() [inline]
```

Creates a virtual network for a network node.

Returns

Created virtual network.

Implements [Services.IVirtualNetworkService](#).

5.161.1.4 CreateVirtualNetworkNode()

```
async Task< VirtualNetworkNodeEntityDTO > Services.Impl.VirtualNetworkService.CreateVirtual↔
NetworkNode (
    string? name) [inline]
```

Creates a new virtual network node.

Parameters

<i>name</i>	Node name.
-------------	------------

Returns

Created network node.

Implements [Services.IVirtualNetworkService](#).

5.161.1.5 GetAllVirtualNetworks()

```
Task< List< VirtualNetworkModel > > Services.Impl.VirtualNetworkService.GetAllVirtualNetworks  
( ) [inline]
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implements [Services.IVirtualNetworkService](#).

5.161.1.6 GetVirtualNetworkConnections()

```
async Task< List< VirtualNetworkConnectionDTO > > Services.Impl.VirtualNetworkService.GetVirtualNetworkConnections ( ) [inline]
```

Retrieves a list of all connections in the virtual network.

Returns

List of connections.

Implements [Services.IVirtualNetworkService](#).

5.161.1.7 GetVirtualNetworkUsingBridgeName()

```
Task< VirtualNetworkModel > Services.Impl.VirtualNetworkService.GetVirtualNetworkUsingBridgeName (   
    string bridgeName) [inline]
```

Retrieves a virtual network based on bridge name.

Parameters

<i>bridgeName</i>	Network bridge name.
-------------------	----------------------

Returns

Virtual network.

Implements [Services.IVirtualNetworkService](#).

5.161.1.8 GetVisibleVirtualNetworkNodeEntities()

```
async Task< List< VirtualNetworkNodeEntityDTO > > Services.Impl.VirtualNetworkService.Get↔  
VisibleVirtualNetworkNodeEntities () [inline]
```

Retrieves a list of visible virtual network nodes.

Returns

List of visible nodes.

Implements [Services.IVirtualNetworkService](#).

5.161.1.9 UpdateNetworkForVirtualMachineNetworkInterface()

```
async Task Services.Impl.VirtualNetworkService.UpdateNetworkForVirtualMachineNetworkInterface  
(  
    int vmId,  
    int connectionId,  
    string networkName) [inline]
```

Updates the network for a virtual machine network interface.

Parameters

<i>vmId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.
<i>networkName</i>	New network name.

Implements [Services.IVirtualNetworkService](#).

5.161.1.10 UpdateVirtualNetworkNodeEntityPosition()

```
async Task< VirtualNetworkNodeEntityDTO > Services.Impl.VirtualNetworkService.UpdateVirtual↔  
NetworkNodeEntityPosition (  
    int entityId,  
    int x,  
    int y) [inline]
```

Updates the position of a virtual network node.

Parameters

<i>entity↔ Id</i>	Node identifier.
<i>x</i>	New X coordinate.
<i>y</i>	New Y coordinate.

Returns

Updated node.

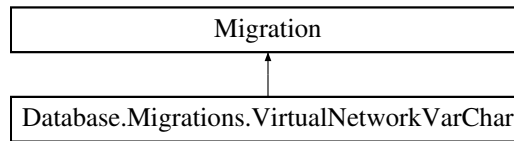
Implements [Services.IVirtualNetworkService](#).

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

- Services/Impl/VirtualNetworkService.cs

5.162 Database.Migrations.VirtualNetworkVarChar Class Reference

Inheritance diagram for Database.Migrations.VirtualNetworkVarChar:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

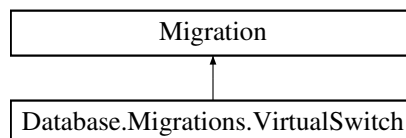
5.162.1 Detailed Description

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

- Database/Migrations/20250927161142_VirtualNetworkVarChar.cs
- Database/Migrations/20250927161142_VirtualNetworkVarChar.Designer.cs

5.163 Database.Migrations.VirtualSwitch Class Reference

Inheritance diagram for Database.Migrations.VirtualSwitch:



Protected Member Functions

- override void **Up** (MigrationBuilder migrationBuilder)
- override void **Down** (MigrationBuilder migrationBuilder)
- override void **BuildTargetModel** (ModelBuilder modelBuilder)

5.163.1 Detailed Description

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

- Database/Migrations/20250906192140_VirtualSwitch.cs
- Database/Migrations/20250906192140_VirtualSwitch.Designer.cs

Index

Add
Repositories.Impl.VirtualMachineConsoleStreamRepository, 144
Repositories.Impl.VirtualMachineEntityRepository, 155
Repositories.IVirtualMachineConsoleStreamRepository, 87
Repositories.IVirtualMachineEntityRepository, 93
AddDataToConsole
Repositories.Impl.VirtualMachineConsoleDataRepository, 138
Repositories.IVirtualMachineConsoleDataRepository, 84
AddSniffer
Repositories.Impl.PacketSnifferRepository, 120
Repositories.IPacketSnifferRepository, 73
ALIVE
Models.Enums, 17
AttachDeviceToVirtualMachine
NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 132
NativeLibrary.Wrappers.IVirtualizationWrapper, 79
AttachNetworkInterfaceToVirtualMachine
Services.Impl.VirtualNetworkService, 189
Services.IVirtualNetworkService, 111
AttachVirtualNetworkInterfaceToVirtualMachine
Services.Impl.VirtualMachineManagerService, 162
Services.IVirtualMachineManagerService, 98
BackgroundServices, 13
BackgroundServices.Impl, 13
BackgroundServices.Impl.PacketSnifferBackgroundService, 118
ExecuteAsync, 119
PacketSnifferBackgroundService, 119
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, 136
ExecuteAsync, 137
VirtualMachineConsoleBackgroundService, 137
Booted
Models.Enums, 19
CloseStream
NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 132
NativeLibrary.Wrappers.IVirtualizationWrapper, 79
Services.Impl.VirtualMachineConsoleService, 141
Services.IVirtualMachineConsoleService, 85
ConnectEntities
Controllers.VirtualNetworkController, 170
ConnectionInfo
Controllers.HypervisorConnectionController, 48
ConnectionStatus
Models.Enums, 17
ConnectTwoEntities
Services.IEntitiesConnectorService, 53
Services.Impl.EntitiesConnectorService, 39
ConnectTwoVirtualMachines
Services.IEntitiesConnectorService, 53
Services.Impl.EntitiesConnectorService, 40
ConnectTwoVirtualNetworkNodes
Services.Impl.VirtualNetworkNodeConnector, 178
Services.IVirtualNetworkNodeConnector, 102
ConnectVirtualMachineToInternet
Services.IEntitiesConnectorService, 53
Services.Impl.EntitiesConnectorService, 40
ConnectVirtualMachineToVirtualNetworkNode
Services.IEntitiesConnectorService, 54
Services.Impl.EntitiesConnectorService, 40
ConnectVirtualNetworkNodeToInternet
Services.Impl.VirtualNetworkNodeConnector, 178
Services.IVirtualNetworkNodeConnector, 102
Controllers, 13
Controllers.EntityController, 43
CreateInternetEntity, 44
CreateVirtualMachineEntity, 44
CreateVirtualNetworkNodeEntity, 44
UpdateEntityPosition, 45
Controllers.HypervisorConnectionController, 47
ConnectionInfo, 48
Controllers.InfoController, 62
GetSystemInfo, 63
Controllers.VirtualMachineController, 146
GetAvailableBootableDisks, 146
Controllers.VirtualNetworkController, 169
ConnectEntities, 170
DisconnectEntities, 170
GetAllConnections, 170
ControllersTests, 14
ControllersTests.HypervisorConnectionControllerTests, 48
ControllersTests.InfoControllerTests, 63
Create
Repositories.IInternetEntityRepository, 59
Repositories.Impl.InternetEntityRepository, 70
Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 153
Repositories.Impl.VirtualNetworkConnectionRepository, 166

- Repositories.Impl.VirtualNetworkNodeEntityRepository, 181
- Repositories.Impl.VirtualNetworkRepository, 186
- Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 89
- Repositories.IVirtualNetworkConnectionRepository, 100
- Repositories.IVirtualNetworkNodeEntityRepository, 104
- Repositories.IVirtualNetworkRepository, 108
- CreateEntity
 - Services.Impl.VirtualMachineEntityService, 158
 - Services.IVirtualMachineEntityService, 95
- CreateInternet
 - Services.IInternetEntityService, 61
 - Services.Impl.InternetEntityService, 72
- CreateInternetEntity
 - Controllers.EntityController, 44
- CreateInternetVirtualNetwork
 - Services.Impl.VirtualNetworkService, 189
 - Services.IVirtualNetworkService, 111
- CreateInvisible
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 181
 - Repositories.IVirtualNetworkNodeEntityRepository, 105
- CreateNodeVirtualNetwork
 - Services.Impl.VirtualNetworkService, 189
 - Services.IVirtualNetworkService, 112
- CreateVirtualMachine
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 132
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 79
 - Services.Impl.VirtualMachineManagerService, 163
 - Services.IVirtualMachineManagerService, 98
- CreateVirtualMachineEntity
 - Controllers.EntityController, 44
- CreateVirtualNetwork
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 133
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 80
- CreateVirtualNetworkNode
 - Services.Impl.VirtualNetworkService, 189
 - Services.IVirtualNetworkService, 112
- CreateVirtualNetworkNodeEntity
 - Controllers.EntityController, 44
- Database, 14
- Database.InterconnectDbContext, 66
- Database.Migrations, 14
- Database.Migrations.AppDbContextModelSnapshot, 25
- Database.Migrations.BootableDiskModel, 29
- Database.Migrations.ChangesToConnection, 34
- Database.Migrations.ChangeTypeOfVMEntityStrings, 34
- Database.Migrations.DeviceDefinition, 38
- Database.Migrations.InitialCreate, 65
- Database.Migrations.InternetEntity, 68
- Database.Migrations.ModificationToVirtualNetwork, 114
- Database.Migrations.RemoveOsType, 126
- Database.Migrations.RenameSwitchToNode, 126
- Database.Migrations.RenameUuidToVmUuid, 127
- Database.Migrations.UpdateBootableDiskModelPath, 130
- Database.Migrations.UpdateOperatingSystemType, 131
- Database.Migrations.VirtualMachineEntityInterfaces, 150
- Database.Migrations.VirtualMachineType, 165
- Database.Migrations.VirtualNetwork, 165
- Database.Migrations.VirtualNetworkEntities, 173
- Database.Migrations.VirtualNetworkIpAddress, 176
- Database.Migrations.VirtualNetworkVarChar, 192
- Database.Migrations.VirtualSwitch, 192
- DEAD
 - Models.Enums, 17
- DestroyNetwork
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 133
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 80
- DetachDeviceFromVirtualMachine
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 133
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 80
- DisconnectEntities
 - Controllers.VirtualNetworkController, 170
 - Services.IEntitiesDisconnectorService, 55
 - Services.Impl.EntitiesDisconnectorService, 42
- DisconnectTwoVirtualNetworkNodes
 - Services.Impl.VirtualNetworkNodeConnector, 178
 - Services.IVirtualNetworkNodeConnector, 103
- DisconnectVirtualMachineFromInternet
 - Services.IEntitiesDisconnectorService, 55
 - Services.Impl.EntitiesDisconnectorService, 42
- DisconnectVirtualMachineFromVirtualMachine
 - Services.IEntitiesDisconnectorService, 55
 - Services.Impl.EntitiesDisconnectorService, 42
- DisconnectVirtualMachineFromVirtualNetworkNode
 - Services.IEntitiesDisconnectorService, 56
 - Services.Impl.EntitiesDisconnectorService, 43
- DisconnectVirtualNetworkNodeFromInternet
 - Services.Impl.VirtualNetworkNodeConnector, 179
 - Services.IVirtualNetworkNodeConnector, 103
- EntityType
 - Models.Enums, 17
- ExecuteAsync
 - BackgroundServices.Impl.PacketSnifferBackgroundService, 119
 - BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, 137
- GetAll
 - Repositories.IInternetEntityRepository, 59
 - Repositories.Impl.InternetEntityRepository, 70
 - Repositories.Impl.VirtualMachineEntityRepository, 155
 - Repositories.Impl.VirtualNetworkConnectionRepository, 167

- Repositories.Impl.VirtualNetworkNodeEntityRepository, 182
- Repositories.Impl.VirtualNetworkRepository, 186
- Repositories.IVirtualMachineEntityRepository, 93
- Repositories.IVirtualNetworkConnectionRepository, 100
- Repositories.IVirtualNetworkNodeEntityRepository, 105
- Repositories.IVirtualNetworkRepository, 108
- GetAllConnections
 - Controllers.VirtualNetworkController, 170
- GetAllStreams
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, 144
 - Repositories.IVirtualMachineConsoleStreamRepository, 88
- GetAllVirtualNetworks
 - Services.Impl.VirtualNetworkService, 190
 - Services.IVirtualNetworkService, 112
- GetAvailableBootableDiskModels
 - Services.IBootableDiskProviderService, 50
 - Services.Impl.BootableDiskProviderService, 30
- GetAvailableBootableDisks
 - Controllers.VirtualMachineController, 146
- GetBootableDiskPathById
 - Services.IBootableDiskProviderService, 50
 - Services.Impl.BootableDiskProviderService, 30
- GetByBridgeName
 - Repositories.Impl.VirtualNetworkRepository, 186
 - Repositories.IVirtualNetworkRepository, 108
- GetById
 - Repositories.IBootableDiskRepository, 51
 - Repositories.IInternetEntityRepository, 60
 - Repositories.Impl.BootableDiskRepository, 31
 - Repositories.Impl.IInternetEntityRepository, 70
 - Repositories.Impl.VirtualMachineEntityRepository, 156
 - Repositories.Impl.VirtualNetworkConnectionRepository, 167
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 182
 - Repositories.Impl.VirtualNetworkRepository, 186
 - Repositories.IVirtualMachineEntityRepository, 93
 - Repositories.IVirtualNetworkConnectionRepository, 100
 - Repositories.IVirtualNetworkNodeEntityRepository, 105
 - Repositories.IVirtualNetworkRepository, 109
 - Services.Impl.VirtualMachineEntityService, 159
 - Services.IVirtualMachineEntityService, 96
- GetByIds
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 153
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 89
- GetByNetworkName
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 153
- Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 91
- GetBytesFromConsole
 - Services.Impl.VirtualMachineConsoleService, 142
 - Services.IVirtualMachineConsoleService, 85
- GetByUuid
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, 144
 - Repositories.Impl.VirtualNetworkRepository, 187
 - Repositories.IVirtualMachineConsoleStreamRepository, 88
 - Repositories.IVirtualNetworkRepository, 109
- GetByUuidWithVirtualNetworkNodes
 - Repositories.Impl.VirtualNetworkRepository, 187
 - Repositories.IVirtualNetworkRepository, 109
- GetByVirtualMachineId
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 154
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 91
- GetCapturedPacket
 - Services.Impl.PacketSnifferService, 122
 - Services.IPacketSnifferService, 75
- GetConnectionInfo
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 134
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 80
 - Services.IHypervisorConnectionService, 57
 - Services.Impl.HypervisorConnectionService, 49
- GetConnectionStatus
 - Services.IHypervisorConnectionService, 57
 - Services.Impl.HypervisorConnectionService, 49
- GetData
 - Repositories.Impl.VirtualMachineConsoleDataRepository, 138
 - Repositories.IVirtualMachineConsoleDataRepository, 84
- GetDataFromStream
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 134
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 81
- GetEntities
 - Services.Impl.VirtualMachineEntityService, 159
 - Services.IVirtualMachineEntityService, 96
- GetInitialConsoleData
 - Services.Impl.VirtualMachineConsoleService, 142
 - Services.IVirtualMachineConsoleService, 85
- GetInitialDataForConsole
 - Hubs.VirtualMachineConsoleHub, 139
- GetInternetEntities
 - Services.IInternetEntityService, 61
 - Services.Impl.IInternetEntityService, 72
- GetListOfVirtualMachines
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 134
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 81
 - Services.Impl.VirtualMachineManagerService, 163
 - Services.IVirtualMachineManagerService, 99

- GetMacAddresses
 - Services.Impl.VirtualMachineEntityService, 159
 - Services.IVirtualMachineEntityService, 96
- GetNumberOfPackets
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, 124
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 77
- GetOnlyWithNotNullablePath
 - Repositories.IBootableDiskRepository, 52
 - Repositories.Impl.BootableDiskRepository, 32
- GetOpenedSniffers
 - Repositories.Impl.PacketSnifferRepository, 121
 - Repositories.IPacketSnifferRepository, 74
 - Services.Impl.PacketSnifferService, 122
 - Services.IPacketSnifferService, 75
- GetPacket
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, 124
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 77
- GetStreams
 - Services.Impl.VirtualMachineConsoleService, 142
 - Services.IVirtualMachineConsoleService, 86
- GetSystemInfo
 - Controllers.InfoController, 63
 - Services.IInfoService, 58
 - Services.Impl.InfoService, 64
- GetUsingEntityId
 - Repositories.Impl.VirtualNetworkConnectionRepository, 167
 - Repositories.IVirtualNetworkConnectionRepository, 101
- GetVirtualMachineInfo
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 134
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 81
- GetVirtualMachineNameWithPrefix
 - Models.VirtualMachineCreateDefinition, 147
- GetVirtualNetworkConnections
 - Services.Impl.VirtualNetworkService, 190
 - Services.IVirtualNetworkService, 112
- GetVirtualNetworkUsingBridgeName
 - Services.Impl.VirtualNetworkService, 190
 - Services.IVirtualNetworkService, 113
- GetVisible
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 182
 - Repositories.IVirtualNetworkNodeEntityRepository, 106
- GetVisibleVirtualNetworkNodeEntities
 - Services.Impl.VirtualNetworkService, 190
 - Services.IVirtualNetworkService, 113
- Hubs, 14
- Hubs.ConnectionStatusHub, 37
 - Ping, 37
- Hubs.PacketSnifferHub, 119
- Hubs.VirtualMachineConsoleHub, 139
 - GetInitialDataForConsole, 139
 - JoinConsoleGroup, 140
 - LeaveConsoleGroup, 140
 - SendDataToConsole, 140
- HubsTests, 15
 - HubsTests.ConnectionStatusHubTests, 37
 - HubsTests.VirtualMachineConsoleHubTests, 141
- InitializeConnection
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 135
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 82
 - Services.IHypervisorConnectionService, 57
 - Services.Impl.HypervisorConnectionService, 49
- Interconnect, 15
 - Interconnect.Middlewares, 15
 - Interconnect.Middlewares.ExceptionHandlerMiddleware, 45
 - InvokeAsync, 46
 - Interconnect.Program, 125
 - Main, 125
 - InterconnectTests, 15
 - InterconnectTests.ExceptionHandlerMiddlewareTests, 46
- Internet
 - Models.Enums, 17
- InvokeAsync
 - Interconnect.Middlewares.ExceptionHandlerMiddleware, 46
- IsConnectionAlive
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 135
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 82
- JoinConsoleGroup
 - Hubs.VirtualMachineConsoleHub, 140
- LeaveConsoleGroup
 - Hubs.VirtualMachineConsoleHub, 140
- ListenForPacket
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 77
 - Services.Impl.PacketSnifferService, 122
 - Services.IPacketSnifferService, 75
- Main
 - Interconnect.Program, 125
- Mappers, 15
- Models, 16
 - Models.BaseEntity, 26
 - Models.Config, 16
 - Models.Config.InterconnectConfig, 66
 - Models.ConnectionInfo, 35
 - Models.Database, 16
 - Models.Database.BootableDiskModel, 29
 - Models.Database.InternetEntityModel, 68
 - Models.Database.VirtualMachineEntityModel, 151
 - Models.Database.VirtualMachineEntityNetworkInterfaceModel, 152
 - Models.Database.VirtualNetworkEntityConnectionModel, 173
 - Models.Database.VirtualNetworkModel, 176

- Models.Database.VirtualNetworkNodeEntityModel, 180
- Models.DTO, 17
- Models.DTO.BootableDiskModelDTO, 30
- Models.DTO.InternetEntityModelDTO, 69
- Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO, 149
- Models.DTO.VirtualMachineEntityDTO, 150
- Models.DTO.VirtualMachineEntityMacAddressDTO, 151
- Models.DTO.VirtualNetworkConnectionDTO, 166
- Models.DTO.VirtualNetworkNodeEntityDTO, 179
- Models.Enums, 17
 - ALIVE, 17
 - Booted, 19
 - ConnectionStatus, 17
 - DEAD, 17
 - EntityType, 17
 - Internet, 17
 - Network, 17
 - Unknown, 19
 - VirtualMachine, 17
 - VirtualMachineState, 17
 - VirtualNetworkNode, 17
- Models.InformationModel, 63
- Models.Packet, 117
- Models.PacketSniffer, 118
- Models.Requests, 19
- Models.Requests.ConnectEntitiesRequest, 35
- Models.Requests.CreateVirtualMachineEntityRequest, 38
- Models.Requests.CreateVirtualNetworkNodeEntityRequest, 38
- Models.Requests.SendDataToConsoleRequest, 127
- Models.Requests.UpdateEntityPositionRequest, 130
- Models.Requests.VirtualNetworkEntityConnectionRequest, 174
- Models.Responses, 19
- Models.Responses.BaseResponse< TResponse, TSelf >, 27
 - WithEmptySuccess, 28
 - WithSuccess, 28
- Models.Responses.BootableDisksResponse, 32
- Models.Responses.CapturedPacketResponse, 33
- Models.Responses.ConnectionInfoResponse, 36
- Models.Responses.ExceptionResponse, 46
- Models.Responses.InternetEntitiesResponse, 67
- Models.Responses.StringResponse, 128
- Models.Responses.SystemInfoResponse, 129
- Models.Responses.VirtualMachineEntitiesResponse, 149
- Models.Responses.VirtualMachineEntityResponse, 157
- Models.Responses.VirtualMachineInfoResponse, 161
- Models.Responses.VirtualMachinesInfoResponse, 164
- Models.Responses.VirtualNetworkConnectionsResponse, 169
- Models.Responses.VirtualNetworkNodesEntitiesResponse, 184
- Models.StreamData, 127
- Models.StreamInfo, 128
- Models.TerminalData, 129
- Models.VirtualMachineCreateDefinition, 146
 - GetVirtualMachineNameWithPrefix, 147
- Models.VirtualMachineInfo, 161
- Models.VirtualNetworkCreateDefinition, 171
- Models.VirtualNetworkInterfaceCreateDefinition, 174
- NativeLibrary, 19
 - NativeLibrary.Interop, 20
 - NativeLibrary.Structs, 20
 - NativeLibrary.Structs.NativeConnectionInfo, 115
 - NativeLibrary.Structs.NativeExecutionInfo, 115
 - NativeLibrary.Structs.NativePacket, 116
 - NativeLibrary.Structs.NativeStreamData, 116
 - NativeLibrary.Structs.NativeStreamInfo, 116
 - NativeLibrary.Structs.NativeVirtualMachineInfo, 117
 - NativeLibrary.Utils, 20
 - NativeLibrary.Utils.Impl, 20
 - NativeLibrary.Utils.Impl.NativeList< T >, 115
 - NativeLibrary.Utils.INativeList< T >, 62
 - NativeLibrary.Wrappers, 20
 - NativeLibrary.Wrappers.Impl, 20
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, 124
 - GetNumberOfPackets, 124
 - GetPacket, 124
 - OpenSnifferHandler, 125
- NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 131
 - AttachDeviceToVirtualMachine, 132
 - CloseStream, 132
 - CreateVirtualMachine, 132
 - CreateVirtualNetwork, 133
 - DestroyNetwork, 133
 - DetachDeviceFromVirtualMachine, 133
 - GetConnectionInfo, 134
 - GetDataFromStream, 134
 - GetListOfVirtualMachines, 134
 - GetVirtualMachineInfo, 134
 - InitializeConnection, 135
 - IsConnectionAlive, 135
 - OpenVirtualMachineConsole, 135
 - SendDataToStream, 135
 - UpdateVmDevice, 136
- NativeLibrary.Wrappers.IPacketSnifferWrapper, 76
 - GetNumberOfPackets, 77
 - GetPacket, 77
 - ListenForPacket, 77
 - OpenSnifferHandler, 77
- NativeLibrary.Wrappers.IVirtualizationWrapper, 78
 - AttachDeviceToVirtualMachine, 79
 - CloseStream, 79
 - CreateVirtualMachine, 79
 - CreateVirtualNetwork, 80
 - DestroyNetwork, 80
 - DetachDeviceFromVirtualMachine, 80
 - GetConnectionInfo, 80
 - GetDataFromStream, 81
 - GetListOfVirtualMachines, 81
 - GetVirtualMachineInfo, 81
 - InitializeConnection, 82

- IsConnectionAlive, [82](#)
- OpenVirtualMachineConsole, [82](#)
- SendDataToStream, [82](#)
- UpdateVmDevice, [83](#)
- Network
 - Models.Enums, [17](#)
- OpenSnifferHandler
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, [125](#)
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, [77](#)
- OpenVirtualMachineConsole
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [135](#)
 - NativeLibrary.Wrappers.IVirtualizationWrapper, [82](#)
 - Services.Impl.VirtualMachineConsoleService, [142](#)
 - Services.IVirtualMachineConsoleService, [86](#)
- PacketSnifferBackgroundService
 - BackgroundServices.Impl.PacketSnifferBackgroundService, [119](#)
- Ping
 - Hubs.ConnectionStatusHub, [37](#)
- Remove
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, [145](#)
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, [154](#)
 - Repositories.Impl.VirtualNetworkConnectionRepository, [168](#)
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, [183](#)
 - Repositories.Impl.VirtualNetworkRepository, [187](#)
 - Repositories.IVirtualMachineConsoleStreamRepository, [88](#)
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, [91](#)
 - Repositories.IVirtualNetworkConnectionRepository, [101](#)
 - Repositories.IVirtualNetworkNodeEntityRepository, [106](#)
 - Repositories.IVirtualNetworkRepository, [110](#)
- RemoveSniffer
 - Repositories.Impl.PacketSnifferRepository, [121](#)
 - Repositories.IPacketSnifferRepository, [74](#)
- Repositories, [21](#)
- Repositories.IBootableDiskRepository, [51](#)
 - GetById, [51](#)
 - GetOnlyWithNotNullablePath, [52](#)
- Repositories.IInternetEntityRepository, [59](#)
 - Create, [59](#)
 - GetAll, [59](#)
 - GetById, [60](#)
 - UpdatePosition, [60](#)
- Repositories.Impl, [21](#)
- Repositories.Impl.BootableDiskRepository, [31](#)
 - GetById, [31](#)
 - GetOnlyWithNotNullablePath, [32](#)
- Repositories.Impl.InternetEntityRepository, [70](#)
 - Create, [70](#)
 - GetAll, [70](#)
 - GetById, [70](#)
 - UpdatePosition, [71](#)
- Repositories.Impl.PacketSnifferRepository, [120](#)
 - AddSniffer, [120](#)
 - GetOpenedSniffers, [121](#)
 - RemoveSniffer, [121](#)
- Repositories.Impl.VirtualMachineConsoleDataRepository, [138](#)
 - AddDataToConsole, [138](#)
 - GetData, [138](#)
- Repositories.Impl.VirtualMachineConsoleStreamRepository, [144](#)
 - Add, [144](#)
 - GetAllStreams, [144](#)
 - GetByUuid, [144](#)
 - Remove, [145](#)
- Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, [152](#)
 - Create, [153](#)
 - GetByIds, [153](#)
 - GetByNetworkName, [153](#)
 - GetByVirtualMachineId, [154](#)
 - Remove, [154](#)
 - Update, [154](#)
- Repositories.Impl.VirtualMachineEntityRepository, [155](#)
 - Add, [155](#)
 - GetAll, [155](#)
 - GetById, [156](#)
 - Update, [156](#)
- Repositories.Impl.VirtualNetworkConnectionRepository, [166](#)
 - Create, [166](#)
 - GetAll, [167](#)
 - GetById, [167](#)
 - GetUsingEntityId, [167](#)
 - Remove, [168](#)
- Repositories.Impl.VirtualNetworkNodeEntityRepository, [181](#)
 - Create, [181](#)
 - CreateInvisible, [181](#)
 - GetAll, [182](#)
 - GetById, [182](#)
 - GetVisible, [182](#)
 - Remove, [183](#)
 - UpdateEntityPosition, [183](#)
 - UpdateNetwork, [183](#)
- Repositories.Impl.VirtualNetworkRepository, [185](#)
 - Create, [186](#)
 - GetAll, [186](#)
 - GetByBridgeName, [186](#)
 - GetById, [186](#)
 - GetByUuid, [187](#)
 - GetByUuidWithVirtualNetworkNodes, [187](#)
 - Remove, [187](#)
- Repositories.IPacketSnifferRepository, [73](#)

- AddSniffer, 73
- GetOpenedSniffers, 74
- RemoveSniffer, 74
- Repositories.IVirtualMachineConsoleDataRepository, 83
 - AddDataToConsole, 84
 - GetData, 84
- Repositories.IVirtualMachineConsoleStreamRepository, 87
 - Add, 87
 - GetAllStreams, 88
 - GetByUuid, 88
 - Remove, 88
- Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 89
 - Create, 89
 - GetByIds, 89
 - GetByNetworkName, 91
 - GetByVirtualMachineId, 91
 - Remove, 91
 - Update, 92
- Repositories.IVirtualMachineEntityRepository, 92
 - Add, 93
 - GetAll, 93
 - GetById, 93
 - Update, 93
- Repositories.IVirtualNetworkConnectionRepository, 99
 - Create, 100
 - GetAll, 100
 - GetById, 100
 - GetUsingEntityId, 101
 - Remove, 101
- Repositories.IVirtualNetworkNodeEntityRepository, 104
 - Create, 104
 - CreateInvisible, 105
 - GetAll, 105
 - GetById, 105
 - GetVisible, 106
 - Remove, 106
 - UpdateEntityPosition, 106
 - UpdateNetwork, 106
- Repositories.IVirtualNetworkRepository, 107
 - Create, 108
 - GetAll, 108
 - GetByBridgeName, 108
 - GetById, 109
 - GetByUuid, 109
 - GetByUuidWithVirtualNetworkNodes, 109
 - Remove, 110
- RepositoriesTests, 21
- RepositoriesTests.BootableDiskRepositoryTests, 32
- RepositoriesTests.InternetEntityRepositoryTests, 71
- RepositoriesTests.VirtualMachineConsoleDataRepositoryTests, 139
- RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests, 145
- RepositoriesTests.VirtualMachineEntityRepositoryTests, 157
- RepositoriesTests.VirtualNetworkConnectionRepositoryTests, 168
- RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests, 184
- SendBytesToVirtualMachineConsoleByUuid
 - Services.Impl.VirtualMachineConsoleService, 143
 - Services.IVirtualMachineConsoleService, 86
- SendDataToConsole
 - Hubs.VirtualMachineConsoleHub, 140
- SendDataToStream
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 135
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 82
- Services, 22
- Services.IBootableDiskProviderService, 50
 - GetAvailableBootableDiskModels, 50
 - GetBootableDiskPathById, 50
- Services.IEntitiesConnectorService, 52
 - ConnectTwoEntities, 53
 - ConnectTwoVirtualMachines, 53
 - ConnectVirtualMachineToInternet, 53
 - ConnectVirtualMachineToVirtualNetworkNode, 54
- Services.IEntitiesDisconnectorService, 54
 - DisconnectEntities, 55
 - DisconnectVirtualMachineFromInternet, 55
 - DisconnectVirtualMachineFromVirtualMachine, 55
 - DisconnectVirtualMachineFromVirtualNetworkNode, 56
- Services.IHypervisorConnectionService, 56
 - GetConnectionInfo, 57
 - GetConnectionStatus, 57
 - InitializeConnection, 57
- Services.IInfoService, 58
 - GetSystemInfo, 58
- Services.IInternetEntityService, 61
 - CreateInternet, 61
 - GetInternetEntities, 61
 - UpdateInternetEntityPosition, 61
- Services.Impl, 22
- Services.Impl.BootableDiskProviderService, 30
 - GetAvailableBootableDiskModels, 30
 - GetBootableDiskPathById, 30
- Services.Impl.EntitiesConnectorService, 39
 - ConnectTwoEntities, 39
 - ConnectTwoVirtualMachines, 40
 - ConnectVirtualMachineToInternet, 40
 - ConnectVirtualMachineToVirtualNetworkNode, 40
- Services.Impl.EntitiesDisconnectorService, 41
 - DisconnectEntities, 42
 - DisconnectVirtualMachineFromInternet, 42
 - DisconnectVirtualMachineFromVirtualMachine, 42
 - DisconnectVirtualMachineFromVirtualNetworkNode, 43
- Services.Impl.HypervisorConnectionService, 48
 - GetConnectionInfo, 49
 - GetConnectionStatus, 49
 - InitializeConnection, 49
- Services.Impl.InfoService, 64

- GetSystemInfo, 64
- Services.Impl.InitializationService, 65
- Services.Impl.InternetEntityService, 72
 - CreateInternet, 72
 - GetInternetEntities, 72
 - UpdateInternetEntityPosition, 72
- Services.Impl.PacketSnifferService, 121
 - GetCapturedPacket, 122
 - GetOpenedSniffers, 122
 - ListenForPacket, 122
 - StartListeningForBridge, 122
- Services.Impl.VirtualMachineConsoleService, 141
 - CloseStream, 141
 - GetBytesFromConsole, 142
 - GetInitialConsoleData, 142
 - GetStreams, 142
 - OpenVirtualMachineConsole, 142
 - SendBytesToVirtualMachineConsoleByUuid, 143
- Services.Impl.VirtualMachineEntityService, 158
 - CreateEntity, 158
 - GetById, 159
 - GetEntities, 159
 - GetMacAddresses, 159
 - UpdateEntityPosition, 159
 - UpdateEntityVmUUID, 160
- Services.Impl.VirtualMachineManagerService, 162
 - AttachVirtualNetworkInterfaceToVirtualMachine, 162
 - CreateVirtualMachine, 163
 - GetListOfVirtualMachines, 163
- Services.Impl.VirtualNetworkNodeConnector, 177
 - ConnectTwoVirtualNetworkNodes, 178
 - ConnectVirtualNetworkNodeToInternet, 178
 - DisconnectTwoVirtualNetworkNodes, 178
 - DisconnectVirtualNetworkNodeFromInternet, 179
- Services.Impl.VirtualNetworkService, 188
 - AttachNetworkInterfaceToVirtualMachine, 189
 - CreateInternetVirtualNetwork, 189
 - CreateNodeVirtualNetwork, 189
 - CreateVirtualNetworkNode, 189
 - GetAllVirtualNetworks, 190
 - GetVirtualNetworkConnections, 190
 - GetVirtualNetworkUsingBridgeName, 190
 - GetVisibleVirtualNetworkNodeEntities, 190
 - UpdateNetworkForVirtualMachineNetworkInterface, 191
 - UpdateVirtualNetworkNodeEntityPosition, 191
- Services.IPacketSnifferService, 74
 - GetCapturedPacket, 75
 - GetOpenedSniffers, 75
 - ListenForPacket, 75
 - StartListeningForBridge, 76
- Services.IVirtualMachineConsoleService, 84
 - CloseStream, 85
 - GetBytesFromConsole, 85
 - GetInitialConsoleData, 85
 - GetStreams, 86
 - OpenVirtualMachineConsole, 86
- SendBytesToVirtualMachineConsoleByUuid, 86
- Services.IVirtualMachineEntityService, 94
 - CreateEntity, 95
 - GetById, 96
 - GetEntities, 96
 - GetMacAddresses, 96
 - UpdateEntityPosition, 97
 - UpdateEntityVmUUID, 97
- Services.IVirtualMachineManagerService, 98
 - AttachVirtualNetworkInterfaceToVirtualMachine, 98
 - CreateVirtualMachine, 98
 - GetListOfVirtualMachines, 99
- Services.IVirtualNetworkNodeConnector, 102
 - ConnectTwoVirtualNetworkNodes, 102
 - ConnectVirtualNetworkNodeToInternet, 102
 - DisconnectTwoVirtualNetworkNodes, 103
 - DisconnectVirtualNetworkNodeFromInternet, 103
- Services.IVirtualNetworkService, 110
 - AttachNetworkInterfaceToVirtualMachine, 111
 - CreateInternetVirtualNetwork, 111
 - CreateNodeVirtualNetwork, 112
 - CreateVirtualNetworkNode, 112
 - GetAllVirtualNetworks, 112
 - GetVirtualNetworkConnections, 112
 - GetVirtualNetworkUsingBridgeName, 113
 - GetVisibleVirtualNetworkNodeEntities, 113
 - UpdateNetworkForVirtualMachineNetworkInterface, 113
 - UpdateVirtualNetworkNodeEntityPosition, 114
- Services.Utills, 23
- Services.Utills.BaseDefinitionBuilder< TDefinition >, 25
- Services.Utills.VirtualMachineDefinitionBuilder, 147
- Services.Utills.VirtualNetworkCreateDefinitionBuilder, 172
- Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder, 175
- ServicesTests, 23
- ServicesTests.BootableDiskProviderServiceTests, 31
- ServicesTests.HypervisorConnectionServiceTests, 50
- ServicesTests.InfoServiceTests, 65
- ServicesTests.InitializationServiceTests, 66
- ServicesTests.Utills, 23
- ServicesTests.Utills.VirtualMachineDefinitionBuilderTests, 148
- ServicesTests.Utills.VirtualNetworkDefinitionBuilderTests, 173
- ServicesTests.Utills.VirtualNetworkInterfaceCreateDefinitionBuilderTests, 176
- ServicesTests.VirtualMachineConsoleServiceTests, 143
- ServicesTests.VirtualMachineEntityServiceTests, 161
- ServicesTests.VirtualMachineManagerServiceTests, 164
- StartListeningForBridge
 - Services.Impl.PacketSnifferService, 122
 - Services.IPacketSnifferService, 76
- TestUtils, 23

- Unknown
 - Models.Enums, [19](#)
- Update
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, [154](#)
 - Repositories.Impl.VirtualMachineEntityRepository, [156](#)
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, [92](#)
 - Repositories.IVirtualMachineEntityRepository, [93](#)
- UpdateEntityPosition
 - Controllers.EntityController, [45](#)
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, [183](#)
 - Repositories.IVirtualNetworkNodeEntityRepository, [106](#)
 - Services.Impl.VirtualMachineEntityService, [159](#)
 - Services.IVirtualMachineEntityService, [97](#)
- UpdateEntityVmUUID
 - Services.Impl.VirtualMachineEntityService, [160](#)
 - Services.IVirtualMachineEntityService, [97](#)
- UpdateInternetEntityPosition
 - Services.IInternetEntityService, [61](#)
 - Services.Impl.InternetEntityService, [72](#)
- UpdateNetwork
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, [183](#)
 - Repositories.IVirtualNetworkNodeEntityRepository, [106](#)
- UpdateNetworkForVirtualMachineNetworkInterface
 - Services.Impl.VirtualNetworkService, [191](#)
 - Services.IVirtualNetworkService, [113](#)
- UpdatePosition
 - Repositories.IInternetEntityRepository, [60](#)
 - Repositories.Impl.InternetEntityRepository, [71](#)
- UpdateVirtualNetworkNodeEntityPosition
 - Services.Impl.VirtualNetworkService, [191](#)
 - Services.IVirtualNetworkService, [114](#)
- UpdateVmDevice
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [136](#)
 - NativeLibrary.Wrappers.IVirtualizationWrapper, [83](#)
- VirtualMachine
 - Models.Enums, [17](#)
- VirtualMachineConsoleBackgroundService
 - BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, [137](#)
- VirtualMachineState
 - Models.Enums, [17](#)
- VirtualNetworkNode
 - Models.Enums, [17](#)
- WithEmptySuccess
 - Models.Responses.BaseResponse< TResponse, TSelf >, [28](#)
- WithSuccess
 - Models.Responses.BaseResponse< TResponse, TSelf >, [28](#)