

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.Utills Namespace Reference	23
4.33 ServicesTests Namespace Reference	23
4.34 ServicesTests.Utills Namespace Reference	24
4.35 TestUtills Namespace Reference	24
5 Class Documentation	25
5.1 Database.Migrations.AppDbContextModelSnapshot Class Reference	25
5.2 Services.Utills.BaseDefinitionBuilder< TDefinition > Class Template Reference	25
5.2.1 Detailed Description	26
5.2.2 Member Function Documentation	26
5.2.2.1 Build()	26
5.2.2.2 BuildingBlock()	26
5.2.2.3 CheckIsEverythingIsProvided()	27
5.2.2.4 SetFromCreateDefinition()	27
5.2.3 Member Data Documentation	27
5.2.3.1 XmlWriter	27
5.3 Models.BaseEntity Class Reference	28
5.3.1 Detailed Description	28
5.4 Models.Responses.BaseResponse< TResponse, TSelf > Class Template Reference	28
5.4.1 Detailed Description	29
5.4.2 Member Function Documentation	30
5.4.2.1 WithEmptySuccess()	30
5.4.2.2 WithSuccess()	30
5.5 Database.Migrations.BootableDiskModel Class Reference	30
5.5.1 Detailed Description	31
5.6 Models.Database.BootableDiskModel Class Reference	31
5.6.1 Detailed Description	31
5.7 Models.DTO.BootableDiskModelDTO Class Reference	31
5.8 Services.Impl.BootableDiskProviderService Class Reference	32
5.8.1 Detailed Description	32
5.8.2 Constructor & Destructor Documentation	32
5.8.2.1 BootableDiskProviderService()	32
5.8.3 Member Function Documentation	32
5.8.3.1 GetAvailableBootableDiskModels()	32
5.8.3.2 GetBootableDiskPathById()	33
5.9 ServicesTests.BootableDiskProviderServiceTests Class Reference	33
5.10 Repositories.Impl.BootableDiskRepository Class Reference	33
5.10.1 Member Function Documentation	34
5.10.1.1 GetById()	34
5.10.1.2 GetOnlyWithNotNullablePath()	34

5.11 RepositoriesTests.BootableDiskRepositoryTests Class Reference	35
5.12 Models.Responses.BootableDisksResponse Class Reference	35
5.13 Models.Responses.CapturedPacketResponse Class Reference	36
5.14 Database.Migrations.ChangesToConnection Class Reference	36
5.14.1 Detailed Description	37
5.15 Database.Migrations.ChangeTypeOfVMEntityStrings Class Reference	37
5.15.1 Detailed Description	37
5.16 Models.Requests.ConnectEntitiesRequest Class Reference	37
5.17 Models.ConnectionInfo Class Reference	38
5.17.1 Detailed Description	38
5.18 Models.Responses.ConnectionInfoResponse Class Reference	38
5.19 Hubs.ConnectionStatusHub Class Reference	39
5.19.1 Detailed Description	39
5.19.2 Member Function Documentation	39
5.19.2.1 Ping()	39
5.20 HubsTests.ConnectionStatusHubTests Class Reference	40
5.21 Models.Requests.CreateVirtualMachineEntityRequest Class Reference	40
5.22 Models.Requests.CreateVirtualNetworkNodeEntityRequest Class Reference	40
5.23 Models.Requests.DeleteEntityRequest Class Reference	40
5.24 Services.Impl.DeleteEntityService Class Reference	41
5.24.1 Detailed Description	41
5.24.2 Constructor & Destructor Documentation	41
5.24.2.1 DeleteEntityService()	41
5.24.3 Member Function Documentation	42
5.24.3.1 DeleteInternetEntity()	42
5.24.3.2 DeleteVirtualMachineEntity()	42
5.24.3.3 DeleteVirtualNetworkNodeEntity()	42
5.25 Database.Migrations.DeviceDefinition Class Reference	43
5.25.1 Detailed Description	43
5.26 Services.Impl.EntitiesConnectorService Class Reference	43
5.26.1 Detailed Description	44
5.26.2 Member Function Documentation	44
5.26.2.1 ConnectTwoEntities()	44
5.26.2.2 ConnectTwoVirtualMachines()	44
5.26.2.3 ConnectVirtualMachineToInternet()	45
5.26.2.4 ConnectVirtualMachineToVirtualNetworkNode()	46
5.27 Services.Impl.EntitiesDisconnectorService Class Reference	46
5.27.1 Detailed Description	47
5.27.2 Constructor & Destructor Documentation	47
5.27.2.1 EntitiesDisconnectorService()	47
5.27.3 Member Function Documentation	48
5.27.3.1 DisconnectEntities()	48

5.27.3.2 DisconnectVirtualMachineFromInternet()	48
5.27.3.3 DisconnectVirtualMachineFromVirtualMachine()	48
5.27.3.4 DisconnectVirtualMachineFromVirtualNetworkNode()	49
5.28 Controllers.EntityController Class Reference	49
5.28.1 Detailed Description	50
5.28.2 Constructor & Destructor Documentation	50
5.28.2.1 EntityController()	50
5.28.3 Member Function Documentation	51
5.28.3.1 CreateInternetEntity()	51
5.28.3.2 CreateVirtualMachineEntity()	51
5.28.3.3 CreateVirtualNetworkNodeEntity()	51
5.28.3.4 DeleteEntity()	51
5.28.3.5 GetAllInternetEntities()	52
5.28.3.6 GetAllVirtualMachineEntities()	52
5.28.3.7 GetAllVirtualNetworkNodeEntities()	52
5.28.3.8 UpdateEntityPosition()	52
5.29 Interconnect.Middlewares.ExceptionHandlerMiddleware Class Reference	53
5.29.1 Detailed Description	53
5.29.2 Member Function Documentation	53
5.29.2.1 InvokeAsync()	53
5.30 InterconnectTests.ExceptionHandlerMiddlewareTests Class Reference	54
5.31 Models.Responses.ExceptionResponse Class Reference	54
5.32 Controllers.HypervisorConnectionController Class Reference	55
5.32.1 Detailed Description	55
5.32.2 Constructor & Destructor Documentation	55
5.32.2.1 HypervisorConnectionController()	55
5.32.3 Member Function Documentation	55
5.32.3.1 ConnectionInfo()	55
5.33 ControllersTests.HypervisorConnectionControllerTests Class Reference	56
5.34 Services.Impl.HypervisorConnectionService Class Reference	56
5.34.1 Detailed Description	56
5.34.2 Constructor & Destructor Documentation	56
5.34.2.1 HypervisorConnectionService()	56
5.34.3 Member Function Documentation	57
5.34.3.1 GetConnectionInfo()	57
5.34.3.2 GetConnectionStatus()	57
5.34.3.3 InitializeConnection()	57
5.35 ServicesTests.HypervisorConnectionServiceTests Class Reference	58
5.36 Services.IBootableDiskProviderService Interface Reference	58
5.36.1 Detailed Description	58
5.36.2 Member Function Documentation	58
5.36.2.1 GetAvailableBootableDiskModels()	58

5.36.2.2 GetBootableDiskPathById()	59
5.37 Repositories.IBootableDiskRepository Interface Reference	59
5.37.1 Detailed Description	59
5.37.2 Member Function Documentation	59
5.37.2.1 GetById()	59
5.37.2.2 GetOnlyWithNotNullablePath()	60
5.38 Services.IDeleteEntityService Interface Reference	60
5.38.1 Member Function Documentation	61
5.38.1.1 DeleteInternetEntity()	61
5.38.1.2 DeleteVirtualMachineEntity()	61
5.38.1.3 DeleteVirtualNetworkNodeEntity()	61
5.39 Services.IEntitiesConnectorService Interface Reference	61
5.39.1 Detailed Description	62
5.39.2 Member Function Documentation	62
5.39.2.1 ConnectTwoEntities()	62
5.39.2.2 ConnectTwoVirtualMachines()	62
5.39.2.3 ConnectVirtualMachineToInternet()	62
5.39.2.4 ConnectVirtualMachineToVirtualNetworkNode()	63
5.40 Services.IEntitiesDisconnectorService Interface Reference	63
5.40.1 Detailed Description	64
5.40.2 Member Function Documentation	64
5.40.2.1 DisconnectEntities()	64
5.40.2.2 DisconnectVirtualMachineFromInternet()	64
5.40.2.3 DisconnectVirtualMachineFromVirtualMachine()	65
5.40.2.4 DisconnectVirtualMachineFromVirtualNetworkNode()	65
5.41 Services.IHypervisorConnectionService Interface Reference	65
5.41.1 Detailed Description	66
5.41.2 Member Function Documentation	66
5.41.2.1 GetConnectionInfo()	66
5.41.2.2 GetConnectionStatus()	66
5.41.2.3 InitializeConnection()	66
5.42 Services.IInfoService Interface Reference	67
5.42.1 Detailed Description	67
5.42.2 Member Function Documentation	67
5.42.2.1 GetSystemInfo()	67
5.43 Repositories.IInternetEntityRepository Interface Reference	68
5.43.1 Detailed Description	68
5.43.2 Member Function Documentation	68
5.43.2.1 Create()	68
5.43.2.2 GetAll()	69
5.43.2.3 GetById()	69
5.43.2.4 Remove()	69

5.43.2.5 UpdatePosition()	69
5.44 Services.IInternetEntityService Interface Reference	70
5.44.1 Detailed Description	70
5.44.2 Member Function Documentation	71
5.44.2.1 CreateInternet()	71
5.44.2.2 GetInternetEntities()	71
5.44.2.3 UpdateInternetEntityPosition()	71
5.45 NativeLibrary.Utils.INativeList< T > Interface Template Reference	72
5.46 Controllers.InfoController Class Reference	72
5.46.1 Detailed Description	72
5.46.2 Constructor & Destructor Documentation	72
5.46.2.1 InfoController()	72
5.46.3 Member Function Documentation	73
5.46.3.1 GetSystemInfo()	73
5.47 ControllersTests.InfoControllerTests Class Reference	73
5.48 Models.InformationModel Class Reference	73
5.48.1 Detailed Description	74
5.49 Services.Impl.InfoService Class Reference	74
5.49.1 Detailed Description	74
5.49.2 Member Function Documentation	74
5.49.2.1 GetSystemInfo()	74
5.50 ServicesTests.InfoServiceTests Class Reference	75
5.51 Database.Migrations.InitialCreate Class Reference	75
5.51.1 Detailed Description	75
5.52 Services.Impl.InitializationService Class Reference	75
5.52.1 Detailed Description	76
5.52.2 Constructor & Destructor Documentation	76
5.52.2.1 InitializationService()	76
5.52.3 Member Function Documentation	76
5.52.3.1 StartAsync()	76
5.52.3.2 StopAsync()	76
5.53 ServicesTests.InitializationServiceTests Class Reference	77
5.54 Models.Config.InterconnectConfig Class Reference	77
5.55 Database.InterconnectDbContext Class Reference	77
5.55.1 Detailed Description	78
5.56 Models.Responses.InternetEntitiesResponse Class Reference	78
5.57 Database.Migrations.InternetEntity Class Reference	79
5.57.1 Detailed Description	79
5.58 Models.Database.InternetEntityModel Class Reference	79
5.58.1 Detailed Description	80
5.59 Models.DTO.InternetEntityModelDTO Class Reference	80
5.60 Repositories.Impl.InternetEntityRepository Class Reference	81

5.60.1 Member Function Documentation	81
5.60.1.1 Create()	81
5.60.1.2 GetAll()	82
5.60.1.3 GetById()	82
5.60.1.4 Remove()	82
5.60.1.5 UpdatePosition()	82
5.61 Repositories.Tests.InternetEntityRepositoryTests Class Reference	83
5.62 Services.Impl.InternetEntityService Class Reference	83
5.62.1 Detailed Description	84
5.62.2 Constructor & Destructor Documentation	84
5.62.2.1 InternetEntityService()	84
5.62.3 Member Function Documentation	84
5.62.3.1 CreateInternet()	84
5.62.3.2 GetInternetEntities()	85
5.62.3.3 UpdateInternetEntityPosition()	85
5.63 Repositories.IPacketSnifferRepository Interface Reference	85
5.63.1 Detailed Description	86
5.63.2 Member Function Documentation	86
5.63.2.1 AddSniffer()	86
5.63.2.2 GetOpenedSniffers()	86
5.63.2.3 RemoveSniffer()	86
5.64 Services.IPacketSnifferService Interface Reference	87
5.64.1 Detailed Description	87
5.64.2 Member Function Documentation	87
5.64.2.1 GetCapturedPacket()	87
5.64.2.2 GetOpenedSniffers()	88
5.64.2.3 ListenForPacket()	88
5.64.2.4 StartListeningForBridge()	88
5.65 NativeLibrary.Wrappers.IPacketSnifferWrapper Interface Reference	89
5.65.1 Detailed Description	89
5.65.2 Member Function Documentation	89
5.65.2.1 GetNumberOfPackets()	89
5.65.2.2 GetPacket()	89
5.65.2.3 ListenForPacket()	90
5.65.2.4 OpenSnifferHandler()	90
5.66 NativeLibrary.Wrappers.IVirtualizationWrapper Interface Reference	90
5.66.1 Detailed Description	91
5.66.2 Member Function Documentation	91
5.66.2.1 AttachDeviceToVirtualMachine()	91
5.66.2.2 CloseStream()	92
5.66.2.3 CreateVirtualMachine()	92
5.66.2.4 CreateVirtualNetwork()	92

5.66.2.5 DestroyNetwork()	92
5.66.2.6 DetachDeviceFromVirtualMachine()	93
5.66.2.7 GetConnectionInfo()	93
5.66.2.8 GetDataFromStream()	93
5.66.2.9 GetListOfVirtualMachines()	94
5.66.2.10 GetVirtualMachineInfo()	94
5.66.2.11 InitializeConnection()	94
5.66.2.12 IsConnectionAlive()	94
5.66.2.13 OpenVirtualMachineConsole()	94
5.66.2.14 SendDataToStream()	95
5.66.2.15 UpdateVmDevice()	95
5.67 Repositories.IVirtualMachineConsoleDataRepository Interface Reference	96
5.67.1 Detailed Description	96
5.67.2 Member Function Documentation	96
5.67.2.1 AddDataToConsole()	96
5.67.2.2 GetData()	96
5.68 Services.IVirtualMachineConsoleService Interface Reference	97
5.68.1 Detailed Description	97
5.68.2 Member Function Documentation	97
5.68.2.1 CloseStream()	97
5.68.2.2 GetBytesFromConsole()	98
5.68.2.3 GetInitialConsoleData()	98
5.68.2.4 GetStreams()	98
5.68.2.5 OpenVirtualMachineConsole()	98
5.68.2.6 SendBytesToVirtualMachineConsoleByUuid()	99
5.69 Repositories.IVirtualMachineConsoleStreamRepository Interface Reference	99
5.69.1 Detailed Description	100
5.69.2 Member Function Documentation	100
5.69.2.1 Add()	100
5.69.2.2 GetAllStreams()	100
5.69.2.3 GetByUuid()	100
5.69.2.4 Remove()	100
5.70 Repositories.IVirtualMachineEntityNetworkInterfaceRepository Interface Reference	101
5.70.1 Detailed Description	101
5.70.2 Member Function Documentation	101
5.70.2.1 Create()	101
5.70.2.2 GetByIds()	102
5.70.2.3 GetByNetworkName()	102
5.70.2.4 GetByVirtualMachineId()	102
5.70.2.5 Remove()	103
5.70.2.6 Update()	103
5.71 Repositories.IVirtualMachineEntityRepository Interface Reference	103

5.71.1 Detailed Description	104
5.71.2 Member Function Documentation	104
5.71.2.1 Add()	104
5.71.2.2 GetAll()	104
5.71.2.3 GetById()	104
5.71.2.4 Remove()	105
5.71.2.5 Update()	105
5.72 Services.IVirtualMachineEntityService Interface Reference	105
5.72.1 Detailed Description	106
5.72.2 Member Function Documentation	106
5.72.2.1 CreateEntity()	106
5.72.2.2 GetById()	107
5.72.2.3 GetEntities()	107
5.72.2.4 GetMacAddresses()	107
5.72.2.5 UpdateEntityPosition()	107
5.72.2.6 UpdateEntityVmUUID()	108
5.73 Services.IVirtualMachineManagerService Interface Reference	108
5.73.1 Detailed Description	109
5.73.2 Member Function Documentation	109
5.73.2.1 AttachVirtualNetworkInterfaceToVirtualMachine()	109
5.73.2.2 CreateVirtualMachine()	109
5.73.2.3 GetListOfVirtualMachines()	110
5.74 Repositories.IVirtualNetworkConnectionRepository Interface Reference	110
5.74.1 Detailed Description	110
5.74.2 Member Function Documentation	111
5.74.2.1 Create()	111
5.74.2.2 GetAll()	111
5.74.2.3 GetById()	111
5.74.2.4 GetUsingEntityId()	112
5.74.2.5 Remove()	112
5.75 Services.IVirtualNetworkNodeConnector Interface Reference	112
5.75.1 Detailed Description	113
5.75.2 Member Function Documentation	113
5.75.2.1 ConnectTwoVirtualNetworkNodes()	113
5.75.2.2 ConnectVirtualNetworkNodeToInternet()	113
5.75.2.3 DisconnectTwoVirtualNetworkNodes()	114
5.75.2.4 DisconnectVirtualNetworkNodeFromInternet()	114
5.76 Repositories.IVirtualNetworkNodeEntityRepository Interface Reference	115
5.76.1 Detailed Description	115
5.76.2 Member Function Documentation	115
5.76.2.1 Create()	115
5.76.2.2 CreateInvisible()	116

5.76.2.3 GetAll()	116
5.76.2.4 GetById()	116
5.76.2.5 GetVisible()	117
5.76.2.6 Remove()	117
5.76.2.7 UpdateEntityPosition()	117
5.76.2.8 UpdateNetwork()	118
5.77 Repositories.IVirtualNetworkRepository Interface Reference	118
5.77.1 Detailed Description	119
5.77.2 Member Function Documentation	119
5.77.2.1 Create()	119
5.77.2.2 GetAll()	119
5.77.2.3 GetByBridgeName()	119
5.77.2.4 GetById()	120
5.77.2.5 GetByUuid()	120
5.77.2.6 GetByUuidWithVirtualNetworkNodes()	120
5.77.2.7 Remove()	121
5.78 Services.IVirtualNetworkService Interface Reference	121
5.78.1 Detailed Description	122
5.78.2 Member Function Documentation	122
5.78.2.1 AttachNetworkInterfaceToVirtualMachine()	122
5.78.2.2 CreateInternetVirtualNetwork()	123
5.78.2.3 CreateNodeVirtualNetwork()	123
5.78.2.4 CreateVirtualNetworkNode()	123
5.78.2.5 GetAllVirtualNetworks()	123
5.78.2.6 GetVirtualNetworkConnections()	124
5.78.2.7 GetVirtualNetworkUsingBridgeName()	124
5.78.2.8 GetVisibleVirtualNetworkNodeEntities()	124
5.78.2.9 UpdateNetworkForVirtualMachineNetworkInterface()	124
5.78.2.10 UpdateVirtualNetworkNodeEntityPosition()	125
5.79 Database.Migrations.ModificationToVirtualNetwork Class Reference	125
5.79.1 Detailed Description	126
5.80 NativeLibrary.Structs.NativeConnectionInfo Struct Reference	126
5.81 NativeLibrary.Structs.NativeExecutionInfo Struct Reference	126
5.82 NativeLibrary.Utils.Impl.NativeList< T > Class Template Reference	126
5.83 NativeLibrary.Structs.NativePacket Struct Reference	127
5.84 NativeLibrary.Structs.NativeStreamData Struct Reference	127
5.85 NativeLibrary.Structs.NativeStreamInfo Struct Reference	127
5.86 NativeLibrary.Structs.NativeVirtualMachineInfo Struct Reference	128
5.87 Models.Packet Class Reference	128
5.87.1 Detailed Description	128
5.88 Models.PacketSniffer Class Reference	129
5.88.1 Detailed Description	129

5.89 BackgroundServices.Impl.PacketSnifferBackgroundService Class Reference	129
5.89.1 Detailed Description	130
5.89.2 Constructor & Destructor Documentation	130
5.89.2.1 PacketSnifferBackgroundService()	130
5.89.3 Member Function Documentation	130
5.89.3.1 ExecuteAsync()	130
5.90 Hubs.PacketSnifferHub Class Reference	130
5.90.1 Detailed Description	131
5.91 Repositories.Impl.PacketSnifferRepository Class Reference	131
5.91.1 Member Function Documentation	131
5.91.1.1 AddSniffer()	131
5.91.1.2 GetOpenedSniffers()	132
5.91.1.3 RemoveSniffer()	132
5.92 Services.Impl.PacketSnifferService Class Reference	132
5.92.1 Detailed Description	133
5.92.2 Constructor & Destructor Documentation	133
5.92.2.1 PacketSnifferService()	133
5.92.3 Member Function Documentation	133
5.92.3.1 GetCapturedPacket()	133
5.92.3.2 GetOpenedSniffers()	134
5.92.3.3 ListenForPacket()	134
5.92.3.4 StartListeningForBridge()	134
5.93 NativeLibrary.Wrappers.Impl.PacketSnifferWrapper Class Reference	135
5.93.1 Member Function Documentation	135
5.93.1.1 GetNumberOfPackets()	135
5.93.1.2 GetPacket()	135
5.93.1.3 OpenSnifferHandler()	135
5.94 Interconnect.Program Class Reference	136
5.94.1 Detailed Description	136
5.94.2 Member Function Documentation	136
5.94.2.1 Main()	136
5.95 Database.Migrations.RemoveOsType Class Reference	137
5.95.1 Detailed Description	137
5.96 Database.Migrations.RenameSwitchToNode Class Reference	137
5.96.1 Detailed Description	137
5.97 Database.Migrations.RenameUuidToVmUuid Class Reference	138
5.97.1 Detailed Description	138
5.98 Models.Requests.SendDataToConsoleRequest Class Reference	138
5.99 Models.StreamData Class Reference	138
5.99.1 Detailed Description	139
5.100 Models.StreamInfo Class Reference	139
5.100.1 Detailed Description	139

5.101 Models.Responses.StringResponse Class Reference	139
5.102 Models.Responses.SystemInfoResponse Class Reference	140
5.103 Models.TerminalData Class Reference	140
5.103.1 Detailed Description	140
5.104 Database.Migrations.UpdateBootableDiskModelPath Class Reference	141
5.104.1 Detailed Description	141
5.105 Models.Requests.UpdateEntityPositionRequest Class Reference	141
5.106 Database.Migrations.UpdateOperatingSystemType Class Reference	141
5.106.1 Detailed Description	142
5.107 NativeLibrary.Wrappers.Impl.VirtualizationWrapper Class Reference	142
5.107.1 Member Function Documentation	143
5.107.1.1 AttachDeviceToVirtualMachine()	143
5.107.1.2 CloseStream()	143
5.107.1.3 CreateVirtualMachine()	143
5.107.1.4 CreateVirtualNetwork()	144
5.107.1.5 DestroyNetwork()	144
5.107.1.6 DetachDeviceFromVirtualMachine()	144
5.107.1.7 GetConnectionInfo()	145
5.107.1.8 GetDataFromStream()	145
5.107.1.9 GetListOfVirtualMachines()	145
5.107.1.10 GetVirtualMachineInfo()	145
5.107.1.11 InitializeConnection()	146
5.107.1.12 IsConnectionAlive()	146
5.107.1.13 OpenVirtualMachineConsole()	146
5.107.1.14 SendDataToStream()	146
5.107.1.15 UpdateVmDevice()	147
5.108 BackgroundServices.Impl.VirtualMachineConsoleBackgroundService Class Reference	147
5.108.1 Detailed Description	148
5.108.2 Constructor & Destructor Documentation	148
5.108.2.1 VirtualMachineConsoleBackgroundService()	148
5.108.3 Member Function Documentation	148
5.108.3.1 ExecuteAsync()	148
5.109 Repositories.Impl.VirtualMachineConsoleDataRepository Class Reference	149
5.109.1 Member Function Documentation	149
5.109.1.1 AddDataToConsole()	149
5.109.1.2 GetData()	149
5.110 RepositoriesTests.VirtualMachineConsoleDataRepositoryTests Class Reference	150
5.111 Hubs.VirtualMachineConsoleHub Class Reference	150
5.111.1 Detailed Description	150
5.111.2 Member Function Documentation	150
5.111.2.1 GetInitialDataForConsole()	150
5.111.2.2 JoinConsoleGroup()	151

5.111.2.3 LeaveConsoleGroup()	151
5.111.2.4 SendDataToConsole()	151
5.112 HubsTests.VirtualMachineConsoleHubTests Class Reference	152
5.113 Services.Impl.VirtualMachineConsoleService Class Reference	152
5.113.1 Detailed Description	152
5.113.2 Constructor & Destructor Documentation	153
5.113.2.1 VirtualMachineConsoleService()	153
5.113.3 Member Function Documentation	153
5.113.3.1 CloseStream()	153
5.113.3.2 GetBytesFromConsole()	153
5.113.3.3 GetInitialConsoleData()	153
5.113.3.4 GetStreams()	154
5.113.3.5 OpenVirtualMachineConsole()	154
5.113.3.6 SendBytesToVirtualMachineConsoleByUuid()	154
5.114 ServicesTests.VirtualMachineConsoleServiceTests Class Reference	155
5.115 Repositories.Impl.VirtualMachineConsoleStreamRepository Class Reference	155
5.115.1 Member Function Documentation	156
5.115.1.1 Add()	156
5.115.1.2 GetAllStreams()	156
5.115.1.3 GetByUuid()	156
5.115.1.4 Remove()	157
5.116 RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests Class Reference	157
5.117 Controllers.VirtualMachineController Class Reference	158
5.117.1 Detailed Description	158
5.117.2 Constructor & Destructor Documentation	158
5.117.2.1 VirtualMachineController()	158
5.117.3 Member Function Documentation	158
5.117.3.1 GetAvailableBootableDisks()	158
5.118 Models.VirtualMachineCreateDefinition Class Reference	159
5.118.1 Detailed Description	159
5.118.2 Member Function Documentation	159
5.118.2.1 GetVirtualMachineNameWithPrefix()	159
5.119 Services.Utils.VirtualMachineDefinitionBuilder Class Reference	160
5.119.1 Detailed Description	161
5.119.2 Member Function Documentation	161
5.119.2.1 Build()	161
5.119.2.2 SetFromCreateDefinition()	161
5.119.2.3 SetPrefix()	161
5.120 ServicesTests.Utils.VirtualMachineDefinitionBuilderTests Class Reference	162
5.121 Models.Responses.VirtualMachineEntitiesResponse Class Reference	162
5.122 Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO Class Reference	163
5.123 Models.DTO.VirtualMachineEntityDTO Class Reference	163

5.124 Database.Migrations.VirtualMachineEntityInterfaces Class Reference	164
5.124.1 Detailed Description	164
5.125 Models.DTO.VirtualMachineEntityMacAddressDTO Class Reference	164
5.126 Models.Database.VirtualMachineEntityModel Class Reference	164
5.126.1 Detailed Description	165
5.127 Models.Database.VirtualMachineEntityNetworkInterfaceModel Class Reference	165
5.127.1 Detailed Description	166
5.128 Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository Class Reference	166
5.128.1 Member Function Documentation	166
5.128.1.1 Create()	166
5.128.1.2 GetByIds()	167
5.128.1.3 GetByNetworkName()	167
5.128.1.4 GetByVirtualMachineId()	167
5.128.1.5 Remove()	168
5.128.1.6 Update()	168
5.129 Repositories.Impl.VirtualMachineEntityRepository Class Reference	168
5.129.1 Member Function Documentation	169
5.129.1.1 Add()	169
5.129.1.2 GetAll()	169
5.129.1.3 GetById()	169
5.129.1.4 Remove()	170
5.129.1.5 Update()	170
5.130 RepositoriesTests.VirtualMachineEntityRepositoryTests Class Reference	170
5.131 Models.Responses.VirtualMachineEntityResponse Class Reference	171
5.132 Services.Impl.VirtualMachineEntityService Class Reference	171
5.132.1 Detailed Description	172
5.132.2 Constructor & Destructor Documentation	172
5.132.2.1 VirtualMachineEntityService()	172
5.132.3 Member Function Documentation	173
5.132.3.1 CreateEntity()	173
5.132.3.2 GetById()	173
5.132.3.3 GetEntities()	174
5.132.3.4 GetMacAddresses()	174
5.132.3.5 UpdateEntityPosition()	174
5.132.3.6 UpdateEntityVmUUID()	174
5.133 ServicesTests.VirtualMachineEntityServiceTests Class Reference	175
5.134 Models.VirtualMachineInfo Class Reference	175
5.134.1 Detailed Description	176
5.135 Models.Responses.VirtualMachineInfoResponse Class Reference	176
5.136 Services.Impl.VirtualMachineManagerService Class Reference	177
5.136.1 Detailed Description	177
5.136.2 Constructor & Destructor Documentation	177

5.136.2.1 VirtualMachineManagerService()	177
5.136.3 Member Function Documentation	178
5.136.3.1 AttachVirtualNetworkInterfaceToVirtualMachine()	178
5.136.3.2 CreateVirtualMachine()	179
5.136.3.3 GetListOfVirtualMachines()	179
5.137 ServicesTests.VirtualMachineManagerServiceTests Class Reference	180
5.138 Models.Responses.VirtualMachinesInfoResponse Class Reference	180
5.139 Database.Migrations.VirtualMachineType Class Reference	181
5.139.1 Detailed Description	181
5.140 Database.Migrations.VirtualNetwork Class Reference	181
5.140.1 Detailed Description	181
5.141 Models.DTO.VirtualNetworkConnectionDTO Class Reference	182
5.142 Repositories.Impl.VirtualNetworkConnectionRepository Class Reference	182
5.142.1 Member Function Documentation	182
5.142.1.1 Create()	182
5.142.1.2 GetAll()	183
5.142.1.3 GetByld()	183
5.142.1.4 GetUsingEntityId()	183
5.142.1.5 Remove()	184
5.143 RepositoriesTests.VirtualNetworkConnectionRepositoryTests Class Reference	184
5.144 Models.Responses.VirtualNetworkConnectionsResponse Class Reference	185
5.145 Controllers.VirtualNetworkController Class Reference	185
5.145.1 Detailed Description	186
5.145.2 Constructor & Destructor Documentation	186
5.145.2.1 VirtualNetworkController()	186
5.145.3 Member Function Documentation	186
5.145.3.1 ConnectEntities()	186
5.145.3.2 DisconnectEntities()	187
5.145.3.3 GetAllConnections()	188
5.146 Models.VirtualNetworkCreateDefinition Class Reference	188
5.146.1 Detailed Description	189
5.147 Services.Utils.VirtualNetworkCreateDefinitionBuilder Class Reference	189
5.147.1 Detailed Description	190
5.147.2 Member Function Documentation	190
5.147.2.1 Build()	190
5.147.2.2 SetFromCreateDefinition()	190
5.148 ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests Class Reference	191
5.149 Database.Migrations.VirtualNetworkEntities Class Reference	191
5.149.1 Detailed Description	191
5.150 Models.Database.VirtualNetworkEntityConnectionModel Class Reference	191
5.150.1 Detailed Description	192
5.151 Models.Requests.VirtualNetworkEntityConnectionRequest Class Reference	192

5.152 Models.VirtualNetworkInterfaceCreateDefinition Class Reference	192
5.152.1 Detailed Description	193
5.153 Services.Utls.VirtualNetworkInterfaceCreateDefinitionBuilder Class Reference	193
5.153.1 Detailed Description	194
5.153.2 Member Function Documentation	194
5.153.2.1 Build()	194
5.153.2.2 SetFromCreateDefinition()	194
5.153.2.3 SetFromXml()	194
5.153.2.4 SetNetworkName()	195
5.154 ServicesTests.Utls.VirtualNetworkInterfaceCreateDefinitionBuilderTests Class Reference	195
5.155 Database.Migrations.VirtualNetworkIpAddress Class Reference	196
5.155.1 Detailed Description	196
5.156 Models.Database.VirtualNetworkModel Class Reference	196
5.156.1 Detailed Description	197
5.157 Services.Impl.VirtualNetworkNodeConnector Class Reference	197
5.157.1 Detailed Description	197
5.157.2 Member Function Documentation	197
5.157.2.1 ConnectTwoVirtualNetworkNodes()	197
5.157.2.2 ConnectVirtualNetworkNodeToInternet()	198
5.157.2.3 DisconnectTwoVirtualNetworkNodes()	198
5.157.2.4 DisconnectVirtualNetworkNodeFromInternet()	198
5.158 Models.DTO.VirtualNetworkNodeEntityDTO Class Reference	199
5.159 Models.Database.VirtualNetworkNodeEntityModel Class Reference	199
5.159.1 Detailed Description	200
5.160 Repositories.Impl.VirtualNetworkNodeEntityRepository Class Reference	200
5.160.1 Member Function Documentation	201
5.160.1.1 Create()	201
5.160.1.2 CreateInvisible()	201
5.160.1.3 GetAll()	201
5.160.1.4 GetById()	201
5.160.1.5 GetVisible()	202
5.160.1.6 Remove()	202
5.160.1.7 UpdateEntityPosition()	202
5.160.1.8 UpdateNetwork()	203
5.161 RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests Class Reference	203
5.162 Models.Responses.VirtualNetworkNodesEntitiesResponse Class Reference	203
5.163 Repositories.Impl.VirtualNetworkRepository Class Reference	204
5.163.1 Member Function Documentation	205
5.163.1.1 Create()	205
5.163.1.2 GetAll()	205
5.163.1.3 GetByBridgeName()	205
5.163.1.4 GetById()	205

5.163.1.5 GetByUuid()	206
5.163.1.6 GetByUuidWithVirtualNetworkNodes()	206
5.163.1.7 Remove()	206
5.164 Services.Impl.VirtualNetworkService Class Reference	207
5.164.1 Detailed Description	208
5.164.2 Constructor & Destructor Documentation	208
5.164.2.1 VirtualNetworkService()	208
5.164.3 Member Function Documentation	208
5.164.3.1 AttachNetworkInterfaceToVirtualMachine()	208
5.164.3.2 CreateInternetVirtualNetwork()	209
5.164.3.3 CreateNodeVirtualNetwork()	209
5.164.3.4 CreateVirtualNetworkNode()	209
5.164.3.5 GetAllVirtualNetworks()	209
5.164.3.6 GetVirtualNetworkConnections()	210
5.164.3.7 GetVirtualNetworkUsingBridgeName()	210
5.164.3.8 GetVisibleVirtualNetworkNodeEntities()	210
5.164.3.9 UpdateNetworkForVirtualMachineNetworkInterface()	210
5.164.3.10 UpdateVirtualNetworkNodeEntityPosition()	211
5.165 Database.Migrations.VirtualNetworkVarChar Class Reference	211
5.165.1 Detailed Description	212
5.166 Database.Migrations.VirtualSwitch Class Reference	212
5.166.1 Detailed Description	212
Index	213

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	24
TestUtils	24

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

BackgroundService	
BackgroundServices.Impl.PacketSnifferBackgroundService	129
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService	147
Services.Utills.BaseDefinitionBuilder< TDefinition >	25
Services.Utills.VirtualMachineDefinitionBuilder	160
Services.Utills.VirtualNetworkCreateDefinitionBuilder	189
Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder	193
Models.BaseEntity	28
Models.DTO.InternetEntityModelDTO	80
Models.DTO.VirtualMachineEntityDTO	163
Models.DTO.VirtualNetworkNodeEntityDTO	199
Models.Responses.BaseResponse< TResponse, TSelf >	28
Models.Responses.BootableDisksResponse	35
Models.Responses.CapturedPacketResponse	36
Models.Responses.ConnectionInfoResponse	38
Models.Responses.ExceptionResponse	54
Models.Responses.InternetEntitiesResponse	78
Models.Responses.StringResponse	139
Models.Responses.VirtualMachineEntitiesResponse	162
Models.Responses.VirtualMachineEntityResponse	171
Models.Responses.VirtualMachineInfoResponse	176
Models.Responses.VirtualMachinesInfoResponse	180
Models.Responses.VirtualNetworkConnectionsResponse	185
Models.Responses.VirtualNetworkNodesEntitiesResponse	203
Models.Database.BootableDiskModel	31
Models.DTO.BootableDiskModelDTO	31
ServicesTests.BootableDiskProviderServiceTests	33
RepositoriesTests.BootableDiskRepositoryTests	35
Models.Requests.ConnectEntitiesRequest	37
Models.ConnectionInfo	38
HubsTests.ConnectionStatusHubTests	40
ControllerBase	
Controllers.EntityController	49
Controllers.HypervisorConnectionController	55

Controllers.InfoController	72
Controllers.VirtualMachineController	158
Controllers.VirtualNetworkController	185
Models.Requests.CreateVirtualMachineEntityRequest	40
Models.Requests.CreateVirtualNetworkNodeEntityRequest	40
DbContext	
Database.InterconnectDbContext	77
Models.Requests.DeleteEntityRequest	40
Interconnect.Middlewares.ExceptionHandlerMiddleware	53
InterconnectTests.ExceptionHandlerMiddlewareTests	54
Hub	
Hubs.ConnectionStatusHub	39
Hubs.PacketSnifferHub	130
Hubs.VirtualMachineConsoleHub	150
ControllersTests.HypervisorConnectionControllerTests	56
ServicesTests.HypervisorConnectionServiceTests	58
Services.IBootableDiskProviderService	58
Services.Impl.BootableDiskProviderService	32
Repositories.IBootableDiskRepository	59
Repositories.Impl.BootableDiskRepository	33
Services.IDeleteEntityService	60
Services.Impl.DeleteEntityService	41
Services.IEntitiesConnectorService	61
Services.Impl.EntitiesConnectorService	43
Services.IEntitiesDisconnectorService	63
Services.Impl.EntitiesDisconnectorService	46
IHostedService	
Services.IHypervisorConnectionService	65
Services.Impl.HypervisorConnectionService	56
Services.IInfoService	67
Services.Impl.InfoService	74
Repositories.IInternetEntityRepository	68
Repositories.Impl.InternetEntityRepository	81
Services.IInternetEntityService	70
Services.Impl.InternetEntityService	83
ControllersTests.InfoControllerTests	73
Models.InformationModel	73
ServicesTests.InfoServiceTests	75
ServicesTests.InitializationServiceTests	77
Models.Config.InterconnectConfig	77
Models.Database.InternetEntityModel	79
RepositoriesTests.InternetEntityRepositoryTests	83
Repositories.IPacketSnifferRepository	85
Repositories.Impl.PacketSnifferRepository	131
Services.IPacketSnifferService	87
Services.Impl.PacketSnifferService	132
NativeLibrary.Wrappers.IPacketSnifferWrapper	89
NativeLibrary.Wrappers.Impl.PacketSnifferWrapper	135
IReadOnlyList	
NativeLibrary.Utils.INativeList< T >	72
NativeLibrary.Utils.Impl.NativeList< T >	126
NativeLibrary.Wrappers.IVirtualizationWrapper	90
NativeLibrary.Wrappers.Impl.VirtualizationWrapper	142
Repositories.IVirtualMachineConsoleDataRepository	96

Repositories.Impl.VirtualMachineConsoleDataRepository	149
Services.IVirtualMachineConsoleService	97
Services.Impl.VirtualMachineConsoleService	152
Repositories.IVirtualMachineConsoleStreamRepository	99
Repositories.Impl.VirtualMachineConsoleStreamRepository	155
Repositories.IVirtualMachineEntityNetworkInterfaceRepository	101
Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository	166
Repositories.IVirtualMachineEntityRepository	103
Repositories.Impl.VirtualMachineEntityRepository	168
Services.IVirtualMachineEntityService	105
Services.Impl.VirtualMachineEntityService	171
Services.IVirtualMachineManagerService	108
Services.Impl.VirtualMachineManagerService	177
Repositories.IVirtualNetworkConnectionRepository	110
Repositories.Impl.VirtualNetworkConnectionRepository	182
Services.IVirtualNetworkNodeConnector	112
Services.Impl.VirtualNetworkNodeConnector	197
Repositories.IVirtualNetworkNodeEntityRepository	115
Repositories.Impl.VirtualNetworkNodeEntityRepository	200
Repositories.IVirtualNetworkRepository	118
Repositories.Impl.VirtualNetworkRepository	204
Services.IVirtualNetworkService	121
Services.Impl.VirtualNetworkService	207
Migration	
Database.Migrations.BootableDiskModel	30
Database.Migrations.ChangeTypeOfVmEntityStrings	37
Database.Migrations.ChangesToConnection	36
Database.Migrations.DeviceDefinition	43
Database.Migrations.InitialCreate	75
Database.Migrations.InternetEntity	79
Database.Migrations.ModificationToVirtualNetwork	125
Database.Migrations.RemoveOsType	137
Database.Migrations.RenameSwitchToNode	137
Database.Migrations.RenameUuidToVmUuid	138
Database.Migrations.UpdateBootableDiskModelPath	141
Database.Migrations.UpdateOperatingSystemType	141
Database.Migrations.VirtualMachineEntityInterfaces	164
Database.Migrations.VirtualMachineType	181
Database.Migrations.VirtualNetwork	181
Database.Migrations.VirtualNetworkEntities	191
Database.Migrations.VirtualNetworkIpAddress	196
Database.Migrations.VirtualNetworkVarChar	211
Database.Migrations.VirtualSwitch	212
ModelSnapshot	
Database.Migrations.AppDbContextModelSnapshot	25
NativeLibrary.Structs.NativeConnectionInfo	126
NativeLibrary.Structs.NativeExecutionInfo	126
NativeLibrary.Structs.NativePacket	127
NativeLibrary.Structs.NativeStreamData	127
NativeLibrary.Structs.NativeStreamInfo	127
NativeLibrary.Structs.NativeVirtualMachineInfo	128
Models.Packet	128
Models.PacketSniffer	129
Interconnect.Program	136
Models.Requests.SendDataToConsoleRequest	138

Models.StreamData	138
Models.StreamInfo	139
Models.Responses.SystemInfoResponse	140
Models.TerminalData	140
Models.Requests.UpdateEntityPositionRequest	141
RepositoriesTests.VirtualMachineConsoleDataRepositoryTests	150
HubsTests.VirtualMachineConsoleHubTests	152
ServicesTests.VirtualMachineConsoleServiceTests	155
RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests	157
Models.VirtualMachineCreateDefinition	159
ServicesTests.Utils.VirtualMachineDefinitionBuilderTests	162
Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO	163
Models.DTO.VirtualMachineEntityMacAddressDTO	164
Models.Database.VirtualMachineEntityModel	164
Models.Database.VirtualMachineEntityNetworkInterfaceModel	165
RepositoriesTests.VirtualMachineEntityRepositoryTests	170
ServicesTests.VirtualMachineEntityServiceTests	175
Models.VirtualMachineInfo	175
ServicesTests.VirtualMachineManagerServiceTests	180
Models.DTO.VirtualNetworkConnectionDTO	182
RepositoriesTests.VirtualNetworkConnectionRepositoryTests	184
Models.VirtualNetworkCreateDefinition	188
ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests	191
Models.Database.VirtualNetworkEntityConnectionModel	191
Models.Requests.VirtualNetworkEntityConnectionRequest	192
Models.VirtualNetworkInterfaceCreateDefinition	192
ServicesTests.Utils.VirtualNetworkInterfaceCreateDefinitionBuilderTests	195
Models.Database.VirtualNetworkModel	196
Models.Database.VirtualNetworkNodeEntityModel	199
RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests	203

Chapter 3

Class Index

3.1 Class List

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

Database.Migrations.AppDbContextModelSnapshot	25
Services.Utls.BaseDefinitionBuilder< TDefinition >	
Base class for building XML definitions from model definitions	25
Models.BaseEntity	
Base class representing an entity with a position on the board	28
Models.Responses.BaseResponse< TResponse, TSelf >	
Base class for API responses	28
Database.Migrations.BootableDiskModel	
30	
Models.Database.BootableDiskModel	
Database model representing a bootable disk	31
Models.DTO.BootableDiskModelDTO	31
Services.Impl.BootableDiskProviderService	
Service providing information about available bootable disks	32
ServicesTests.BootableDiskProviderServiceTests	33
Repositories.Impl.BootableDiskRepository	33
RepositoriesTests.BootableDiskRepositoryTests	35
Models.Responses.BootableDisksResponse	35
Models.Responses.CapturedPacketResponse	36
Database.Migrations.ChangesToConnection	
36	
Database.Migrations.ChangeTypeOfVMEntityStrings	
37	
Models.Requests.ConnectEntitiesRequest	37
Models.ConnectionInfo	
Information about the connection to the virtualization hypervisor	38
Models.Responses.ConnectionInfoResponse	38
Hubs.ConnectionStatusHub	
SignalR hub for checking connection status	39
HubsTests.ConnectionStatusHubTests	40
Models.Requests.CreateVirtualMachineEntityRequest	40
Models.Requests.CreateVirtualNetworkNodeEntityRequest	40
Models.Requests.DeleteEntityRequest	40
Services.Impl.DeleteEntityService	
Service responsible for deleting entities from the system	41

Database.Migrations.DeviceDefinition	
43	
Services.Impl.EntitiesConnectorService	
Service responsible for connecting network entities	43
Services.Impl.EntitiesDisconnectService	
Service responsible for disconnecting network entities	46
Controllers.EntityController	
Controller managing entities in the system	49
Interconnect.Middlewares.ExceptionHandlerMiddleware	
Middleware for global exception handling in the application	53
InterconnectTests.ExceptionHandlerMiddlewareTests	54
Models.Responses.ExceptionResponse	54
Controllers.HypervisorConnectionController	
Controller managing hypervisor connection	55
ControllersTests.HypervisorConnectionControllerTests	56
Services.Impl.HypervisorConnectionService	
Service managing connection to the virtualization hypervisor	56
ServicesTests.HypervisorConnectionServiceTests	58
Services.IBootableDiskProviderService	
Service providing information about available bootable disks	58
Repositories.IBootableDiskRepository	
Repository managing bootable disks in the database	59
Services.IDeleteEntityService	60
Services.IEntitiesConnectorService	
Service responsible for connecting network entities	61
Services.IEntitiesDisconnectService	
Service responsible for disconnecting network entities	63
Services.IHypervisorConnectionService	
Service managing connection to the virtualization hypervisor	65
Services.IInfoService	
Service providing system information	67
Repositories.IInternetEntityRepository	
Repository managing Internet entities in the database	68
Services.IInternetEntityService	
Service managing Internet entities	70
NativeLibrary.Utils.INativeList< T >	72
Controllers.InfoController	
Controller providing system information	72
ControllersTests.InfoControllerTests	73
Models.InformationModel	
Model containing operating system information	73
Services.Impl.InfoService	
Service providing system information	74
ServicesTests.InfoServiceTests	75
Database.Migrations.InitialCreate	
75	
Services.Impl.InitializationService	
Service responsible for application initialization during startup	75
ServicesTests.InitializationServiceTests	77
Models.Config.InterconnectConfig	77
Database.InterconnectDbContext	
Database context for the Interconnect application	77
Models.Responses.InternetEntitiesResponse	78
Database.Migrations.InternetEntity	
79	
Models.Database.InternetEntityModel	
Database model representing an Internet entity connected to a virtual network	79
Models.DTO.InternetEntityModelDTO	80

Repositories.Impl.InternetEntityRepository	81
RepositoriesTests.InternetEntityRepositoryTests	83
Services.Impl.InternetEntityService	
Service managing Internet entities	83
Repositories.IPacketSnifferRepository	
Repository managing packet analyzers in memory	85
Services.IPacketSnifferService	
Service for capturing network packets	87
NativeLibrary.Wrappers.IPacketSnifferWrapper	
Interface wrapper for native packet analyzer library	89
NativeLibrary.Wrappers.IVirtualizationWrapper	
Interface wrapper for native virtualization library	90
Repositories.IVirtualMachineConsoleDataRepository	
Repository managing virtual machine console data in memory	96
Services.IVirtualMachineConsoleService	
Service managing virtual machine console	97
Repositories.IVirtualMachineConsoleStreamRepository	
Repository managing virtual machine console streams in memory	99
Repositories.IVirtualMachineEntityNetworkInterfaceRepository	
Repository managing virtual machine network interfaces in the database	101
Repositories.IVirtualMachineEntityRepository	
Repository managing virtual machine entities in the database	103
Services.IVirtualMachineEntityService	
Service managing virtual machine entities	105
Services.IVirtualMachineManagerService	
Service managing virtual machines in the hypervisor	108
Repositories.IVirtualNetworkConnectionRepository	
Repository managing entity connections in the virtual network	110
Services.IVirtualNetworkNodeConnector	
Service responsible for connecting virtual network nodes	112
Repositories.IVirtualNetworkNodeEntityRepository	
Repository managing virtual network nodes in the database	115
Repositories.IVirtualNetworkRepository	
Repository managing virtual networks in the database	118
Services.IVirtualNetworkService	
Service managing virtual networks	121
Database.Migrations.ModificationToVirtualNetwork	125
NativeLibrary.Structs.NativeConnectionInfo	126
NativeLibrary.Structs.NativeExecutionInfo	126
NativeLibrary.Utils.Impl.NativeList< T >	126
NativeLibrary.Structs.NativePacket	127
NativeLibrary.Structs.NativeStreamData	127
NativeLibrary.Structs.NativeStreamInfo	127
NativeLibrary.Structs.NativeVirtualMachineInfo	128
Models.Packet	
Represents a captured network packet	128
Models.PacketSniffer	
Represents a packet sniffer for a specific network bridge	129
BackgroundServices.Impl.PacketSnifferBackgroundService	
Background service for managing network packet sniffing and broadcasting	129
Hubs.PacketSnifferHub	
SignalR hub for transmitting captured network packets	130
Repositories.Impl.PacketSnifferRepository	131
Services.Impl.PacketSnifferService	
Service for capturing network packets	132
NativeLibrary.Wrappers.Impl.PacketSnifferWrapper	135

Interconnect.Program	
Main class of the Interconnect application	136
Database.Migrations.RemoveOsType	
137	
Database.Migrations.RenameSwitchToNode	
137	
Database.Migrations.RenameUuidToVmUuid	
138	
Models.Requests.SendDataToConsoleRequest	138
Models.StreamData	
Represents data read from a stream	138
Models.StreamInfo	
Information about a virtual machine console stream	139
Models.Responses.StringResponse	139
Models.Responses.SystemInfoResponse	140
Models.TerminalData	
Virtual machine terminal data	140
Database.Migrations.UpdateBootableDiskModelPath	
141	
Models.Requests.UpdateEntityPositionRequest	141
Database.Migrations.UpdateOperatingSystemType	
141	
NativeLibrary.Wrappers.Impl.VirtualizationWrapper	142
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService	
Background service for managing virtual machine console data streaming	147
Repositories.Impl.VirtualMachineConsoleDataRepository	149
RepositoriesTests.VirtualMachineConsoleDataRepositoryTests	150
Hubs.VirtualMachineConsoleHub	
SignalR hub for communicating with virtual machine console	150
HubsTests.VirtualMachineConsoleHubTests	152
Services.Impl.VirtualMachineConsoleService	
Service managing virtual machine console connections and data	152
ServicesTests.VirtualMachineConsoleServiceTests	155
Repositories.Impl.VirtualMachineConsoleStreamRepository	155
RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests	157
Controllers.VirtualMachineController	
Controller managing virtual machines	158
Models.VirtualMachineCreateDefinition	
Definition for creating a new virtual machine	159
Services.Utils.VirtualMachineDefinitionBuilder	
Builder class for creating virtual machine XML definitions	160
ServicesTests.Utils.VirtualMachineDefinitionBuilderTests	162
Models.Responses.VirtualMachineEntitiesResponse	162
Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO	163
Models.DTO.VirtualMachineEntityDTO	163
Database.Migrations.VirtualMachineEntityInterfaces	
164	
Models.DTO.VirtualMachineEntityMacAddressDTO	164
Models.Database.VirtualMachineEntityModel	
Database model representing a virtual machine entity	164
Models.Database.VirtualMachineEntityNetworkInterfaceModel	
Database model representing a virtual machine network interface	165
Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository	166
Repositories.Impl.VirtualMachineEntityRepository	168
RepositoriesTests.VirtualMachineEntityRepositoryTests	170
Models.Responses.VirtualMachineEntityResponse	171
Services.Impl.VirtualMachineEntityService	
Service managing virtual machine entities	171

ServicesTests.VirtualMachineEntityServiceTests	175
Models.VirtualMachineInfo	
Information about a virtual machine	175
Models.Responses.VirtualMachineInfoResponse	176
Services.Impl.VirtualMachineManagerService	
Service managing virtual machines in the hypervisor	177
ServicesTests.VirtualMachineManagerServiceTests	180
Models.Responses.VirtualMachinesInfoResponse	180
Database.Migrations.VirtualMachineType	
181	
Database.Migrations.VirtualNetwork	
181	
Models.DTO.VirtualNetworkConnectionDTO	182
Repositories.Impl.VirtualNetworkConnectionRepository	182
RepositoriesTests.VirtualNetworkConnectionRepositoryTests	184
Models.Responses.VirtualNetworkConnectionsResponse	185
Controllers.VirtualNetworkController	
Controller managing virtual networks and connections	185
Models.VirtualNetworkCreateDefinition	
Definition for creating a new virtual network	188
Services.Utills.VirtualNetworkCreateDefinitionBuilder	
Builder class for creating virtual network XML definitions	189
ServicesTests.Utills.VirtualNetworkDefinitionBuilderTests	191
Database.Migrations.VirtualNetworkEntities	
191	
Models.Database.VirtualNetworkEntityConnectionModel	
Database model representing a connection between entities in a virtual network	191
Models.Requests.VirtualNetworkEntityConnectionRequest	192
Models.VirtualNetworkInterfaceCreateDefinition	
Definition for creating a network interface for a virtual machine	192
Services.Utills.VirtualNetworkInterfaceCreateDefinitionBuilder	
Builder class for creating virtual network interface XML definitions	193
ServicesTests.Utills.VirtualNetworkInterfaceCreateDefinitionBuilderTests	195
Database.Migrations.VirtualNetworkIpAddress	
196	
Models.Database.VirtualNetworkModel	
Database model representing a virtual network	196
Services.Impl.VirtualNetworkNodeConnector	
Service responsible for connecting virtual network nodes	197
Models.DTO.VirtualNetworkNodeEntityDTO	199
Models.Database.VirtualNetworkNodeEntityModel	
Database model representing a virtual network node (switch/router)	199
Repositories.Impl.VirtualNetworkNodeEntityRepository	200
RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests	203
Models.Responses.VirtualNetworkNodesEntitiesResponse	203
Repositories.Impl.VirtualNetworkRepository	204
Services.Impl.VirtualNetworkService	
Service managing virtual networks	207
Database.Migrations.VirtualNetworkVarChar	
211	
Database.Migrations.VirtualSwitch	
212	

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**
Provides functionality for analyzing network packets from native packet data.
- 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 [DeleteEntityRequest](#)
- 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 [IDeleteEntityService](#)
- 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 **InitializationService**
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](#)
Service providing information about available bootable disks.
- class [DeleteEntityService](#)
Service responsible for deleting entities from the system.
- class [EntitiesConnectorService](#)
Service responsible for connecting network entities.
- class [EntitiesDisconnectService](#)
Service responsible for disconnecting network entities.
- class [HypervisorConnectionService](#)
Service managing connection to the virtualization hypervisor.
- class [InfoService](#)

- Service providing system information.*
- class [InitializationService](#)
 - Service responsible for application initialization during startup.*
- class [InternetEntityService](#)
 - Service managing Internet entities.*
- class [PacketSnifferService](#)
 - Service for capturing network packets.*
- class [VirtualMachineConsoleService](#)
 - Service managing virtual machine console connections and data.*
- class [VirtualMachineEntityService](#)
 - Service managing virtual machine entities.*
- class [VirtualMachineManagerService](#)
 - Service managing virtual machines in the hypervisor.*
- class [VirtualNetworkNodeConnector](#)
 - Service responsible for connecting virtual network nodes.*
- class [VirtualNetworkService](#)
 - Service managing virtual networks.*

4.32 Services.Utils Namespace Reference

Classes

- class [BaseDefinitionBuilder< TDefinition >](#)
 - Base class for building XML definitions from model definitions.*
- class **EntitiesUtils**
 - Utility class for entity-related operations and transformations.*
- class **MacAddressGenerator**
 - Utility class for generating random MAC addresses.*
- class [VirtualMachineDefinitionBuilder](#)
 - Builder class for creating virtual machine XML definitions.*
- class [VirtualNetworkCreateDefinitionBuilder](#)
 - Builder class for creating virtual network XML definitions.*
- class **VirtualNetworkDefinitionUtils**
 - Utility class for extracting information from virtual network definitions.*
- class [VirtualNetworkInterfaceCreateDefinitionBuilder](#)
 - Builder class for creating virtual network interface XML definitions.*
- class **VirtualNetworkUtils**
 - Utility class for virtual network name and UUID conversions.*

4.33 ServicesTests Namespace Reference

Classes

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

4.34 ServicesTests.Utils Namespace Reference

Classes

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

4.35 TestUtils Namespace Reference

Classes

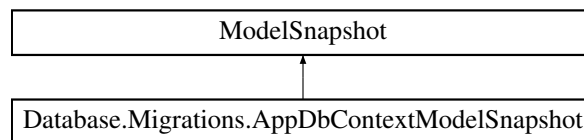
- class **ResponseExtractor**
- class **TestMocks**

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)

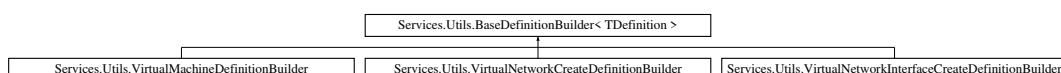
The documentation for this class was generated from the following file:

- Database/Migrations/AppDbContextModelSnapshot.cs

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

Base class for building XML definitions from model definitions.

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



Public Member Functions

- `BaseDefinitionBuilder< TDefinition > SetFromCreateDefinition (TDefinition definition)`
Sets the builder state from a create definition object.
- `string Build ()`
Builds the XML definition string.

Protected Member Functions

- `delegate void BuildingBlock (XmlWriter writer)`
Delegate for XML building blocks.
- `void ChecksEverythingIsProvided (params object?[] args)`
Validates that all required parameters are provided.
- `StringWriter CreateXmlWriter ()`

Protected Attributes

- `XmlWriter`
Creates an XML writer with appropriate settings for building definitions.

5.2.1 Detailed Description

Base class for building XML definitions from model definitions.

Template Parameters

<i>TDefinition</i>	The type of definition to build from.
--------------------	---------------------------------------

5.2.2 Member Function Documentation

5.2.2.1 Build()

```
string Services.Utills.BaseDefinitionBuilder< TDefinition >.Build () [abstract]
```

Builds the XML definition string.

Returns

The XML definition as a string.

5.2.2.2 BuildingBlock()

```
delegate void Services.Utills.BaseDefinitionBuilder< TDefinition >.BuildingBlock (
    XmlWriter writer) [protected]
```

Delegate for XML building blocks.

Parameters

<i>writer</i>	XML writer for building the definition.
---------------	---

5.2.2.3 CheckIsEverythingIsProvided()

```
void Services.Utills.BaseDefinitionBuilder< TDefinition >.CheckIsEverythingIsProvided (
    params object?[] args) [inline], [protected]
```

Validates that all required parameters are provided.

Parameters

<i>args</i>	Parameters to validate for non-null values.
-------------	---

Exceptions

<i>Exception</i>	Thrown when any parameter is null.
------------------	------------------------------------

5.2.2.4 SetFromCreateDefinition()

```
BaseDefinitionBuilder< TDefinition > Services.Utills.BaseDefinitionBuilder< TDefinition >.Set↔
FromCreateDefinition (
    TDefinition definition) [abstract]
```

Sets the builder state from a create definition object.

Parameters

<i>definition</i>	The definition to set from.
-------------------	-----------------------------

Returns

The builder instance for method chaining.

5.2.3 Member Data Documentation

5.2.3.1 XmlWriter

```
Services.Utills.BaseDefinitionBuilder< TDefinition >.XmlWriter [protected]
```

Creates an XML writer with appropriate settings for building definitions.

Returns

A tuple containing the XML writer and underlying string writer.

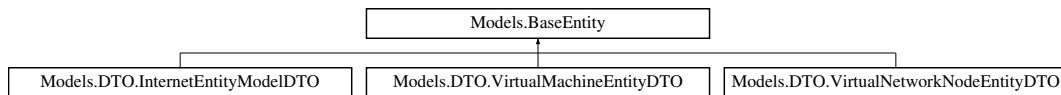
The documentation for this class was generated from the following file:

- Services/Utills/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.

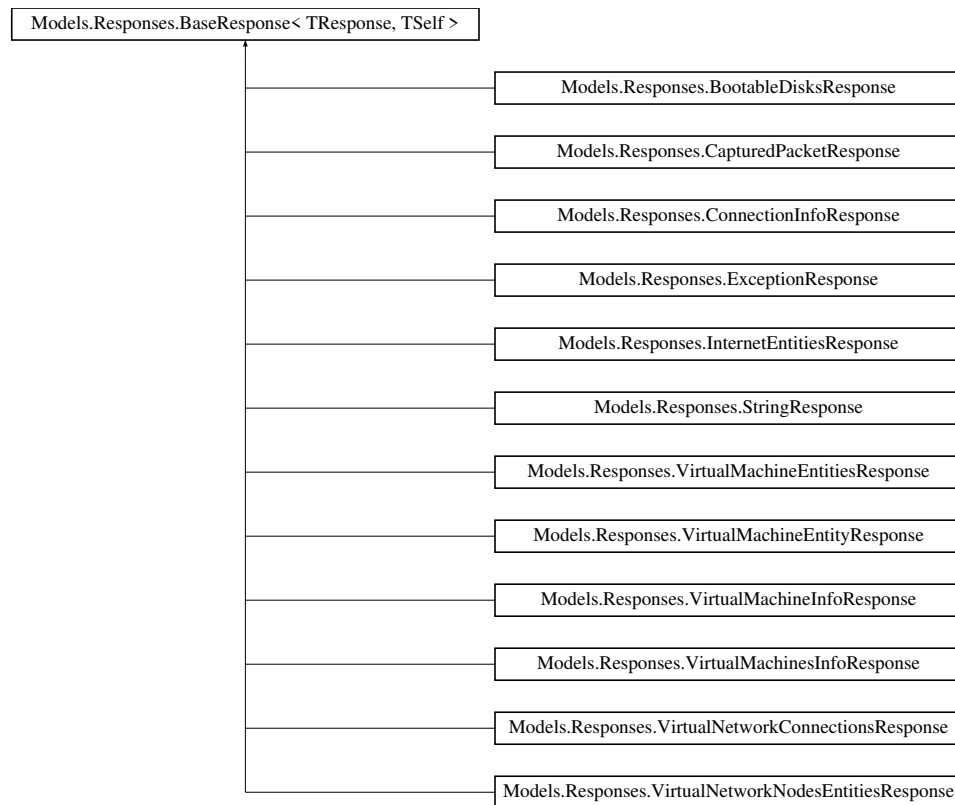
The documentation for this class was generated from the following file:

- 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

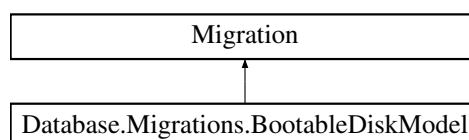
Successful response.

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]

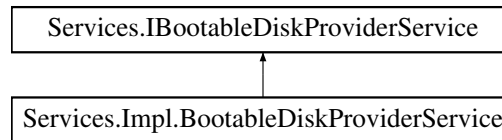
The documentation for this class was generated from the following file:

- Models/DTO/BootableDiskModelDTO.cs

5.8 Services.Impl.BootableDiskProviderService Class Reference

Service providing information about available bootable disks.

Inheritance diagram for Services.Impl.BootableDiskProviderService:



Public Member Functions

- [BootableDiskProviderService](#) ([IBootableDiskRepository](#) repository)
Initializes a new instance of the [BootableDiskProviderService](#).
- `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 Detailed Description

Service providing information about available bootable disks.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 BootableDiskProviderService()

```
Services.Impl.BootableDiskProviderService.BootableDiskProviderService (
    IBootableDiskRepository repository) [inline]
```

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

Parameters

<i>repository</i>	Repository for bootable disk models.
-------------------	--------------------------------------

5.8.3 Member Function Documentation

5.8.3.1 GetAvailableBootableDiskModels()

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

Retrieves a list of available bootable disk models.

Returns

List of bootable disks.

Implements [Services.IBootableDiskProviderService](#).

5.8.3.2 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

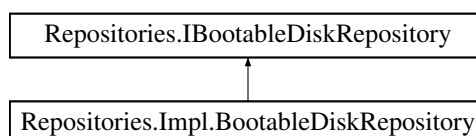
- void **SetUp** ()
- void **TearDown** ()
- async Task **GetAvailableBootableDiskModels_WhenInvoked_ShouldGetAllBootableDisksWithPath** ()
- async Task **GetBootableDiskPathById_WhenInvokedWithCorrectId_ShouldReturnBootableDiskPath** ()

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.GetOnly↵  
WithNotNullablePath () [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

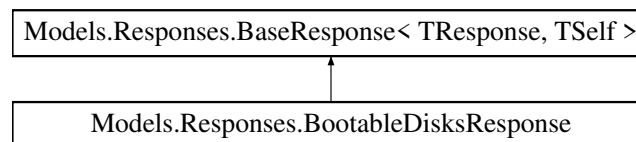
- void **SetUp** ()
- void **TearDown** ()
- async Task **GetOnlyWithNotNullablePath_WhenInvoked_ShouldReturnBootableDisksWithNotNullablePaths** ()
- async Task **GetById_WhenInvoked_ShouldGetBootableDiskUsingProvidedId** ()

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 >](#)

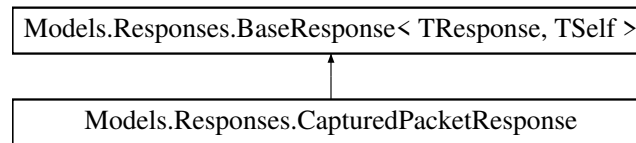
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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 >

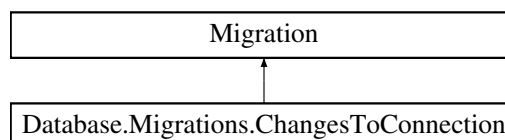
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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)

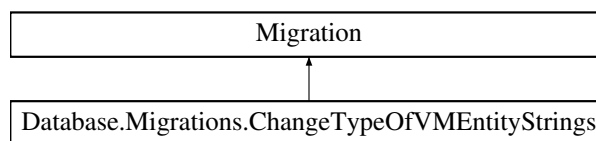
5.14.1 Detailed Description

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

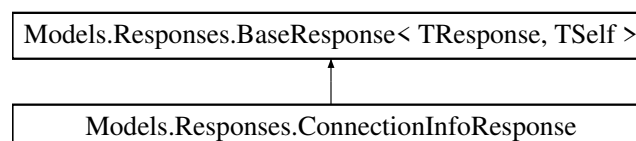
Information about the connection to the virtualization hypervisor.

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.

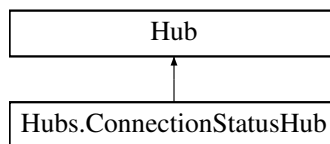
The documentation for this class was generated from the following file:

- 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 Models.Requests.DeleteEntityRequest Class Reference

Properties

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

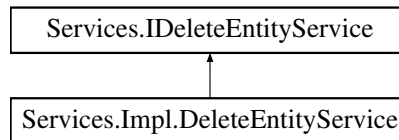
The documentation for this class was generated from the following file:

- Models/Requests/DeleteEntityRequest.cs

5.24 Services.Impl.DeleteEntityService Class Reference

Service responsible for deleting entities from the system.

Inheritance diagram for Services.Impl.DeleteEntityService:



Public Member Functions

- [DeleteEntityService](#) ([IInternetEntityRepository](#) internetEntityRepository, [IVirtualMachineEntityRepository](#) virtualMachineEntityRepository, [IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeEntityRepository, [IVirtualNetworkConnectionRepository](#) virtualNetworkConnectionRepository, [IEntitiesDisconnectorService](#) entitiesDisconnectorService)
Initializes a new instance of the [DeleteEntityService](#).
- async Task [DeleteInternetEntity](#) (int id)
Deletes an Internet entity.
- async Task [DeleteVirtualNetworkNodeEntity](#) (int id)
Deletes a virtual network node entity.
- async Task [DeleteVirtualMachineEntity](#) (int id)
Deletes a virtual machine entity.

5.24.1 Detailed Description

Service responsible for deleting entities from the system.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 DeleteEntityService()

```

Services.Impl.DeleteEntityService.DeleteEntityService (
    IInternetEntityRepository internetEntityRepository,
    IVirtualMachineEntityRepository virtualMachineEntityRepository,
    IVirtualNetworkNodeEntityRepository virtualNetworkNodeEntityRepository,
    IVirtualNetworkConnectionRepository virtualNetworkConnectionRepository,
    IEntitiesDisconnectorService entitiesDisconnectorService) [inline]
  
```

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

Parameters

<i>internetEntityRepository</i>	Repository for Internet entities.
<i>virtualMachineEntityRepository</i>	Repository for virtual machine entities.
<i>virtualNetworkNodeEntityRepository</i>	Repository for virtual network node entities.
<i>virtualNetworkConnectionRepository</i>	Repository for virtual network connections.
<i>entitiesDisconnectorService</i>	Service for disconnecting entities.

5.24.3 Member Function Documentation

5.24.3.1 DeleteInternetEntity()

```
async Task Services.Impl.DeleteEntityService.DeleteInternetEntity (
    int id) [inline]
```

Deletes an Internet entity.

Parameters

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

Implements [Services.IDeleteEntityService](#).

5.24.3.2 DeleteVirtualMachineEntity()

```
async Task Services.Impl.DeleteEntityService.DeleteVirtualMachineEntity (
    int id) [inline]
```

Deletes a virtual machine entity.

Parameters

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

Implements [Services.IDeleteEntityService](#).

5.24.3.3 DeleteVirtualNetworkNodeEntity()

```
async Task Services.Impl.DeleteEntityService.DeleteVirtualNetworkNodeEntity (
    int id) [inline]
```

Deletes a virtual network node entity.

Parameters

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

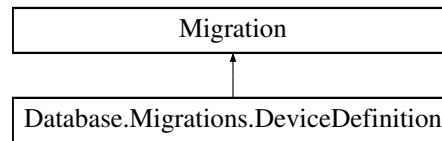
Implements [Services.IDeleteEntityService](#).

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

- Services/Impl/DeleteEntityService.cs

5.25 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)

5.25.1 Detailed Description

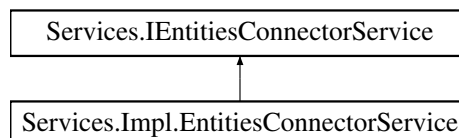
The documentation for this class was generated from the following files:

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

5.26 Services.Impl.EntitiesConnectorService Class Reference

Service responsible for connecting network entities.

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.

5.26.1 Detailed Description

Service responsible for connecting network entities.

5.26.2 Member Function Documentation

5.26.2.1 ConnectTwoEntities()

```
async Task< VirtualNetworkConnectionDTO > Services.Impl.EntitiesConnectorService.ConnectTwo↔  
Entities (   
    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](#).

5.26.2.2 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](#).

5.26.2.3 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>destinationEntityId</i>	Internet entity identifier.

Returns

Created connection data.

Implements [Services.IEntitiesConnectorService](#).

5.26.2.4 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>destinationEntityId</i>	Network node identifier.

Returns

Created connection data.

Implements [Services.IEntitiesConnectorService](#).

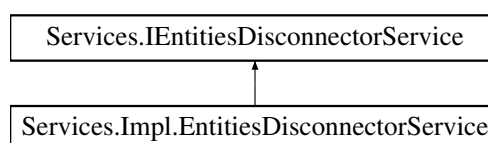
The documentation for this class was generated from the following file:

- Services/Impl/EntitiesConnectorService.cs

5.27 Services.Impl.EntitiesDisconnectService Class Reference

Service responsible for disconnecting network entities.

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

5.27.1 Detailed Description

Service responsible for disconnecting network entities.

5.27.2 Constructor & Destructor Documentation

5.27.2.1 EntitiesDisconnectService()

```
Services.Impl.EntitiesDisconnectService.EntitiesDisconnectService (
    IVirtualizationWrapper wrapper,
    IVirtualMachineEntityRepository vmEntityRepository,
    IVirtualNetworkConnectionRepository connectionRepository,
    IVirtualNetworkNodeEntityRepository virtualNetworkNodeRepository,
    IVirtualNetworkRepository networkRepository,
    IVirtualNetworkNodeConnector virtualNetworkNodeConnector,
    IVirtualMachineEntityNetworkInterfaceRepository networkInterfaceRepository) [inline]
```

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

Parameters

<i>wrapper</i>	Virtualization wrapper for hypervisor operations.
<i>vmEntityRepository</i>	Repository for virtual machine entities.
<i>connectionRepository</i>	Repository for network connections.
<i>virtualNetworkNodeRepository</i>	Repository for network nodes.
<i>networkRepository</i>	Repository for virtual networks.
<i>virtualNetworkNodeConnector</i>	Service for network node connections.
<i>networkInterfaceRepository</i>	Repository for network interfaces.

5.27.3 Member Function Documentation

5.27.3.1 DisconnectEntities()

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

Disconnects entities based on connection identifier.

Parameters

<i>connection↵ Id</i>	Connection identifier to remove.
---------------------------	----------------------------------

Returns

Removed connection data.

Implements [Services.IEntitiesDisconnectorService](#).

5.27.3.2 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>virtualMachine↵ EntityId</i>	Virtual machine identifier.
<i>internetEntityId</i>	Internet entity identifier.

Implements [Services.IEntitiesDisconnectorService](#).

5.27.3.3 DisconnectVirtualMachineFromVirtualMachine()

```
async Task Services.Impl.EntitiesDisconnectorService.DisconnectVirtualMachineFromVirtual↵
Machine (
    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](#).

5.27.3.4 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](#).

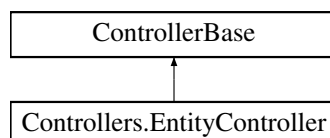
The documentation for this class was generated from the following file:

- Services/Impl/EntitiesDisconnectorService.cs

5.28 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, [IDeleteEntityService](#) deleteEntityService)
Initializes a new instance of the [EntityController](#).
- 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< [StringResponse](#) > > [DeleteEntity](#) ([DeleteEntityRequest](#) request)
Deletes an entity from the system.
- async Task< ActionResult< [VirtualMachineEntitiesResponse](#) > > [GetAllVirtualMachineEntities](#) ()
Retrieves all virtual machine entities with their MAC addresses.
- async Task< ActionResult< [VirtualNetworkNodesEntitiesResponse](#) > > [GetAllVirtualNetworkNodeEntities](#) ()
Retrieves all virtual network node entities.
- async Task< ActionResult< [InternetEntitiesResponse](#) > > [GetAllInternetEntities](#) ()
Retrieves all Internet entities.

5.28.1 Detailed Description

Controller managing entities in the system.

5.28.2 Constructor & Destructor Documentation

5.28.2.1 EntityController()

```
Controllers.EntityController.EntityController (
    IVirtualMachineEntityService vmEntityService,
    IVirtualNetworkService virtualNetworkService,
    IInternetEntityService internetEntityService,
    IDeleteEntityService deleteEntityService) [inline]
```

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

Parameters

<i>vmEntityService</i>	Service for managing virtual machine entities.
<i>virtualNetworkService</i>	Service for managing virtual networks.
<i>internetEntityService</i>	Service for managing Internet entities.
<i>deleteEntityService</i>	Service for deleting entities.

5.28.3 Member Function Documentation

5.28.3.1 CreateInternetEntity()

```
async Task< ActionResult< InternetEntitiesResponse > > Controllers.EntityController.Create↵  
InternetEntity () [inline]
```

Creates a new Internet entity.

Returns

Created Internet entity.

5.28.3.2 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.

5.28.3.3 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.

5.28.3.4 DeleteEntity()

```
async Task< ActionResult< StringResponse > > Controllers.EntityController.DeleteEntity (  
    DeleteEntityRequest request) [inline]
```

Deletes an entity from the system.

Parameters

<i>request</i>	Data for entity deletion.
----------------	---------------------------

Returns

Operation confirmation.

5.28.3.5 GetAllInternetEntities()

```
async Task< ActionResult< InternetEntitiesResponse > > Controllers.EntityController.GetAll↵  
InternetEntities () [inline]
```

Retrieves all Internet entities.

Returns

List of Internet entities.

5.28.3.6 GetAllVirtualMachineEntities()

```
async Task< ActionResult< VirtualMachineEntitiesResponse > > Controllers.EntityController.↵  
GetAllVirtualMachineEntities () [inline]
```

Retrieves all virtual machine entities with their MAC addresses.

Returns

List of virtual machine entities and MAC addresses.

5.28.3.7 GetAllVirtualNetworkNodeEntities()

```
async Task< ActionResult< VirtualNetworkNodesEntitiesResponse > > Controllers.EntityController.↵  
GetAllVirtualNetworkNodeEntities () [inline]
```

Retrieves all virtual network node entities.

Returns

List of virtual network nodes.

5.28.3.8 UpdateEntityPosition()

```
async Task< ActionResult< StringResponse > > Controllers.EntityController.UpdateEntity↵  
Position (↵  
    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

5.29 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.

5.29.1 Detailed Description

Middleware for global exception handling in the application.

5.29.2 Member Function Documentation

5.29.2.1 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

5.30 InterconnectTests.ExceptionHandlerMiddlewareTests Class Reference

Public Member Functions

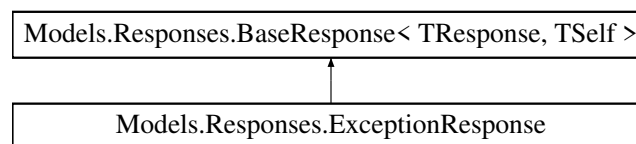
- async Task **InvokeAsync_NoException_CallsNextMiddleware** ()
- async Task **InvokeAsync_WhenExceptionThrown_ReturnsErrorResponse** ()

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

- InterconnectTests/ExceptionHandlerMiddlewareTests.cs

5.31 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.

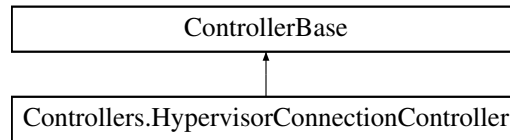
The documentation for this class was generated from the following file:

- Models/Responses/ExceptionResponse.cs

5.32 Controllers.HypervisorConnectionController Class Reference

Controller managing hypervisor connection.

Inheritance diagram for Controllers.HypervisorConnectionController:



Public Member Functions

- [HypervisorConnectionController](#) ([IHypervisorConnectionService](#) hypervisorConnectionService)
Initializes a new instance of the [HypervisorConnectionController](#).
- [ActionResult](#)< [ConnectionInfoResponse](#) > [ConnectionInfo](#) ()
Retrieves hypervisor connection information.

5.32.1 Detailed Description

Controller managing hypervisor connection.

5.32.2 Constructor & Destructor Documentation

5.32.2.1 HypervisorConnectionController()

```
Controllers.HypervisorConnectionController.HypervisorConnectionController (
    IHypervisorConnectionService hypervisorConnectionService) [inline]
```

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

Parameters

<i>hypervisorConnectionService</i>	Service for managing hypervisor connection.
------------------------------------	---

5.32.3 Member Function Documentation

5.32.3.1 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

5.33 ControllersTests.HypervisorConnectionControllerTests Class Reference

Public Member Functions

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

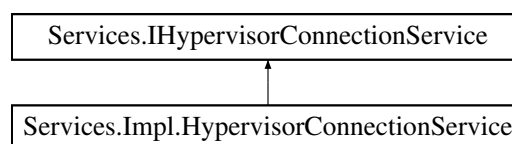
The documentation for this class was generated from the following file:

- ControllersTests/HypervisorConnectionControllerTests.cs

5.34 Services.Impl.HypervisorConnectionService Class Reference

Service managing connection to the virtualization hypervisor.

Inheritance diagram for Services.Impl.HypervisorConnectionService:



Public Member Functions

- [HypervisorConnectionService](#) (IVirtualizationWrapper vmManager)
Initializes a new instance of the [HypervisorConnectionService](#).
- 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.

5.34.1 Detailed Description

Service managing connection to the virtualization hypervisor.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 HypervisorConnectionService()

```
Services.Impl.HypervisorConnectionService.HypervisorConnectionService (
    IVirtualizationWrapper vmManager) [inline]
```

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

Parameters

<i>vmManager</i>	The virtualization wrapper for hypervisor operations.
------------------	---

5.34.3 Member Function Documentation

5.34.3.1 GetConnectionInfo()

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

Retrieves connection and hypervisor information.

Returns

Connection information.

Implements [Services.IHypervisorConnectionService](#).

5.34.3.2 GetConnectionStatus()

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

Retrieves the current connection status with the hypervisor.

Returns

Connection status.

Implements [Services.IHypervisorConnectionService](#).

5.34.3.3 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

5.35 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** ()

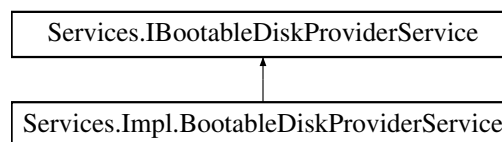
The documentation for this class was generated from the following file:

- ServicesTests/HypervisorConnectionServiceTests.cs

5.36 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.

5.36.1 Detailed Description

Service providing information about available bootable disks.

5.36.2 Member Function Documentation

5.36.2.1 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](#).

5.36.2.2 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](#).

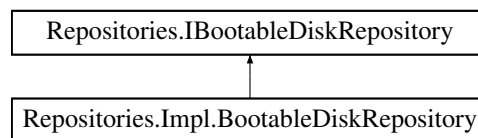
The documentation for this interface was generated from the following file:

- Services/IBootableDiskProviderService.cs

5.37 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.

5.37.1 Detailed Description

Repository managing bootable disks in the database.

5.37.2 Member Function Documentation

5.37.2.1 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](#).

5.37.2.2 GetOnlyWithNotNullablePath()

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

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

Returns

List of disks with paths.

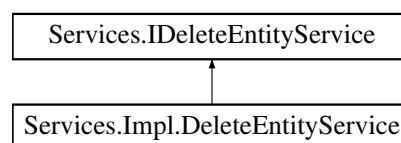
Implemented in [Repositories.Impl.BootableDiskRepository](#).

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

- Repositories/IBootableDiskRepository.cs

5.38 Services.IDeleteEntityService Interface Reference

Inheritance diagram for Services.IDeleteEntityService:

**Public Member Functions**

- Task [DeleteInternetEntity](#) (int id)
- Task [DeleteVirtualMachineEntity](#) (int id)
- Task [DeleteVirtualNetworkNodeEntity](#) (int id)

5.38.1 Member Function Documentation

5.38.1.1 DeleteInternetEntity()

```
Task Services.IDeleteEntityService.DeleteInternetEntity (
    int id)
```

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

5.38.1.2 DeleteVirtualMachineEntity()

```
Task Services.IDeleteEntityService.DeleteVirtualMachineEntity (
    int id)
```

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

5.38.1.3 DeleteVirtualNetworkNodeEntity()

```
Task Services.IDeleteEntityService.DeleteVirtualNetworkNodeEntity (
    int id)
```

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

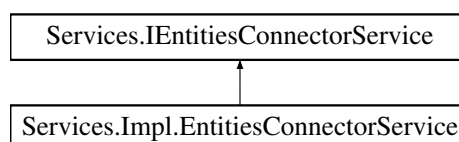
The documentation for this interface was generated from the following file:

- Services/IDeleteEntityService.cs

5.39 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.

5.39.1 Detailed Description

Service responsible for connecting network entities.

5.39.2 Member Function Documentation

5.39.2.1 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](#).

5.39.2.2 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](#).

5.39.2.3 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](#).

5.39.2.4 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](#).

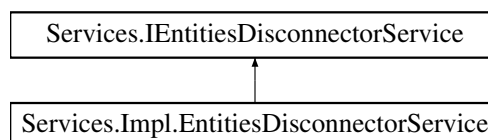
The documentation for this interface was generated from the following file:

- Services/IEntitiesConnectorService.cs

5.40 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.

5.40.1 Detailed Description

Service responsible for disconnecting network entities.

5.40.2 Member Function Documentation

5.40.2.1 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](#).

5.40.2.2 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](#).

5.40.2.3 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](#).

5.40.2.4 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](#).

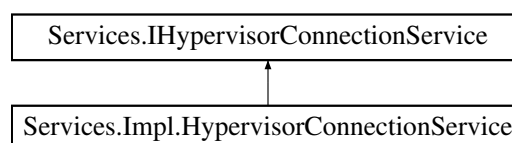
The documentation for this interface was generated from the following file:

- Services/IEntitiesDisconnectorService.cs

5.41 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.

5.41.1 Detailed Description

Service managing connection to the virtualization hypervisor.

5.41.2 Member Function Documentation

5.41.2.1 [GetConnectionInfo\(\)](#)

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

Retrieves connection and hypervisor information.

Returns

Connection information.

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

5.41.2.2 [GetConnectionStatus\(\)](#)

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

Retrieves the current connection status with the hypervisor.

Returns

Connection status.

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

5.41.2.3 [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](#).

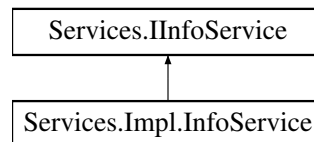
The documentation for this interface was generated from the following file:

- `Services/IHypervisorConnectionService.cs`

5.42 Services.IInfoService Interface Reference

Service providing system information.

Inheritance diagram for Services.IInfoService:



Public Member Functions

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

5.42.1 Detailed Description

Service providing system information.

5.42.2 Member Function Documentation

5.42.2.1 GetSystemInfo()

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

Retrieves operating system information.

Returns

System information.

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

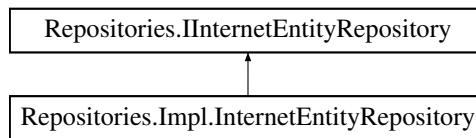
The documentation for this interface was generated from the following file:

- `Services/IInfoService.cs`

5.43 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.
- Task [Remove](#) (int id)
Deletes the entity with the specified identifier.

5.43.1 Detailed Description

Repository managing Internet entities in the database.

5.43.2 Member Function Documentation

5.43.2.1 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](#).

5.43.2.2 GetAll()

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

Retrieves all Internet entities.

Returns

List of Internet entities.

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

5.43.2.3 GetById()

```
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](#).

5.43.2.4 Remove()

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

Deletes the entity with the specified identifier.

Parameters

<i>id</i>	The unique identifier of the entity to delete.
-----------	--

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

5.43.2.5 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](#).

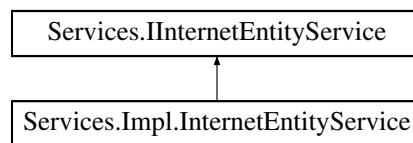
The documentation for this interface was generated from the following file:

- [Repositories/InternetEntityRepository.cs](#)

5.44 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.

5.44.1 Detailed Description

Service managing Internet entities.

5.44.2 Member Function Documentation

5.44.2.1 CreateInternet()

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

Creates a new Internet entity.

Returns

Created Internet entity.

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

5.44.2.2 GetInternetEntities()

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

Retrieves a list of all Internet entities.

Returns

List of Internet entities.

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

5.44.2.3 UpdateInternetEntityPosition()

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

Updates the position of an Internet entity on the board.

Parameters

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

Returns

Updated Internet entity.

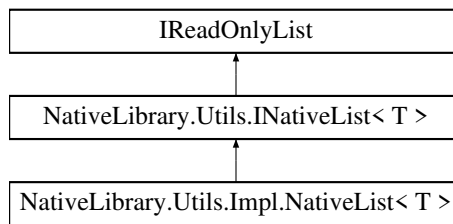
Implemented in [Services.Impl.InternetEntityService](#).

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

- Services/IInternetEntityService.cs

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

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



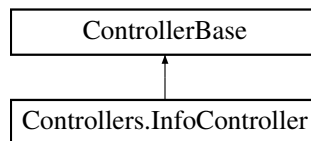
The documentation for this interface was generated from the following file:

- NativeLibrary/Utls/INativeList.cs

5.46 Controllers.InfoController Class Reference

Controller providing system information.

Inheritance diagram for Controllers.InfoController:



Public Member Functions

- [InfoController](#) ([IInfoService](#) infoService)
Initializes a new instance of the [InfoController](#).
- [ActionResult](#)< [SystemInfoResponse](#) > [GetSystemInfo](#) ()
Retrieves operating system information.

5.46.1 Detailed Description

Controller providing system information.

5.46.2 Constructor & Destructor Documentation

5.46.2.1 InfoController()

```

Controllers.InfoController.InfoController (
    IInfoService infoService) [inline]
  
```

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

Parameters

<i>infoService</i>	Service for providing system information.
--------------------	---

5.46.3 Member Function Documentation

5.46.3.1 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

5.47 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

5.48 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.

5.48.1 Detailed Description

Model containing operating system information.

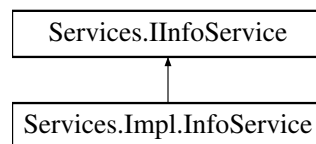
The documentation for this class was generated from the following file:

- Models/InformationModel.cs

5.49 Services.Impl.InfoService Class Reference

Service providing system information.

Inheritance diagram for Services.Impl.InfoService:



Public Member Functions

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

5.49.1 Detailed Description

Service providing system information.

5.49.2 Member Function Documentation

5.49.2.1 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

5.50 ServicesTests.InfoServiceTests Class Reference

Public Member Functions

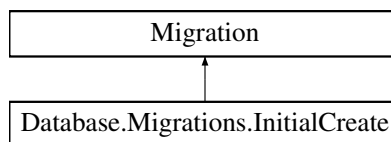
- void **GetOperatingSystem_Invoke_ShouldReturnOperatingSystem** ()

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

- ServicesTests/InfoServiceTests.cs

5.51 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)

5.51.1 Detailed Description

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

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

5.52 Services.Impl.InitializationService Class Reference

Service responsible for application initialization during startup.

Public Member Functions

- **InitializationService** (IHypervisorConnectionService connService, IOptions< [InterconnectConfig](#) > config)
Initializes a new instance of the [InitializationService](#).
- void **Initialize** ()
Initializes necessary application components.
- Task **StartAsync** (CancellationToken cancellationTokentoken)
Starts the initialization service asynchronously.
- Task **StopAsync** (CancellationToken cancellationTokentoken)
Stops the initialization service asynchronously.

5.52.1 Detailed Description

Service responsible for application initialization during startup.

5.52.2 Constructor & Destructor Documentation

5.52.2.1 InitializationService()

```
Services.Impl.InitializationService.InitializationService (  
    IHypervisorConnectionService connService,  
    IOptions< InterconnectConfig > config) [inline]
```

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

Parameters

<i>connService</i>	Service for managing hypervisor connection.
<i>config</i>	Configuration options for interconnect.

5.52.3 Member Function Documentation

5.52.3.1 StartAsync()

```
Task Services.Impl.InitializationService.StartAsync (  
    CancellationToken cancellationToken) [inline]
```

Starts the initialization service asynchronously.

Parameters

<i>cancellationToken</i>	Cancellation token for the operation.
--------------------------	---------------------------------------

Returns

A task representing the asynchronous start operation.

5.52.3.2 StopAsync()

```
Task Services.Impl.InitializationService.StopAsync (  
    CancellationToken cancellationToken) [inline]
```

Stops the initialization service asynchronously.

Parameters

<i>cancellationToken</i>	Cancellation token for the operation.
--------------------------	---------------------------------------

Returns

A task representing the asynchronous stop operation.

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

- Services/Impl/InitializationService.cs

5.53 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

5.54 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]
- required string **InternetEntityDefaultIp** [get, set]
- required string **InternetEntityDefaultNetmask** [get, set]

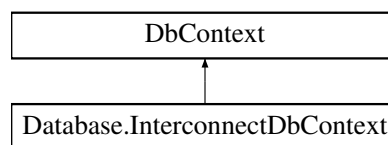
The documentation for this class was generated from the following file:

- Models/Config/InterconnectConfig.cs

5.55 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.

5.55.1 Detailed Description

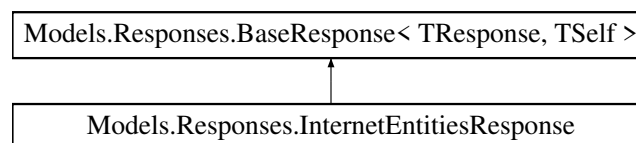
[Database](#) context for the [Interconnect](#) application.

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

- Database/InterconnectDbContext.cs

5.56 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 >](#)

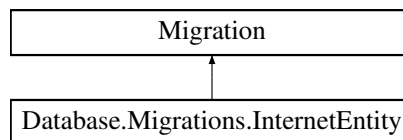
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/InternetEntitiesResponse.cs

5.57 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)

5.57.1 Detailed Description

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

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

5.58 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.

5.58.1 Detailed Description

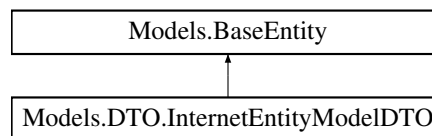
[Database](#) model representing an Internet entity connected to a virtual network.

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

- Models/Database/InternetEntityModel.cs

5.59 Models.DTO.InternetEntityModelDTO Class Reference

Inheritance diagram for Models.DTO.InternetEntityModelDTO:



Properties

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

Properties inherited from [Models.BaseEntity](#)

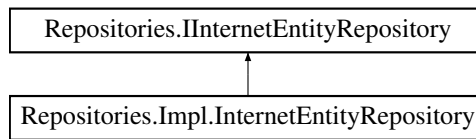
- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

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

- Models/DTO/InternetEntityModelDTO.cs

5.60 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 **Remove** (int id)
Deletes the entity with the specified identifier.
- 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.

5.60.1 Member Function Documentation

5.60.1.1 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](#).

5.60.1.2 GetAll()

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

Retrieves all Internet entities.

Returns

List of Internet entities.

Implements [Repositories.IInternetEntityRepository](#).

5.60.1.3 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](#).

5.60.1.4 Remove()

```
async Task Repositories.Impl.InternetEntityRepository.Remove (   
    int id) [inline]
```

Deletes the entity with the specified identifier.

Parameters

<i>id</i>	The unique identifier of the entity to delete.
-----------	--

Implements [Repositories.IInternetEntityRepository](#).

5.60.1.5 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

5.61 RepositoriesTests.InternetEntityRepositoryTests Class Reference

Public Member Functions

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

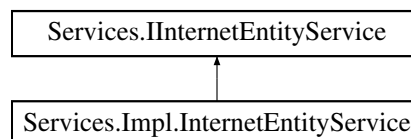
The documentation for this class was generated from the following file:

- RepositoriesTests/InternetEntityRepositoryTests.cs

5.62 Services.Impl.InternetEntityService Class Reference

Service managing Internet entities.

Inheritance diagram for Services.Impl.InternetEntityService:



Public Member Functions

- [InternetEntityService](#) ([IInternetEntityRepository](#) internetRepository, [IVirtualNetworkService](#) virtualNetworkService)
Initializes a new instance of the [InternetEntityService](#).
- 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.

5.62.1 Detailed Description

Service managing Internet entities.

5.62.2 Constructor & Destructor Documentation

5.62.2.1 InternetEntityService()

```
Services.Impl.InternetEntityService.InternetEntityService (
    IInternetEntityRepository internetRepository,
    IVirtualNetworkService virtualNetworkService) [inline]
```

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

Parameters

<i>internetRepository</i>	Repository for Internet entities.
<i>virtualNetworkService</i>	Service for managing virtual networks.

5.62.3 Member Function Documentation

5.62.3.1 CreateInternet()

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

Creates a new Internet entity.

Returns

Created Internet entity.

Implements [Services.IInternetEntityService](#).

5.62.3.2 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](#).

5.62.3.3 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](#).

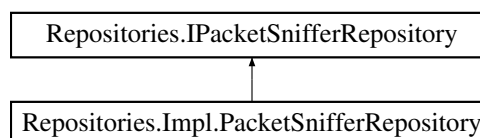
The documentation for this class was generated from the following file:

- Services/Impl/InternetEntityService.cs

5.63 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.

5.63.1 Detailed Description

Repository managing packet analyzers in memory.

5.63.2 Member Function Documentation

5.63.2.1 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](#).

5.63.2.2 GetOpenedSniffers()

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

Retrieves a list of opened packet analyzers.

Returns

List of opened analyzers.

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

5.63.2.3 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](#).

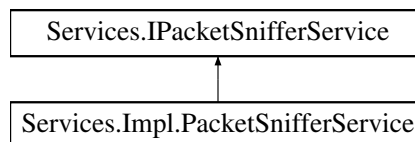
The documentation for this interface was generated from the following file:

- Repositories/IPacketSnifferRepository.cs

5.64 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.

5.64.1 Detailed Description

Service for capturing network packets.

5.64.2 Member Function Documentation

5.64.2.1 GetCapturedPacket()

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

Retrieves the last captured packet.

Returns

Captured packet or null if no packet available.

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

5.64.2.2 GetOpenedSniffers()

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

Retrieves a list of active packet analyzers.

Returns

List of opened packet sniffers.

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

5.64.2.3 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](#).

5.64.2.4 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](#).

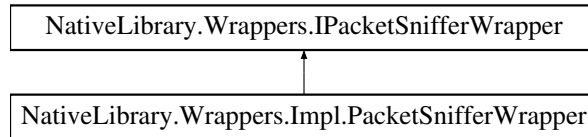
The documentation for this interface was generated from the following file:

- Services/IPacketSnifferService.cs

5.65 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.

5.65.1 Detailed Description

Interface wrapper for native packet analyzer library.

5.65.2 Member Function Documentation

5.65.2.1 GetNumberOfPackets()

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

Gets the number of captured packets.

Returns

Number of packets.

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

5.65.2.2 GetPacket()

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

Gets the last captured packet.

Returns

Captured packet.

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

5.65.2.3 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.

5.65.2.4 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](#).

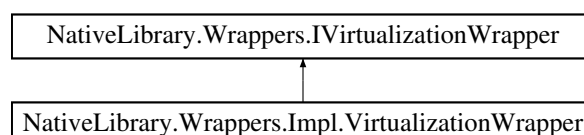
The documentation for this interface was generated from the following file:

- NativeLibrary/Wrappers/IPacketSnifferWrapper.cs

5.66 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.

5.66.1 Detailed Description

Interface wrapper for native virtualization library.

5.66.2 Member Function Documentation

5.66.2.1 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](#).

5.66.2.2 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](#).

5.66.2.3 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](#).

5.66.2.4 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](#).

5.66.2.5 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](#).

5.66.2.6 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](#).

5.66.2.7 GetConnectionInfo()

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

Gets connection information from the hypervisor.

Returns

Connection information.

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

5.66.2.8 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](#).

5.66.2.9 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](#).

5.66.2.10 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](#).

5.66.2.11 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](#).

5.66.2.12 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](#).

5.66.2.13 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](#).

5.66.2.14 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](#).

5.66.2.15 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](#).

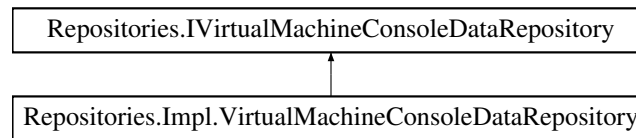
The documentation for this interface was generated from the following file:

- NativeLibrary/Wrappers/IVirtualizationWrapper.cs

5.67 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.

5.67.1 Detailed Description

Repository managing virtual machine console data in memory.

5.67.2 Member Function Documentation

5.67.2.1 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](#).

5.67.2.2 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](#).

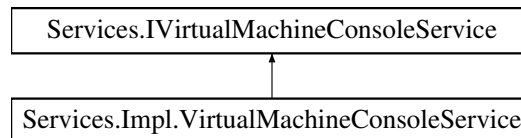
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualMachineConsoleDataRepository.cs](#)

5.68 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.

5.68.1 Detailed Description

Service managing virtual machine console.

5.68.2 Member Function Documentation

5.68.2.1 CloseStream()

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

Closes a console stream.

Parameters

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

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

5.68.2.2 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](#).

5.68.2.3 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](#).

5.68.2.4 GetStreams()

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

Retrieves a list of all opened streams.

Returns

List of opened streams.

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

5.68.2.5 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](#).

5.68.2.6 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](#).

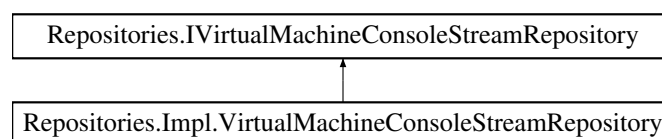
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineConsoleService.cs

5.69 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.

5.69.1 Detailed Description

Repository managing virtual machine console streams in memory.

5.69.2 Member Function Documentation

5.69.2.1 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](#).

5.69.2.2 GetAllStreams()

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

Retrieves all opened streams.

Returns

List of streams.

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

5.69.2.3 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](#).

5.69.2.4 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](#).

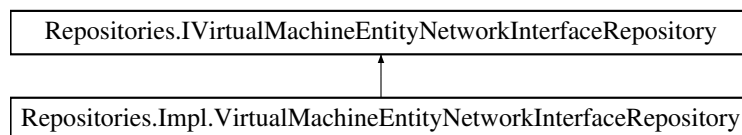
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualMachineConsoleStreamRepository.cs](#)

5.70 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.

5.70.1 Detailed Description

Repository managing virtual machine network interfaces in the database.

5.70.2 Member Function Documentation

5.70.2.1 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](#).

5.70.2.2 GetByIds()

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

Retrieves a network interface by virtual machine and connection identifiers.

Parameters

<i>virtual↔ MachineId</i>	Virtual machine identifier.
<i>connectionId</i>	Connection identifier.

Returns

Network interface or null.

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

5.70.2.3 GetByNetworkName()

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

Retrieves network interfaces connected to a specific network.

Parameters

<i>virtual↔ MachineId</i>	Virtual machine identifier.
<i>networkName</i>	Network name.

Returns

List of network interfaces.

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

5.70.2.4 GetByVirtualMachineId()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.IVirtualMachine↔  
EntityNetworkInterfaceRepository.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](#).

5.70.2.5 Remove()

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

Removes a network interface.

Parameters

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

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

5.70.2.6 Update()

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

Updates a network interface.

Parameters

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

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

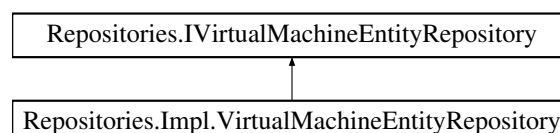
The documentation for this interface was generated from the following file:

- Repositories/IVirtualMachineEntityNetworkInterfaceRepository.cs

5.71 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.
- Task [Remove](#) (int id)
Removes the item with the specified identifier.

5.71.1 Detailed Description

Repository managing virtual machine entities in the database.

5.71.2 Member Function Documentation

5.71.2.1 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](#).

5.71.2.2 GetAll()

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

Retrieves all virtual machine entities.

Returns

List of entities.

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

5.71.2.3 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](#).

5.71.2.4 Remove()

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

Removes the item with the specified identifier.

Parameters

<i>id</i>	The unique identifier of the item to remove.
-----------	--

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

5.71.2.5 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](#).

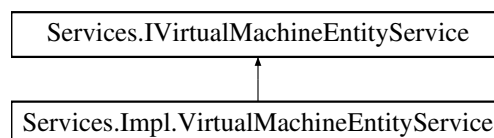
The documentation for this interface was generated from the following file:

- Repositories/IVirtualMachineEntityRepository.cs

5.72 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.

5.72.1 Detailed Description

Service managing virtual machine entities.

5.72.2 Member Function Documentation

5.72.2.1 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>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.

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

5.72.2.2 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](#).

5.72.2.3 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](#).

5.72.2.4 GetMacAddresses()

```
Task< List< VirtualMachineEntityMacAddressDTO > > Services.IVirtualMachineEntityService.Get↵
MacAddresses ()
```

Retrieves a list of virtual machine MAC addresses.

Returns

List of MAC addresses.

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

5.72.2.5 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](#).

5.72.2.6 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](#).

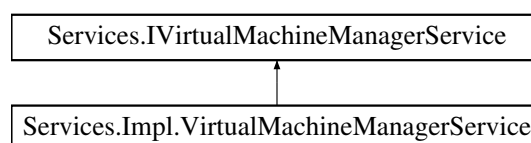
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineEntityService.cs

5.73 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.

5.73.1 Detailed Description

Service managing virtual machines in the hypervisor.

5.73.2 Member Function Documentation

5.73.2.1 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](#).

5.73.2.2 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](#).

5.73.2.3 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](#).

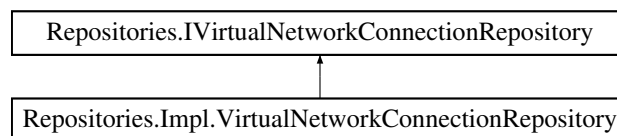
The documentation for this interface was generated from the following file:

- Services/IVirtualMachineManagerService.cs

5.74 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.

5.74.1 Detailed Description

Repository managing entity connections in the virtual network.

5.74.2 Member Function Documentation

5.74.2.1 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](#).

5.74.2.2 GetAll()

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

Retrieves all connections.

Returns

List of connections.

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

5.74.2.3 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](#).

5.74.2.4 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](#).

5.74.2.5 Remove()

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

Removes a connection.

Parameters

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

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

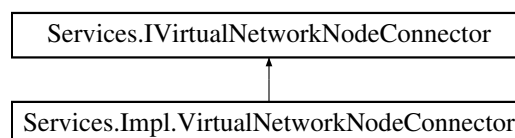
The documentation for this interface was generated from the following file:

- Repositories/IVirtualNetworkConnectionRepository.cs

5.75 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.

5.75.1 Detailed Description

Service responsible for connecting virtual network nodes.

5.75.2 Member Function Documentation

5.75.2.1 ConnectTwoVirtualNetworkNodes()

```
Task< VirtualNetworkConnectionDTO > Services.IVirtualNetworkNodeConnector.ConnectTwoVirtual↵  
NetworkNodes (  
    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](#).

5.75.2.2 ConnectVirtualNetworkNodeToInternet()

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

Connects a virtual network node to the Internet.

Parameters

<i>virtualNetworkNodeId</i>	Network node identifier.
<i>internetId</i>	Internet entity identifier.

Returns

Created connection data.

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

5.75.2.3 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](#).

5.75.2.4 DisconnectVirtualNetworkNodeFromInternet()

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

Disconnects a virtual network node from the Internet.

Parameters

<i>connectionId</i>	Connection identifier.
<i>virtualNetworkNodeId</i>	Network node identifier.

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

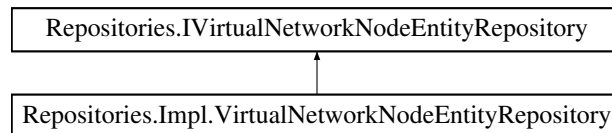
The documentation for this interface was generated from the following file:

- Services/IVirtualNetworkNodeConnector.cs

5.76 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.

5.76.1 Detailed Description

Repository managing virtual network nodes in the database.

5.76.2 Member Function Documentation

5.76.2.1 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](#).

5.76.2.2 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](#).

5.76.2.3 GetAll()

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

Retrieves all virtual network nodes.

Returns

List of nodes.

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

5.76.2.4 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](#).

5.76.2.5 GetVisible()

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

Retrieves visible virtual network nodes.

Returns

List of visible nodes.

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

5.76.2.6 Remove()

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

Removes a virtual network node.

Parameters

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

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

5.76.2.7 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](#).

5.76.2.8 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](#).

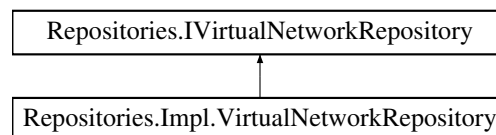
The documentation for this interface was generated from the following file:

- [Repositories/IVirtualNetworkNodeEntityRepository.cs](#)

5.77 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.

5.77.1 Detailed Description

Repository managing virtual networks in the database.

5.77.2 Member Function Documentation

5.77.2.1 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](#).

5.77.2.2 GetAll()

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

Retrieves all virtual networks.

Returns

List of virtual networks.

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

5.77.2.3 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](#).

5.77.2.4 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](#).

5.77.2.5 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](#).

5.77.2.6 GetByUuidWithVirtualNetworkNodes()

```
Task< VirtualNetworkModel > Repositories.IVirtualNetworkRepository.GetByUuidWithVirtual↔
NetworkNodes (
    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](#).

5.77.2.7 Remove()

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

Removes a virtual network.

Parameters

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

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

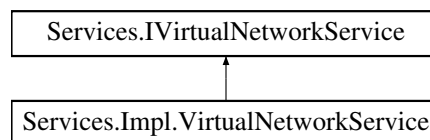
The documentation for this interface was generated from the following file:

- Repositories/IVirtualNetworkRepository.cs

5.78 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.

5.78.1 Detailed Description

Service managing virtual networks.

5.78.2 Member Function Documentation

5.78.2.1 [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](#).

5.78.2.2 CreateInternetVirtualNetwork()

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

Creates a virtual network for Internet connection.

Returns

Created virtual network.

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

5.78.2.3 CreateNodeVirtualNetwork()

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

Creates a virtual network for a network node.

Returns

Created virtual network.

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

5.78.2.4 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](#).

5.78.2.5 GetAllVirtualNetworks()

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

Retrieves all virtual networks.

Returns

List of virtual networks.

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

5.78.2.6 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](#).

5.78.2.7 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](#).

5.78.2.8 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](#).

5.78.2.9 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](#).

5.78.2.10 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.

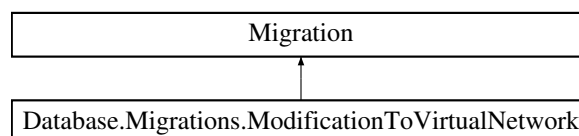
Implemented in [Services.Impl.VirtualNetworkService](#).

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

- Services/IVirtualNetworkService.cs

5.79 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)

5.79.1 Detailed Description

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

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

5.80 NativeLibrary.Structs.NativeConnectionInfo Struct Reference

Public Attributes

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

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

- NativeLibrary/Structs/NativeConnectionInfo.cs

5.81 NativeLibrary.Structs.NativeExecutionInfo Struct Reference

Public Attributes

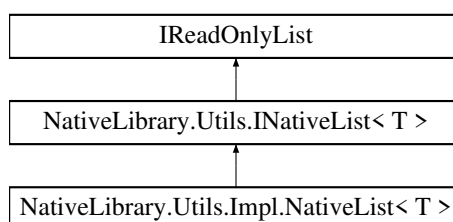
- bool **ErrorOccurred**
- string **Msg**

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

- NativeLibrary/Structs/NativeExecutionInfo.cs

5.82 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

5.83 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

5.84 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

5.85 NativeLibrary.Structs.NativeStreamInfo Struct Reference

Public Attributes

- string **Uuid**

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

- NativeLibrary/Structs/NativeStreamInfo.cs

5.86 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

5.87 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.

5.87.1 Detailed Description

Represents a captured network packet.

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

- Models/Packet.cs

5.88 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.

5.88.1 Detailed Description

Represents a packet sniffer for a specific network bridge.

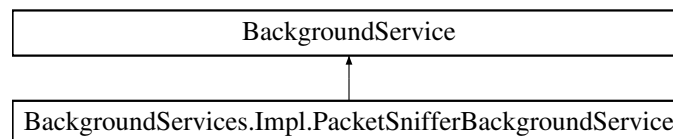
The documentation for this class was generated from the following file:

- Models/PacketSniffer.cs

5.89 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](#) (CancellationTokens stoppingToken)
Executes the background service to listen for and broadcast captured packets.

5.89.1 Detailed Description

Background service for managing network packet sniffing and broadcasting.

5.89.2 Constructor & Destructor Documentation

5.89.2.1 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.

5.89.3 Member Function Documentation

5.89.3.1 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.
----------------------	--

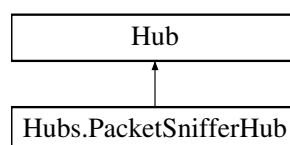
The documentation for this class was generated from the following file:

- BackgroundServices/Impl/PacketSnifferBackgroundService.cs

5.90 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.

5.90.1 Detailed Description

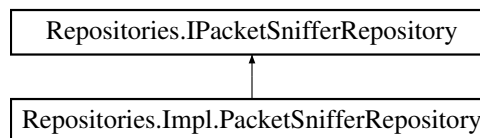
SignalR hub for transmitting captured network packets.

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

- Hubs/PackageSnifferHub.cs

5.91 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.

5.91.1 Member Function Documentation

5.91.1.1 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](#).

5.91.1.2 GetOpenedSniffers()

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

Retrieves a list of opened packet analyzers.

Returns

List of opened analyzers.

Implements [Repositories.IPacketSnifferRepository](#).

5.91.1.3 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](#).

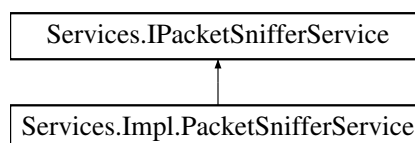
The documentation for this class was generated from the following file:

- Repositories/Impl/PacketSnifferRepository.cs

5.92 Services.Impl.PacketSnifferService Class Reference

Service for capturing network packets.

Inheritance diagram for Services.Impl.PacketSnifferService:



Public Member Functions

- [PacketSnifferService](#) ([IPacketSnifferWrapper](#) snifferWrapper, [IPacketSnifferRepository](#) packetSnifferRepository)
Initializes a new instance of the [PacketSnifferService](#).
- 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.

5.92.1 Detailed Description

Service for capturing network packets.

5.92.2 Constructor & Destructor Documentation

5.92.2.1 PacketSnifferService()

```
Services.Impl.PacketSnifferService.PacketSnifferService (
    IPacketSnifferWrapper snifferWrapper,
    IPacketSnifferRepository packetSnifferRepository) [inline]
```

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

Parameters

<i>snifferWrapper</i>	Wrapper for packet sniffing functionality.
<i>packetSnifferRepository</i>	Repository for managing packet sniffers.

5.92.3 Member Function Documentation

5.92.3.1 GetCapturedPacket()

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

Retrieves the last captured packet.

Returns

Captured packet or null if no packet available.

Implements [Services.IPacketSnifferService](#).

5.92.3.2 GetOpenedSniffers()

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

Retrieves a list of active packet analyzers.

Returns

List of opened packet sniffers.

Implements [Services.IPacketSnifferService](#).

5.92.3.3 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](#).

5.92.3.4 StartListeningForBridge()

```
void Services.Impl.PacketSnifferService.StartListeningForBridge (
    string bridgeName) [inline]
```

Starts listening for packets on a specified network bridge.

Parameters

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

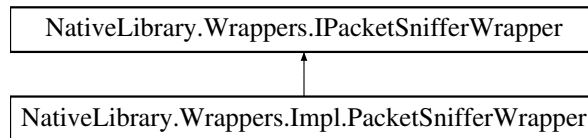
Implements [Services.IPacketSnifferService](#).

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

- Services/Impl/PacketSnifferService.cs

5.93 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** (nint handler, string bridgeName)

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

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

5.93.1 Member Function Documentation

5.93.1.1 GetNumberOfPackets()

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

Gets the number of captured packets.

Returns

Number of packets.

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

5.93.1.2 GetPacket()

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

Gets the last captured packet.

Returns

Captured packet.

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

5.93.1.3 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](#)

5.94 Interconnect.Program Class Reference

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

Static Public Member Functions

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

5.94.1 Detailed Description

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

5.94.2 Member Function Documentation

5.94.2.1 Main()

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

Application entry point.

Parameters

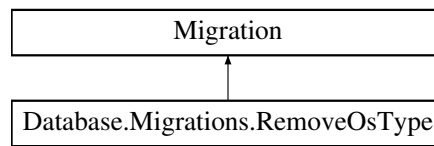
<i>args</i>	Command line arguments.
-------------	-------------------------

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

- [Interconnect/Program.cs](#)

5.95 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)

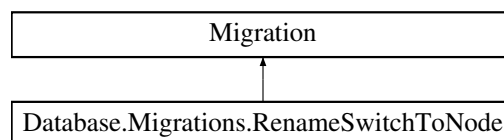
5.95.1 Detailed Description

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

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

5.96 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)

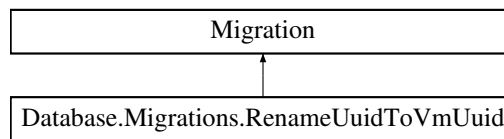
5.96.1 Detailed Description

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

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

5.97 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)

5.97.1 Detailed Description

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

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

5.98 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

5.99 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.

5.99.1 Detailed Description

Represents data read from a stream.

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

- Models/StreamData.cs

5.100 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.

5.100.1 Detailed Description

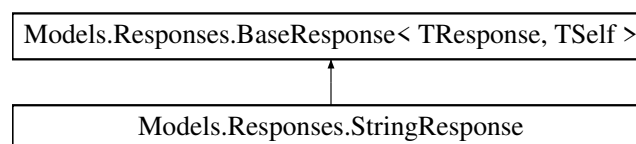
Information about a virtual machine console stream.

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

- Models/StreamInfo.cs

5.101 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

5.102 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

5.103 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.

5.103.1 Detailed Description

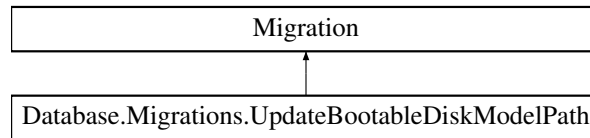
Virtual machine terminal data.

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

- Models/TerminalData.cs

5.104 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)

5.104.1 Detailed Description

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

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

5.105 Models.Requests.UpdateEntityPositionRequest Class Reference

Properties

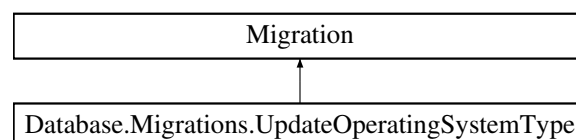
- required int **Id** [get, set]
- required [EntityType](#) **Type** [get, set]
- required int **X** [get, set]
- required int **Y** [get, set]

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

- Models/Requests/UpdateEntityPositionRequest.cs

5.106 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)

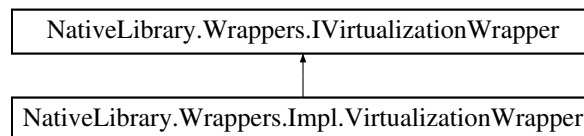
5.106.1 Detailed Description

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

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

5.107 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.

5.107.1 Member Function Documentation

5.107.1.1 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](#).

5.107.1.2 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](#).

5.107.1.3 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](#).

5.107.1.4 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](#).

5.107.1.5 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](#).

5.107.1.6 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](#).

5.107.1.7 GetConnectionInfo()

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

Gets connection information from the hypervisor.

Returns

Connection information.

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

5.107.1.8 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](#).

5.107.1.9 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](#).

5.107.1.10 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.IVirtualizationWrapper](#).

5.107.1.11 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.IVirtualizationWrapper](#).

5.107.1.12 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.IVirtualizationWrapper](#).

5.107.1.13 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.IVirtualizationWrapper](#).

5.107.1.14 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](#).

5.107.1.15 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](#).

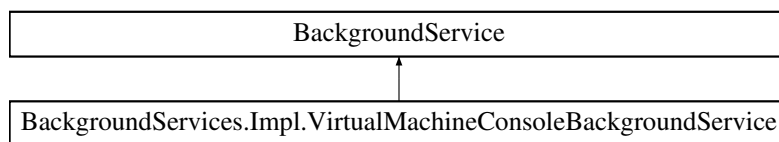
The documentation for this class was generated from the following file:

- NativeLibrary/Wrappers/Impl/VirtualizationWrapper.cs

5.108 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, IHttpContext< [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.

5.108.1 Detailed Description

Background service for managing virtual machine console data streaming.

5.108.2 Constructor & Destructor Documentation

5.108.2.1 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.

5.108.3 Member Function Documentation

5.108.3.1 ExecuteAsync()

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

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

Parameters

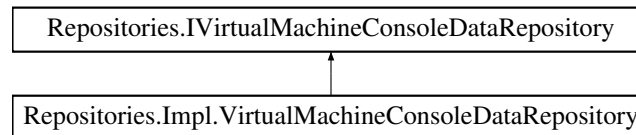
<i>stoppingToken</i>	Cancellation token for stopping the service.
----------------------	--

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

- BackgroundServices/Impl/VirtualMachineConsoleBackgroundService.cs

5.109 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.

5.109.1 Member Function Documentation

5.109.1.1 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](#).

5.109.1.2 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

5.110 RepositoriesTests.VirtualMachineConsoleDataRepositoryTests Class Reference

Public Member Functions

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

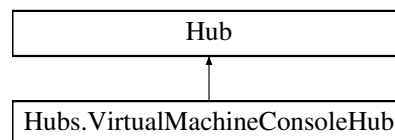
The documentation for this class was generated from the following file:

- RepositoriesTests/VirtualMachineConsoleDataRepositoryTests.cs

5.111 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.

5.111.1 Detailed Description

SignalR hub for communicating with virtual machine console.

5.111.2 Member Function Documentation

5.111.2.1 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.

5.111.2.2 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.
-------------	-----------------------

5.111.2.3 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.
-------------	-----------------------

5.111.2.4 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

5.112 HubsTests.VirtualMachineConsoleHubTests Class Reference

Public Member Functions

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

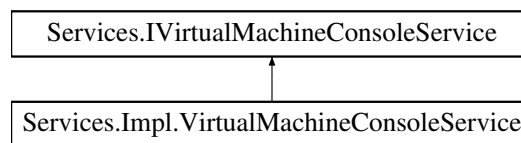
The documentation for this class was generated from the following file:

- HubsTests/VirtualMachineConsoleHubTests.cs

5.113 Services.Impl.VirtualMachineConsoleService Class Reference

Service managing virtual machine console connections and data.

Inheritance diagram for Services.Impl.VirtualMachineConsoleService:



Public Member Functions

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

5.113.1 Detailed Description

Service managing virtual machine console connections and data.

5.113.2 Constructor & Destructor Documentation

5.113.2.1 VirtualMachineConsoleService()

```
Services.Impl.VirtualMachineConsoleService.VirtualMachineConsoleService (
    IVirtualizationWrapper virtualizationWrapper,
    IVirtualMachineConsoleDataRepository consoleDataRepository,
    IVirtualMachineConsoleStreamRepository consoleStreamRepository) [inline]
```

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

Parameters

<i>virtualizationWrapper</i>	Virtualization wrapper for console operations.
<i>consoleDataRepository</i>	Repository for console data.
<i>consoleStreamRepository</i>	Repository for console streams.

5.113.3 Member Function Documentation

5.113.3.1 CloseStream()

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

Closes a console stream connection.

Parameters

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

Implements [Services.IVirtualMachineConsoleService](#).

5.113.3.2 GetBytesFromConsole()

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

Gets bytes from the console stream.

Parameters

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

Returns

Data retrieved from the console stream.

Implements [Services.IVirtualMachineConsoleService](#).

5.113.3.3 GetInitialConsoleData()

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

Retrieves initial console data for a virtual machine.

Parameters

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

Returns

Array of console data bytes.

Implements [Services.IVirtualMachineConsoleService](#).

5.113.3.4 GetStreams()

```
List< StreamInfo > Services.Impl.VirtualMachineConsoleService.GetStreams () [inline]
```

Retrieves all active console streams.

Returns

List of active console streams.

Implements [Services.IVirtualMachineConsoleService](#).

5.113.3.5 OpenVirtualMachineConsole()

```
StreamInfo Services.Impl.VirtualMachineConsoleService.OpenVirtualMachineConsole (
    Guid uuid) [inline]
```

Opens a console connection for a virtual machine.

Parameters

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

Returns

Stream information for the opened console.

Implements [Services.IVirtualMachineConsoleService](#).

5.113.3.6 SendBytesToVirtualMachineConsoleByUuid()

```
void Services.Impl.VirtualMachineConsoleService.SendBytesToVirtualMachineConsoleByUuid (
    Guid uuid,
    string data) [inline]
```

Sends bytes to a virtual machine console.

Parameters

<i>uuid</i>	Virtual machine UUID.
<i>data</i>	Data to send to the console.

Implements [Services.IVirtualMachineConsoleService](#).

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

- Services/Impl/VirtualMachineConsoleService.cs

5.114 ServicesTests.VirtualMachineConsoleServiceTests Class Reference

Public Member Functions

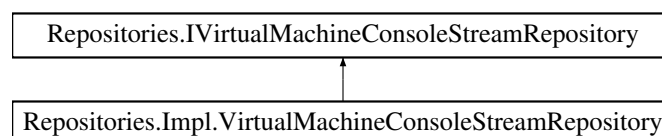
- void **Setup** ()
- void **GetInitialConsoleData_WhenInvoked_ShouldReturnInitialConsoleData** ()
- void **GetBytesFromConsole_WhenInvoked_ShouldGetBytesFromConsole** ()
- void **GetBytesFromConsole_WhenInvoked_ShouldSaveBytesToRepository** ()
- void **OpenVirtualMachineConsole_WhenInvoked_ShouldOpenVirtualMachineConsole** ()
- void **OpenVirtualMachineConsole_WhenInvokedAndVmCreated_ShouldAddStreamToRepository** ()
- void **SendBytesToVirtualMachineConsoleByUuid_WhenInvoked_ShouldSendDataToStream** ()
- void **SendBytesToVirtualMachineConsoleByUuid_WhenInvokedWithNotOpenedConsole_Should↵
OpenConsole** ()
- 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

5.115 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.

5.115.1 Member Function Documentation

5.115.1.1 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](#).

5.115.1.2 GetAllStreams()

```
List< StreamInfo > Repositories.Impl.VirtualMachineConsoleStreamRepository.GetAllStreams ()  
[inline]
```

Retrieves all opened streams.

Returns

List of streams.

Implements [Repositories.IVirtualMachineConsoleStreamRepository](#).

5.115.1.3 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](#).

5.115.1.4 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

5.116 RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests Class Reference

Public Member Functions

- void **Setup** ()
- void **Add_WhenInvoked_ShouldAddNewStream** ()
- void **GetAllStreams_WhenInvoked_ShouldGetAllStreams** ()
- void **Remove_WhenInvoked_ShouldRemoveStream** ()

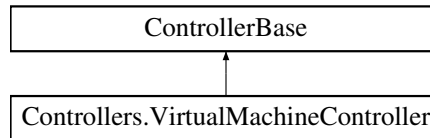
The documentation for this class was generated from the following file:

- RepositoriesTests/VirtualMachineConsoleStreamRepositoryTests.cs

5.117 Controllers.VirtualMachineController Class Reference

Controller managing virtual machines.

Inheritance diagram for Controllers.VirtualMachineController:



Public Member Functions

- [VirtualMachineController](#) ([IBootableDiskProviderService](#) bootableDiskProviderService)
Initializes a new instance of the [VirtualMachineController](#).
- `async Task< ActionResult< BootableDisksResponse > > GetAvailableBootableDisks ()`
Retrieves a list of available bootable disks.

5.117.1 Detailed Description

Controller managing virtual machines.

5.117.2 Constructor & Destructor Documentation

5.117.2.1 VirtualMachineController()

```

Controllers.VirtualMachineController.VirtualMachineController (
    IBootableDiskProviderService bootableDiskProviderService) [inline]
  
```

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

Parameters

<i>bootableDiskProviderService</i>	Service for providing bootable disk information.
------------------------------------	--

5.117.3 Member Function Documentation

5.117.3.1 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

5.118 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.

5.118.1 Detailed Description

Definition for creating a new virtual machine.

5.118.2 Member Function Documentation

5.118.2.1 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.

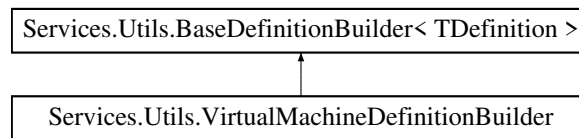
The documentation for this class was generated from the following file:

- Models/VirtualMachineCreateDefinition.cs

5.119 Services.Utls.VirtualMachineDefinitionBuilder Class Reference

Builder class for creating virtual machine XML definitions.

Inheritance diagram for Services.Utls.VirtualMachineDefinitionBuilder:



Public Member Functions

- [VirtualMachineDefinitionBuilder SetPrefix](#) (string prefix)
Sets the VM name prefix.
- override [VirtualMachineDefinitionBuilder SetFromCreateDefinition](#) ([VirtualMachineCreateDefinition](#) definition)
Sets the builder state from a virtual machine create definition.
- override string [Build](#) ()
Builds the complete virtual machine XML definition.

Public Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition > SetFromCreateDefinition](#) (TDefinition definition)
Sets the builder state from a create definition object.
- string [Build](#) ()
Builds the XML definition string.

Additional Inherited Members

Protected Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- delegate void [BuildingBlock](#) ([XmlWriter](#) writer)
Delegate for XML building blocks.
- void [CheckIsEverythingIsProvided](#) (params object?[] args)
Validates that all required parameters are provided.
- [StringWriter CreateXmlWriter](#) ()

Protected Attributes inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [XmlWriter](#)
Creates an XML writer with appropriate settings for building definitions.

5.119.1 Detailed Description

Builder class for creating virtual machine XML definitions.

5.119.2 Member Function Documentation

5.119.2.1 Build()

```
override string Services.Utills.VirtualMachineDefinitionBuilder.Build () [inline]
```

Builds the complete virtual machine XML definition.

Returns

The XML definition as a string.

5.119.2.2 SetFromCreateDefinition()

```
override VirtualMachineDefinitionBuilder Services.Utills.VirtualMachineDefinitionBuilder.SetFromCreateDefinition (
    VirtualMachineCreateDefinition definition) [inline]
```

Sets the builder state from a virtual machine create definition.

Parameters

<i>definition</i>	The VM creation definition.
-------------------	-----------------------------

Returns

The builder instance for method chaining.

5.119.2.3 SetPrefix()

```
VirtualMachineDefinitionBuilder Services.Utills.VirtualMachineDefinitionBuilder.SetPrefix (
    string prefix) [inline]
```

Sets the VM name prefix.

Parameters

<i>prefix</i>	The prefix for VM names.
---------------	--------------------------

Returns

The builder instance for method chaining.

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

- Services/Utills/VirtualMachineDefinitionBuilder.cs

5.120 ServicesTests.Utils.VirtualMachineDefinitionBuilderTests Class Reference

Public Member Functions

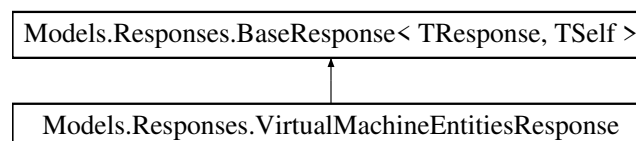
- void **OnSetup** ()
- void **Build_WhenNothingsProvided_ShouldThrowException** ()
- void **SetFromCreateDefinition_WhenPrefixIsNotProvided_ShouldThrowException** ()
- void **Build_WhenBootableDiskPathProvided_ShouldIncludeBootableDiskPathInXml** ()
- void **Build_WhenMemoryProvided_ShouldIncludeMemoryInXml** ()
- void **Build_WhenNameProvided_ShouldIncludeNameInXml** ()
- void **Build_WhenVirtualCpusProvided_ShouldIncludeVirtualCpusInXml** ()

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

- ServicesTests/Utils/VirtualMachineDefinitionBuilderTests.cs

5.121 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

5.122 Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO Class Reference

Properties

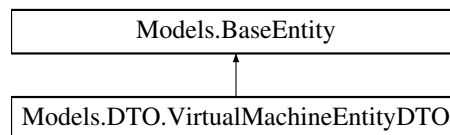
- required List< [VirtualMachineEntityDTO](#) > **VirtualMachineEntities** [get, set]
- required List< [VirtualMachineEntityMacAddressDTO](#) > **MacAddressEntities** [get, set]

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

- Models/DTO/VirtualMachineEntitiesWithMacAddressesDTO.cs

5.123 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](#)

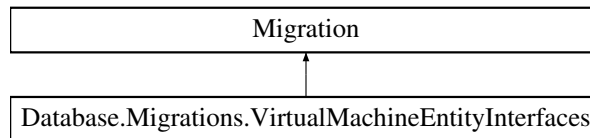
- required int **Id** [get, set]
Entity identifier.
- required int **X** [get, set]
X coordinate of the entity position on the board.
- required int **Y** [get, set]
Y coordinate of the entity position on the board.

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

- Models/DTO/VirtualMachineEntityDTO.cs

5.124 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)

5.124.1 Detailed Description

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

- Database/Migrations/20260118124006_VirtualMachineEntityInterfaces.cs
- Database/Migrations/20260118124006_VirtualMachineEntityInterfaces.Designer.cs

5.125 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

5.126 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.

5.126.1 Detailed Description

[Database](#) model representing a virtual machine entity.

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

- Models/Database/VirtualMachineEntityModel.cs

5.127 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.

5.127.1 Detailed Description

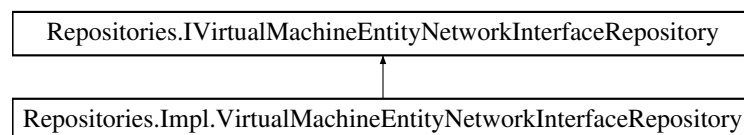
[Database](#) model representing a virtual machine network interface.

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

- Models/Database/VirtualMachineEntityNetworkInterfaceModel.cs

5.128 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.

5.128.1 Member Function Documentation

5.128.1.1 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](#).

5.128.1.2 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](#).

5.128.1.3 GetByNetworkName()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.Impl.VirtualMachine↵  
EntityNetworkInterfaceRepository.GetByNetworkName (   
    int virtualMachineId,   
    string networkName)
```

Retrieves network interfaces connected to a specific network.

Parameters

<i>virtual↵ MachineId</i>	Virtual machine identifier.
<i>networkName</i>	Network name.

Returns

List of network interfaces.

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.128.1.4 GetByVirtualMachineId()

```
Task< List< VirtualMachineEntityNetworkInterfaceModel > > Repositories.Impl.VirtualMachine↵  
EntityNetworkInterfaceRepository.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](#).

5.128.1.5 Remove()

```
Task Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.Remove (
    VirtualMachineEntityNetworkInterfaceModel networkInterface) [inline]
```

Removes a network interface.

Parameters

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

Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

5.128.1.6 Update()

```
Task Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository.Update (
    VirtualMachineEntityNetworkInterfaceModel networkInterface) [inline]
```

Updates a network interface.

Parameters

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

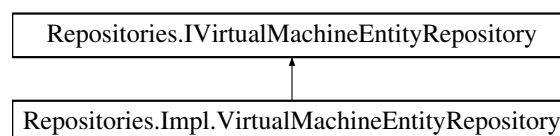
Implements [Repositories.IVirtualMachineEntityNetworkInterfaceRepository](#).

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

- Repositories/Impl/VirtualMachineEntityNetworkInterfaceRepository.cs

5.129 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 [Remove](#) (int id)
Removes the item with the specified identifier.
- async Task [Update](#) ([VirtualMachineEntityModel](#) model)
Updates a virtual machine entity.

5.129.1 Member Function Documentation

5.129.1.1 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](#).

5.129.1.2 GetAll()

```
async Task< List< VirtualMachineEntityModel > > Repositories.Impl.VirtualMachineEntity↵
Repository.GetAll () [inline]
```

Retrieves all virtual machine entities.

Returns

List of entities.

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.129.1.3 GetById()

```
async Task< VirtualMachineEntityModel > Repositories.Impl.VirtualMachineEntityRepository.Get↵
ById (
    int id) [inline]
```

Retrieves a virtual machine entity by identifier.

Parameters

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

Returns

Virtual machine entity.

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.129.1.4 Remove()

```
async Task Repositories.Impl.VirtualMachineEntityRepository.Remove (  
    int id) [inline]
```

Removes the item with the specified identifier.

Parameters

<i>id</i>	The unique identifier of the item to remove.
-----------	--

Implements [Repositories.IVirtualMachineEntityRepository](#).

5.129.1.5 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

5.130 RepositoriesTests.VirtualMachineEntityRepositoryTests Class Reference

Public Member Functions

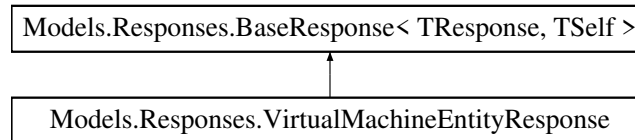
- void **Setup** ()
- void **Teardown** ()
- async Task **Add_WhenInvoked_ShouldAddEntity** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllEntities** ()
- async Task **GetById_WhenInvoked_ShouldGetEntityById** ()
- async Task **Update_WhenInvoked_ShouldUpdateEntity** ()

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

- RepositoriesTests/VirtualMachineEntityRepositoryTests.cs

5.131 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.

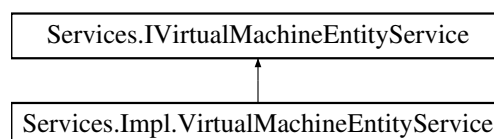
The documentation for this class was generated from the following file:

- Models/Responses/VirtualMachineEntityResponse.cs

5.132 Services.Impl.VirtualMachineEntityService Class Reference

Service managing virtual machine entities.

Inheritance diagram for Services.Impl.VirtualMachineEntityService:



Public Member Functions

- [VirtualMachineEntityService](#) ([IVirtualMachineEntityRepository](#) repository, [IBootableDiskProviderService](#) bootableDiskProviderService, [IVirtualMachineManagerService](#) vmManagerService, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterfaceRepository)
Initializes a new instance of the [VirtualMachineEntityService](#).
- 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.

5.132.1 Detailed Description

Service managing virtual machine entities.

5.132.2 Constructor & Destructor Documentation

5.132.2.1 VirtualMachineEntityService()

```
Services.Impl.VirtualMachineEntityService.VirtualMachineEntityService (
    IVirtualMachineEntityRepository repository,
    IBootableDiskProviderService bootableDiskProviderService,
    IVirtualMachineManagerService vmManagerService,
    IVirtualMachineEntityNetworkInterfaceRepository networkInterfaceRepository) [inline]
```

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

Parameters

<i>repository</i>	Repository for virtual machine entities.
<i>bootableDiskProviderService</i>	Service for managing bootable disks.
<i>vmManagerService</i>	Service for managing virtual machines.
<i>networkInterfaceRepository</i>	Repository for network interfaces.

5.132.3 Member Function Documentation

5.132.3.1 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](#).

5.132.3.2 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](#).

5.132.3.3 GetEntities()

```
async Task< List< VirtualMachineEntityDTO > > Services.Impl.VirtualMachineEntityService.GetEntities () [inline]
```

Retrieves a list of all virtual machine entities.

Returns

List of virtual machine entities.

Implements [Services.IVirtualMachineEntityService](#).

5.132.3.4 GetMacAddresses()

```
async Task< List< VirtualMachineEntityMacAddressDTO > > Services.Impl.VirtualMachineEntityService.GetMacAddresses () [inline]
```

Retrieves a list of virtual machine MAC addresses.

Returns

List of MAC addresses.

Implements [Services.IVirtualMachineEntityService](#).

5.132.3.5 UpdateEntityPosition()

```
async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.UpdateEntityPosition (
    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](#).

5.132.3.6 UpdateEntityVmUUID()

```
async Task< VirtualMachineEntityDTO > Services.Impl.VirtualMachineEntityService.UpdateEntityVmUUID (
    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

5.133 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

5.134 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.

5.134.1 Detailed Description

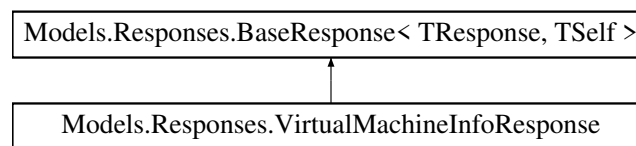
Information about a virtual machine.

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

- Models/VirtualMachineInfo.cs

5.135 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.

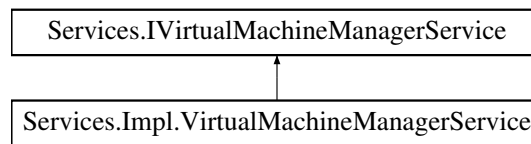
The documentation for this class was generated from the following file:

- Models/Responses/VirtualMachineInfoResponse.cs

5.136 Services.Impl.VirtualMachineManagerService Class Reference

Service managing virtual machines in the hypervisor.

Inheritance diagram for Services.Impl.VirtualMachineManagerService:



Public Member Functions

- [VirtualMachineManagerService](#) ([IVirtualizationWrapper](#) vmManager, [IOptions< InterconnectConfig >](#) config, [IVirtualMachineConsoleService](#) vmConsoleService)
Initializes a new instance of the [VirtualMachineManagerService](#).
- [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.

5.136.1 Detailed Description

Service managing virtual machines in the hypervisor.

5.136.2 Constructor & Destructor Documentation

5.136.2.1 VirtualMachineManagerService()

```

Services.Impl.VirtualMachineManagerService.VirtualMachineManagerService (
    IVirtualizationWrapper vmManager,
    IOptions< InterconnectConfig > config,
    IVirtualMachineConsoleService vmConsoleService) [inline]
  
```

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

Parameters

<i>vmManager</i>	Virtualization wrapper for hypervisor operations.
<i>config</i>	Configuration options for interconnect.
<i>vmConsoleService</i>	Service for managing virtual machine consoles.

5.136.3 Member Function Documentation

5.136.3.1 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](#).

5.136.3.2 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](#).

5.136.3.3 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

5.137 ServicesTests.VirtualMachineManagerServiceTests Class Reference

Public Member Functions

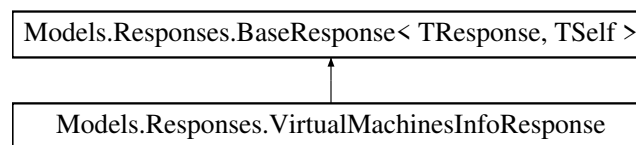
- void **SetUp** ()
- void **CreateVirtualMachine_WhenInvoked_ShouldCreateVirtualMachineUsingNativeLibrary** ()
- void **CreateVirtualMachine_AfterVirtualMachineCreated_ShouldOpenVirtualMachineConsole** ()
- void **GetListOfVirtualMachines_WhenInvoked_ShouldGetListOfVirtualMachinesUsingNativeLibrary** ()
- void **AttachVirtualNetworkInterfaceToVirtualMachine_WhenInvoked_ShouldAttachVirtualNetworkInterfaceToVirtualMachine** ()

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

- ServicesTests/VirtualMachineManagerServiceTests.cs

5.138 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 >](#)

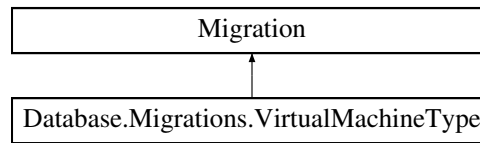
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualMachinesInfoResponse.cs

5.139 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)

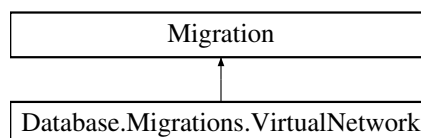
5.139.1 Detailed Description

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

- Database/Migrations/20260117231943_VirtualMachineType.cs
- Database/Migrations/20260117231943_VirtualMachineType.Designer.cs

5.140 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)

5.140.1 Detailed Description

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

- Database/Migrations/20250914132927_VirtualNetwork.cs
- Database/Migrations/20250914132927_VirtualNetwork.Designer.cs

5.141 Models.DTO.VirtualNetworkConnectionDTO Class Reference

Properties

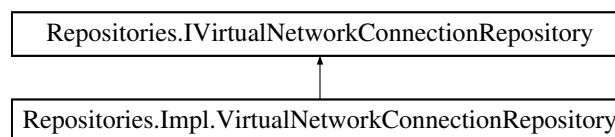
- required int **Id** [get, set]
- required int **SourceEntityId** [get, set]
- required [EntityType](#) **SourceEntityType** [get, set]
- required int **DestinationEntityId** [get, set]
- required [EntityType](#) **DestinationEntityType** [get, set]

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

- Models/DTO/VirtualNetworkConnectionDTO.cs

5.142 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.

5.142.1 Member Function Documentation

5.142.1.1 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](#).

5.142.1.2 GetAll()

```
Task< List< VirtualNetworkEntityConnectionModel > > Repositories.Impl.VirtualNetworkConnection↵  
Repository.GetAll () [inline]
```

Retrieves all connections.

Returns

List of connections.

Implements [Repositories.IVirtualNetworkConnectionRepository](#).

5.142.1.3 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](#).

5.142.1.4 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](#).

5.142.1.5 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

5.143 RepositoriesTests.VirtualNetworkConnectionRepositoryTests

Class Reference

Public Member Functions

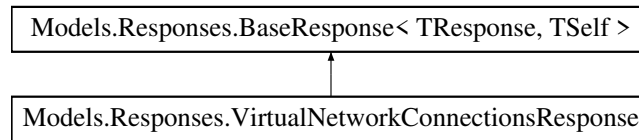
- void **SetUp** ()
- void **TearDown** ()
- async Task **Create_WhenInvoked_ShouldCreateVirtualNetworkConnectionEntity** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllVirtualNetworkConnectionEntities** ()
- async Task **GetUsingEntityId_WhenInvokedWithSourceEntity_ShouldGetVirtualNetworkConnectionEntity** ↔
- async Task **GetUsingEntityId_WhenInvokedWithDestinationEntity_ShouldGetVirtualNetworkConnectionEntity** ↔

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

- RepositoriesTests/VirtualNetworkConnectionRepositoryTests.cs

5.144 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.

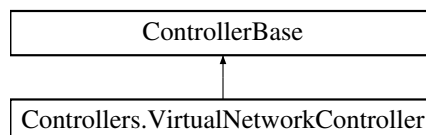
The documentation for this class was generated from the following file:

- Models/Responses/VirtualNetworkConnectionsResponse.cs

5.145 Controllers.VirtualNetworkController Class Reference

Controller managing virtual networks and connections.

Inheritance diagram for Controllers.VirtualNetworkController:



Public Member Functions

- [VirtualNetworkController](#) ([IVirtualNetworkService](#) virtualNetworkService, [IEntitiesConnectorService](#) entitiesConnectorService, [IEntitiesDisconnectorService](#) entitiesDisconnectorService)
Initializes a new instance of the [VirtualNetworkController](#).
- 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.

5.145.1 Detailed Description

Controller managing virtual networks and connections.

5.145.2 Constructor & Destructor Documentation

5.145.2.1 VirtualNetworkController()

```

Controllers.VirtualNetworkController.VirtualNetworkController (
    IVirtualNetworkService virtualNetworkService,
    IEntitiesConnectorService entitiesConnectorService,
    IEntitiesDisconnectorService entitiesDisconnectorService) [inline]

```

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

Parameters

<i>virtualNetworkService</i>	Service for managing virtual networks.
<i>entitiesConnectorService</i>	Service for connecting entities.
<i>entitiesDisconnectorService</i>	Service for disconnecting entities.

5.145.3 Member Function Documentation

5.145.3.1 ConnectEntities()

```

async Task< VirtualNetworkConnectionsResponse > Controllers.VirtualNetworkController.Connect↔
Entities (
    ConnectEntitiesRequest req) [inline]

```

Connects two entities in the virtual network.

Parameters

<i>req</i>	Connection data.
------------	------------------

Returns

Created connection.

5.145.3.2 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.

5.145.3.3 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

5.146 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.

5.146.1 Detailed Description

Definition for creating a new virtual network.

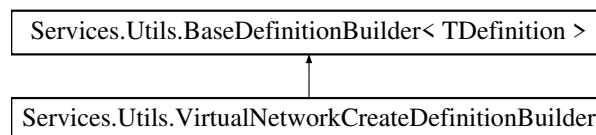
The documentation for this class was generated from the following file:

- Models/VirtualNetworkCreateDefinition.cs

5.147 Services.Utls.VirtualNetworkCreateDefinitionBuilder Class Reference

Builder class for creating virtual network XML definitions.

Inheritance diagram for Services.Utls.VirtualNetworkCreateDefinitionBuilder:



Public Member Functions

- override [VirtualNetworkCreateDefinitionBuilder SetFromCreateDefinition](#) ([VirtualNetworkCreateDefinition](#) definition)
Sets the builder state from a virtual network create definition.
- override string [Build](#) ()
Builds the complete virtual network XML definition.

Public Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition > SetFromCreateDefinition](#) ([TDefinition](#) definition)
Sets the builder state from a create definition object.
- string [Build](#) ()
Builds the XML definition string.

Additional Inherited Members

Protected Member Functions inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- delegate void [BuildingBlock](#) ([XmlWriter](#) writer)
Delegate for XML building blocks.
- void [CheckIsEverythingIsProvided](#) (params object?[] args)
Validates that all required parameters are provided.
- [StringWriter CreateXmlWriter](#) ()

Protected Attributes inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [XmlWriter](#)

Creates an XML writer with appropriate settings for building definitions.

5.147.1 Detailed Description

Builder class for creating virtual network XML definitions.

5.147.2 Member Function Documentation

5.147.2.1 Build()

```
override string Services.Utls.VirtualNetworkCreateDefinitionBuilder.Build () [inline]
```

Builds the complete virtual network XML definition.

Returns

The XML definition as a string.

5.147.2.2 SetFromCreateDefinition()

```
override VirtualNetworkCreateDefinitionBuilder Services.Utls.VirtualNetworkCreateDefinition←  
Builder.SetFromCreateDefinition (   
    VirtualNetworkCreateDefinition definition) [inline]
```

Sets the builder state from a virtual network create definition.

Parameters

<i>definition</i>	The network creation definition.
-------------------	----------------------------------

Returns

The builder instance for method chaining.

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

- [Services/Utls/VirtualNetworkCreateDefinitionBuilder.cs](#)

5.148 ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests Class Reference

Public Member Functions

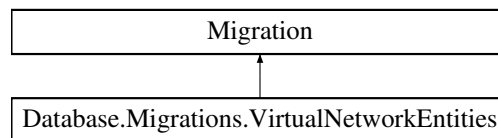
- void **OnSetup** ()
- void **Build_WhenNothingsProvided_ShouldThrowException** ()
- void **Build_WhenInvoked_ShouldIncludeNetworkName** ()
- void **Build_WhenInvoked_ShouldIncludeMacAddress** ()
- void **Build_WhenInvoked_ShouldIncludeIpAddress** ()
- void **Build_WhenInvoked_ShouldIncludeNetmask** ()
- 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

5.149 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)

5.149.1 Detailed Description

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

- Database/Migrations/20250906140923_VirtualNetworkEntities.cs
- Database/Migrations/20250906140923_VirtualNetworkEntities.Designer.cs

5.150 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.

5.150.1 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

5.151 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

5.152 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.

5.152.1 Detailed Description

Definition for creating a network interface for a virtual machine.

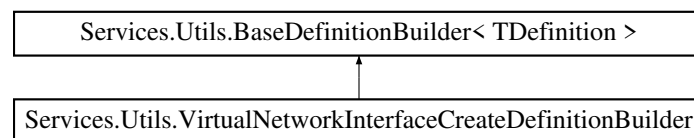
The documentation for this class was generated from the following file:

- Models/VirtualNetworkInterfaceCreateDefinition.cs

5.153 Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder Class Reference

Builder class for creating virtual network interface XML definitions.

Inheritance diagram for Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder:



Public Member Functions

- override [VirtualNetworkInterfaceCreateDefinitionBuilder SetFromCreateDefinition](#) ([VirtualNetworkInterfaceCreateDefinition](#) definition)
Sets the builder state from a network interface create definition.
- [VirtualNetworkInterfaceCreateDefinitionBuilder SetFromXml](#) (string xml)
Sets the builder state from an existing network interface XML definition.
- [VirtualNetworkInterfaceCreateDefinitionBuilder SetNetworkName](#) (string networkName)
Sets the network name for the interface.
- override string [Build](#) ()
Builds the complete network interface XML definition.

Public Member Functions inherited from [Services.Utils.BaseDefinitionBuilder< TDefinition >](#)

- [BaseDefinitionBuilder< TDefinition > SetFromCreateDefinition](#) (TDefinition definition)
Sets the builder state from a create definition object.
- string [Build](#) ()
Builds the XML definition string.

Additional Inherited Members

Protected Member Functions inherited from [Services.Utils.BaseDefinitionBuilder< TDefinition >](#)

- delegate void [BuildingBlock](#) ([XmlWriter](#) writer)
Delegate for XML building blocks.
- void [CheckIsEverythingIsProvided](#) (params object?[] args)
Validates that all required parameters are provided.
- [StringWriter](#) [CreateXmlWriter](#) ()

Protected Attributes inherited from [Services.Utls.BaseDefinitionBuilder< TDefinition >](#)

- [XmlWriter](#)

Creates an XML writer with appropriate settings for building definitions.

5.153.1 Detailed Description

Builder class for creating virtual network interface XML definitions.

5.153.2 Member Function Documentation

5.153.2.1 Build()

```
override string Services.Utls.VirtualNetworkInterfaceCreateDefinitionBuilder.Build () [inline]
```

Builds the complete network interface XML definition.

Returns

The XML definition as a string.

5.153.2.2 SetFromCreateDefinition()

```
override VirtualNetworkInterfaceCreateDefinitionBuilder Services.Utls.VirtualNetworkInterfaceCreateDefinitionBuilder.SetFromCreateDefinition (
    VirtualNetworkInterfaceCreateDefinition definition) [inline]
```

Sets the builder state from a network interface create definition.

Parameters

<i>definition</i>	The interface creation definition.
-------------------	------------------------------------

Returns

The builder instance for method chaining.

5.153.2.3 SetFromXml()

```
VirtualNetworkInterfaceCreateDefinitionBuilder Services.Utls.VirtualNetworkInterfaceCreateDefinitionBuilder.SetFromXml (
    string xml) [inline]
```

Sets the builder state from an existing network interface XML definition.

Parameters

<i>xml</i>	The existing interface XML definition.
------------	--

Returns

The builder instance for method chaining.

5.153.2.4 SetNetworkName()

```
VirtualNetworkInterfaceCreateDefinitionBuilder Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder.SetNetworkName (  
    string networkName) [inline]
```

Sets the network name for the interface.

Parameters

<i>networkName</i>	The name of the network to attach to.
--------------------	---------------------------------------

Returns

The builder instance for method chaining.

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

- Services/Utils/VirtualNetworkInterfaceDefinitionBuilder.cs

5.154 ServicesTests.Utils.VirtualNetworkInterfaceCreateDefinitionBuilderTests Class Reference**Public Member Functions**

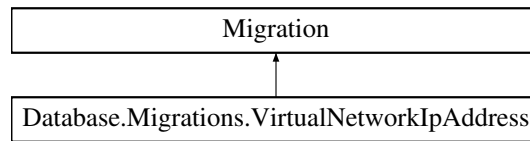
- void **OnSetup** ()
- void **Build_WhenNothingsProvided_ShouldThrowException** ()
- void **Build_WhenInvoked_ShouldContainNetworkName** ()
- void **Build_WhenInvoked_ShouldContainMacAddress** ()

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

- ServicesTests/Utils/VirtualNetworkInterfaceCreateDefinitionBuilderTests.cs

5.155 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)

5.155.1 Detailed Description

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

- Database/Migrations/20250927161009_VirtualNetworkIpAddress.cs
- Database/Migrations/20250927161009_VirtualNetworkIpAddress.Designer.cs

5.156 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.

5.156.1 Detailed Description

[Database](#) model representing a virtual network.

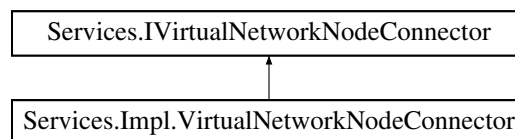
The documentation for this class was generated from the following file:

- Models/Database/VirtualNetworkModel.cs

5.157 Services.Impl.VirtualNetworkNodeConnector Class Reference

Service responsible for connecting virtual network nodes.

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.

5.157.1 Detailed Description

Service responsible for connecting virtual network nodes.

5.157.2 Member Function Documentation

5.157.2.1 ConnectTwoVirtualNetworkNodes()

```

async Task< VirtualNetworkConnectionDTO > Services.Impl.VirtualNetworkNodeConnector.ConnectTwoVirtualNetworkNodes (
    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](#).

5.157.2.2 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](#).

5.157.2.3 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](#).

5.157.2.4 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>virtualNetwork↔ NodeId</i>	Network node identifier.

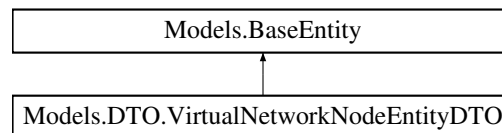
Implements [Services.IVirtualNetworkNodeConnector](#).

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

- Services/Impl/VirtualNetworkNodeConnector.cs

5.158 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

5.159 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.

5.159.1 Detailed Description

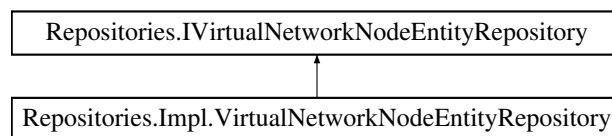
[Database](#) model representing a virtual network node (switch/router).

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

- Models/Database/VirtualNetworkNodeEntityModel.cs

5.160 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.

5.160.1 Member Function Documentation

5.160.1.1 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](#).

5.160.1.2 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](#).

5.160.1.3 GetAll()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.Impl.VirtualNetworkNodeEntity↔  
Repository.GetAll () [inline]
```

Retrieves all virtual network nodes.

Returns

List of nodes.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.160.1.4 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](#).

5.160.1.5 GetVisible()

```
Task< List< VirtualNetworkNodeEntityModel > > Repositories.Impl.VirtualNetworkNodeEntityRepository.GetVisible () [inline]
```

Retrieves visible virtual network nodes.

Returns

List of visible nodes.

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.160.1.6 Remove()

```
async Task Repositories.Impl.VirtualNetworkNodeEntityRepository.Remove (
    int id) [inline]
```

Removes a virtual network node.

Parameters

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

Implements [Repositories.IVirtualNetworkNodeEntityRepository](#).

5.160.1.7 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](#).

5.160.1.8 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

5.161 RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests Class Reference

Public Member Functions

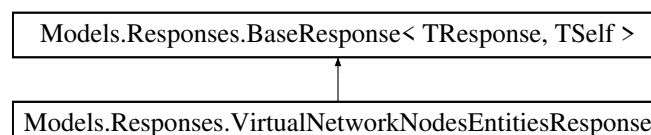
- void **SetUp** ()
- void **TearDown** ()
- async Task **Create_WhenInvoked_ShouldCreateVisibleVirtualNetworkNode** ()
- async Task **Create_WhenInvoked_ShouldCreateInvisibleVirtualNetworkNode** ()
- async Task **GetAll_WhenInvoked_ShouldGetAllVirtualNetworkNodes** ()
- async Task **GetById_WhenInvokedWithCorrectId_ShouldGetVirtualNetworkNodeById** ()
- async Task **UpdateEntityPosition_WhenInvoked_ShouldUpdateEntityPosition** ()

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

- RepositoriesTests/VirtualNetworkNodeEntityRepositoryTests.cs

5.162 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 >

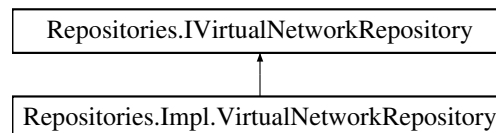
- bool **Success** [get, set]
Indicates whether the operation was successful.
- string? **ErrorMessage** [get, set]
Error message, if any error occurred.
- TResponse? **Data** [get, set]
Response data.

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

- Models/Responses/VirtualNetworkNodesEntitiesResponse.cs

5.163 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.

5.163.1 Member Function Documentation

5.163.1.1 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](#).

5.163.1.2 GetAll()

```
Task< List< VirtualNetworkModel > > Repositories.Impl.VirtualNetworkRepository.GetAll ()
[inline]
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implements [Repositories.IVirtualNetworkRepository](#).

5.163.1.3 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](#).

5.163.1.4 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](#).

5.163.1.5 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](#).

5.163.1.6 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](#).

5.163.1.7 Remove()

```
async Task Repositories.Impl.VirtualNetworkRepository.Remove (
    int id) [inline]
```

Removes a virtual network.

Parameters

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

Implements [Repositories.IVirtualNetworkRepository](#).

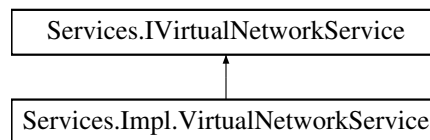
The documentation for this class was generated from the following file:

- `Repositories/Impl/VirtualNetworkRepository.cs`

5.164 Services.Impl.VirtualNetworkService Class Reference

Service managing virtual networks.

Inheritance diagram for Services.Impl.VirtualNetworkService:



Public Member Functions

- [VirtualNetworkService](#) (IOptions< [InterconnectConfig](#) > config, [IVirtualizationWrapper](#) wrapper, [IVirtualMachineEntityRepository](#) vmEntityRepository, [IVirtualNetworkConnectionRepository](#) connectionRepository, [IVirtualNetworkNodeEntityRepository](#) virtualNetworkNodeRepository, [IVirtualNetworkRepository](#) networkRepository, [IPacketSnifferService](#) packetSnifferService, [IVirtualMachineEntityNetworkInterfaceRepository](#) networkInterfaceRepository)
Initializes a new instance of the [VirtualNetworkService](#).
- 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](#) > [CreateInternetVirtualNetwork](#) ()
Creates a virtual network for Internet connection.
- async 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.

5.164.1 Detailed Description

Service managing virtual networks.

5.164.2 Constructor & Destructor Documentation

5.164.2.1 VirtualNetworkService()

```
Services.Impl.VirtualNetworkService.VirtualNetworkService (
    IOptions< InterconnectConfig > config,
    IVirtualizationWrapper wrapper,
    IVirtualMachineEntityRepository vmEntityRepository,
    IVirtualNetworkConnectionRepository connectionRepository,
    IVirtualNetworkNodeEntityRepository virtualNetworkNodeRepository,
    IVirtualNetworkRepository networkRepository,
    IPacketSnifferService packetSnifferService,
    IVirtualMachineEntityNetworkInterfaceRepository networkInterfaceRepository) [inline]
```

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

Parameters

<i>config</i>	Configuration options for interconnect.
<i>wrapper</i>	Virtualization wrapper for hypervisor operations.
<i>vmEntityRepository</i>	Repository for virtual machine entities.
<i>connectionRepository</i>	Repository for network connections.
<i>virtualNetworkNodeRepository</i>	Repository for network node entities.
<i>networkRepository</i>	Repository for virtual networks.
<i>packetSnifferService</i>	Service for packet sniffing.
<i>networkInterfaceRepository</i>	Repository for network interfaces.

5.164.3 Member Function Documentation

5.164.3.1 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](#).

5.164.3.2 CreateInternetVirtualNetwork()

```
async Task< VirtualNetworkModel > Services.Impl.VirtualNetworkService.CreateInternetVirtualNetwork () [inline]
```

Creates a virtual network for Internet connection.

Returns

Created virtual network.

Implements [Services.IVirtualNetworkService](#).

5.164.3.3 CreateNodeVirtualNetwork()

```
async Task< VirtualNetworkModel > Services.Impl.VirtualNetworkService.CreateNodeVirtualNetwork () [inline]
```

Creates a virtual network for a network node.

Returns

Created virtual network.

Implements [Services.IVirtualNetworkService](#).

5.164.3.4 CreateVirtualNetworkNode()

```
async Task< VirtualNetworkNodeEntityDTO > Services.Impl.VirtualNetworkService.CreateVirtualNetworkNode ( string? name) [inline]
```

Creates a new virtual network node.

Parameters

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

Returns

Created network node.

Implements [Services.IVirtualNetworkService](#).

5.164.3.5 GetAllVirtualNetworks()

```
Task< List< VirtualNetworkModel > > Services.Impl.VirtualNetworkService.GetAllVirtualNetworks () [inline]
```

Retrieves all virtual networks.

Returns

List of virtual networks.

Implements [Services.IVirtualNetworkService](#).

5.164.3.6 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](#).

5.164.3.7 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](#).

5.164.3.8 GetVisibleVirtualNetworkNodeEntities()

```
async Task< List< VirtualNetworkNodeEntityDTO > > Services.Impl.VirtualNetworkService.GetVisibleVirtualNetworkNodeEntities () [inline]
```

Retrieves a list of visible virtual network nodes.

Returns

List of visible nodes.

Implements [Services.IVirtualNetworkService](#).

5.164.3.9 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](#).

5.164.3.10 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.

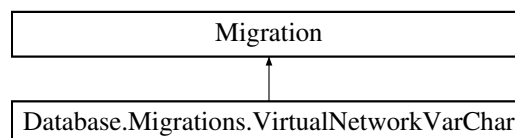
Implements [Services.IVirtualNetworkService](#).

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

- Services/Impl/VirtualNetworkService.cs

5.165 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)

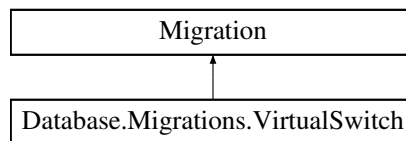
5.165.1 Detailed Description

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

- Database/Migrations/20250927161142_VirtualNetworkVarChar.cs
- Database/Migrations/20250927161142_VirtualNetworkVarChar.Designer.cs

5.166 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)

5.166.1 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, [194](#)
[156](#)
Repositories.Impl.VirtualMachineEntityRepository, [169](#)
Repositories.IVirtualMachineConsoleStreamRepository, [100](#)
Repositories.IVirtualMachineEntityRepository, [104](#)
AddDataToConsole
Repositories.Impl.VirtualMachineConsoleDataRepository, [149](#)
Repositories.IVirtualMachineConsoleDataRepository, [96](#)
AddSniffer
Repositories.Impl.PacketSnifferRepository, [131](#)
Repositories.IPacketSnifferRepository, [86](#)
ALIVE
Models.Enums, [17](#)
AttachDeviceToVirtualMachine
NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [143](#)
NativeLibrary.Wrappers.IVirtualizationWrapper, [91](#)
AttachNetworkInterfaceToVirtualMachine
Services.Impl.VirtualNetworkService, [208](#)
Services.IVirtualNetworkService, [122](#)
AttachVirtualNetworkInterfaceToVirtualMachine
Services.Impl.VirtualMachineManagerService, [178](#)
Services.IVirtualMachineManagerService, [109](#)

BackgroundServices, [13](#)
BackgroundServices.Impl, [13](#)
BackgroundServices.Impl.PacketSnifferBackgroundService, [129](#)
ExecuteAsync, [130](#)
PacketSnifferBackgroundService, [130](#)
BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, [147](#)
ExecuteAsync, [148](#)
VirtualMachineConsoleBackgroundService, [148](#)
BootableDiskProviderService
Services.Impl.BootableDiskProviderService, [32](#)
Booted
Models.Enums, [19](#)
Build
Services.Utils.BaseDefinitionBuilder< TDefinition
>, [26](#)
Services.Utils.VirtualMachineDefinitionBuilder, [161](#)
Services.Utils.VirtualNetworkCreateDefinitionBuilder, [190](#)
Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder, [194](#)
BuildingBlock
Services.Utils.BaseDefinitionBuilder< TDefinition
>, [26](#)
ChecksEverythingIsProvided
Services.Utils.BaseDefinitionBuilder< TDefinition
>, [27](#)
CloseStream
NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [143](#)
NativeLibrary.Wrappers.IVirtualizationWrapper, [92](#)
Services.Impl.VirtualMachineConsoleService, [153](#)
Services.IVirtualMachineConsoleService, [97](#)
ConnectEntities
Controllers.VirtualNetworkController, [186](#)
ConnectionInfo
Controllers.HypervisorConnectionController, [55](#)
ConnectionStatus
Models.Enums, [17](#)
ConnectTwoEntities
Services.IEntitiesConnectorService, [62](#)
Services.Impl.EntitiesConnectorService, [44](#)
ConnectTwoVirtualMachines
Services.IEntitiesConnectorService, [62](#)
Services.Impl.EntitiesConnectorService, [44](#)
ConnectTwoVirtualNetworkNodes
Services.Impl.VirtualNetworkNodeConnector, [197](#)
Services.IVirtualNetworkNodeConnector, [113](#)
ConnectVirtualMachineToInternet
Services.IEntitiesConnectorService, [62](#)
Services.Impl.EntitiesConnectorService, [44](#)
ConnectVirtualMachineToVirtualNetworkNode
Services.IEntitiesConnectorService, [63](#)
Services.Impl.EntitiesConnectorService, [46](#)
ConnectVirtualNetworkNodeToInternet
Services.Impl.VirtualNetworkNodeConnector, [198](#)
Services.IVirtualNetworkNodeConnector, [113](#)
Controllers, [13](#)
Controllers.EntityController, [49](#)
CreateInternetEntity, [51](#)
CreateVirtualMachineEntity, [51](#)
CreateVirtualNetworkNodeEntity, [51](#)
DeleteEntity, [51](#)
EntityController, [50](#)
GetAllInternetEntities, [52](#)
GetAllVirtualMachineEntities, [52](#)
GetAllVirtualNetworkNodeEntities, [52](#)
UpdateEntityPosition, [52](#)

- Controllers.HypervisorConnectionController, 55
 - ConnectionInfo, 55
 - HypervisorConnectionController, 55
- Controllers.InfoController, 72
 - GetSystemInfo, 73
 - InfoController, 72
- Controllers.VirtualMachineController, 158
 - GetAvailableBootableDisks, 158
 - VirtualMachineController, 158
- Controllers.VirtualNetworkController, 185
 - ConnectEntities, 186
 - DisconnectEntities, 186
 - GetAllConnections, 188
 - VirtualNetworkController, 186
- ControllersTests, 14
- ControllersTests.HypervisorConnectionControllerTests, 56
- ControllersTests.InfoControllerTests, 73
- Create
 - Repositories.IInternetEntityRepository, 68
 - Repositories.Impl.InternetEntityRepository, 81
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 166
 - Repositories.Impl.VirtualNetworkConnectionRepository, 182
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 201
 - Repositories.Impl.VirtualNetworkRepository, 205
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 101
 - Repositories.IVirtualNetworkConnectionRepository, 111
 - Repositories.IVirtualNetworkNodeEntityRepository, 115
 - Repositories.IVirtualNetworkRepository, 119
- CreateEntity
 - Services.Impl.VirtualMachineEntityService, 173
 - Services.IVirtualMachineEntityService, 106
- CreateInternet
 - Services.IInternetEntityService, 71
 - Services.Impl.InternetEntityService, 84
- CreateInternetEntity
 - Controllers.EntityController, 51
- CreateInternetVirtualNetwork
 - Services.Impl.VirtualNetworkService, 208
 - Services.IVirtualNetworkService, 122
- CreateInvisible
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 201
 - Repositories.IVirtualNetworkNodeEntityRepository, 116
- CreateNodeVirtualNetwork
 - Services.Impl.VirtualNetworkService, 209
 - Services.IVirtualNetworkService, 123
- CreateVirtualMachine
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 143
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 92
- Services.Impl.VirtualMachineManagerService, 179
- Services.IVirtualMachineManagerService, 109
- CreateVirtualMachineEntity
 - Controllers.EntityController, 51
- CreateVirtualNetwork
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 144
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 92
- CreateVirtualNetworkNode
 - Services.Impl.VirtualNetworkService, 209
 - Services.IVirtualNetworkService, 123
- CreateVirtualNetworkNodeEntity
 - Controllers.EntityController, 51
- Database, 14
- Database.InterconnectDbContext, 77
- Database.Migrations, 14
- Database.Migrations.AppDbContextModelSnapshot, 25
- Database.Migrations.BootableDiskModel, 30
- Database.Migrations.ChangesToConnection, 36
- Database.Migrations.ChangeTypeOfVmEntityStrings, 37
- Database.Migrations.DeviceDefinition, 43
- Database.Migrations.InitialCreate, 75
- Database.Migrations.InternetEntity, 79
- Database.Migrations.ModificationToVirtualNetwork, 125
- Database.Migrations.RemoveOsType, 137
- Database.Migrations.RenameSwitchToNode, 137
- Database.Migrations.RenameUuidToVmUuid, 138
- Database.Migrations.UpdateBootableDiskModelPath, 141
- Database.Migrations.UpdateOperatingSystemType, 141
- Database.Migrations.VirtualMachineEntityInterfaces, 164
- Database.Migrations.VirtualMachineType, 181
- Database.Migrations.VirtualNetwork, 181
- Database.Migrations.VirtualNetworkEntities, 191
- Database.Migrations.VirtualNetworkIpAddress, 196
- Database.Migrations.VirtualNetworkVarChar, 211
- Database.Migrations.VirtualSwitch, 212
- DEAD
 - Models.Enums, 17
- DeleteEntity
 - Controllers.EntityController, 51
- DeleteEntityService
 - Services.Impl.DeleteEntityService, 41
- DeleteInternetEntity
 - Services.IDeleteEntityService, 61
 - Services.Impl.DeleteEntityService, 42
- DeleteVirtualMachineEntity
 - Services.IDeleteEntityService, 61
 - Services.Impl.DeleteEntityService, 42
- DeleteVirtualNetworkNodeEntity
 - Services.IDeleteEntityService, 61
 - Services.Impl.DeleteEntityService, 42
- DestroyNetwork
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 144
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 92

- DetachDeviceFromVirtualMachine
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 144
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 93
- DisconnectEntities
 - Controllers.VirtualNetworkController, 186
 - Services.IEntitiesDisconnectService, 64
 - Services.Impl.EntitiesDisconnectService, 48
- DisconnectTwoVirtualNetworkNodes
 - Services.Impl.VirtualNetworkNodeConnector, 198
 - Services.IVirtualNetworkNodeConnector, 114
- DisconnectVirtualMachineFromInternet
 - Services.IEntitiesDisconnectService, 64
 - Services.Impl.EntitiesDisconnectService, 48
- DisconnectVirtualMachineFromVirtualMachine
 - Services.IEntitiesDisconnectService, 64
 - Services.Impl.EntitiesDisconnectService, 48
- DisconnectVirtualMachineFromVirtualNetworkNode
 - Services.IEntitiesDisconnectService, 65
 - Services.Impl.EntitiesDisconnectService, 49
- DisconnectVirtualNetworkNodeFromInternet
 - Services.Impl.VirtualNetworkNodeConnector, 198
 - Services.IVirtualNetworkNodeConnector, 114
- EntitiesDisconnectService
 - Services.Impl.EntitiesDisconnectService, 47
- EntityController
 - Controllers.EntityController, 50
- EntityType
 - Models.Enums, 17
- ExecuteAsync
 - BackgroundServices.Impl.PacketSnifferBackgroundService, 130
 - BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, 148
- GetAll
 - Repositories.IInternetEntityRepository, 68
 - Repositories.Impl.InternetEntityRepository, 81
 - Repositories.Impl.VirtualMachineEntityRepository, 169
 - Repositories.Impl.VirtualNetworkConnectionRepository, 183
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 201
 - Repositories.Impl.VirtualNetworkRepository, 205
 - Repositories.IVirtualMachineEntityRepository, 104
 - Repositories.IVirtualNetworkConnectionRepository, 111
 - Repositories.IVirtualNetworkNodeEntityRepository, 116
 - Repositories.IVirtualNetworkRepository, 119
- GetAllConnections
 - Controllers.VirtualNetworkController, 188
- GetAllInternetEntities
 - Controllers.EntityController, 52
- GetAllStreams
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, 156
- Repositories.IVirtualMachineConsoleStreamRepository, 100
- GetAllVirtualMachineEntities
 - Controllers.EntityController, 52
- GetAllVirtualNetworkNodeEntities
 - Controllers.EntityController, 52
- GetAllVirtualNetworks
 - Services.Impl.VirtualNetworkService, 209
 - Services.IVirtualNetworkService, 123
- GetAvailableBootableDiskModels
 - Services.IBootableDiskProviderService, 58
 - Services.Impl.BootableDiskProviderService, 32
- GetAvailableBootableDisks
 - Controllers.VirtualMachineController, 158
- GetBootableDiskPathById
 - Services.IBootableDiskProviderService, 58
 - Services.Impl.BootableDiskProviderService, 32
- GetByBridgeName
 - Repositories.Impl.VirtualNetworkRepository, 205
 - Repositories.IVirtualNetworkRepository, 119
- GetById
 - Repositories.IBootableDiskRepository, 59
 - Repositories.IInternetEntityRepository, 69
 - Repositories.Impl.BootableDiskRepository, 34
 - Repositories.Impl.InternetEntityRepository, 82
 - Repositories.Impl.VirtualMachineEntityRepository, 169
 - Repositories.Impl.VirtualNetworkConnectionRepository, 183
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 201
 - Repositories.Impl.VirtualNetworkRepository, 205
 - Repositories.IVirtualMachineEntityRepository, 104
 - Repositories.IVirtualNetworkConnectionRepository, 111
 - Repositories.IVirtualNetworkNodeEntityRepository, 116
 - Repositories.IVirtualNetworkRepository, 120
 - Services.Impl.VirtualMachineEntityService, 173
 - Services.IVirtualMachineEntityService, 106
- GetByIds
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 167
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 102
- GetByNetworkName
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 167
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 102
- GetBytesFromConsole
 - Services.Impl.VirtualMachineConsoleService, 153
 - Services.IVirtualMachineConsoleService, 98
- GetByUuid
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, 156
 - Repositories.Impl.VirtualNetworkRepository, 206
 - Repositories.IVirtualMachineConsoleStreamRepository,

- 100
- Repositories.IVirtualNetworkRepository, 120
- GetByUuidWithVirtualNetworkNodes
 - Repositories.Impl.VirtualNetworkRepository, 206
 - Repositories.IVirtualNetworkRepository, 120
- GetByVirtualMachineId
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 167
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 102
- GetCapturedPacket
 - Services.Impl.PacketSnifferService, 133
 - Services.IPacketSnifferService, 87
- GetConnectionInfo
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 144
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 93
 - Services.IHypervisorConnectionService, 66
 - Services.Impl.HypervisorConnectionService, 57
- GetConnectionStatus
 - Services.IHypervisorConnectionService, 66
 - Services.Impl.HypervisorConnectionService, 57
- GetData
 - Repositories.Impl.VirtualMachineConsoleDataRepository, 149
 - Repositories.IVirtualMachineConsoleDataRepository, 96
- GetDataFromStream
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 145
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 93
- GetEntities
 - Services.Impl.VirtualMachineEntityService, 173
 - Services.IVirtualMachineEntityService, 107
- GetInitialConsoleData
 - Services.Impl.VirtualMachineConsoleService, 153
 - Services.IVirtualMachineConsoleService, 98
- GetInitialDataForConsole
 - Hubs.VirtualMachineConsoleHub, 150
- GetInternetEntities
 - Services.IInternetEntityService, 71
 - Services.Impl.InternetEntityService, 84
- GetListOfVirtualMachines
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 145
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 93
 - Services.Impl.VirtualMachineManagerService, 179
 - Services.IVirtualMachineManagerService, 109
- GetMacAddresses
 - Services.Impl.VirtualMachineEntityService, 174
 - Services.IVirtualMachineEntityService, 107
- GetNumberOfPackets
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, 135
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 89
- GetOnlyWithNotNullablePath
 - Repositories.IBootableDiskRepository, 60
 - Repositories.Impl.BootableDiskRepository, 34
- GetOpenedSniffers
 - Repositories.Impl.PacketSnifferRepository, 132
 - Repositories.IPacketSnifferRepository, 86
 - Services.Impl.PacketSnifferService, 133
 - Services.IPacketSnifferService, 87
- GetPacket
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, 135
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 89
- GetStreams
 - Services.Impl.VirtualMachineConsoleService, 154
 - Services.IVirtualMachineConsoleService, 98
- GetSystemInfo
 - Controllers.InfoController, 73
 - Services.IInfoService, 67
 - Services.Impl.InfoService, 74
- GetUsingEntityId
 - Repositories.Impl.VirtualNetworkConnectionRepository, 183
 - Repositories.IVirtualNetworkConnectionRepository, 111
- GetVirtualMachineInfo
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 145
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 94
- GetVirtualMachineNameWithPrefix
 - Models.VirtualMachineCreateDefinition, 159
- GetVirtualNetworkConnections
 - Services.Impl.VirtualNetworkService, 209
 - Services.IVirtualNetworkService, 123
- GetVirtualNetworkUsingBridgeName
 - Services.Impl.VirtualNetworkService, 210
 - Services.IVirtualNetworkService, 124
- GetVisible
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 202
 - Repositories.IVirtualNetworkNodeEntityRepository, 117
- GetVisibleVirtualNetworkNodeEntities
 - Services.Impl.VirtualNetworkService, 210
 - Services.IVirtualNetworkService, 124
- Hubs, 14
 - Hubs.ConnectionStatusHub, 39
 - Ping, 39
 - Hubs.PacketSnifferHub, 130
 - Hubs.VirtualMachineConsoleHub, 150
 - GetInitialDataForConsole, 150
 - JoinConsoleGroup, 151
 - LeaveConsoleGroup, 151
 - SendDataToConsole, 151
- HubsTests, 15
 - HubsTests.ConnectionStatusHubTests, 40
 - HubsTests.VirtualMachineConsoleHubTests, 152
- HypervisorConnectionController
 - Controllers.HypervisorConnectionController, 55
- HypervisorConnectionService
 - Services.Impl.HypervisorConnectionService, 56

- InfoController
 - Controllers.InfoController, 72
- InitializationService
 - Services.Impl.InitializationService, 76
- InitializeConnection
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 146
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 94
 - Services.IHypervisorConnectionService, 66
 - Services.Impl.HypervisorConnectionService, 57
- Interconnect, 15
- Interconnect.Middlewares, 15
- Interconnect.Middlewares.ExceptionHandlerMiddleware, 53
 - InvokeAsync, 53
- Interconnect.Program, 136
 - Main, 136
- InterconnectTests, 15
- InterconnectTests.ExceptionHandlerMiddlewareTests, 54
- Internet
 - Models.Enums, 17
- InternetEntityService
 - Services.Impl.InternetEntityService, 84
- InvokeAsync
 - Interconnect.Middlewares.ExceptionHandlerMiddlewareTests, 53
- IsConnectionAlive
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 146
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 94
- JoinConsoleGroup
 - Hubs.VirtualMachineConsoleHub, 151
- LeaveConsoleGroup
 - Hubs.VirtualMachineConsoleHub, 151
- ListenForPacket
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, 89
 - Services.Impl.PacketSnifferService, 134
 - Services.IPacketSnifferService, 88
- Main
 - Interconnect.Program, 136
- Mappers, 15
- Models, 16
- Models.BaseEntity, 28
- Models.Config, 16
- Models.Config.InterconnectConfig, 77
- Models.ConnectionInfo, 38
- Models.Database, 16
- Models.Database.BootableDiskModel, 31
- Models.Database.InternetEntityModel, 79
- Models.Database.VirtualMachineEntityModel, 164
- Models.Database.VirtualMachineEntityNetworkInterfaceModel, 165
- Models.Database.VirtualNetworkEntityConnectionModel, 191
- Models.Database.VirtualNetworkModel, 196
- Models.Database.VirtualNetworkNodeEntityModel, 199
- Models.DTO, 17
- Models.DTO.BootableDiskModelDTO, 31
- Models.DTO.InternetEntityModelDTO, 80
- Models.DTO.VirtualMachineEntitiesWithMacAddressesDTO, 163
- Models.DTO.VirtualMachineEntityDTO, 163
- Models.DTO.VirtualMachineEntityMacAddressDTO, 164
- Models.DTO.VirtualNetworkConnectionDTO, 182
- Models.DTO.VirtualNetworkNodeEntityDTO, 199
- 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, 73
- Models.Packet, 128
- Models.PacketSniffer, 129
- Models.Requests, 19
- Models.Requests.ConnectEntitiesRequest, 37
- Models.Requests.CreateVirtualMachineEntityRequest, 40
- Models.Requests.CreateVirtualNetworkNodeEntityRequest, 40
- Models.Requests.DeleteEntityRequest, 40
- Models.Requests.SendDataToConsoleRequest, 138
- Models.Requests.UpdateEntityPositionRequest, 141
- Models.Requests.VirtualNetworkEntityConnectionRequest, 192
- Models.Responses, 19
- Models.Responses.BaseResponse< TResponse, TSelf >, 28
 - WithEmptySuccess, 30
 - WithSuccess, 30
- Models.Responses.BootableDisksResponse, 35
- Models.Responses.CapturedPacketResponse, 36
- Models.Responses.ConnectionInfoResponse, 38
- Models.Responses.ExceptionResponse, 54
- Models.Responses.InternetEntitiesResponse, 78
- Models.Responses.StringResponse, 139
- Models.Responses.SystemInfoResponse, 140
- Models.Responses.VirtualMachineEntitiesResponse, 162
- Models.Responses.VirtualMachineEntityResponse, 171
- Models.Responses.VirtualMachineInfoResponse, 176
- Models.Responses.VirtualMachinesInfoResponse, 180
- Models.Responses.VirtualNetworkConnectionsResponse, 185
- Models.Responses.VirtualNetworkNodesEntitiesResponse, 203
- Models.StreamData, 138

- Models.StreamInfo, [139](#)
- Models.TerminalData, [140](#)
- Models.VirtualMachineCreateDefinition, [159](#)
 - GetVirtualMachineNameWithPrefix, [159](#)
- Models.VirtualMachineInfo, [175](#)
- Models.VirtualNetworkCreateDefinition, [188](#)
- Models.VirtualNetworkInterfaceCreateDefinition, [192](#)

- NativeLibrary, [19](#)
- NativeLibrary.Interop, [20](#)
- NativeLibrary.Structs, [20](#)
- NativeLibrary.Structs.NativeConnectionInfo, [126](#)
- NativeLibrary.Structs.NativeExecutionInfo, [126](#)
- NativeLibrary.Structs.NativePacket, [127](#)
- NativeLibrary.Structs.NativeStreamData, [127](#)
- NativeLibrary.Structs.NativeStreamInfo, [127](#)
- NativeLibrary.Structs.NativeVirtualMachineInfo, [128](#)
- NativeLibrary.Utils, [20](#)
- NativeLibrary.Utils.Impl, [20](#)
- NativeLibrary.Utils.Impl.NativeList< T >, [126](#)
- NativeLibrary.Utils.INativeList< T >, [72](#)
- NativeLibrary.Wrappers, [20](#)
- NativeLibrary.Wrappers.Impl, [20](#)
- NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, [135](#)
 - GetNumberOfPackets, [135](#)
 - GetPacket, [135](#)
 - OpenSnifferHandler, [135](#)
- NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [142](#)
 - AttachDeviceToVirtualMachine, [143](#)
 - CloseStream, [143](#)
 - CreateVirtualMachine, [143](#)
 - CreateVirtualNetwork, [144](#)
 - DestroyNetwork, [144](#)
 - DetachDeviceFromVirtualMachine, [144](#)
 - GetConnectionInfo, [144](#)
 - GetDataFromStream, [145](#)
 - GetListOfVirtualMachines, [145](#)
 - GetVirtualMachineInfo, [145](#)
 - InitializeConnection, [146](#)
 - IsConnectionAlive, [146](#)
 - OpenVirtualMachineConsole, [146](#)
 - SendDataToStream, [146](#)
 - UpdateVmDevice, [147](#)
- NativeLibrary.Wrappers.IPacketSnifferWrapper, [89](#)
 - GetNumberOfPackets, [89](#)
 - GetPacket, [89](#)
 - ListenForPacket, [89](#)
 - OpenSnifferHandler, [90](#)
- NativeLibrary.Wrappers.IVirtualizationWrapper, [90](#)
 - AttachDeviceToVirtualMachine, [91](#)
 - CloseStream, [92](#)
 - CreateVirtualMachine, [92](#)
 - CreateVirtualNetwork, [92](#)
 - DestroyNetwork, [92](#)
 - DetachDeviceFromVirtualMachine, [93](#)
 - GetConnectionInfo, [93](#)
 - GetDataFromStream, [93](#)
 - GetListOfVirtualMachines, [93](#)
 - GetVirtualMachineInfo, [94](#)
 - InitializeConnection, [94](#)
 - IsConnectionAlive, [94](#)
 - OpenVirtualMachineConsole, [94](#)
 - SendDataToStream, [95](#)
 - UpdateVmDevice, [95](#)
- Network
 - Models.Enums, [17](#)
- OpenSnifferHandler
 - NativeLibrary.Wrappers.Impl.PacketSnifferWrapper, [135](#)
 - NativeLibrary.Wrappers.IPacketSnifferWrapper, [90](#)
- OpenVirtualMachineConsole
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, [146](#)
 - NativeLibrary.Wrappers.IVirtualizationWrapper, [94](#)
 - Services.Impl.VirtualMachineConsoleService, [154](#)
 - Services.IVirtualMachineConsoleService, [98](#)
- PacketSnifferBackgroundService
 - BackgroundServices.Impl.PacketSnifferBackgroundService, [130](#)
- PacketSnifferService
 - Services.Impl.PacketSnifferService, [133](#)
- Ping
 - Hubs.ConnectionStatusHub, [39](#)
- Remove
 - Repositories.IInternetEntityRepository, [69](#)
 - Repositories.Impl.InternetEntityRepository, [82](#)
 - Repositories.Impl.VirtualMachineConsoleStreamRepository, [157](#)
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, [168](#)
 - Repositories.Impl.VirtualMachineEntityRepository, [170](#)
 - Repositories.Impl.VirtualNetworkConnectionRepository, [184](#)
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, [202](#)
 - Repositories.Impl.VirtualNetworkRepository, [206](#)
 - Repositories.IVirtualMachineConsoleStreamRepository, [100](#)
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, [103](#)
 - Repositories.IVirtualMachineEntityRepository, [105](#)
 - Repositories.IVirtualNetworkConnectionRepository, [112](#)
 - Repositories.IVirtualNetworkNodeEntityRepository, [117](#)
 - Repositories.IVirtualNetworkRepository, [121](#)
- RemoveSniffer
 - Repositories.Impl.PacketSnifferRepository, [132](#)
 - Repositories.IPacketSnifferRepository, [86](#)
- Repositories, [21](#)
- Repositories.IBootableDiskRepository, [59](#)
 - GetById, [59](#)
 - GetOnlyWithNotNullablePath, [60](#)
- Repositories.IInternetEntityRepository, [68](#)

- Create, [68](#)
- GetAll, [68](#)
- GetById, [69](#)
- Remove, [69](#)
- UpdatePosition, [69](#)
- Repositories.Impl, [21](#)
- Repositories.Impl.BootableDiskRepository, [33](#)
 - GetById, [34](#)
 - GetOnlyWithNotNullablePath, [34](#)
- Repositories.Impl.InternetEntityRepository, [81](#)
 - Create, [81](#)
 - GetAll, [81](#)
 - GetById, [82](#)
 - Remove, [82](#)
 - UpdatePosition, [82](#)
- Repositories.Impl.PacketSnifferRepository, [131](#)
 - AddSniffer, [131](#)
 - GetOpenedSniffers, [132](#)
 - RemoveSniffer, [132](#)
- Repositories.Impl.VirtualMachineConsoleDataRepository, [149](#)
 - AddDataToConsole, [149](#)
 - GetData, [149](#)
- Repositories.Impl.VirtualMachineConsoleStreamRepository, [155](#)
 - Add, [156](#)
 - GetAllStreams, [156](#)
 - GetByUuid, [156](#)
 - Remove, [157](#)
- Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, [166](#)
 - Create, [166](#)
 - GetByIds, [167](#)
 - GetByNetworkName, [167](#)
 - GetByVirtualMachineId, [167](#)
 - Remove, [168](#)
 - Update, [168](#)
- Repositories.Impl.VirtualMachineEntityRepository, [168](#)
 - Add, [169](#)
 - GetAll, [169](#)
 - GetById, [169](#)
 - Remove, [170](#)
 - Update, [170](#)
- Repositories.Impl.VirtualNetworkConnectionRepository, [182](#)
 - Create, [182](#)
 - GetAll, [183](#)
 - GetById, [183](#)
 - GetUsingEntityId, [183](#)
 - Remove, [184](#)
- Repositories.Impl.VirtualNetworkNodeEntityRepository, [200](#)
 - Create, [201](#)
 - CreateInvisible, [201](#)
 - GetAll, [201](#)
 - GetById, [201](#)
 - GetVisible, [202](#)
 - Remove, [202](#)
 - UpdateEntityPosition, [202](#)
 - UpdateNetwork, [202](#)
- Repositories.Impl.VirtualNetworkRepository, [204](#)
 - Create, [205](#)
 - GetAll, [205](#)
 - GetByBridgeName, [205](#)
 - GetById, [205](#)
 - GetByUuid, [206](#)
 - GetByUuidWithVirtualNetworkNodes, [206](#)
 - Remove, [206](#)
- Repositories.IPacketSnifferRepository, [85](#)
 - AddSniffer, [86](#)
 - GetOpenedSniffers, [86](#)
 - RemoveSniffer, [86](#)
- Repositories.IVirtualMachineConsoleDataRepository, [96](#)
 - AddDataToConsole, [96](#)
 - GetData, [96](#)
- Repositories.IVirtualMachineConsoleStreamRepository, [99](#)
 - Add, [100](#)
 - GetAllStreams, [100](#)
 - GetByUuid, [100](#)
 - Remove, [100](#)
- Repositories.IVirtualMachineEntityNetworkInterfaceRepository, [101](#)
 - Create, [101](#)
 - GetByIds, [102](#)
 - GetByNetworkName, [102](#)
 - GetByVirtualMachineId, [102](#)
 - Remove, [103](#)
 - Update, [103](#)
- Repositories.IVirtualMachineEntityRepository, [103](#)
 - Add, [104](#)
 - GetAll, [104](#)
 - GetById, [104](#)
 - Remove, [105](#)
 - Update, [105](#)
- Repositories.IVirtualNetworkConnectionRepository, [110](#)
 - Create, [111](#)
 - GetAll, [111](#)
 - GetById, [111](#)
 - GetUsingEntityId, [111](#)
 - Remove, [112](#)
- Repositories.IVirtualNetworkNodeEntityRepository, [115](#)
 - Create, [115](#)
 - CreateInvisible, [116](#)
 - GetAll, [116](#)
 - GetById, [116](#)
 - GetVisible, [117](#)
 - Remove, [117](#)
 - UpdateEntityPosition, [117](#)
 - UpdateNetwork, [117](#)
- Repositories.IVirtualNetworkRepository, [118](#)
 - Create, [119](#)
 - GetAll, [119](#)
 - GetByBridgeName, [119](#)
 - GetById, [120](#)

- GetByUuid, 120
- GetByUuidWithVirtualNetworkNodes, 120
- Remove, 121
- RepositoriesTests, 21
- RepositoriesTests.BootableDiskRepositoryTests, 35
- RepositoriesTests.InternetEntityRepositoryTests, 83
- RepositoriesTests.VirtualMachineConsoleDataRepositoryTests, 150
- RepositoriesTests.VirtualMachineConsoleStreamRepositoryTests, 157
- RepositoriesTests.VirtualMachineEntityRepositoryTests, 170
- RepositoriesTests.VirtualNetworkConnectionRepositoryTests, 184
- RepositoriesTests.VirtualNetworkNodeEntityRepositoryTests, 203
- SendBytesToVirtualMachineConsoleByUuid
 - Services.Impl.VirtualMachineConsoleService, 154
 - Services.IVirtualMachineConsoleService, 99
- SendDataToConsole
 - Hubs.VirtualMachineConsoleHub, 151
- SendDataToStream
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 146
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 95
- Services, 22
- Services.IBootableDiskProviderService, 58
 - GetAvailableBootableDiskModels, 58
 - GetBootableDiskPathById, 58
- Services.IDeleteEntityService, 60
 - DeleteInternetEntity, 61
 - DeleteVirtualMachineEntity, 61
 - DeleteVirtualNetworkNodeEntity, 61
- Services.IEntitiesConnectorService, 61
 - ConnectTwoEntities, 62
 - ConnectTwoVirtualMachines, 62
 - ConnectVirtualMachineToInternet, 62
 - ConnectVirtualMachineToVirtualNetworkNode, 63
- Services.IEntitiesDisconnectorService, 63
 - DisconnectEntities, 64
 - DisconnectVirtualMachineFromInternet, 64
 - DisconnectVirtualMachineFromVirtualMachine, 64
 - DisconnectVirtualMachineFromVirtualNetworkNode, 65
- Services.IHypervisorConnectionService, 65
 - GetConnectionInfo, 66
 - GetConnectionStatus, 66
 - InitializeConnection, 66
- Services.IInfoService, 67
 - GetSystemInfo, 67
- Services.IInternetEntityService, 70
 - CreateInternet, 71
 - GetInternetEntities, 71
 - UpdateInternetEntityPosition, 71
- Services.Impl, 22
- Services.Impl.BootableDiskProviderService, 32
 - BootableDiskProviderService, 32
 - GetAvailableBootableDiskModels, 32
 - GetBootableDiskPathById, 32
- Services.Impl.DeleteEntityService, 41
 - DeleteEntityService, 41
 - DeleteInternetEntity, 42
 - DeleteVirtualMachineEntity, 42
 - DeleteVirtualNetworkNodeEntity, 42
- Services.Impl.EntitiesConnectorService, 43
 - ConnectTwoEntities, 44
 - ConnectTwoVirtualMachines, 44
 - ConnectVirtualMachineToInternet, 44
 - ConnectVirtualMachineToVirtualNetworkNode, 46
- Services.Impl.EntitiesDisconnectorService, 46
 - DisconnectEntities, 48
 - DisconnectVirtualMachineFromInternet, 48
 - DisconnectVirtualMachineFromVirtualMachine, 48
 - DisconnectVirtualMachineFromVirtualNetworkNode, 49
- Services.Impl.EntitiesDisconnectorService, 47
- Services.Impl.HypervisorConnectionService, 56
 - GetConnectionInfo, 57
 - GetConnectionStatus, 57
 - HypervisorConnectionService, 56
 - InitializeConnection, 57
- Services.Impl.InfoService, 74
 - GetSystemInfo, 74
- Services.Impl.InitializationService, 75
 - InitializationService, 76
 - StartAsync, 76
 - StopAsync, 76
- Services.Impl.InternetEntityService, 83
 - CreateInternet, 84
 - GetInternetEntities, 84
 - InternetEntityService, 84
 - UpdateInternetEntityPosition, 85
- Services.Impl.PacketSnifferService, 132
 - GetCapturedPacket, 133
 - GetOpenedSniffers, 133
 - ListenForPacket, 134
 - PacketSnifferService, 133
 - StartListeningForBridge, 134
- Services.Impl.VirtualMachineConsoleService, 152
 - CloseStream, 153
 - GetBytesFromConsole, 153
 - GetInitialConsoleData, 153
 - GetStreams, 154
 - OpenVirtualMachineConsole, 154
 - SendBytesToVirtualMachineConsoleByUuid, 154
 - VirtualMachineConsoleService, 153
- Services.Impl.VirtualMachineEntityService, 171
 - CreateEntity, 173
 - GetById, 173
 - GetEntities, 173
 - GetMacAddresses, 174
 - UpdateEntityPosition, 174
 - UpdateEntityVmUUID, 174
 - VirtualMachineEntityService, 172
- Services.Impl.VirtualMachineManagerService, 177

- AttachVirtualNetworkInterfaceToVirtualMachine, 178
- CreateVirtualMachine, 179
- GetListOfVirtualMachines, 179
- VirtualMachineManagerService, 177
- Services.Impl.VirtualNetworkNodeConnector, 197
 - ConnectTwoVirtualNetworkNodes, 197
 - ConnectVirtualNetworkNodeToInternet, 198
 - DisconnectTwoVirtualNetworkNodes, 198
 - DisconnectVirtualNetworkNodeFromInternet, 198
- Services.Impl.VirtualNetworkService, 207
 - AttachNetworkInterfaceToVirtualMachine, 208
 - CreateInternetVirtualNetwork, 208
 - CreateNodeVirtualNetwork, 209
 - CreateVirtualNetworkNode, 209
 - GetAllVirtualNetworks, 209
 - GetVirtualNetworkConnections, 209
 - GetVirtualNetworkUsingBridgeName, 210
 - GetVisibleVirtualNetworkNodeEntities, 210
 - UpdateNetworkForVirtualMachineNetworkInterface, 210
 - UpdateVirtualNetworkNodeEntityPosition, 211
- VirtualNetworkService, 208
- Services.IPacketSnifferService, 87
 - GetCapturedPacket, 87
 - GetOpenedSniffers, 87
 - ListenForPacket, 88
 - StartListeningForBridge, 88
- Services.IVirtualMachineConsoleService, 97
 - CloseStream, 97
 - GetBytesFromConsole, 98
 - GetInitialConsoleData, 98
 - GetStreams, 98
 - OpenVirtualMachineConsole, 98
 - SendBytesToVirtualMachineConsoleByUuid, 99
- Services.IVirtualMachineEntityService, 105
 - CreateEntity, 106
 - GetById, 106
 - GetEntities, 107
 - GetMacAddresses, 107
 - UpdateEntityPosition, 107
 - UpdateEntityVmUUID, 108
- Services.IVirtualMachineManagerService, 108
 - AttachVirtualNetworkInterfaceToVirtualMachine, 109
 - CreateVirtualMachine, 109
 - GetListOfVirtualMachines, 109
- Services.IVirtualNetworkNodeConnector, 112
 - ConnectTwoVirtualNetworkNodes, 113
 - ConnectVirtualNetworkNodeToInternet, 113
 - DisconnectTwoVirtualNetworkNodes, 114
 - DisconnectVirtualNetworkNodeFromInternet, 114
- Services.IVirtualNetworkService, 121
 - AttachNetworkInterfaceToVirtualMachine, 122
 - CreateInternetVirtualNetwork, 122
 - CreateNodeVirtualNetwork, 123
 - CreateVirtualNetworkNode, 123
 - GetAllVirtualNetworks, 123
 - GetVirtualNetworkConnections, 123
 - GetVirtualNetworkUsingBridgeName, 124
 - GetVisibleVirtualNetworkNodeEntities, 124
 - UpdateNetworkForVirtualMachineNetworkInterface, 124
 - UpdateVirtualNetworkNodeEntityPosition, 125
- Services.Utils, 23
- Services.Utils.BaseDefinitionBuilder< TDefinition >, 25
 - Build, 26
 - BuildingBlock, 26
 - ChecksEverythingIsProvided, 27
 - SetFromCreateDefinition, 27
 - XmlWriter, 27
- Services.Utils.VirtualMachineDefinitionBuilder, 160
 - Build, 161
 - SetFromCreateDefinition, 161
 - SetPrefix, 161
- Services.Utils.VirtualNetworkCreateDefinitionBuilder, 189
 - Build, 190
 - SetFromCreateDefinition, 190
- Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder, 193
 - Build, 194
 - SetFromCreateDefinition, 194
 - SetFromXml, 194
 - SetNetworkName, 195
- ServicesTests, 23
- ServicesTests.BootableDiskProviderServiceTests, 33
- ServicesTests.HypervisorConnectionServiceTests, 58
- ServicesTests.InfoServiceTests, 75
- ServicesTests.InitializationServiceTests, 77
- ServicesTests.Utils, 24
- ServicesTests.Utils.VirtualMachineDefinitionBuilderTests, 162
- ServicesTests.Utils.VirtualNetworkDefinitionBuilderTests, 191
- ServicesTests.Utils.VirtualNetworkInterfaceCreateDefinitionBuilderTests, 195
- ServicesTests.VirtualMachineConsoleServiceTests, 155
- ServicesTests.VirtualMachineEntityServiceTests, 175
- ServicesTests.VirtualMachineManagerServiceTests, 180
- SetFromCreateDefinition
 - Services.Utils.BaseDefinitionBuilder< TDefinition >, 27
 - Services.Utils.VirtualMachineDefinitionBuilder, 161
 - Services.Utils.VirtualNetworkCreateDefinitionBuilder, 190
 - Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder, 194
- SetFromXml
 - Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder, 194
- SetNetworkName
 - Services.Utils.VirtualNetworkInterfaceCreateDefinitionBuilder, 195
- SetPrefix

- Services.Utils.VirtualMachineDefinitionBuilder, 161
- StartAsync
 - Services.Impl.InitializationService, 76
- StartListeningForBridge
 - Services.Impl.PacketSnifferService, 134
 - Services.IPacketSnifferService, 88
- StopAsync
 - Services.Impl.InitializationService, 76
- TestUtils, 24
- Unknown
 - Models.Enums, 19
- Update
 - Repositories.Impl.VirtualMachineEntityNetworkInterfaceRepository, 168
 - Repositories.Impl.VirtualMachineEntityRepository, 170
 - Repositories.IVirtualMachineEntityNetworkInterfaceRepository, 103
 - Repositories.IVirtualMachineEntityRepository, 105
- UpdateEntityPosition
 - Controllers.EntityController, 52
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 202
 - Repositories.IVirtualNetworkNodeEntityRepository, 117
 - Services.Impl.VirtualMachineEntityService, 174
 - Services.IVirtualMachineEntityService, 107
- UpdateEntityVmUUID
 - Services.Impl.VirtualMachineEntityService, 174
 - Services.IVirtualMachineEntityService, 108
- UpdateInternetEntityPosition
 - Services.IInternetEntityService, 71
 - Services.Impl.InternetEntityService, 85
- UpdateNetwork
 - Repositories.Impl.VirtualNetworkNodeEntityRepository, 202
 - Repositories.IVirtualNetworkNodeEntityRepository, 117
- UpdateNetworkForVirtualMachineNetworkInterface
 - Services.Impl.VirtualNetworkService, 210
 - Services.IVirtualNetworkService, 124
- UpdatePosition
 - Repositories.IInternetEntityRepository, 69
 - Repositories.Impl.InternetEntityRepository, 82
- UpdateVirtualNetworkNodeEntityPosition
 - Services.Impl.VirtualNetworkService, 211
 - Services.IVirtualNetworkService, 125
- UpdateVmDevice
 - NativeLibrary.Wrappers.Impl.VirtualizationWrapper, 147
 - NativeLibrary.Wrappers.IVirtualizationWrapper, 95
- VirtualMachine
 - Models.Enums, 17
- VirtualMachineConsoleBackgroundService
 - BackgroundServices.Impl.VirtualMachineConsoleBackgroundService, 148
- VirtualMachineConsoleService
 - Services.Impl.VirtualMachineConsoleService, 153
- VirtualMachineController
 - Controllers.VirtualMachineController, 158
- VirtualMachineEntityService
 - Services.Impl.VirtualMachineEntityService, 172
- VirtualMachineManagerService
 - Services.Impl.VirtualMachineManagerService, 177
- VirtualMachineState
 - Models.Enums, 17
- VirtualNetworkController
 - Controllers.VirtualNetworkController, 186
- VirtualNetworkNode
 - Models.Enums, 17
- VirtualNetworkService
 - Services.Impl.VirtualNetworkService, 208
- WithEmptySuccess
 - Models.Responses.BaseResponse< TResponse, TSelf >, 30
- WithSuccess
 - Models.Responses.BaseResponse< TResponse, TSelf >, 30
- XmlWriter
 - Services.Utils.BaseDefinitionBuilder< TDefinition >, 27