

SWEENEYTHREADS

ACTORBASE

A NoSQL DB BASED ON THE ACTOR MODEL

Specifica Tecnica

Redattori:

Bortolazzo Matteo

Maino Elia

Nicoletti Luca

Approvazione:

Bonato Paolo

Verifica:

Padovan Tommaso

Tommasin Davide



Versione 2.0.2

15 giugno 2016

Indice

1	Introduzione	13
1.1	Scopo del documento	13
1.2	Scopo del prodotto	13
1.3	Glossario	13
1.4	Riferimenti	13
1.4.1	Normativi	13
2	Tecnologie utilizzate	14
2.1	Scala	14
2.2	Akka	14
3	Legenda	15
4	Descrizione dell'architettura	15
4.1	Metodo e formalismo di specifica	15
4.2	Architettura generale	15
4.2.1	Server	16
4.2.2	Client	16
4.2.3	Driver	16
5	Componenti e Classi	17
5.1	Actorbase	17
5.1.1	Descrizione	17
5.1.2	Package Figli	17
5.2	Actorbase.server	17
5.2.1	Descrizione	17
5.2.2	Package Figli	18
5.2.3	Classi	18
5.3	Actorbase.server.Server	18
5.3.1	Descrizione	18
5.3.2	Utilizzo	18
5.3.3	Relazione con altre classi	18
5.4	Actorbase.server.utils	19
5.4.1	Descrizione	19
5.4.2	Classi	19
5.4.3	Trait	19
5.5	Actorbase.server.utils.Parser	19
5.5.1	Descrizione	19
5.5.2	Utilizzo	19
5.5.3	Relazione con altre classi	20
5.6	Actorbase.server.utils.ReplyBuilder	20
5.6.1	Descrizione	20
5.6.2	Utilizzo	20
5.6.3	Relazione con altre classi	20
5.7	Actorbase.server.utils.Serializer	20
5.7.1	Descrizione	20
5.7.2	Utilizzo	20
5.7.3	Relazione con altre classi	20
5.8	Actorbase.server.utils.FileManager	20
5.8.1	Descrizione	20
5.8.2	Utilizzo	20
5.8.3	Relazione con altre classi	20
5.9	Actorbase.server.utils.ConfigurationManager	20
5.9.1	Descrizione	20
5.9.2	Utilizzo	20
5.9.3	Relazione con altre classi	21
5.10	Actorbase.server.utils.Helper	21

5.10.1	Descrizione	21
5.10.2	Utilizzo	21
5.10.3	Relazione con altre classi	21
5.11	Actorbase.server.actors	21
5.11.1	Descrizione	21
5.11.2	Classi	22
5.11.3	Trait	22
5.12	Actorbase.server.actors.Doorkeeper	22
5.12.1	Descrizione	22
5.12.2	Utilizzo	22
5.12.3	Relazione con altre classi	22
5.13	Actorbase.server.actors.Usermanager	22
5.13.1	Descrizione	22
5.13.2	Utilizzo	22
5.13.3	Relazione con altre classi	22
5.14	Actorbase.server.actors.Main	23
5.14.1	Descrizione	23
5.14.2	Utilizzo	23
5.14.3	Relazione con altre classi	23
5.15	Actorbase.server.actors.MapManager	23
5.15.1	Descrizione	23
5.15.2	Utilizzo	23
5.15.3	Relazione con altre classi	23
5.16	Actorbase.server.actors.IndexManager	24
5.16.1	Descrizione	24
5.16.2	Utilizzo	24
5.16.3	Relazione con altre classi	24
5.17	Actorbase.server.actors.Storemanager	24
5.17.1	Descrizione	24
5.17.2	Utilizzo	24
5.17.3	Relazione con altre classi	24
5.18	Actorbase.server.actors.Warehouseman	24
5.18.1	Descrizione	24
5.18.2	Utilizzo	24
5.18.3	Relazione con altre classi	25
5.19	Actorbase.server.actors.ClusterAwareActor	25
5.19.1	Descrizione	25
5.19.2	Utilizzo	25
5.19.3	Interfacce figlie	25
5.20	Actorbase.server.actors.ReplyActor	25
5.20.1	Descrizione	25
5.20.2	Utilizzo	25
5.20.3	Classi figlie	25
5.21	Actorbase.server.enums	26
5.21.1	Descrizione	26
5.21.2	Package Figli	26
5.22	Actorbase.server.enums.EnumPermissions	26
5.22.1	Descrizione	26
5.22.2	Trait	26
5.22.3	Classi	27
5.23	Actorbase.server.enums.EnumPermissions.Permission	27
5.23.1	Descrizione	27
5.23.2	Utilizzo	27
5.23.3	Relazione con altre classi	27
5.23.4	Classi figlie	27
5.24	Actorbase.server.enums.EnumPermissions.Read	27
5.24.1	Descrizione	27
5.24.2	Utilizzo	27

5.24.3	Relazione con altre classi	27
5.24.4	Trait implementati	27
5.25	Actorbase.server.enums.EnumPermissions.ReadWrite	27
5.25.1	Descrizione	27
5.25.2	Utilizzo	27
5.25.3	Relazione con altre classi	27
5.25.4	Trait implementati	27
5.26	Actorbase.server.enums.EnumReplyResults	28
5.26.1	Descrizione	28
5.26.2	Trait	28
5.26.3	Classi	28
5.27	Actorbase.server.enums.EnumReplyResults.ReplyResult	28
5.27.1	Descrizione	28
5.27.2	Utilizzo	28
5.27.3	Relazione con altre classi	28
5.27.4	Classi figlie	28
5.28	Actorbase.server.enums.EnumReplyResults.Done	28
5.28.1	Descrizione	28
5.28.2	Utilizzo	28
5.28.3	Relazione con altre classi	29
5.28.4	Trait implementati	29
5.29	Actorbase.server.enums.EnumReplyResults.Error	29
5.29.1	Descrizione	29
5.29.2	Utilizzo	29
5.29.3	Relazione con altre classi	29
5.29.4	Trait implementati	29
5.30	Actorbase.server.enums.EnumStoremanagerTypes	29
5.30.1	Descrizione	29
5.30.2	Trait	29
5.30.3	Classi	30
5.31	Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType	30
5.31.1	Descrizione	30
5.31.2	Utilizzo	30
5.31.3	Relazione con altre classi	30
5.31.4	Classi figlie	30
5.32	Actorbase.server.enums.EnumStoremanagerTypes.StorefinderType	30
5.32.1	Descrizione	30
5.32.2	Utilizzo	30
5.32.3	Relazione con altre classi	30
5.32.4	Trait implementati	30
5.33	Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperType	30
5.33.1	Descrizione	30
5.33.2	Utilizzo	30
5.33.3	Relazione con altre classi	30
5.33.4	Trait implementati	31
5.34	Actorbase.server.enums.EnumStoremanagerTypes.StorefinderNinjaType	31
5.34.1	Descrizione	31
5.34.2	Utilizzo	31
5.34.3	Relazione con altre classi	31
5.34.4	Trait implementati	31
5.35	Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperNinjaType	31
5.35.1	Descrizione	31
5.35.2	Utilizzo	31
5.35.3	Relazione con altre classi	31
5.35.4	Trait implementati	31
5.36	Actorbase.server.messages	32
5.36.1	Descrizione	32
5.36.2	Package Figli	32

5.37	Actorbase.server.messages.internal	33
5.37.1	Descrizione	33
5.37.2	Package Figli	33
5.38	Actorbase.server.messages.internal.AskMessages	34
5.38.1	Descrizione	34
5.38.2	Classi	34
5.39	Actorbase.server.messages.internal.AskMessages.AskMapMessage	34
5.39.1	Descrizione	34
5.39.2	Utilizzo	34
5.39.3	Relazione con altre classi	34
5.40	Actorbase.server.messages.internal.AskMessages.AskDatabaseMessage	34
5.40.1	Descrizione	34
5.40.2	Utilizzo	34
5.40.3	Relazione con altre classi	34
5.41	Actorbase.server.messages.internal.ScalabilityMessages	35
5.41.1	Descrizione	35
5.41.2	Classi	35
5.42	Actorbase.server.messages.internal.ScalabilityMessages.SendMapMessage	35
5.42.1	Descrizione	35
5.42.2	Utilizzo	35
5.43	Actorbase.server.messages.internal.LinkMessages	36
5.43.1	Descrizione	36
5.43.2	Classi	36
5.43.3	Trait	36
5.44	Actorbase.server.messages.internal.LinkMessages.LinkMessage	36
5.44.1	Descrizione	36
5.44.2	Utilizzo	36
5.44.3	Relazione con altre classi	36
5.44.4	Classi figlie	37
5.45	Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage	37
5.45.1	Descrizione	37
5.45.2	Utilizzo	37
5.45.3	Relazione con altre classi	37
5.45.4	Trait implementati	37
5.46	Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage	37
5.46.1	Descrizione	37
5.46.2	Utilizzo	37
5.46.3	Relazione con altre classi	37
5.46.4	Trait implementati	37
5.47	Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage	37
5.47.1	Descrizione	37
5.47.2	Utilizzo	37
5.47.3	Relazione con altre classi	37
5.47.4	Trait implementati	38
5.48	Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage	38
5.48.1	Descrizione	38
5.48.2	Utilizzo	38
5.48.3	Relazione con altre classi	38
5.48.4	Trait implementati	38
5.49	Actorbase.server.messages.internal.LinkMessages.BecomeStorefinderNinjaMessage	38
5.49.1	Descrizione	38
5.49.2	Utilizzo	38
5.49.3	Relazione con altre classi	38
5.49.4	Trait implementati	38
5.50	Actorbase.server.messages.internal.StorageMessages	38
5.50.1	Descrizione	39
5.50.2	Classi	39
5.50.3	Trait	39

5.51	Actorbase.server.messages.internal.StorageMessages.StorageMessage	39
5.51.1	Descrizione	39
5.51.2	Utilizzo	39
5.51.3	Relazione con altre classi	39
5.51.4	Classi figlie	39
5.52	Actorbase.server.messages.internal.StorageMessages.WriteMapMessage	39
5.52.1	Descrizione	39
5.52.2	Utilizzo	39
5.52.3	Relazione con altre classi	39
5.52.4	Trait implementati	39
5.53	Actorbase.server.messages.internal.StorageMessages.ReadMapMessage	39
5.53.1	Descrizione	39
5.53.2	Utilizzo	39
5.53.3	Relazione con altre classi	40
5.53.4	Trait implementati	40
5.54	Actorbase.server.messages.query	40
5.54.1	Descrizione	40
5.54.2	Package Figli	40
5.54.3	Classi	40
5.54.4	Trait	40
5.55	Actorbase.server.messages.query.QueryMessage	41
5.55.1	Descrizione	41
5.55.2	Utilizzo	41
5.55.3	Relazioni con altre classi	41
5.55.4	Classi figlie	41
5.56	Actorbase.server.messages.query.LoginMessage	41
5.56.1	Descrizione	41
5.56.2	Utilizzo	41
5.56.3	Relazioni con altre classi	41
5.56.4	Trait implementati	41
5.57	Actorbase.server.messages.query.ErrorMessages	42
5.57.1	Descrizione	42
5.57.2	Classi	42
5.57.3	Trait	42
5.58	Actorbase.server.messages.query.ErrorMessages.ErrorMessage	42
5.58.1	Descrizione	42
5.58.2	Utilizzo	42
5.58.3	Relazione con altre classi	42
5.58.4	Classi figlie	42
5.59	Actorbase.server.messages.query.ErrorMessages.InvalidQueryMessage	42
5.59.1	Descrizione	42
5.59.2	Utilizzo	42
5.59.3	Relazione con altre classi	43
5.59.4	Trait implementati	43
5.60	Actorbase.server.messages.query.ErrorMessages.QueryErrorInfo	43
5.60.1	Descrizione	43
5.60.2	Utilizzo	43
5.60.3	Relazione con altre classi	43
5.60.4	Trait implementati	43
5.61	Actorbase.server.messages.query.PermissionMessages	43
5.61.1	Descrizione	43
5.61.2	Trait	44
5.62	Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage	44
5.62.1	Descrizione	44
5.62.2	Utilizzo	44
5.62.3	Classi figlie	44
5.63	Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage	44
5.63.1	Descrizione	44

5.63.2	Utilizzo	44
5.63.3	Classi figlie	44
5.64	Actorbase.server.messages.query.PermissionMessages.ReadMessage	44
5.64.1	Descrizione	44
5.64.2	Utilizzo	44
5.64.3	Classi figlie	44
5.65	Actorbase.server.messages.query.PermissionMessages.ReadWriteMessage	45
5.65.1	Descrizione	45
5.65.2	Utilizzo	45
5.65.3	Classi figlie	45
5.66	Actorbase.server.messages.query.ReplyMessages	45
5.66.1	Descrizione	45
5.66.2	Trait	45
5.66.3	Classi	46
5.67	Actorbase.server.messages.query.ReplyMessages.ReplyInfo	46
5.67.1	Descrizione	46
5.67.2	Utilizzo	46
5.67.3	Classi figlie	46
5.68	Actorbase.server.messages.query.ReplyMessages.ReplyMessage	46
5.68.1	Descrizione	46
5.68.2	Utilizzo	46
5.68.3	Trait implementati	46
5.69	Actorbase.server.messages.query.ReplyMessages.ReplyErrorInfo	46
5.69.1	Descrizione	46
5.69.2	Utilizzo	46
5.69.3	Trait implementati	46
5.70	Actorbase.server.messages.query.admin	47
5.70.1	Descrizione	47
5.70.2	Package Figli	47
5.70.3	Trait	47
5.71	Actorbase.server.messages.query.admin.AdminMessage	47
5.71.1	Descrizione	47
5.71.2	Utilizzo	47
5.71.3	Relazioni con altre classi	47
5.71.4	Classi figlie	48
5.72	Actorbase.server.messages.query.admin.SettingMessages	48
5.72.1	Descrizione	48
5.72.2	Trait	48
5.72.3	Classi	48
5.73	Actorbase.server.messages.query.admin.SettingMessages.SettingMessage	48
5.73.1	Descrizione	48
5.73.2	Utilizzo	48
5.73.3	Classi figlie	48
5.73.4	Trait estesi	48
5.74	Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsMessage	49
5.74.1	Descrizione	49
5.74.2	Utilizzo	49
5.74.3	Trait implementati	49
5.75	Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsInfo	49
5.75.1	Descrizione	49
5.75.2	Utilizzo	49
5.75.3	Trait implementati	49
5.76	Actorbase.server.messages.query.admin.PermissionsManagementMessages	49
5.76.1	Descrizione	49
5.76.2	Classi	49
5.76.3	Trait	49
5.77	Actorbase.server.messages.query.admin	
	.PermissionsManagementMessages.PermissionsManagementMessage	50

5.77.1	Descrizione	50
5.77.2	Utilizzo	50
5.77.3	Relazioni con altre classi	50
5.77.4	Classi figlie	50
5.78	Actorbase.server.messages.query.admin	
	.PermissionsManagementMessages.AddPermissionMessage	50
5.78.1	Descrizione	50
5.78.2	Utilizzo	50
5.78.3	Relazioni con altre classi	50
5.78.4	Trait implementati	50
5.79	Actorbase.server.messages.query.admin	
	.PermissionsManagementMessages.RemovePermissionMessage	50
5.79.1	Descrizione	50
5.79.2	Utilizzo	50
5.79.3	Relazioni con altre classi	51
5.79.4	Trait implementati	51
5.80	Actorbase.server.messages.query.admin	
	.PermissionsManagementMessages.ListPermissionMessage	51
5.80.1	Descrizione	51
5.80.2	Utilizzo	51
5.80.3	Relazioni con altre classi	51
5.80.4	Trait implementati	51
5.81	Actorbase.server.messages.query.admin.UserManagementMessages	51
5.81.1	Descrizione	51
5.81.2	Classi	52
5.81.3	Trait	52
5.82	Actorbase.server.messages.query.admin	
	.UserManagementMessages.UserManagementMessage	52
5.82.1	Descrizione	52
5.82.2	Utilizzo	52
5.82.3	Relazioni con altre classi	52
5.82.4	Classi figlie	52
5.83	Actorbase.server.messages.query.admin	
	.UserManagementMessages.AddUserMessage	52
5.83.1	Descrizione	52
5.83.2	Utilizzo	52
5.83.3	Relazioni con altre classi	52
5.83.4	Trait implementati	52
5.84	Actorbase.server.messages.query.admin	
	.UserManagementMessages.RemoveUserMessage	53
5.84.1	Descrizione	53
5.84.2	Utilizzo	53
5.84.3	Relazioni con altre classi	53
5.84.4	Trait implementati	53
5.85	Actorbase.server.messages.query.admin	
	.UserManagementMessages.ListUserMessage	53
5.85.1	Descrizione	53
5.85.2	Utilizzo	53
5.85.3	Relazioni con altre classi	53
5.85.4	Trait implementati	53
5.86	Actorbase.server.messages.query.user	54
5.86.1	Descrizione	54
5.86.2	Package Figli	54
5.86.3	Trait	54
5.87	Actorbase.server.messages.query.user.UserMessage	55
5.87.1	Descrizione	55
5.87.2	Utilizzo	55
5.87.3	Relazioni con altre classi	55

5.87.4	Classi figlie	55
5.88	Actorbase.server.messages.query.user.RowMessages	56
5.88.1	Descrizione	56
5.88.2	Classi	56
5.88.3	Trait	56
5.89	Actorbase.server.messages.query.user.RowMessages.RowMessage	56
5.89.1	Descrizione	56
5.89.2	Utilizzo	57
5.89.3	Relazioni con altre classi	57
5.89.4	Classi figlie	57
5.90	Actorbase.server.messages.query.user.RowMessages.InsertRowMessage	57
5.90.1	Descrizione	57
5.90.2	Utilizzo	57
5.90.3	Relazioni con altre classi	57
5.90.4	Trait implementati	57
5.91	Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage	57
5.91.1	Descrizione	57
5.91.2	Utilizzo	58
5.91.3	Relazioni con altre classi	58
5.91.4	Trait implementati	58
5.92	Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage	58
5.92.1	Descrizione	58
5.92.2	Utilizzo	58
5.92.3	Relazioni con altre classi	58
5.92.4	Trait implementati	58
5.93	Actorbase.server.messages.query.user.RowMessages.FindRowMessage	58
5.93.1	Descrizione	58
5.93.2	Utilizzo	58
5.93.3	Relazioni con altre classi	59
5.93.4	Trait implementati	59
5.94	Actorbase.server.messages.query.user.RowMessages.ListKeysMessage	59
5.94.1	Descrizione	59
5.94.2	Utilizzo	59
5.94.3	Relazioni con altre classi	59
5.94.4	Trait implementati	59
5.95	Actorbase.server.messages.query.user.MapMessages	60
5.95.1	Descrizione	60
5.95.2	Classi	60
5.95.3	Trait	60
5.96	Actorbase.server.messages.query.user.MapMessages.MapMessage	60
5.96.1	Descrizione	60
5.96.2	Utilizzo	60
5.96.3	Relazioni con altre classi	61
5.96.4	Classi figlie	61
5.97	Actorbase.server.messages.query.user.MapMessages.CreateMapMessage	61
5.97.1	Descrizione	61
5.97.2	Utilizzo	61
5.97.3	Relazioni con altre classi	61
5.97.4	Trait implementati	61
5.98	Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage	61
5.98.1	Descrizione	61
5.98.2	Utilizzo	61
5.98.3	Relazioni con altre classi	62
5.98.4	Trait implementati	62
5.99	Actorbase.server.messages.query.user.MapMessages.SelectMapMessage	62
5.99.1	Descrizione	62
5.99.2	Utilizzo	62
5.99.3	Relazioni con altre classi	62

5.99.4	Trait implementati	62
5.100	Actorbase.server.messages.query.user.MapMessages.ListMapMessage	62
5.100.1	Descrizione	62
5.100.2	Utilizzo	62
5.100.3	Relazioni con altre classi	63
5.100.4	Trait implementati	63
5.101	Actorbase.server.messages.query.user.DatabaseMessages	63
5.101.1	Descrizione	63
5.101.2	Classi	63
5.101.3	Trait	63
5.102	Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage	64
5.102.1	Descrizione	64
5.102.2	Utilizzo	64
5.102.3	Relazioni con altre classi	64
5.102.4	Classi figlie	64
5.103	Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage	64
5.103.1	Descrizione	64
5.103.2	Utilizzo	64
5.103.3	Relazioni con altre classi	64
5.103.4	Trait implementati	64
5.104	Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage	64
5.104.1	Descrizione	64
5.104.2	Utilizzo	64
5.104.3	Relazioni con altre classi	65
5.104.4	Trait implementati	65
5.105	Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage	65
5.105.1	Descrizione	65
5.105.2	Utilizzo	65
5.105.3	Relazioni con altre classi	65
5.105.4	Trait implementati	65
5.106	Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage	65
5.106.1	Descrizione	65
5.106.2	Utilizzo	65
5.106.3	Relazioni con altre classi	65
5.106.4	Trait implementati	65
5.107	Actorbase.server.messages.query.user.HelpMessages	66
5.107.1	Descrizione	66
5.107.2	Classi	66
5.107.3	Trait	66
5.108	Actorbase.server.messages.query.user.HelpMessages.HelpMessage	66
5.108.1	Descrizione	66
5.108.2	Utilizzo	66
5.108.3	Relazioni con altre classi	66
5.108.4	Classi figlie	66
5.109	Actorbase.server.messages.query.user.HelpMessages.CompleteHelp	66
5.109.1	Descrizione	66
5.109.2	Utilizzo	67
5.109.3	Relazioni con altre classi	67
5.109.4	Trait implementati	67
5.110	Actorbase.server.messages.query.user.HelpMessages.SpecificHelp	67
5.110.1	Descrizione	67
5.110.2	Utilizzo	67
5.110.3	Relazioni con altre classi	67
5.110.4	Trait implementati	67
5.111	Actorbase.client	68
5.111.1	Descrizione	68
5.111.2	Classi	68
5.112	Actorbase.client.Client	68

5.112.1	Descrizione	68
5.112.2	Utilizzo	68
5.112.3	Relazioni con altre classi	69
5.113	Actorbase.client.Welcome	69
5.113.1	Descrizione	69
5.113.2	Utilizzo	69
5.113.3	Relazioni con altre classi	69
5.114	Actorbase.driver	69
5.114.1	Descrizione	69
5.114.2	Classi	70
5.114.3	Trait	70
5.115	Actorbase.driver.Connection	70
5.115.1	Descrizione	70
5.115.2	Utilizzo	70
5.115.3	Relazioni con altre classi	70
5.115.4	Classi figlie	70
5.116	Actorbase.driver.ConcreteConnection	70
5.116.1	Descrizione	70
5.116.2	Utilizzo	70
5.116.3	Relazioni con altre classi	70
5.116.4	Trait implementati	70
5.117	Actorbase.driver.Driver	70
5.117.1	Descrizione	70
5.117.2	Utilizzo	70
5.117.3	Relazioni con altre classi	71
6	Diagrammi delle attività	72
6.0.1	Diagramma attività principale	72
7	Struttura del sistema ad attori	73
7.1	Struttura principale	73
7.2	Struttura di un database	74
7.3	Struttura di una mappa in memoria principale	75
7.4	Struttura degli attori di backup	76
8	Tracciamento	77
8.1	Tracciamento componenti-requisiti	77
8.2	Tracciamento requisiti-componenti	77
9	Appendice	79
9.1	Descrizione Design Pattern	79
9.1.1	Broker event-driven	79
9.1.2	Singleton	79
	Elenco delle figure	81
	Elenco delle tabelle	82

Diario delle modifiche

Versione	Data	Autore	Descrizione
2.0.2	2016-05-30	<i>Progettista</i> Maino Elia	Riorganizzazione diagrammi UML e stesura sezione relativa alla struttura del modello ad attori del sistema.
2.0.1	2016-05-26	<i>Progettista</i> Maino Elia	Riorganizzazione delle componenti server in base alle modifiche progettuali recenti.
2.0.0	2016-05-16	<i>Responsabile</i> Bonato Paolo	Approvazione
1.1.1	2016-05-15	<i>Verificatore</i> Biggeri Mattia	Correzioni del documento da verifica: sistematizzati i titoli delle section, aggiunta sezione per descrivere la legenda, aggiunti lati negativi delle tecnologie utilizzate, aggiunta sezione desing pattern utilizzati
1.1.0	2016-05-14	<i>Verificatore</i> Bonato Paolo	Verifica del documento
1.0.5	2016-05-10	<i>Programmatore</i> Nicoletti Luca <i>Programmatore</i> Bortolazzo Matteo	Modificata la sezione Actorbase.driver.
1.0.4	2016-05-08	<i>Programmatore</i> Nicoletti Luca <i>Programmatore</i> Bortolazzo Matteo	Modificata la sezione Actorbase.client.
1.0.3	2016-05-06	<i>Programmatore</i> Nicoletti Luca <i>Programmatore</i> Bortolazzo Matteo	Modificata la sezione Actorbase.server.messages.
1.0.2	2016-05-04	<i>Programmatore</i> Nicoletti Luca <i>Programmatore</i> Bortolazzo Matteo	Modificata la sezione Actorbase.server.actors.
1.0.1	2016-05-01	<i>Programmatore</i> Nicoletti Luca <i>Programmatore</i> Bortolazzo Matteo	Iniziata la revisione, modificata la prima parte del Server.
1.0.0	2016-04-11	<i>Responsabile</i> Maino Elia	Documento approvato
0.3.0	2016-04-11	<i>Verificatore</i> Padovan Tommaso	Verificato il documento
0.2.1	2016-04-11	<i>Progettista</i> Nicoletti Luca	Riviste sezione da 4.8 a 4.17 e modificata sintassi delle classi correlate ad ogni classe. Sostituite vecchie immagini con immagini aggiornate (inserite aggregazioni tra le classi)
0.2.0	2016-04-11	<i>Verificatore</i> Bortolazzo Matteo	Verificato le tabelle inserite in v0.1.2 e il diario delle modifiche
0.1.3	2016-04-10	<i>Progettista</i> Nicoletti Luca	Separata tabella del diario della modifiche in file esterno.
0.1.2	2016-04-10	<i>Progettista</i> Biggeri Mattia	inserita tabella requisito-componente
0.1.1	2016-04-10	<i>Progettista</i> Nicoletti Luca	Sistemazione errori rilevati dalla verifica effettuata in versione 0.0.12
0.1.0	2016-04-10	<i>Verificatore</i> Bortolazzo Matteo	Verificato l'intero documento, segnalati vari errori in tutte le sezioni
0.0.11	2016-04-10	<i>Progettista</i> Maino Elia	Completata la descrizione di classe e interfacce del componente server

Versione	Data	Autore	Descrizione
0.0.10	2016-04-10	<i>Progettista</i> Nicoletti Luca	Modificate e inserite le nuove immagini dei packages e delle classi per la sezione Server (4.2). Modificate le immagini nei diagrammi di sequenza (Sezione 6) e modificate le descrizioni di tutti in modo da uniformarle. Cambiate le immagini dei Design Pattern (Sezione 9.1)
0.0.9	2016-04-10	<i>Progettista</i> Padovan Tommaso	Incremento sezione Componenti e Classi: allineata sezione Driver allo standard, stesura sezione Client.
0.0.8	2016-04-09	<i>Progettisti</i> Bonato Paolo Biggeri Mattia	Stesura sezione tracciamento componenti-requisiti.
0.0.7	2016-04-09	<i>Progettista</i> Nicoletti Luca	Inseriti tutti i diagrammi di sequenza riguardanti l'implementazione di richieste lato server, sostituita immagine dei Packages generale. Inserita una breve spiegazione di ogni diagramma di sequenza
0.0.6	2016-04-09	<i>Progettisti</i> Biggeri Mattia Padovan Tommaso Bonato Paolo	Stesura sezioni Driver, Diagrammi attività, Stime di fattibilità e bisogno di risorse; aggiunta immagine nella sezione 3.2.3, aggiunto Singleton nelle descrizioni dei Design Pattern, aggiornata legenda.
0.0.5	2016-04-08	<i>Progettisti</i> Maino Elia Nicoletti Luca Bortolazzo Matteo	Stesura sezione riguardante le componenti dell'architettura lato server: diagrammi dei package e delle classi e descrizioni testuali
0.0.4	2016-04-06	<i>Progettisti</i> Biggeri Mattia Tommasin Davide	Aggiunta sezione in appendice suoi Design Pattern, contiene al momento la descrizione di: MVC, Event-driven, Command
0.0.3	2016-04-03	<i>Progettista</i> Bonato Paolo	Accorpate le sezioni "Componenti", "Package" e "Classi" in "Componenti e classi". Riadattata la sezione "Metodo e formalismo di specifica" alla nuova struttura. Inserite le immagini 1 e 2. Apportate le correzioni indicate.
0.0.2	2016-03-26	<i>Progettisti</i> Bonato Paolo Biggeri Mattia Padovan Tommaso Tommasin Davide Bortolazzo Matteo	Prima stesura di Architettura generale (sezione 3) e componenti (sezione 4)
0.0.1	2016-03-24	<i>Analisti</i> Bonato Paolo Biggeri Mattia	Creazione scheletro documento, stesura introduzione, definizione di metodo e formalismo di specifica.

Tabella 1: Diario delle modifiche

1 Introduzione

1.1 Scopo del documento

Il documento illustra la progettazione attuale di Actorbase. Verranno presentati l'architettura, le componenti, le classi e i design pattern utilizzati.

1.2 Scopo del prodotto

Il progetto consiste nella realizzazione di un Database NoSQL key-value basato sul modello ad Attori con l'obiettivo di fornire una tecnologia adatta allo sviluppo di moderne applicazioni che richiedono brevissimi tempi di risposta e che elaborano enormi quantità di dati. Lo sviluppo porterà al rilascio del software sotto licenza MIT.

1.3 Glossario

Al fine di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v2.0.0*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo.

1.4 Riferimenti

- **Slide dell'insegnamento Ingegneria del software mod.A:**
<http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/E02.pdf>
- **Scala:**
<http://www.scala-lang.org/>
- **Java:**
<http://www.java.com/>
- **Akka:**
<http://akka.io/>
- **IntelliJ:**
<http://www.jetbrains.com/idea/>

1.4.1 Normativi

- **Norme di progetto:** *Norme di progetto v2.0.0*
- **Capitolato d'appalto Actorbase (C1):**
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1p.pdf>

2 Tecnologie utilizzate

2.1 Scala

Le possibili scelte dettate dal capitolato sono Java e Scala. Si è scelto di utilizzare Scala perché offre i seguenti vantaggi:

- **Completamente Object-Oriented:** A differenza di Java, Scala è completamente orientato agli oggetti. Non c'è distinzione del tipo: oggetto - tipo primitivo, ogni valore è semplicemente un oggetto.
- **Staticamente tipato:** È un linguaggio tipato staticamente, questo permette di effettuare più facilmente i test. Inoltre Scala è in grado di stabilire il tipo di un oggetto per inferenza.
- **Può eseguire codice Java:** Scala può eseguire codice scritto in Java. È dunque possibile utilizzare classi e librerie scritte in Java all'interno di programmi scritti in Scala.
- **Concorrenza e distribuzione:** Ottimo supporto alla programmazione multi-threaded e distribuita, essenziale per la realizzazione di un prodotto responsive e scalabile.
- **Supporto alla definizione di DSL:** Scala supporta nativamente la definizione di DSL.
- **Supporto di Akka:** Il linguaggio supporta la libreria Akka che è richiesta dal capitolato.
- **Preferenze del Committente:** Inoltre il Committente ha espresso esplicitamente la sua preferenza sull'utilizzo di Scala.

Problematiche di scala:

- **Familiarità con la tecnologia:** Nessun membro ha mai avuto esperienza, prima del inizio del progetto, con questo linguaggio di programmazione.
- **Leggibilità:** Alcuni costrutti risultano di difficile lettura.



Figura 1: Scala - logo

2.2 Akka

L'utilizzo della libreria Akka oltre ad essere reso obbligatorio dal committente, fornisce un'eccellente base su cui sviluppare un sistema basato sul modello ad attori. Akka permette di costruire facilmente applicazioni message-driven che siano estremamente concorrenti, distribuite e resilienti. La natura distribuita e asincrona degli attori messi a disposizione da Akka soddisfa pienamente i bisogni del sistema da implementare. Akka esegue l'override di molti operatori rendendo difficile la comprensione agli utenti poco esperti.



Figura 2: Akka - logo

3 Legenda

Tutti gli schemi dei package seguono la legenda sottostante. I livelli di annidamento sono da intendere per la totale struttura dei package e non solo per il singolo schema.

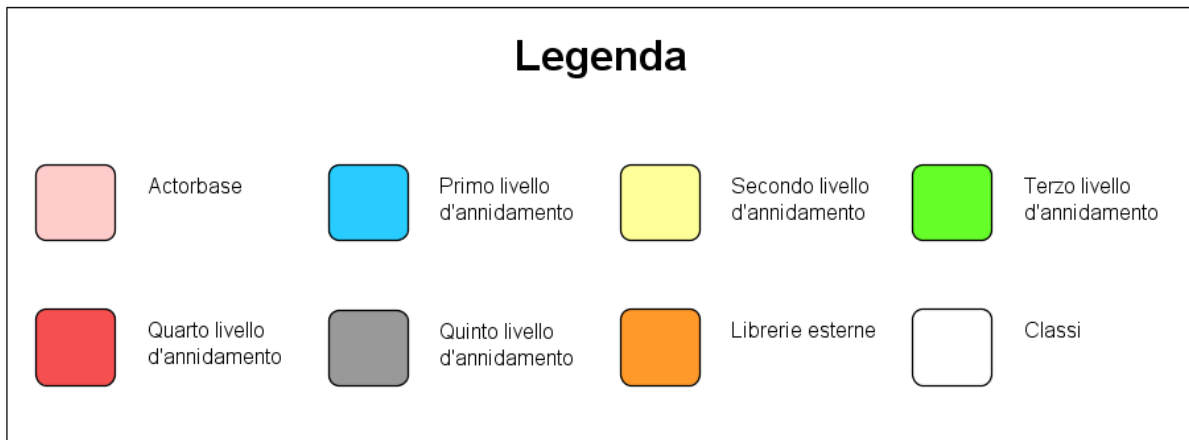


Figura 3: Legenda

4 Descrizione dell'architettura

4.1 Metodo e formalismo di specifica

Nell'esposizione dell'architettura del prodotto si procederà con un approccio di tipo top-down. Inizialmente si descriveranno le tre componenti fondamentali: Client, Server e Driver; poi le componenti più piccole al loro interno, specificando i package e le classi che li compongono.

Per ogni package saranno descritti brevemente il tipo, l'obiettivo e la funzione e saranno specificati eventuali figli, classi ed interazioni con altri package. Ogni classe sarà dotata di una breve descrizione e ne saranno specificate le responsabilità, le classi ereditate, le sottoclassi e le relazioni con altre classi. Successivamente saranno mostrati e descritti i diagrammi delle attività che coinvolgono l'utente. Infine si illustreranno degli esempi di utilizzo dei design pattern nell'architettura del sistema.

4.2 Architettura generale

L'architettura generale del sistema è di tipo client-server.

Il Server ha un'architettura di tipo event-driven basata sul modello ad attori e accetta connessioni in entrata tramite un socket TCP.

Il Client è un semplice programma che espone un'interfaccia a linea di comando per accettare comandi che vengono inoltrati al Server tramite il Driver.

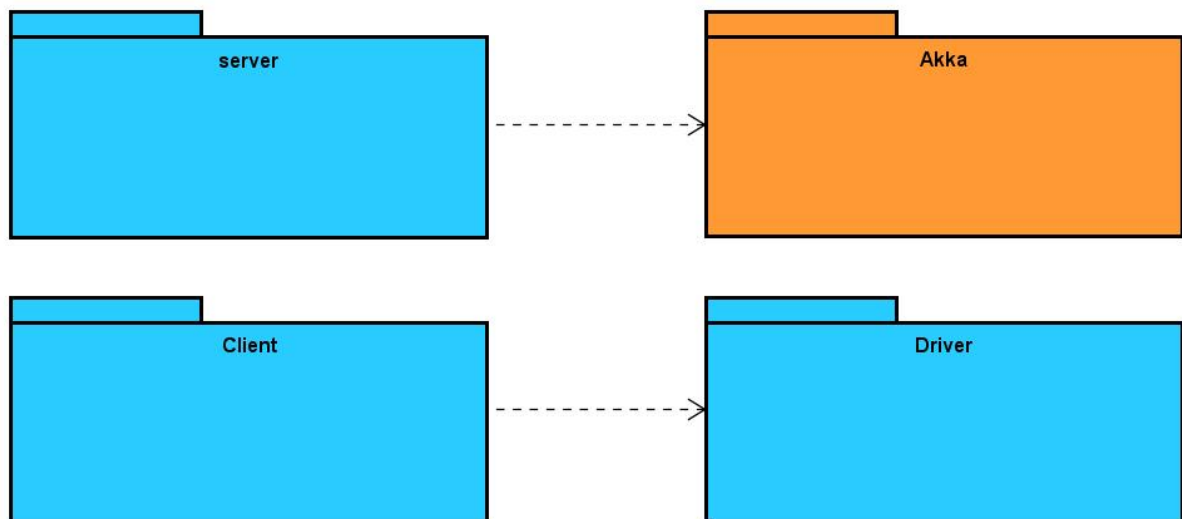


Figura 4: Architettura generale, vista Package

4.2.1 Server

Il Server di *Actorbase* è composto da quattro package principali ed una classe: i package **utils**, **messages**, **actors** ed **enums** e le classi **Server**, **StaticSettings** e **ClusterListener**.

Il package **utils** contiene delle classi di supporto che vengono utilizzate dai vari attori presenti per effettuare operazioni di vario genere.

Il package **messages** contiene tutte le interfacce e le classi dei messaggi che gli attori possono mandarsi tra di loro.

Il package **actors** contiene tutte le classi che definiscono gli attori del sistema.

Il package **enums** contiene le enumerazioni utilizzate per vari scopi.

La classe **Server** è l'entry point del programma e gestisce la configurazione iniziale all'avvio dello stesso.

4.2.2 Client

Il Client di *Actorbase* è composto da due sole classi: **Client** e **Welcome**, entrambe *Singleton*.

4.2.3 Driver

Il Driver è una libreria, invocando i metodi della quale è possibile effettuare richieste TCP sul socket su cui il Server è in ascolto.

Contiene due classi e un'interfaccia: l'interfaccia **Connection**, le classi **Driver** e **ConcreteConnection**. Quest'ultima è una realizzazione dell'interfaccia **Connection**.

5 Componenti e Classi

5.1 Actorbase

5.1.1 Descrizione

È il livello principale del sistema. L'interazione tra le componenti dei package **Server** e **Driver** avviene tramite una connessione di rete TCP di tipo client-server.

5.1.2 Package Figli

- Actorbase.server
- Actorbase.client
- Actorbase.driver

5.2 Actorbase.server

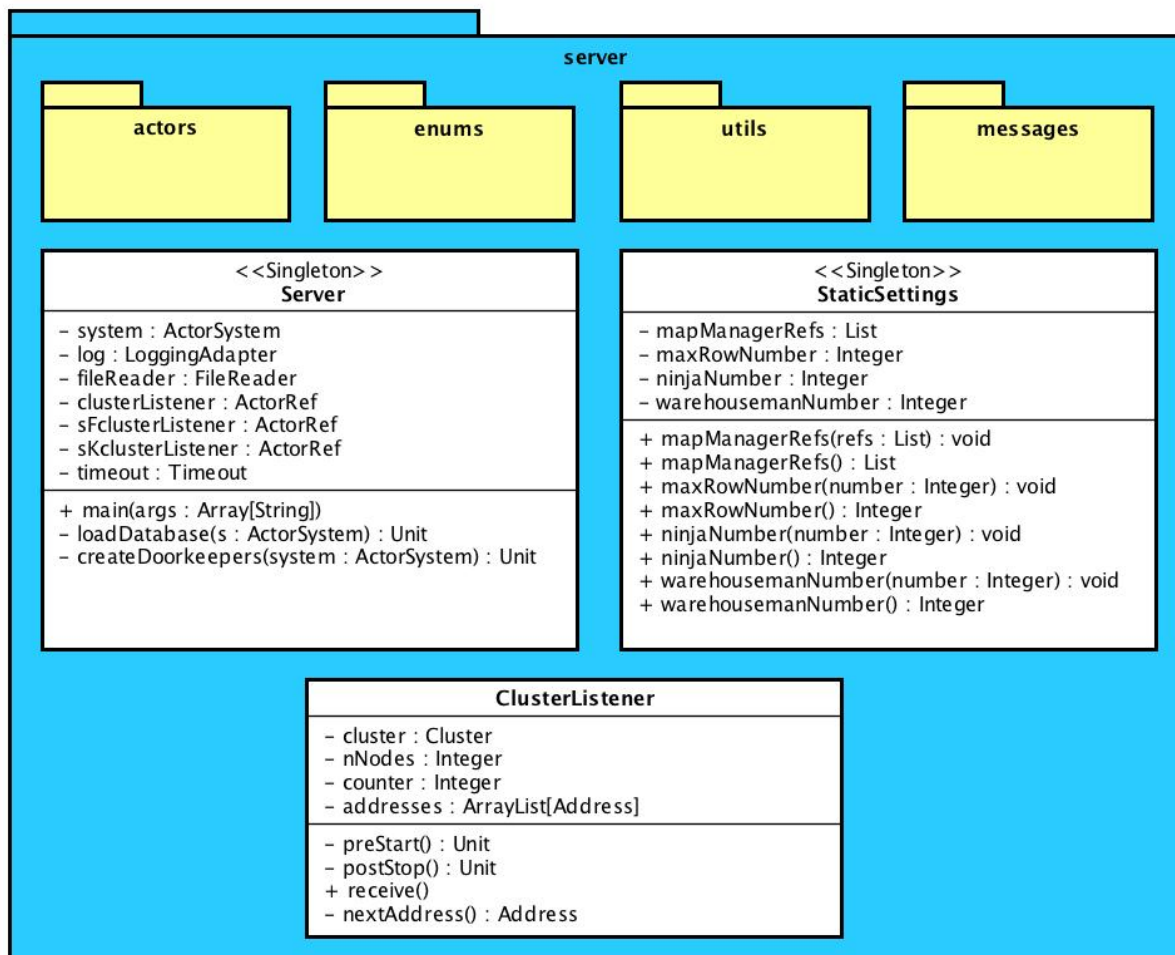


Figura 5: Componente Actorbase.server

5.2.1 Descrizione

Package per la componente lato server del sistema. È composto dai package **utils**, **messages**, **actors** ed **enums** e dalle classi **Server**, **StaticSettings** e **ClusterListener**.

5.2.2 Package Figli

- Actorbase.server.utils
- Actorbase.server.messages
- Actorbase.server.actors
- Actorbase.server.enums

5.2.3 Classi

- Actorbase.server.Server
- Actorbase.server.StaticSettings
- Actorbase.server.ClusterListener

5.3 Actorbase.server.Server

5.3.1 Descrizione

Classe principale della parte **Server** del programma. È di fatto l'entry point dello stesso, gestisce la configurazione iniziale e avvia il sistema. Utilizza il design pattern **Singleton**.

5.3.2 Utilizzo

Classe che fornisce un punto di accesso al programma, la sua esecuzione avvia il server sulla macchina in cui viene lanciata.

5.3.3 Relazione con altre classi

- **Actorbase.server.actors.Doorkeeper:** relazione uscente, creazione.
- **Actorbase.server.actors.MapManager:** relazione uscente, creazione.
- **Actorbase.server.StaticSettings:** relazione di utilizzo.

5.4 Actorbase.server.utils

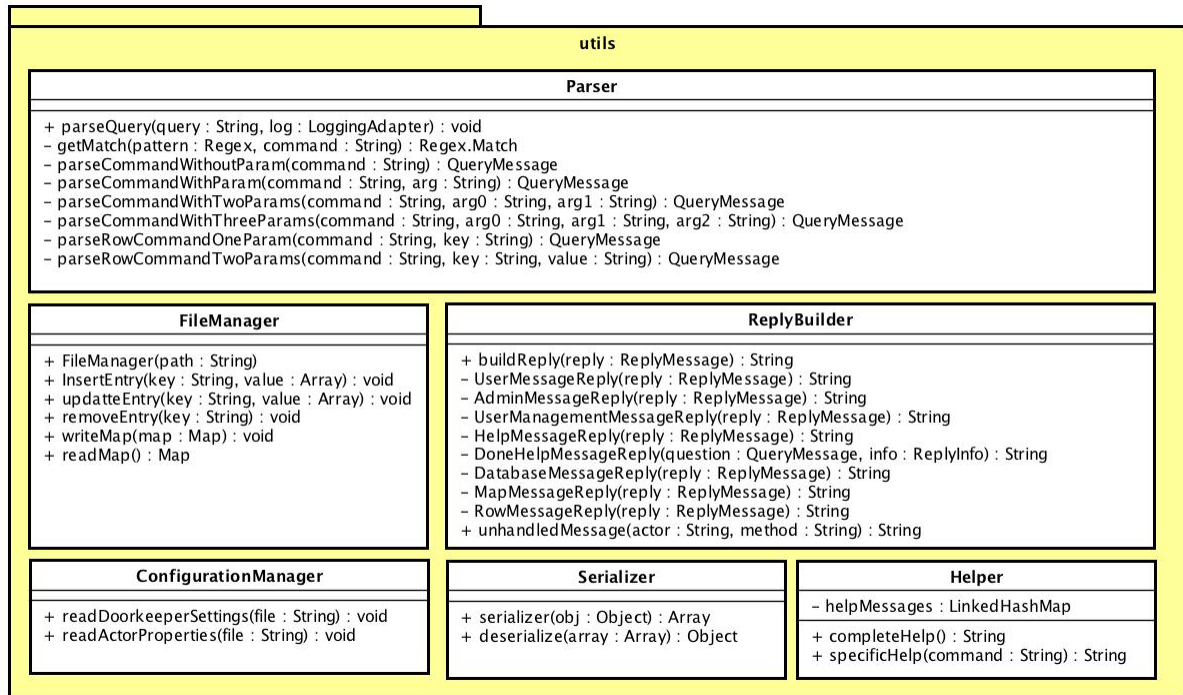


Figura 6: Componente Actorbase.server.utils

5.4.1 Descrizione

Package contenente le classi che effettuano operazioni varie a supporto delle varie componenti del server, e degli attori nello specifico.

5.4.2 Classi

- Actorbase.server.utils.Parser
- Actorbase.server.utils.Helper
- Actorbase.server.utils.ReplyBuilder
- Actorbase.server.utils.Serializer
- Actorbase.server.utils.ConfigurationManager

5.4.3 Trait

- Actorbase.server.utils.FileManager

5.5 Actorbase.server.utils.Parser

5.5.1 Descrizione

Classe utilizzata dal server per tradurre la stringa ricevuta da un client in un messaggio che ne rappresenta la richiesta per la comunicazione tra attori.

5.5.2 Utilizzo

Viene utilizzata dal Server quando esso riceve una richiesta da un Client.

5.5.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.
- **Actorbase.server.messages.QueryMessages.QueryMessage:** relazione uscente, creazione.

5.6 Actorbase.server.utils.ReplyBuilder

5.6.1 Descrizione

Classe che costruisce stringhe di risposta da inviare al client a partire dai messaggi del sistema.

5.6.2 Utilizzo

Viene utilizzata per costruire risposte in formato stringa.

5.6.3 Relazione con altre classi

- **Actorbase.server.actors.ReplyActor:** relazione entrante, creazione.

5.7 Actorbase.server.utils.Serializer

5.7.1 Descrizione

Classe che si occupa di gestire la serializzazione di oggetti in Byte e viceversa.

5.7.2 Utilizzo

Viene utilizzata da attori **Main** e **Usermanager** per serializzare oggetti in array di byte memorizzabili da attori **Storemanager**.

5.7.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, creazione.
- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.

5.8 Actorbase.server.utils.FileManager

5.8.1 Descrizione

Classe utilizzata dagli attori per gestire i file su disco.

5.8.2 Utilizzo

Viene utilizzata dalla maggior parte degli attori quando necessitano di leggere o scrivere su file.

5.8.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.9 Actorbase.server.utils.ConfigurationManager

5.9.1 Descrizione

Classe che fornisce i metodi per leggere i file di configurazione di *Actorbase*.

5.9.2 Utilizzo

Viene utilizzata dalla classe **Server** per caricare le impostazioni all'avvio.

5.9.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione di utilizzo.

5.10 Actorbase.server.utils.Helper

5.10.1 Descrizione

Classe che mette a disposizione dei metodi per visualizzare la lista di comandi disponibili e la loro spiegazione.

5.10.2 Utilizzo

Viene utilizzata dall'attore Main per recuperare la lista di comandi con la loro spiegazione, è anche possibile recuperare la spiegazione di un singolo comando.

5.10.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, creazione.

5.11 Actorbase.server.actors

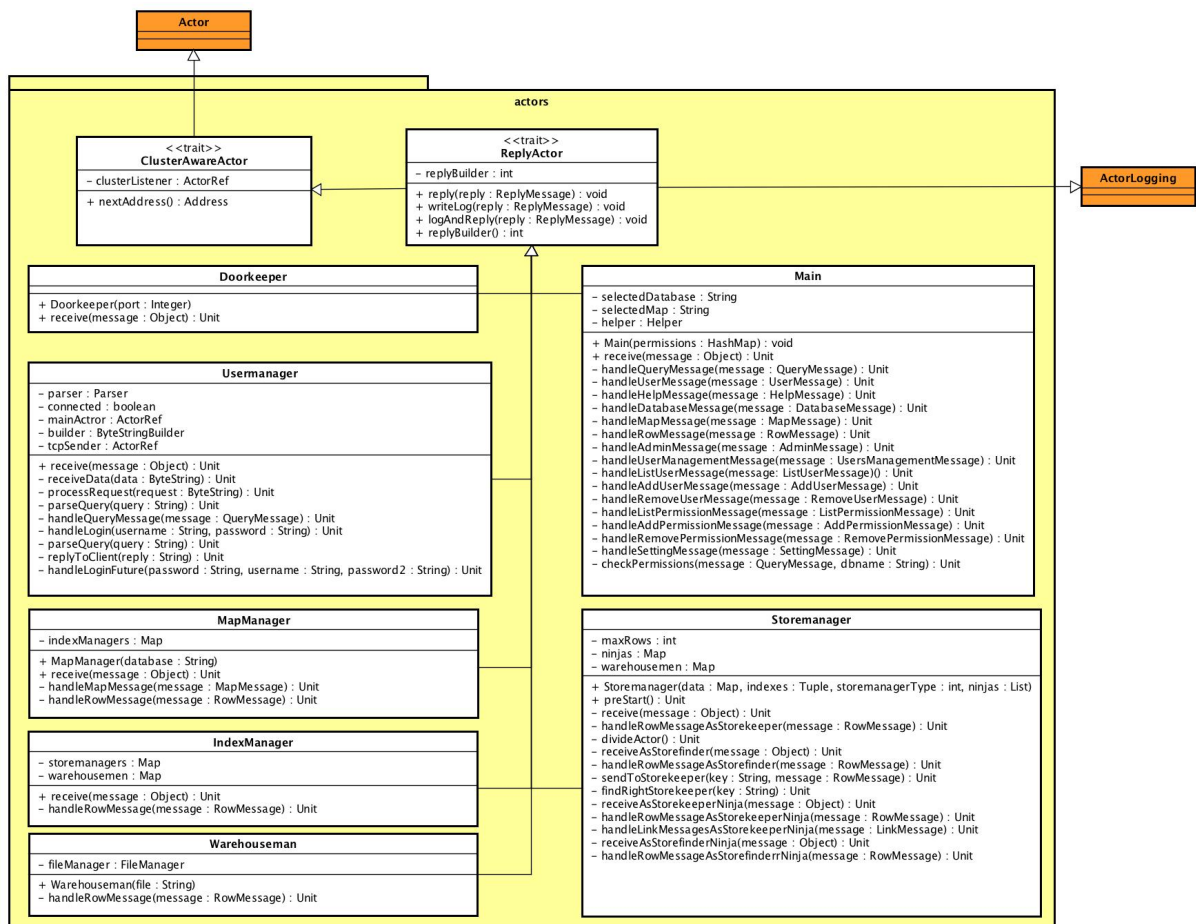


Figura 7: Componente Actorbase.server.actors

5.11.1 Descrizione

Package contenente tutti gli attori che compongono il database. Tutte le classi e le interfacce contenute in questo package rappresentano attori come definiti dalla libreria **Akka**.

5.11.2 Classi

- `Actorbase.server.actors.Doorkeeper`
- `Actorbase.server.actors.Usermanager`
- `Actorbase.server.actors.Main`
- `Actorbase.server.actors.MapManager`
- `Actorbase.server.actors.IndexManager`
- `Actorbase.server.actors.Storemanager`
- `Actorbase.server.actors.Warehouseman`

5.11.3 Trait

- `Actorbase.server.actors.ClusterAwareActor`
- `Actorbase.server.actors.ReplyActor`

5.12 `Actorbase.server.actors.Doorkeeper`

5.12.1 Descrizione

Questo attore apre il socket su cui i client potranno connettersi, crea un attore **Usermanager** per ogni nuova connessione ed inoltra ogni messaggio ricevuto dal client e all'**Usermanager** ad esso associato.

5.12.2 Utilizzo

Viene utilizzato per aprire un socket e resta in attesa di richieste di connessione da parte di client.

5.12.3 Relazione con altre classi

- **`Actorbase.server.Server`**: relazione entrante, creazione.
- **`Actorbase.server.actors.Usermanager`**: relazione uscente, creazione.

5.13 `Actorbase.server.actors.Usermanager`

5.13.1 Descrizione

Questo attore si preoccupa di trasformare le stringhe ricevute dal client in messaggi interpretabili da altri attori, e viceversa.

5.13.2 Utilizzo

Viene utilizzato per gestire una connessione da parte di un client, crea un attore **Main** e invia le richieste ricevute all'attore creato.

5.13.3 Relazione con altre classi

- **`Actorbase.server.actors.Doorkeeper`**: relazione entrante, creazione.
- **`Actorbase.server.actors.Main`**: relazione uscente, creazione.
- **`Actorbase.server.utils.Parser`**: relazione uscente, creazione.
- **`Actorbase.server.messages.query.QueryMessage`**: relazione di utilizzo.
- **`Actorbase.server.utils.ReplyBuilder`**: relazione di utilizzo.

5.14 Actorbase.server.actors.Main

5.14.1 Descrizione

Questo attore è responsabile del controllo dei permessi delle richieste ricevute. Gestisce tutti i messaggi, elaborandoli o inoltrandoli se non di sua competenza. Gestisce l'insieme di database rappresentato da attori **MapManager**.

5.14.2 Utilizzo

Viene utilizzato per controllare i permessi delle richieste ricevute, gestisce i messaggi a livello di database e di help e inoltra tutti gli altri messaggi che non sono di sua competenza. Inoltre si preoccupa di mantenere in memoria le selezioni di database e mappa effettuate dall'utente.

5.14.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.
- **Actorbase.server.actors.MapManager:** relazione uscente, creazione.
- **Actorbase.server.utils.Helper:** relazione uscente, creazione.
- **Actorbase.server.messages.query.QueryMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.ReplyMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.AskMapMessage:** relazione uscente, creazione e utilizzo.

5.15 Actorbase.server.actors.MapManager

5.15.1 Descrizione

Questo attore rappresenta un database di *Actorbase*. Si occupa di indicizzare gli attori che rappresentano le mappe che compongono tale database.

5.15.2 Utilizzo

Viene utilizzato per rappresentare un database, l'attore **Main** lo utilizza per instradare richieste verso uno specifico database. All'avvio il Server crea un **MapManager** per ogni database presente su disco.

5.15.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione entrante di creazione.
- **Actorbase.server.actors.IndexManager:** relazione uscente di creazione.
- **Actorbase.server.messages.query.QueryMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.ReplyMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.ReadMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.users.RowMessages.RowMessage:** relazione di utilizzo.

5.16 Actorbase.server.actors.IndexManager

5.16.1 Descrizione

Questo attore rappresenta una mappa di *Actorbase*. Si occupa di indicizzare gli attori che rappresentano le porzioni di dati di tale mappa, sia su RAM che su disco.

5.16.2 Utilizzo

Viene utilizzato per rappresentare una mappa, attori di tipo **MapManager** creano, interrogano e distruggono questo tipo di attori a seconda delle richieste utente.

5.16.3 Relazione con altre classi

- **Actorbase.server.actors.MapManager**: relazione entrante, creazione.
- **Actorbase.server.actors.Storemanager**: relazione uscente, creazione.
- **Actorbase.server.actors.Warehouseman**: relazione uscente, creazione.
- **Actorbase.server.messages.query.QueryMessage**: relazione di utilizzo.
- **Actorbase.server.messages.query.ReplyMessage**: relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage**: relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.ReadMessage**: relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.ReadWriteMessage**: relazione di utilizzo.
- **Actorbase.server.messages.query.users.RowMessages.RowMessage**: relazione di utilizzo.

5.17 Actorbase.server.actors.Storemanager

5.17.1 Descrizione

Questo attore rappresenta una porzione di mappa in memoria principale, ha più comportamenti: sia come indicizzazione dei dati che come mantenimento di essi.

5.17.2 Utilizzo

Viene utilizzato per gestire le operazioni sugli item, attori di tipo **Storemanager** sono utilizzati dagli **IndexManager** per mantenere indicizzati i dati in memoria principale.

5.17.3 Relazione con altre classi

- **Actorbase.server.actors.IndexManager**: relazione entrante, creazione.
- **Actorbase.server.messages.query.user.UserMessage**: relazione di utilizzo.

5.18 Actorbase.server.actors.Warehouseman

5.18.1 Descrizione

Questo attore permette l'interazione del database con il filesystem.

5.18.2 Utilizzo

Viene utilizzato per mantenere un backup del database su file presenti su disco.

5.18.3 Relazione con altre classi

- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.messages.internal.StorageMessages:** relazione uscente, creazione e utilizzo.

5.19 Actorbase.server.actors.ClusterAwareActor

5.19.1 Descrizione

Interfaccia che definisce un attore che interagisce con altri nodi del cluster.

5.19.2 Utilizzo

Viene estesa da attori che interagiscono con altri nodi del cluster, creando attori su di essi o interagendo con attori presenti su di essi.

5.19.3 Interfacce figlie

- **Actorbase.server.actors.ReplyActor**

5.20 Actorbase.server.actors.ReplyActor

5.20.1 Descrizione

Interfaccia che definisce un attore che risponde con informazioni definite in *Actorbase*.

5.20.2 Utilizzo

Viene estesa da attori che rispondono con informazioni definite dal sistema.

5.20.3 Classi figlie

- **Actorbase.server.actors.Main**
- **Actorbase.server.actors.MapManager**
- **Actorbase.server.actors.IndexManager**
- **Actorbase.server.actors.Storemanager**
- **Actorbase.server.actors.Warehouseman**

5.21 Actorbase.server.enums

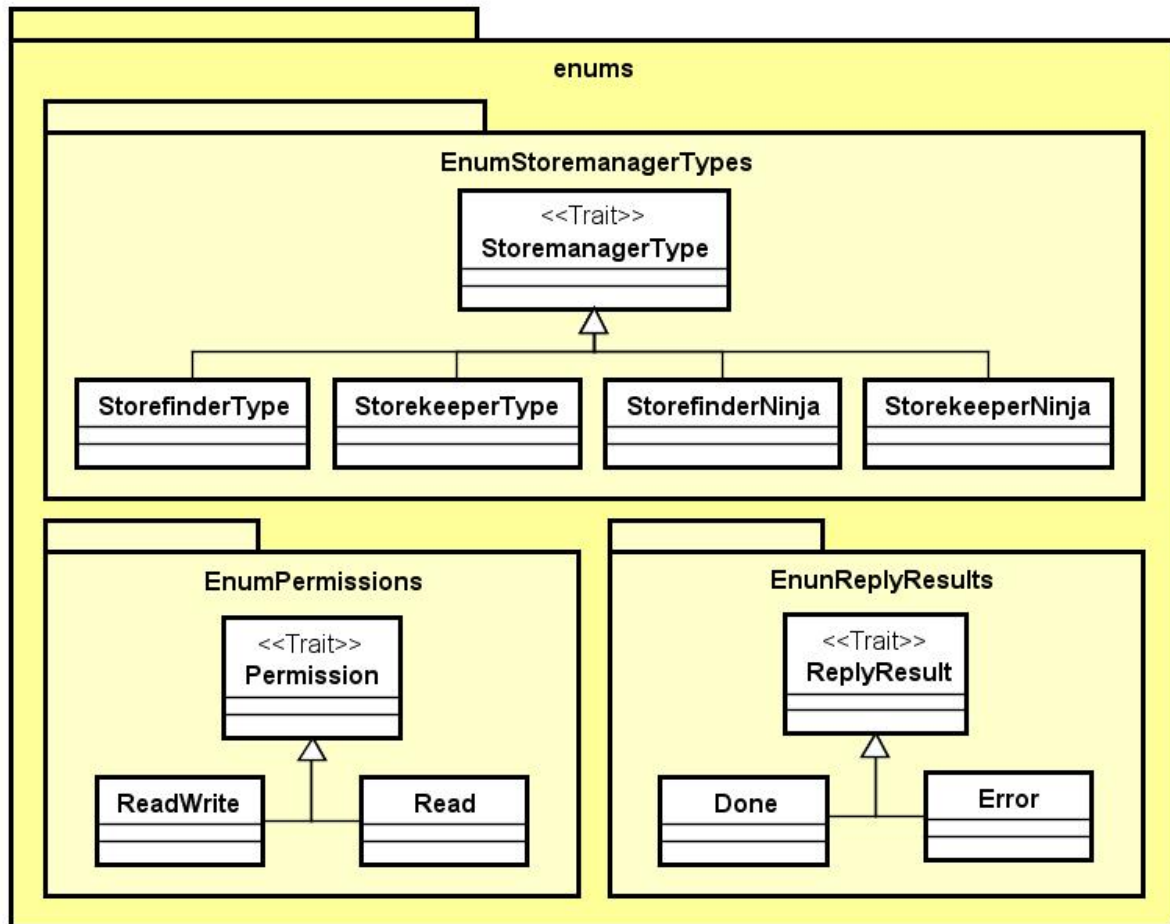


Figura 8: Componente Actorbase.server.enums

5.21.1 Descrizione

Package contenente le enumerazioni utilizzate dagli attori per distinguere i tipi di permessi e di risposte e altre informazioni e caratteristiche del sistema.

5.21.2 Package Figli

- Actorbase.server.enums.EnumStoremanagerTypes
- Actorbase.server.enums.EnumPermissions
- Actorbase.server.enums.EnumReplyResults

5.22 Actorbase.server.enums.EnumPermissions

5.22.1 Descrizione

Package contenente le enumerazioni utilizzate dagli attori per distinguere i tipi di permessi.

5.22.2 Trait

- Actorbase.server.enums.EnumPermissions.Permission

5.22.3 Classi

- `Actorbase.server.enums.EnumPermissions.Read`
- `Actorbase.server.enums.EnumPermissions.ReadWrite`

5.23 `Actorbase.server.enums.EnumPermissions.Permission`

5.23.1 Descrizione

Trait per rappresentare i permessi che la richiesta di un utente può avere.

5.23.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

5.23.3 Relazione con altre classi

- **`Actorbase.server.actors.Main`**: relazione di utilizzo.
- **`Actorbase.server.enums.EnumPermission`**: relazione entrante, aggregazione.

5.23.4 Classi figlie

- `Actorbase.server.enums.EnumPermissions.Read`.
- `Actorbase.server.enums.EnumPermissions.ReadWrite`.

5.24 `Actorbase.server.enums.EnumPermissions.Read`

5.24.1 Descrizione

Questa classe rappresenta il permesso di lettura.

5.24.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

5.24.3 Relazione con altre classi

- **`Actorbase.server.actors.Main`**: relazione di utilizzo.

5.24.4 Trait implementati

- **`Actorbase.server.enums.EnumPermission.Permission`**

5.25 `Actorbase.server.enums.EnumPermissions.ReadWrite`

5.25.1 Descrizione

Questa classe rappresenta il permesso di scrittura.

5.25.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

5.25.3 Relazione con altre classi

- **`Actorbase.server.actors.Main`**: relazione di utilizzo.

5.25.4 Trait implementati

- **`Actorbase.server.enums.EnumPermission.Permission`**

5.26 Actorbase.server.enums.EnumReplyResults

5.26.1 Descrizione

Package contenente le enumerazioni utilizzate dagli attori per distinguere i tipi di risultato di un'operazione.

5.26.2 Trait

- Actorbase.server.enums.EnumReplyResults.ReplyResult

5.26.3 Classi

- Actorbase.server.enums.EnumReplyResults.Done
- Actorbase.server.enums.EnumReplyResults.Error

5.27 Actorbase.server.enums.EnumReplyResults.ReplyResult

5.27.1 Descrizione

Trait per rappresentare le risposte ed il loro stato (avvenuta con successo, errore).

5.27.2 Utilizzo

Viene utilizzata da tutti gli attori per indicare se la richiesta effettuata da un utente è andata a buon fine o no.

5.27.3 Relazione con altre classi

- Actorbase.server.actors.Main: relazione di utilizzo.
- Actorbase.server.actors.Doorkeeper: relazione di utilizzo.
- Actorbase.server.actors.MapManager: relazione di utilizzo.
- Actorbase.server.actors.IndexManager: relazione di utilizzo.
- Actorbase.server.actors.Warehouseman: relazione di utilizzo.
- Actorbase.server.actors.Usermanager: relazione di utilizzo.
- Actorbase.server.actors.Storemanager: relazione di utilizzo.

5.27.4 Classi figlie

- Actorbase.server.enums.EnumReplyResults.Error.
- Actorbase.server.enums.EnumReplyResults.Done.

5.28 Actorbase.server.enums.EnumReplyResults.Done

5.28.1 Descrizione

Questa classe rappresenta l'avvenuto successo di una richiesta effettuata da un utente.

5.28.2 Utilizzo

Viene utilizzata da tutti gli attori come risposta in caso di richiesta effettuata con successo.

5.28.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.

5.28.4 Trait implementati

- **Actorbase.server.enums.EnumReplyResults.ReplyResult**

5.29 Actorbase.server.enums.EnumReplyResults.Error

5.29.1 Descrizione

Questa classe rappresenta il verificarsi di un errore durante l'esecuzione di una richiesta effettuata da un utente.

5.29.2 Utilizzo

Viene utilizzata da tutti gli attori come risposta in caso di errore durante l'esecuzione di una richiesta effettuata da un utente.

5.29.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.

5.29.4 Trait implementati

- **Actorbase.server.enums.EnumReplyResults.ReplyResult**

5.30 Actorbase.server.enums.EnumStoremanagerTypes

5.30.1 Descrizione

Package contenente le enumerazioni utilizzate per definire il comportamento degli attori Storemanager.

5.30.2 Trait

- **Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType**

5.30.3 Classi

- `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderNinjaType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperNinjaType`

5.31 `Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType`

5.31.1 Descrizione

Trait per rappresentare il comportamento di uno **Storemanager**.

5.31.2 Utilizzo

Viene estesa da tutte le classi che rappresentano il comportamento di uno **Storemanager**.

5.31.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione di utilizzo.

5.31.4 Classi figlie

- `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderNinjaType`
- `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperNinjaType`

5.32 `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderType`

5.32.1 Descrizione

Questa classe il comportamento Storefinder di un attore di tipo **Storemanager**.

5.32.2 Utilizzo

Viene utilizzata per definire il comportamento Storefinder.

5.32.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione di utilizzo.

5.32.4 Trait implementati

- `Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType`

5.33 `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperType`

5.33.1 Descrizione

Questa classe il comportamento Storekeeper di un attore di tipo **Storemanager**.

5.33.2 Utilizzo

Viene utilizzata per definire il comportamento Storekeeper.

5.33.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione di utilizzo.

5.33.4 Trait implementati

- `Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType`

5.34 `Actorbase.server.enums.EnumStoremanagerTypes.StorefinderNinjaType`

5.34.1 Descrizione

Questa classe il comportamento `StorefinderNinja` di un attore di tipo `Storemanager`.

5.34.2 Utilizzo

Viene utilizzata per definire il comportamento `StorefinderNinja`.

5.34.3 Relazione con altre classi

- `Actorbase.server.actors.Storemanager`: relazione di utilizzo.

5.34.4 Trait implementati

- `Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType`

5.35 `Actorbase.server.enums.EnumStoremanagerTypes.StorekeeperNinjaType`

5.35.1 Descrizione

Questa classe il comportamento `StorekeeperNinja` di un attore di tipo `Storemanager`.

5.35.2 Utilizzo

Viene utilizzata per definire il comportamento `StorekeeperNinja`.

5.35.3 Relazione con altre classi

- `Actorbase.server.actors.Storemanager`: relazione di utilizzo.

5.35.4 Trait implementati

- `Actorbase.server.enums.EnumStoremanagerTypes.StoremanagerType`

5.36 Actorbase.server.messages

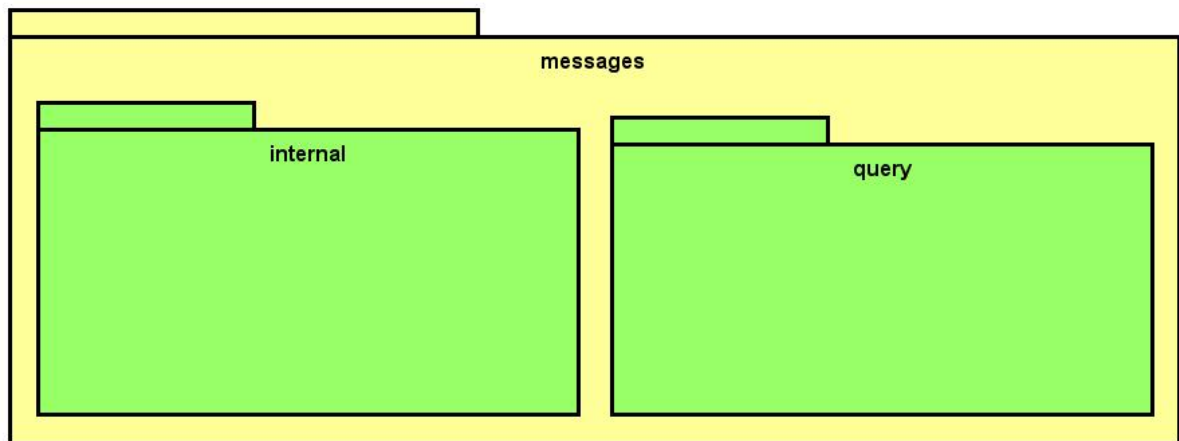


Figura 9: Componente Actorbase.server.messages

5.36.1 Descrizione

Package contenente tutti i messaggi che gli attori si scambiano tra di loro. È composto dai package `internal` e `query`.

5.36.2 Package Figli

- Actorbase.server.messages.internal.
- Actorbase.server.messages.query.

5.37 Actorbase.server.messages.internal

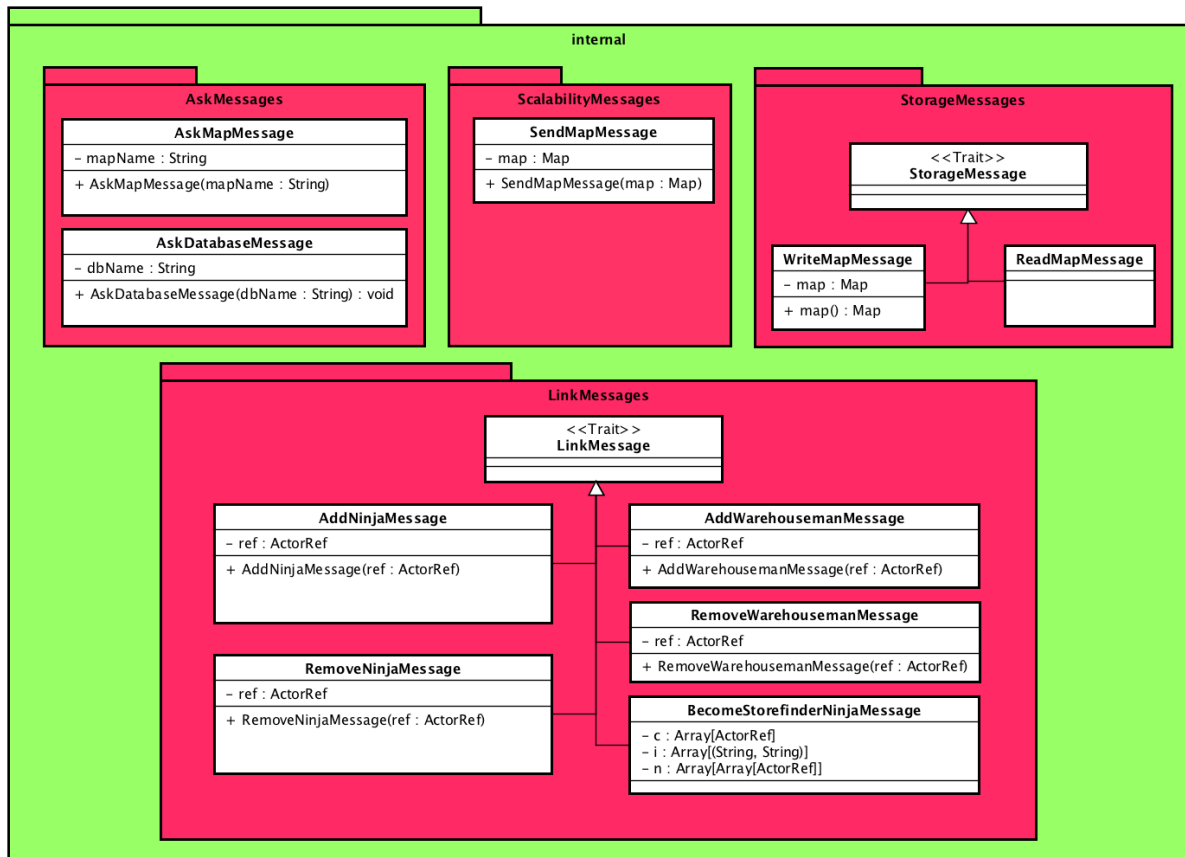


Figura 10: Componente Actorbase.server.messages.internal

5.37.1 Descrizione

Package contenente i messaggi che gli attori si scambiano non riguardanti richieste effettuate dagli utenti ma riguardanti la struttura interna del sistema.

5.37.2 Package Figli

- Actorbase.server.messages.internal.LinkMessages
- Actorbase.server.messages.internal.AskMessages
- Actorbase.server.messages.internal.ScalabilityMessages
- Actorbase.server.messages.internal.StorageMessages

5.38 Actorbase.server.messages.internal.AskMessages

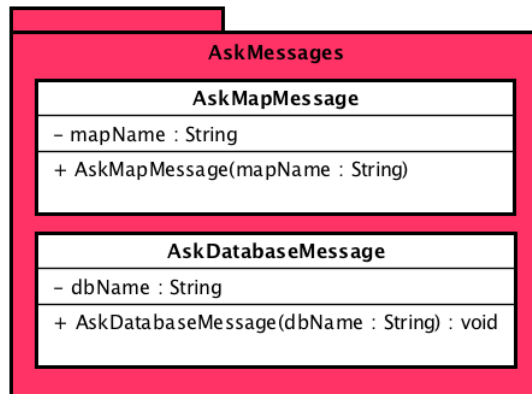


Figura 11: Componente Actorbase.server.messages.internal.AskMessages

5.38.1 Descrizione

Package contenente i messaggi che rappresentano richieste di esistenza di mappe o database all'interno del sistema.

5.38.2 Classi

- Actorbase.server.messages.internal.AskMessages.AskMapMessage
- Actorbase.server.messages.internal.AskMessages.AskDatabaseMessage

5.39 Actorbase.server.messages.internal.AskMessages.AskMapMessage

5.39.1 Descrizione

Questa classe rappresenta un messaggio per verificare la presenza di una mappa in un **MapManager**.

5.39.2 Utilizzo

Viene inviato da un **Main** ad un **MapManager**.

5.39.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.40 Actorbase.server.messages.internal.AskMessages.AskDatabaseMessage

5.40.1 Descrizione

Questa classe rappresenta un messaggio per verificare la presenza di un database.

5.40.2 Utilizzo

Viene inviato da un **Usermanager** ad un **Main**.

5.40.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.41 Actorbase.server.messages.internal.ScalabilityMessages

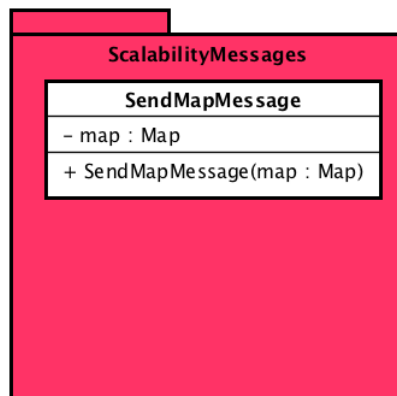


Figura 12: Componente Actorbase.server.messages.internal.ScalabilityMessages

5.41.1 Descrizione

Package contenente i messaggi che rappresentano richieste relative alla scalabilità del sistema.

5.41.2 Classi

- Actorbase.server.messages.internal.ScalabilityMessages.SendMapMessage

5.42 Actorbase.server.messages.internal.ScalabilityMessages.SendMapMessage

5.42.1 Descrizione

Questa classe rappresenta un messaggio contenente una mappa.

5.42.2 Utilizzo

Viene inviato da un per mandare una mappa (più item) da un attore ad un altro.

5.43 Actorbase.server.messages.internal.LinkMessages

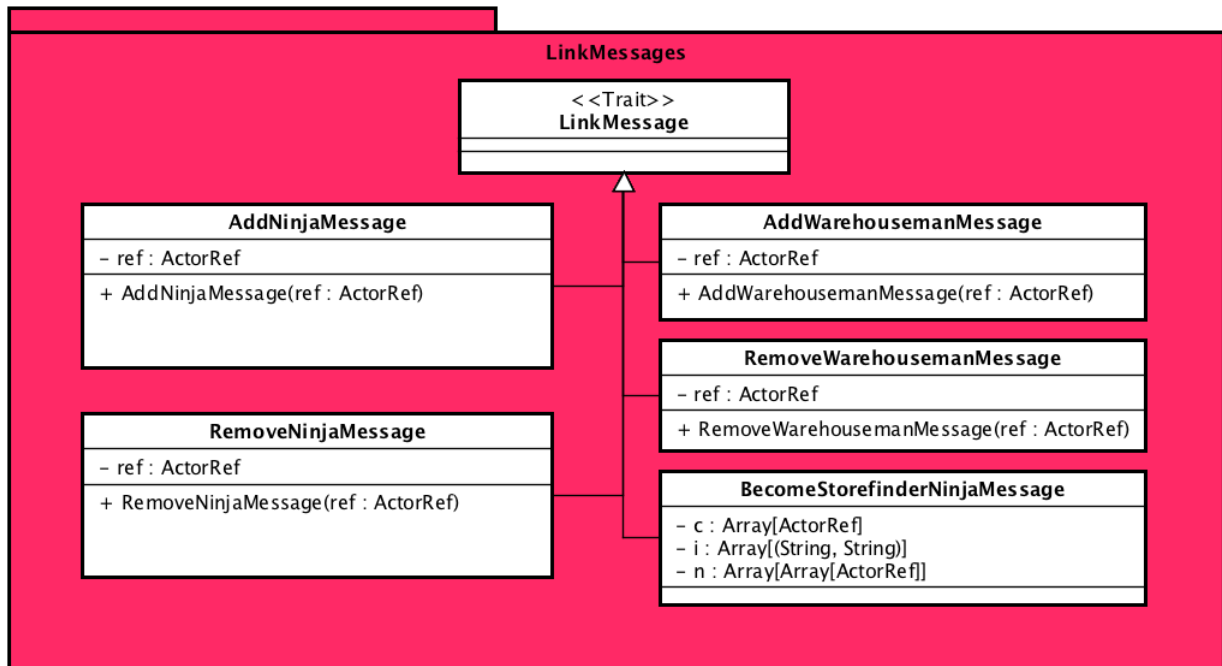


Figura 13: Componente Actorbase.server.messages.internal.LinkMessages

5.43.1 Descrizione

Package contenente i messaggi che gli attori si scambiano non riguardanti richieste effettuate dagli utenti.

5.43.2 Classi

- Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage
- Actorbase.server.messages.internal.LinkMessages.BecomeStorefinderNinjaMessage

5.43.3 Trait

- Actorbase.server.messages.internal.LinkMessages.LinkMessage

5.44 Actorbase.server.messages.internal.LinkMessages.LinkMessage

5.44.1 Descrizione

Trait per la richiesta di aggiunta o rimozione di un attore da parte di un altro attore.

5.44.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.44.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.44.4 Classi figlie

- `Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage`
- `Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage`
- `Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage`
- `Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage`

5.45 `Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage`

5.45.1 Descrizione

Questa classe rappresenta la richiesta di aggiunta di un **Ninja**, ovvero uno **Storekeeper** di backup, da parte di uno **Storemanager**.

5.45.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.45.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.45.4 Trait implementati

- `Actorbase.server.messages.internal.LinkMessages.LinkMessage`

5.46 `Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage`

5.46.1 Descrizione

Questa classe rappresenta la richiesta di aggiunta di un **Warehouseman** da parte di uno **Storemanager**.

5.46.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.46.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.46.4 Trait implementati

- `Actorbase.server.messages.internal.LinkMessages.LinkMessage`

5.47 `Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage`

5.47.1 Descrizione

Questa classe rappresenta la richiesta di rimozione di un **Ninja**, ovvero uno **Storekeeper** di backup, da parte di uno **Storemanager**.

5.47.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.47.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.47.4 Trait implementati

- `Actorbase.server.messages.internal.LinkMessages.LinkMessage`

5.48 `Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage`

5.48.1 Descrizione

Questa classe rappresenta la richiesta di rimozione di un **Warehouseman** da parte di uno **Storemanager**.

5.48.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.48.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.48.4 Trait implementati

- `Actorbase.server.messages.internal.LinkMessages.LinkMessage`

5.49 `Actorbase.server.messages.internal.LinkMessages.BecomeStorefinderNinjaMessage`

5.49.1 Descrizione

Questa classe rappresenta la richiesta di cambio di comportamento in **StorefinderNinja** da parte di uno **Storemanager**.

5.49.2 Utilizzo

Viene inviato da attori di tipo **Storemanager**.

5.49.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager**: relazione entrante, creazione e utilizzo.

5.49.4 Trait implementati

- `Actorbase.server.messages.internal.LinkMessages.LinkMessage`

5.50 `Actorbase.server.messages.internal.StorageMessages`

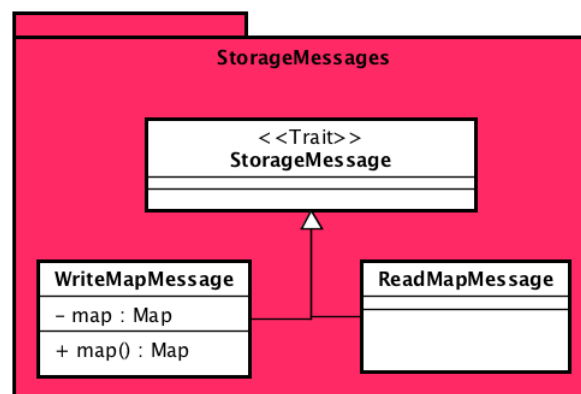


Figura 14: Componente `Actorbase.server.messages.internal.StorageMessages`

5.50.1 Descrizione

Package contenente i messaggi che rappresentano richieste relative alla gestione dei dati su disco.

5.50.2 Classi

- `Actorbase.server.messages.internal.StorageMessages.WriteMapMessage`
- `Actorbase.server.messages.internal.StorageMessages.ReadMapMessage`

5.50.3 Trait

- `Actorbase.server.messages.internal.StorageMessages.StorageMessage`

5.51 `Actorbase.server.messages.internal.StorageMessages.StorageMessage`

5.51.1 Descrizione

Trait che rappresenta i messaggi che gestiscono operazioni di gestione dei dati su disco.

5.51.2 Utilizzo

Viene utilizzato da attori **Warehouseman** per gestire richieste di dati.

5.51.3 Relazione con altre classi

- **`Actorbase.server.actors.Warehouseman`**: relazione entrante, creazione e utilizzo.

5.51.4 Classi figlie

- `Actorbase.server.messages.internal.StorageMessages.WriteMapMessage`
- `Actorbase.server.messages.internal.StorageMessages.ReadMapMessage`

5.52 `Actorbase.server.messages.internal.StorageMessages.WriteMapMessage`

5.52.1 Descrizione

Classe che rappresenta un messaggio per scrivere una mappa su disco.

5.52.2 Utilizzo

Viene utilizzato da attori **Warehouseman** per gestire le richieste di scrittura su disco.

5.52.3 Relazione con altre classi

- **`Actorbase.server.actors.Warehouseman`**: relazione entrante, creazione e utilizzo.

5.52.4 Trait implementati

- `Actorbase.server.messages.internal.StorageMessages.StorageMessage`

5.53 `Actorbase.server.messages.internal.StorageMessages.ReadMapMessage`

5.53.1 Descrizione

Classe che rappresenta un messaggio per leggere una mappa da disco.

5.53.2 Utilizzo

Viene utilizzato da attori **Warehouseman** per gestire le richieste di lettura da disco.

5.53.3 Relazione con altre classi

- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione e utilizzo.

5.53.4 Trait implementati

- Actorbase.server.messages.internal.StorageMessages.StorageMessage

5.54 Actorbase.server.messages.query

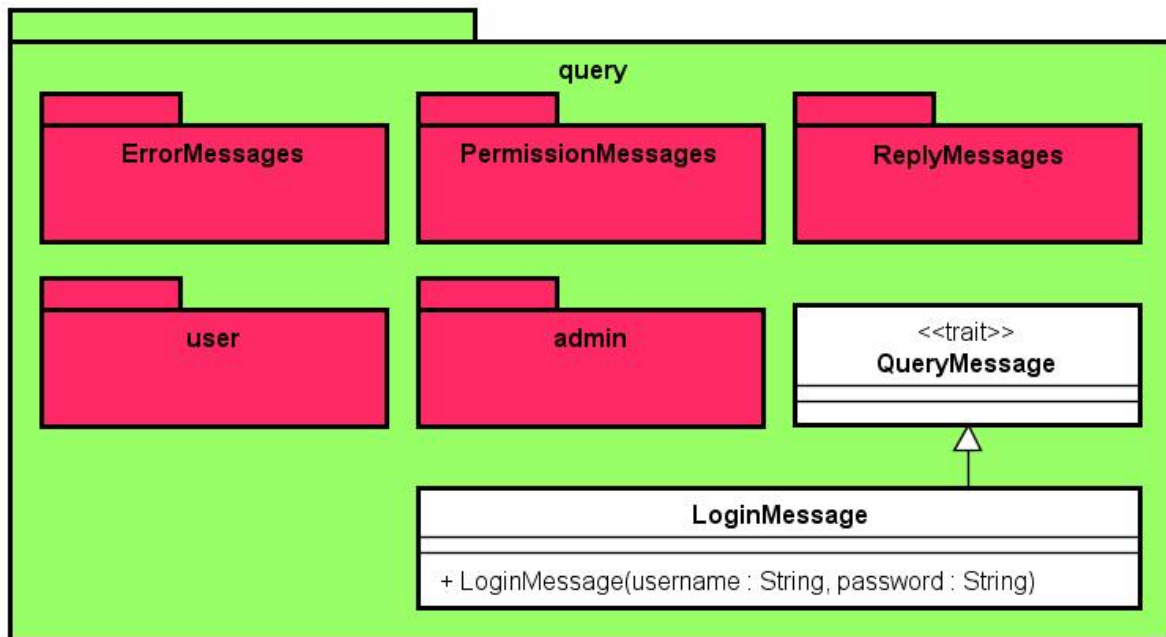


Figura 15: Componente Actorbase.server.messages.query

5.54.1 Descrizione

Package contenente i messaggi che rappresentano le possibili richieste che gli utenti possono effettuare. Questi messaggi vengono scambiati tra gli attori presenti nel database.

5.54.2 Package Figli

- Actorbase.server.messages.query.ErrorMessage
- Actorbase.server.messages.query.PermissionMessages
- Actorbase.server.messages.query.ReplyMessages
- Actorbase.server.messages.query.user
- Actorbase.server.messages.query.admin

5.54.3 Classi

- Actorbase.server.messages.query.LoginMessage

5.54.4 Trait

- Actorbase.server.messages.query.QueryMessage

5.55 Actorbase.server.messages.query.QueryMessage

5.55.1 Descrizione

Interfaccia che rappresenta un messaggio contenente una richiesta effettuata da un utente.

5.55.2 Utilizzo

Viene estesa da tutti i messaggi che rappresentano una query utente ad *Actorbase*.

5.55.3 Relazioni con altre classi

- **Actorbase.server.messages.query.user.UserMessage:** relazione entrante, aggregazione.
- **Actorbase.server.messages.query.user.AdminMessage:** relazione entrante, aggregazione.
- **Actorbase.server.actors.Main:** relazione entrante, utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione entrante, utilizzo.

5.55.4 Classi figlie

- Actorbase.server.messages.query.user.AdminMessage
- Actorbase.server.messages.query.user.UserMessage

5.56 Actorbase.server.messages.query.LoginMessage

5.56.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di login effettuata da parte di un utente.

5.56.2 Utilizzo

Viene utilizzato per formulare una richiesta di login ad attori **Usermanager**.

5.56.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.

5.56.4 Trait implementati

- Actorbase.server.messages.query.QueryMessage

5.57 Actorbase.server.messages.query.ErrorMessage

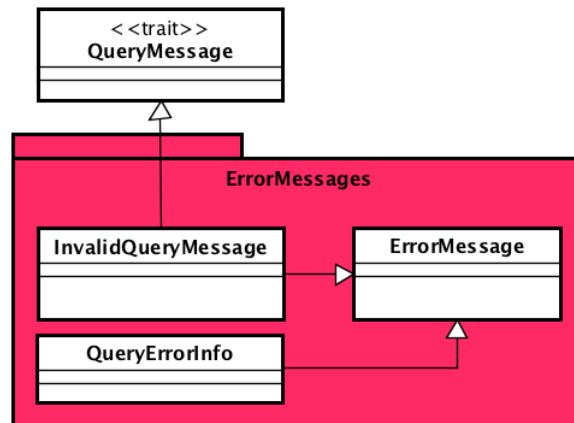


Figura 16: Componente Actorbase.server.messages.query.ErrorMessage

5.57.1 Descrizione

Package contenente i messaggi di errore che vengono utilizzati qualora si verifichi un errore durante la realizzazione di una richiesta effettuata da un utente.

5.57.2 Classi

- Actorbase.server.messages.query.ErrorMessage.InvalidQueryMessage

5.57.3 Trait

- Actorbase.server.messages.query.ErrorMessage.ErrorMessage

5.58 Actorbase.server.messages.query.ErrorMessage.ErrorMessage

5.58.1 Descrizione

Trait per segnalare che la richiesta effettuata non è riconosciuta dal server.

5.58.2 Utilizzo

Viene esteso da tutti i messaggi utilizzati per segnalare una situazione di errore.

5.58.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.utils.Parser:** relazione entrante, creazione.

5.58.4 Classi figlie

- Actorbase.server.messages.query.ErrorMessage.InvalidQueryMessage

5.59 Actorbase.server.messages.query.ErrorMessage.InvalidQueryMessage

5.59.1 Descrizione

Questa classe rappresenta un messaggio per segnalare che la richiesta effettuata non è riconosciuta dal server.

5.59.2 Utilizzo

Viene inviato per segnalare operazioni non riconosciute.

5.59.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.utils.Parser:** relazione entrante, creazione.

5.59.4 Trait implementati

- **Actorbase.server.messages.query.QueryMessage**
- **Actorbase.server.messages.query.ErrorMessages.ErrorMessage**

5.60 Actorbase.server.messages.query.ErrorMessages.QueryErrorInfo

5.60.1 Descrizione

Questa classe rappresenta un messaggio per segnalare che la richiesta effettuata non è stata eseguita correttamente dal server.

5.60.2 Utilizzo

Viene inviato per segnalare operazioni non eseguite correttamente.

5.60.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.utils.Parser:** relazione entrante, creazione.

5.60.4 Trait implementati

- **Actorbase.server.messages.query.QueryMessage**
- **Actorbase.server.messages.query.ReplyMessages.ReplyInfo**

5.61 Actorbase.server.messages.query.PermissionMessages

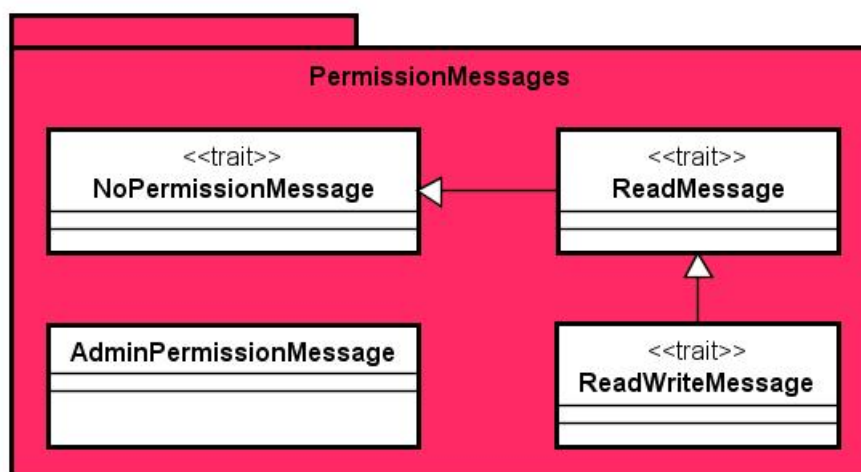


Figura 17: Componente Actorbase.server.messages.query.PermissionMessages

5.61.1 Descrizione

Package contenente i messaggi che rappresentano i vari livelli di permessi necessari agli utenti al fine di poter effettuare una richiesta.

5.61.2 Trait

- Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage
- Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage
- Actorbase.server.messages.query.PermissionMessages.ReadMessage
- Actorbase.server.messages.query.PermissionMessages.ReadWrite

5.62 Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage

5.62.1 Descrizione

Trait che rappresenta un comando che richiede i permessi di amministrazione per la sua esecuzione.

5.62.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

5.62.3 Classi figlie

- Actorbase.server.messages.query.admin.AdminMessage

5.63 Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage

5.63.1 Descrizione

Trait che rappresenta un comando che non richiede alcun permesso per la sua esecuzione.

5.63.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

5.63.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.HelpMessages.CompleteHelpMessage
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelpMessage

5.64 Actorbase.server.messages.query.PermissionMessages.ReadMessage

5.64.1 Descrizione

Trait che rappresenta un comando che richiede almeno i permessi di lettura per la sua esecuzione.

5.64.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

5.64.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

5.65 Actorbase.server.messages.query.PermissionMessages.ReadWriteMessage

5.65.1 Descrizione

Trait che rappresenta un comando che richiede i permessi di scrittura per la sua esecuzione.

5.65.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

5.65.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage

5.66 Actorbase.server.messages.query.ReplyMessages

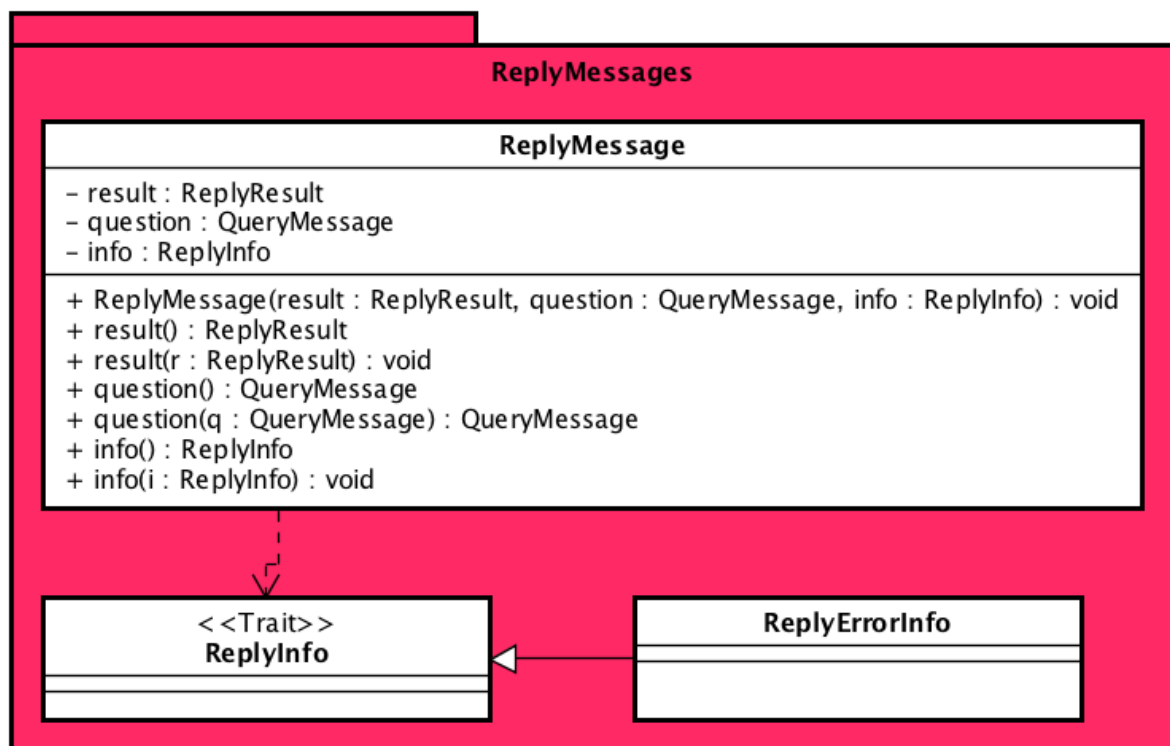


Figura 18: Componente Actorbase.server.messages.query.ReplyMessages

5.66.1 Descrizione

Package contenente i messaggi che rappresentano le risposte tra gli attori.

5.66.2 Trait

- Actorbase.server.messages.query.ReplyMessages.ReplyInfo

5.66.3 Classi

- Actorbase.server.messages.query.ReplyMessages.ReplyMessage
- Actorbase.server.messages.query.ReplyMessages.ReplyErrorInfo

5.67 Actorbase.server.messages.query.ReplyMessages.ReplyInfo

5.67.1 Descrizione

Trait che rappresenta una risposta tra attori del sistema contenente informazioni.

5.67.2 Utilizzo

Trait che viene esteso da tutti i messaggi che rappresentano risposte informative tra attori.

5.67.3 Classi figlie

- Actorbase.server.messages.query.ReplyMessages.ReplyMessage
- Actorbase.server.messages.query.ReplyMessages.ReplyErrorInfo

5.68 Actorbase.server.messages.query.ReplyMessages.ReplyMessage

5.68.1 Descrizione

Messaggio di risposta contenente delle informazioni (ReplyResult).

5.68.2 Utilizzo

Viene inviato come risposta da tutti gli attori che devono rispondere con una qualche informazione.

5.68.3 Trait implementati

- Actorbase.server.messages.query.ReplyMessages.ReplyInfo

5.69 Actorbase.server.messages.query.ReplyMessages.ReplyErrorInfo

5.69.1 Descrizione

Messaggio di risposta contenente un errore.

5.69.2 Utilizzo

Viene inviato come risposta da tutti gli attori che devono rispondere con un errore.

5.69.3 Trait implementati

- Actorbase.server.messages.query.ReplyMessages.ReplyInfo

5.70 Actorbase.server.messages.query.admin

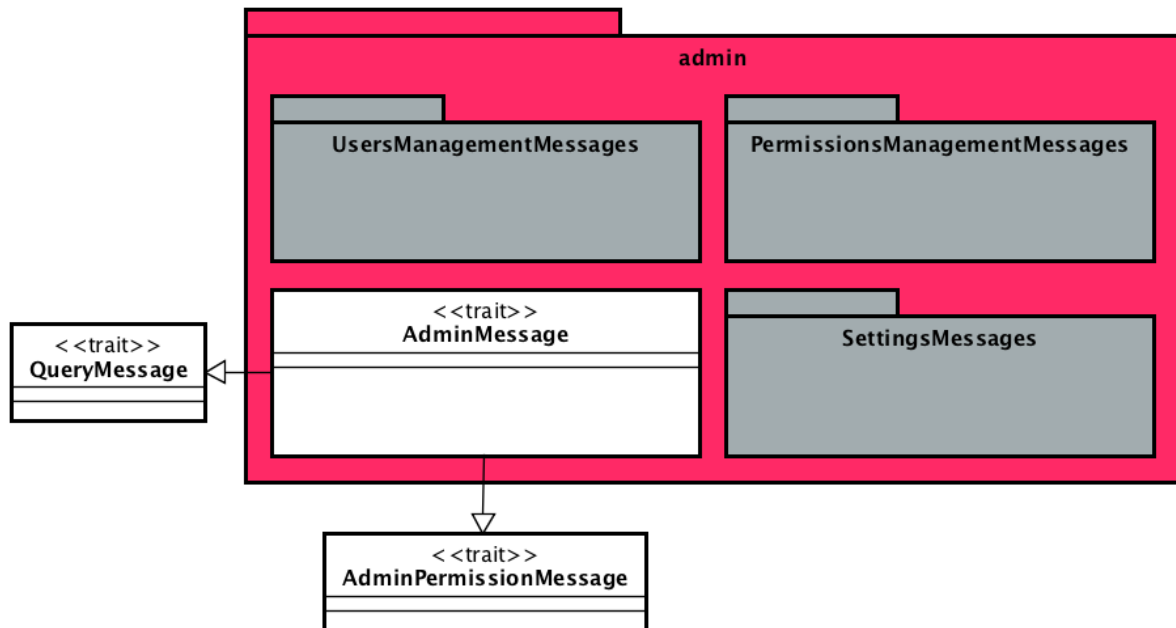


Figura 19: Componente Actorbase.server.messages.query.admin

5.70.1 Descrizione

Package contenente i messaggi che rappresentano i comandi amministrativi.

5.70.2 Package Figli

- Actorbase.server.messages.query.admin.SettingMessages
- Actorbase.server.messages.query.admin.PermissionsManagementMessages
- Actorbase.server.messages.query.admin.UserManagementMessages

5.70.3 Trait

- Actorbase.server.messages.query.admin.AdminMessage

5.71 Actorbase.server.messages.query.admin.AdminMessage

5.71.1 Descrizione

Trait che rappresenta tutti i comandi per effettuare operazioni di tipo amministrativo.

5.71.2 Utilizzo

Viene utilizzato per poter distinguere i comandi di tipo amministrativo da altri comandi.

5.71.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.71.4 Classi figlie

- Actorbase.server.messages.query.admin.ActorPropertiesMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessage
- Actorbase.server.messages.query.admin.UsersManagementMessage

5.72 Actorbase.server.messages.query.admin.SettingMessages

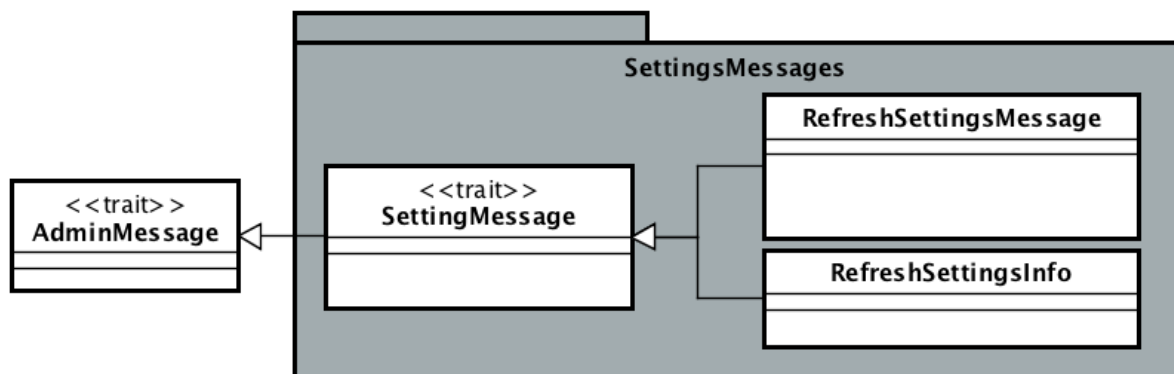


Figura 20: Componente Actorbase.server.messages.query.admin.SettingMessages

5.72.1 Descrizione

Package contenente i messaggi che permettono di gestire i le impostazioni del sistema.

5.72.2 Trait

- Actorbase.server.messages.query.admin.SettingMessages.SettingMessage

5.72.3 Classi

- Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsMessage
- Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsInfo

5.73 Actorbase.server.messages.query.admin.SettingMessages.SettingMessage

5.73.1 Descrizione

Trait che tutti i messaggi amministrativi relativi alle impostazioni devono estendere.

5.73.2 Utilizzo

Viene esteso dai messaggi relativi alle impostazioni del sistema.

5.73.3 Classi figlie

- Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsMessage
- Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsInfo

5.73.4 Trait estesi

- Actorbase.server.messages.query.admin.AdminMessage

5.74 Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsMessage

5.74.1 Descrizione

Messaggio che rappresenta una richiesta di aggiornamento delle impostazioni.

5.74.2 Utilizzo

Viene utilizzato per richiedere un refresh delle impostazioni da parte di diversi attori.

5.74.3 Trait implementati

- Actorbase.server.messages.query.admin.SettingMessages.SettingMessage

5.75 Actorbase.server.messages.query.admin.SettingMessages.RefreshSettingsInfo

5.75.1 Descrizione

Messaggio che rappresenta una risposta ad una richiesta di aggiornamento delle impostazioni.

5.75.2 Utilizzo

Viene utilizzato per rispondere dell'esito di un'operazione di aggiornamento impostazioni.

5.75.3 Trait implementati

- Actorbase.server.messages.query.admin.SettingMessages.SettingMessage

5.76 Actorbase.server.messages.query.admin.PermissionsManagementMessages

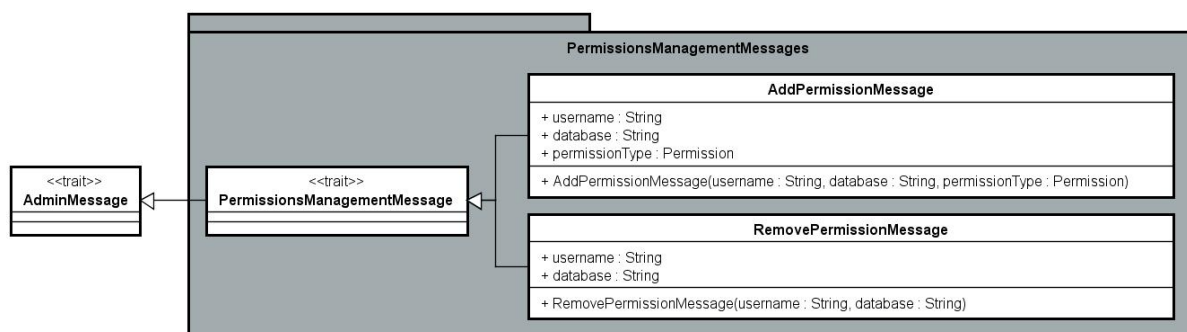


Figura 21: Componente Actorbase.server.messages.query.admin.PermissionsManagementMessages

5.76.1 Descrizione

Pacchetto contenente i messaggi che permettono di gestire i permessi assegnati agli utenti.

5.76.2 Classi

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.AddPermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.RemovePermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.ListPermissionMessage

5.76.3 Trait

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

5.77 Actorbase.server.messages.query.admin .PermissionsManagementMessages.PermissionsManagementMessage

5.77.1 Descrizione

Trait che rappresenta i messaggi per gestire i permessi assegnati agli utenti.

5.77.2 Utilizzo

Viene utilizzato per distinguere i messaggi per gestire i permessi assegnati agli utenti.

5.77.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.77.4 Classi figlie

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.AddPermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.RemovePermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.ListPermissionMessage

5.78 Actorbase.server.messages.query.admin .PermissionsManagementMessages.AddPermissionMessage

5.78.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di aggiunta di permessi ad uno specifico utente.

5.78.2 Utilizzo

Viene utilizzata per aggiungere permessi ad uno specifico utente.

5.78.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.78.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

5.79 Actorbase.server.messages.query.admin .PermissionsManagementMessages.RemovePermissionMessage

5.79.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di permessi ad uno specifico utente.

5.79.2 Utilizzo

Viene utilizzata per rimuovere permessi ad uno specifico utente.

5.79.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.79.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

5.80 Actorbase.server.messages.query.admin. .PermissionsManagementMessages.ListPermissionMessage

5.80.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di permessi di uno specifico utente.

5.80.2 Utilizzo

Viene utilizzata per mostrare i permessi per ogni database di uno specifico utente.

5.80.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.80.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

5.81 Actorbase.server.messages.query.admin.UserManagementMessages

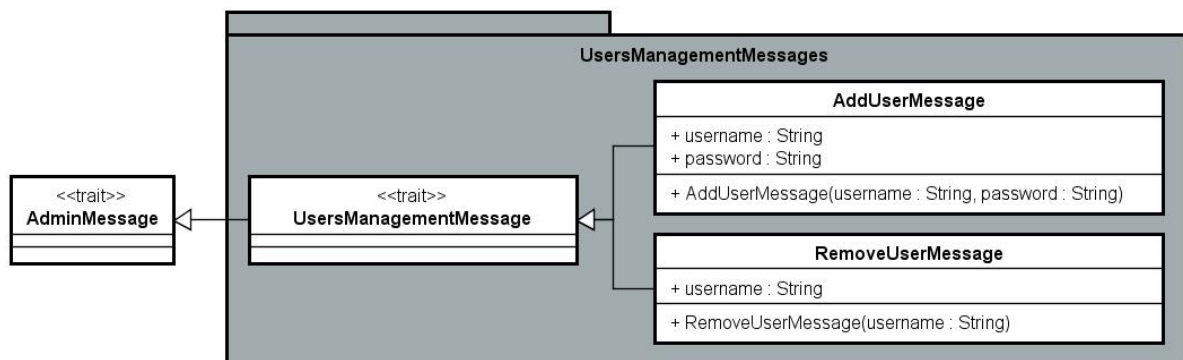


Figura 22: Componente Actorbase.server.messages.query.admin.UserManagementMessages

5.81.1 Descrizione

Pacchetto contenente i messaggi per gestire gli utenti nel database.

5.81.2 Classi

- Actorbase.server.messages.query.admin.UserManagementMessages.AddUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.RemoveUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.ListUserMessage

5.81.3 Trait

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

5.82 Actorbase.server.messages.query.admin .UserManagementMessages.UserManagementMessage

5.82.1 Descrizione

Trait che rappresenta i messaggi per gestire gli utenti nel database.

5.82.2 Utilizzo

Viene utilizzato per distinguere i messaggi per gestire gli utenti nel database.

5.82.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.82.4 Classi figlie

- Actorbase.server.messages.query.admin.UserManagementMessages.AddUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.RemoveUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.ListUserMessage

5.83 Actorbase.server.messages.query.admin .UserManagementMessages.AddUserMessage

5.83.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di aggiunta di un utente.

5.83.2 Utilizzo

Viene utilizzata per aggiungere un utente al sistema.

5.83.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.83.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

5.84 Actorbase.server.messages.query.admin .UserManagementMessages.RemoveUserMessage

5.84.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di un utente.

5.84.2 Utilizzo

Viene utilizzata per rimuovere un utente dal sistema.

5.84.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.84.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

5.85 Actorbase.server.messages.query.admin .UserManagementMessages.ListUserMessage

5.85.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di tutti gli utenti presenti.

5.85.2 Utilizzo

Viene utilizzata per visualizzare tutti gli utenti presenti nel sistema.

5.85.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.85.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

5.86 Actorbase.server.messages.query.user

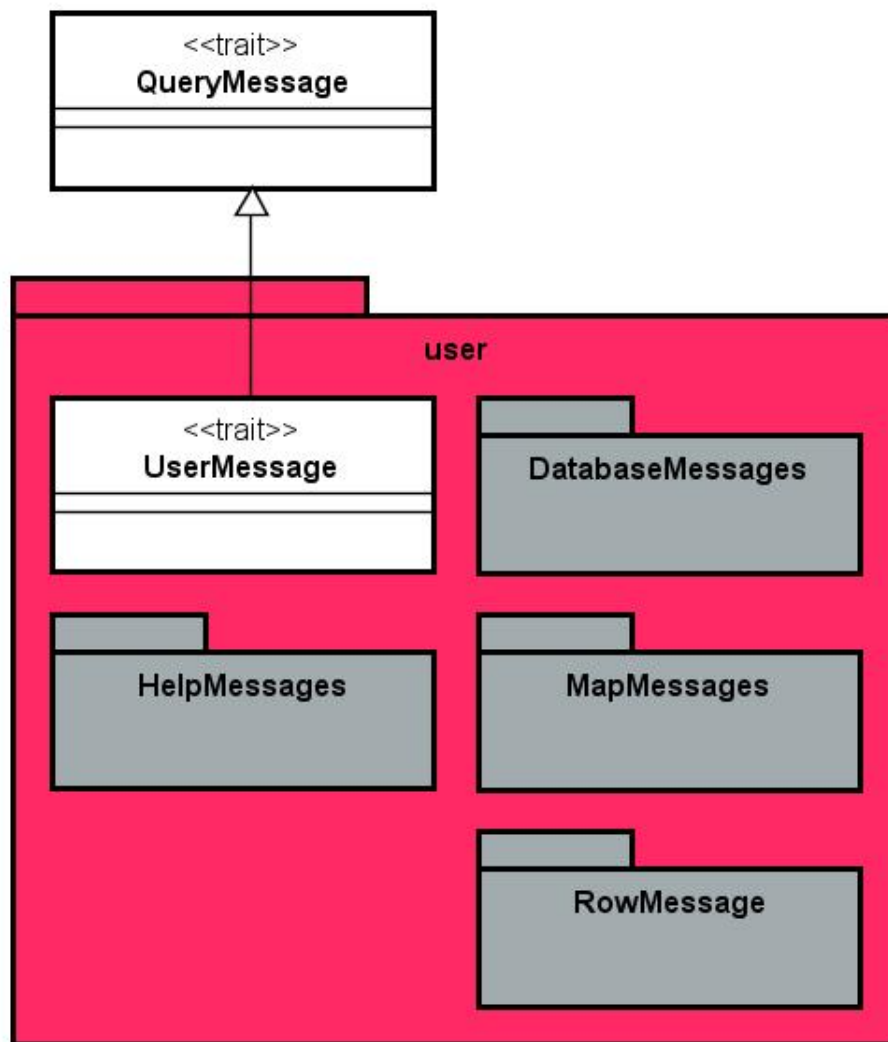


Figura 23: Componente Actorbase.server.messages.query.user

5.86.1 Descrizione

Pacchetto contenente tutti i messaggi che rappresentano richieste non amministrative.

5.86.2 Package Figli

- Actorbase.server.messages.query.user.RowMessages
- Actorbase.server.messages.query.user.MapMessages
- Actorbase.server.messages.query.user.DatabaseMessages
- Actorbase.server.messages.query.user.HelpMessages

5.86.3 Trait

- Actorbase.server.messages.query.user.UserMessage

5.87 Actorbase.server.messages.query.user.UserMessage

5.87.1 Descrizione

Trait che rappresenta richieste non amministrative.

5.87.2 Utilizzo

Viene utilizzato per distinguere i messaggi non amministrativi.

5.87.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.87.4 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.MapMessage
- Actorbase.server.messages.query.user.RowMessages.RowMessage
- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

5.88 Actorbase.server.messages.query.user.RowMessages

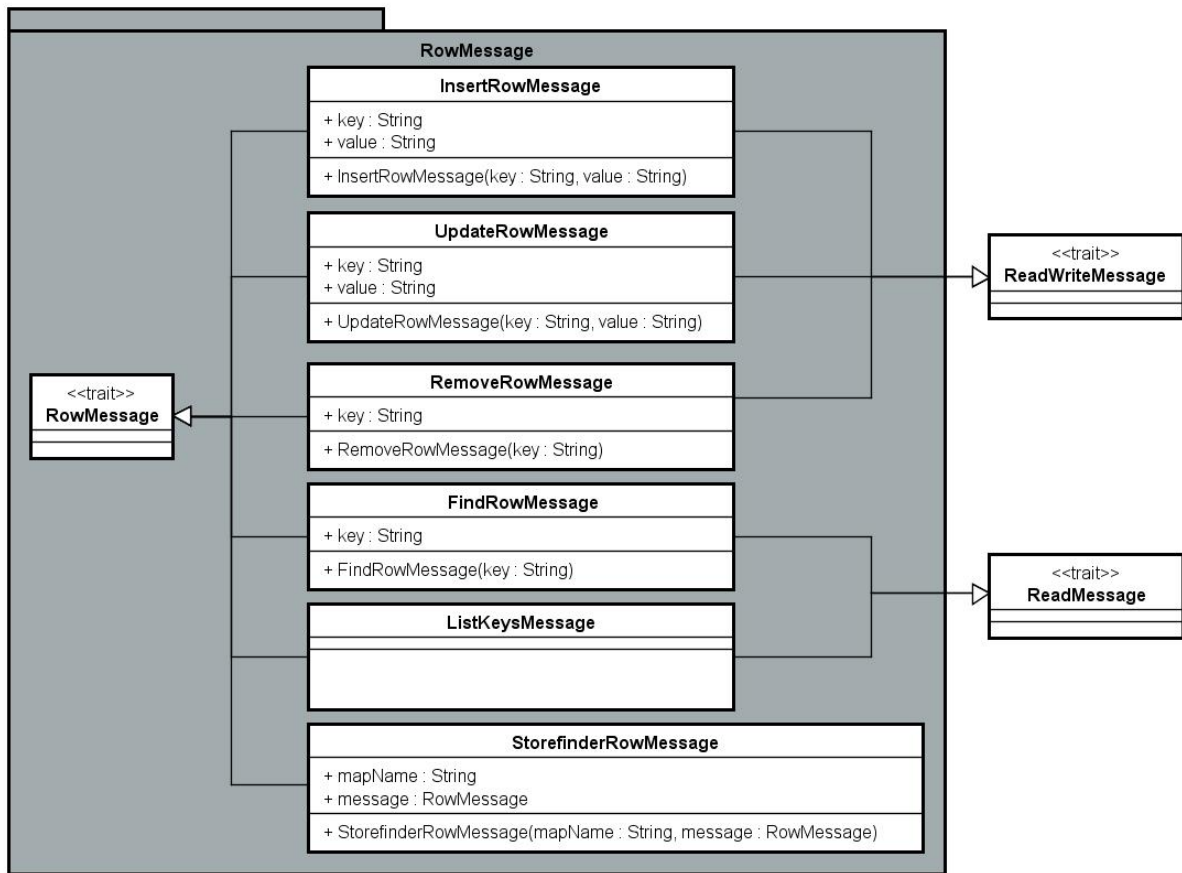


Figura 24: Componente Actorbase.server.messages.query.user.RowMessages

5.88.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di riga.

5.88.2 Classi

- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

5.88.3 Trait

- Actorbase.server.messages.query.user.RowMessages.RowMessage

5.89 Actorbase.server.messages.query.user.RowMessages.RowMessage

5.89.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di riga.

5.89.2 Utilizzo

Viene utilizzato per poter riconoscere i messaggi contenenti richieste a livello di riga.

5.89.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.89.4 Classi figlie

- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

5.90 Actorbase.server.messages.query.user.RowMessages.InsertRowMessage

5.90.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di inserimento di una riga in una mappa.

5.90.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

5.90.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.90.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

5.91 Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage

5.91.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di modifica di una riga in una mappa.

5.91.2 Utilizzo

Effettua l'intero percorso descritto in `Actorbase.server.messages.query.QueryMessage`.

5.91.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.91.4 Trait implementati

- `Actorbase.server.messages.query.user.RowMessages.RowMessage`

5.92 Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage

5.92.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di una riga in una mappa.

5.92.2 Utilizzo

Effettua l'intero percorso descritto in `Actorbase.server.messages.query.QueryMessage`.

5.92.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.92.4 Trait implementati

- `Actorbase.server.messages.query.user.RowMessages.RowMessage`

5.93 Actorbase.server.messages.query.user.RowMessages.FindRowMessage

5.93.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di ricercare una riga in una mappa.

5.93.2 Utilizzo

Effettua l'intero percorso descritto in `Actorbase.server.messages.query.QueryMessage`.

5.93.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.93.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

5.94 Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

5.94.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di tutte le chiavi presenti in una mappa.

5.94.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

5.94.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.94.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

5.95 Actorbase.server.messages.query.user.MapMessages

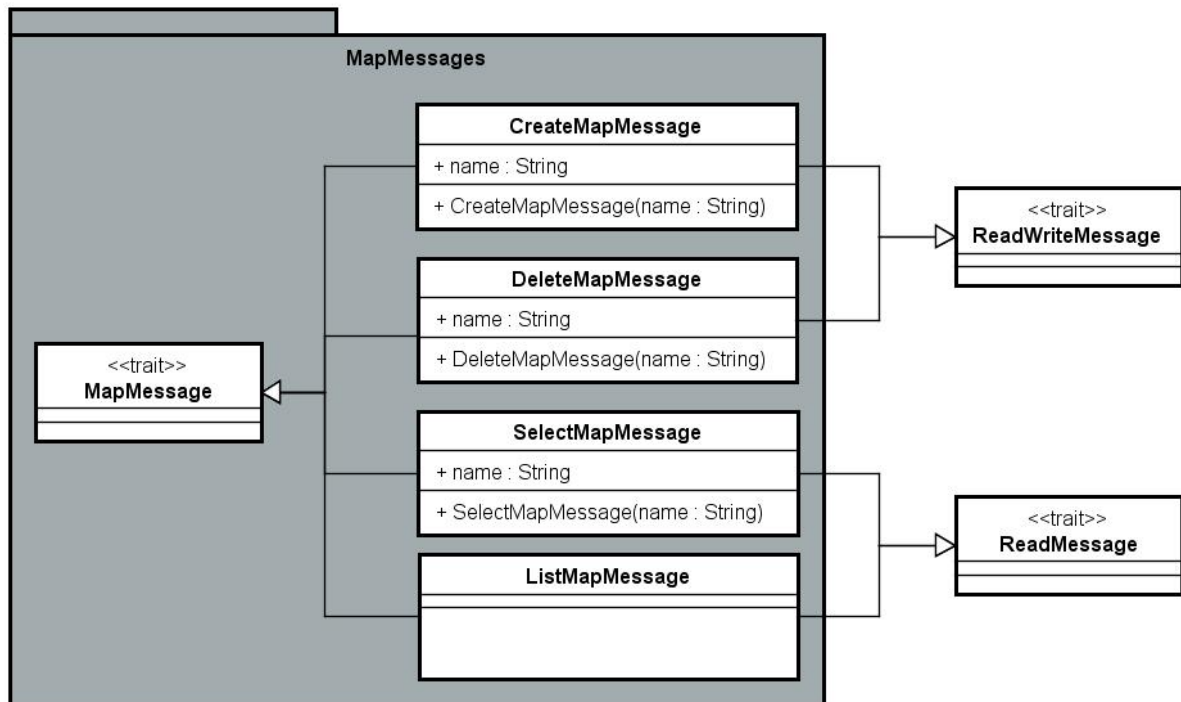


Figura 25: Componente Actorbase.server.messages.query.user.MapMessages

5.95.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di mappa.

5.95.2 Classi

- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage

5.95.3 Trait

- Actorbase.server.messages.query.user.MapMessages.MapMessage

5.96 Actorbase.server.messages.query.user.MapMessages.MapMessage

5.96.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di mappa.

5.96.2 Utilizzo

Viene utilizzato per poter riconoscere i messaggi contenenti richieste a livello di mappa.

5.96.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.96.4 Classi figlie

- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage

5.97 Actorbase.server.messages.query.user.MapMessages.CreateMapMessage

5.97.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di creazione di una mappa in un database.

5.97.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Store-keeper**.

5.97.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.97.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

5.98 Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage

5.98.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di una mappa in un database.

5.98.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Store-keeper**.

5.98.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.98.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

5.99 Actorbase.server.messages.query.user.MapMessages.SelectMapMessage

5.99.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di selezione di una mappa in un database.

5.99.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Store-keeper**.

5.99.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.99.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

5.100 Actorbase.server.messages.query.user.MapMessages.ListMapMessage

5.100.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di tutte le mappe presenti in un database.

5.100.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Store-keeper**.

5.100.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.
- **Actorbase.server.actors.IndexManager:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

5.100.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

5.101 Actorbase.server.messages.query.user.DatabaseMessages

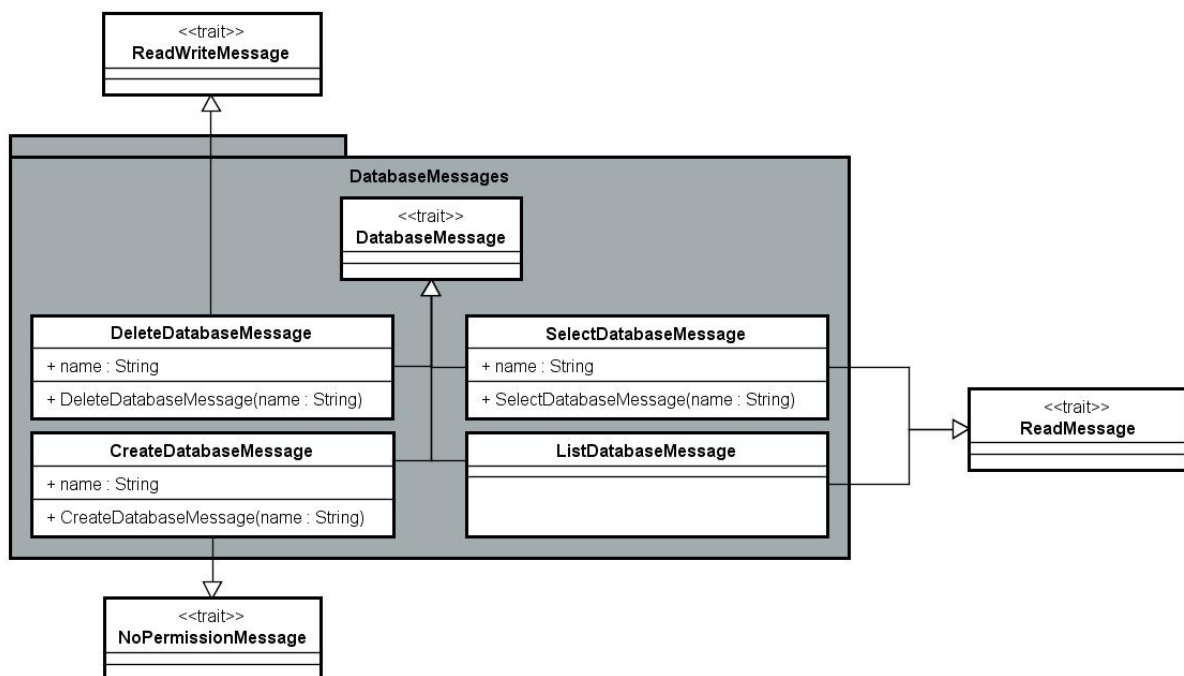


Figura 26: Componente Actorbase.server.messages.query.user.DatabaseMessages

5.101.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di database.

5.101.2 Classi

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

5.101.3 Trait

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.102 Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.102.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di database.

5.102.2 Utilizzo

Viene utilizzato per poter distinguere i messaggi contenenti richieste a livello di database.

5.102.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.102.4 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

5.103 Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage

5.103.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di creazione di un database.

5.103.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

5.103.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.103.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.104 Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage

5.104.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di un database.

5.104.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

5.104.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.104.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.105 Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage

5.105.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di selezione di un database.

5.105.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

5.105.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.105.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.106 Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

5.106.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione dell'intera lista di database presenti.

5.106.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**.

5.106.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.MapManager:** relazione di utilizzo.

5.106.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

5.107 Actorbase.server.messages.query.user.HelpMessages

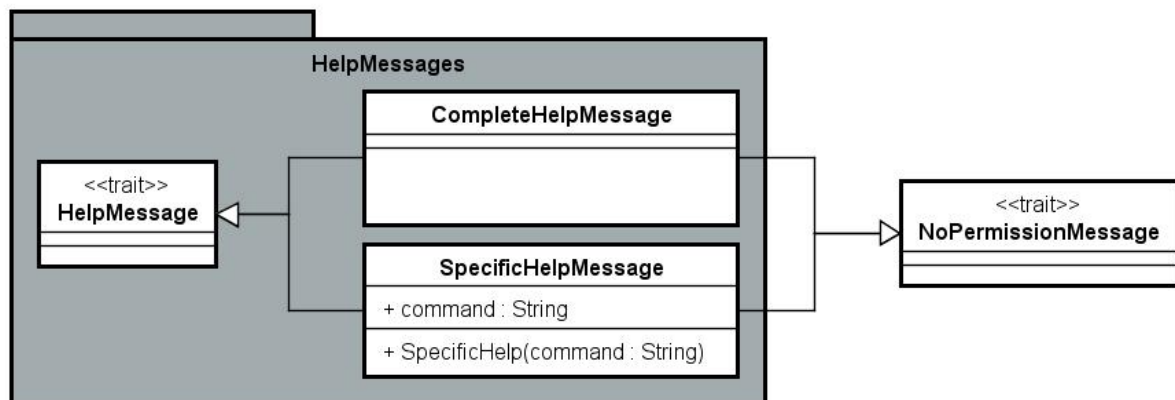


Figura 27: Componente Actorbase.server.messages.query.user.HelpMessages

5.107.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query di aiuto.

5.107.2 Classi

- Actorbase.server.messages.query.user.HelpMessages.CompleteHelp
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelp

5.107.3 Trait

- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

5.108 Actorbase.server.messages.query.user.HelpMessages.HelpMessage

5.108.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste per la visualizzazione di aiuti.

5.108.2 Utilizzo

Viene utilizzata per poter distinguere i messaggi contenenti richieste per la visualizzazione di aiuti.

5.108.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

5.108.4 Classi figlie

- Actorbase.server.messages.query.user.HelpMessages.CompleteHelp
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelp

5.109 Actorbase.server.messages.query.user.HelpMessages.CompleteHelp

5.109.1 Descrizione

Classe che rappresenta i messaggi contenenti richieste per la visualizzazione di tutti i comandi disponibili.

5.109.2 Utilizzo

Effettua il percorso descritto in `Actorbase.server.messages.query.QueryMessage` fermandosi all'attore `Main`.

5.109.3 Relazioni con altre classi

- `Actorbase.server.utils.Parser`: relazione entrante di creazione.
- `Actorbase.server.actors.Usermanager`: relazione di utilizzo.
- `Actorbase.server.actors.Main`: relazione di utilizzo.

5.109.4 Trait implementati

- `Actorbase.server.messages.query.user.HelpMessages.HelpMessage`

5.110 `Actorbase.server.messages.query.user.HelpMessages.SpecificHelp`

5.110.1 Descrizione

Classe che rappresenta i messaggi contenenti richieste per la visualizzazione di aiuto riguardante un comando specifico.

5.110.2 Utilizzo

Effettua il percorso descritto in `Actorbase.server.messages.query.QueryMessage` fermandosi all'attore `Main`.

5.110.3 Relazioni con altre classi

- `Actorbase.server.utils.Parser`: relazione entrante di creazione.
- `Actorbase.server.actors.Usermanager`: relazione di utilizzo.
- `Actorbase.server.actors.Main`: relazione di utilizzo.

5.110.4 Trait implementati

- `Actorbase.server.messages.query.user.HelpMessages.HelpMessage`

5.111 Actorbase.client

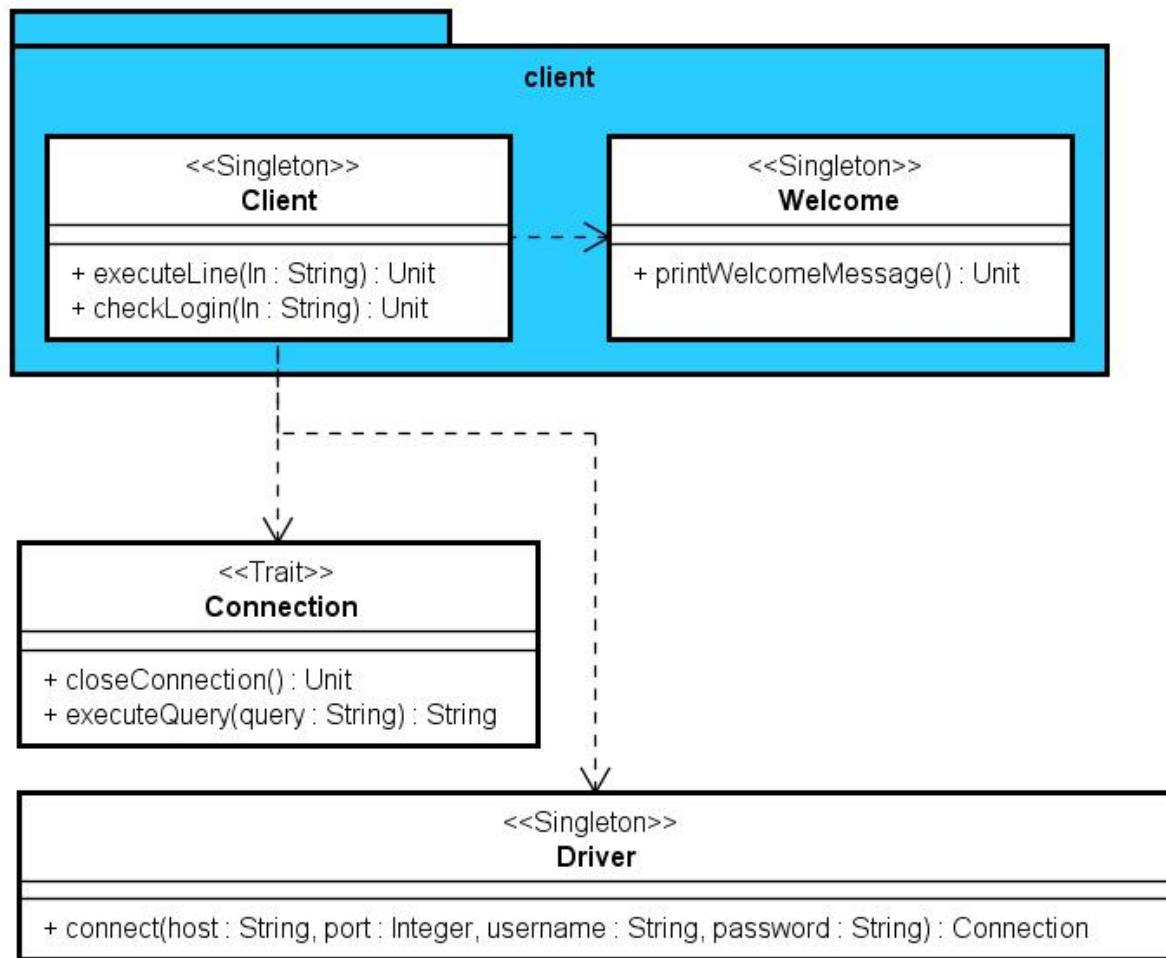


Figura 28: Componente Actorbase.client

5.111.1 Descrizione

Package per la componente lato client del sistema. È composto due classi che implementano il design pattern **Singleton**.

5.111.2 Classi

- Actorbase.client.Client
- Actorbase.client.Welcome

5.112 Actorbase.client.Client

5.112.1 Descrizione

Questa classe fornisce un'interfaccia a linea di comando e permette all'utente di inserire i comandi che verranno inviati al server. Questa classe implementa il design pattern **Singleton**.

5.112.2 Utilizzo

Viene utilizzata per distinguere i comandi di connessione, disconnessione e altri comandi possibili. Inoltre utilizza il driver per comunicare con il server.

5.112.3 Relazioni con altre classi

- **Actorbase.client.Welcome:** relazione di utilizzo.
- **Actorbase.driver.Driver:** relazione di utilizzo.
- **Actorbase.driver.Connection:** relazione di utilizzo.

5.113 Actorbase.client.Welcome

5.113.1 Descrizione

Classe di supporto per stampare un messaggio di benvenuto sulla console del client.

5.113.2 Utilizzo

Viene utilizzata dal client per stampare un messaggio di benvenuto all'avvio dell'interfaccia a linea di comando.

5.113.3 Relazioni con altre classi

- **Actorbase.client.Client**

5.114 Actorbase.driver

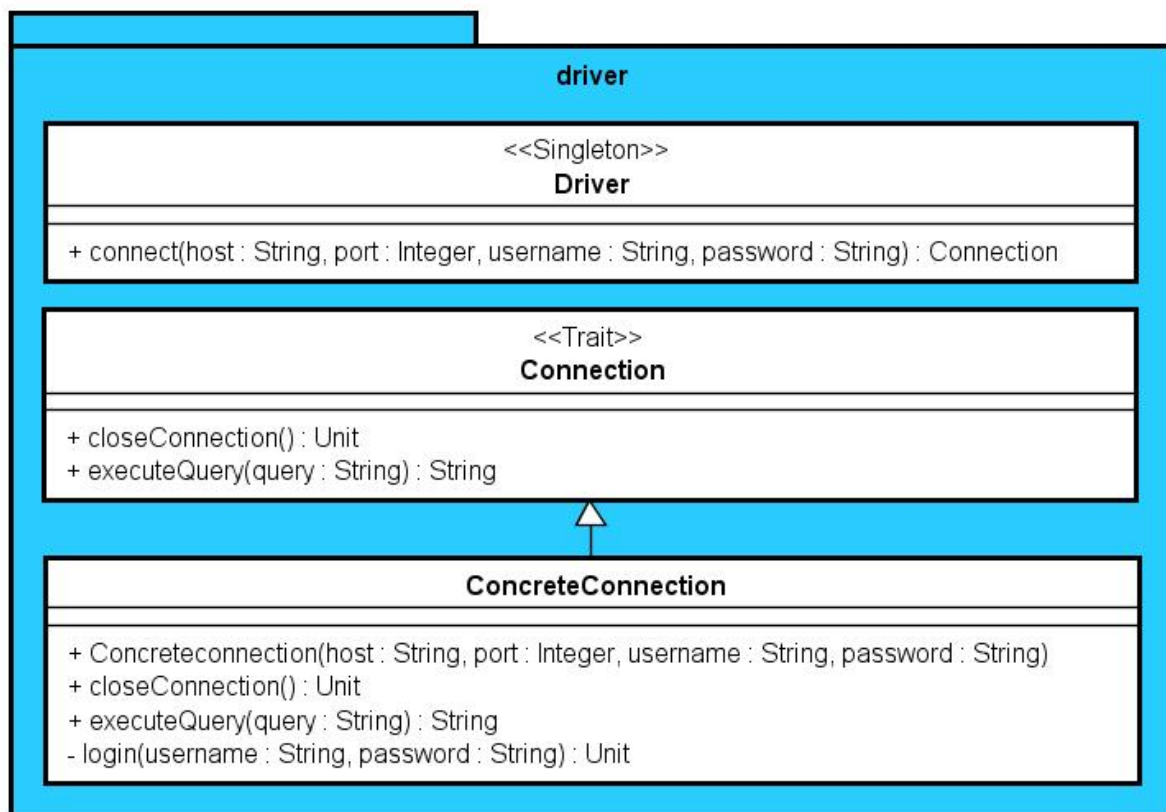


Figura 29: Componente Actorbase.driver

5.114.1 Descrizione

Package per la componente driver del sistema. È composto una classe che implementa il design pattern **Singleton**, un trait e una sua implementazione.

5.114.2 Classi

- Actorbase.driver.Driver
- Actorbase.driver.ConcreteConnection

5.114.3 Trait

- Actorbase.driver.Connection

5.115 Actorbase.driver.Connection

5.115.1 Descrizione

Trait utilizzato per effettuare il login su un determinato server.

5.115.2 Utilizzo

Costruisce una query nel modo corretto ricevendo un comando in formato stringa dal client.

5.115.3 Relazioni con altre classi

- **Actorbase.driver.Driver:** relazione entrante, creazione.
- **Actorbase.driver.Client:** relazione di utilizzo.

5.115.4 Classi figlie

- Actorbase.driver.ConcreteConnection

5.116 Actorbase.driver.ConcreteConnection

5.116.1 Descrizione

Classe utilizzata per effettuare il login su un determinato server. Apre una connessione verso il server specificato.

5.116.2 Utilizzo

Costruisce una query nel modo corretto ricevendo un comando in formato stringa dal client.

5.116.3 Relazioni con altre classi

- **Actorbase.driver.Driver:** relazione entrante, creazione.
- **Actorbase.driver.Client:** relazione di utilizzo.

5.116.4 Trait implementati

- Actorbase.driver.Connection

5.117 Actorbase.driver.Driver

5.117.1 Descrizione

Classe che implementa il design pattern **Singleton**,

5.117.2 Utilizzo

Viene utilizzato per creare una connessione che viene restituita al client.

5.117.3 Relazioni con altre classi

- **Actorbase.client.Client:** relazione di utilizzo.
- **Actorbase.driver.ConcreteConnection:** relazione uscente, creazione.

6 Diagrammi delle attività

Segue il diagramma delle attività che mostra l'interazione dell'utente con il client.

6.0.1 Diagramma attività principale

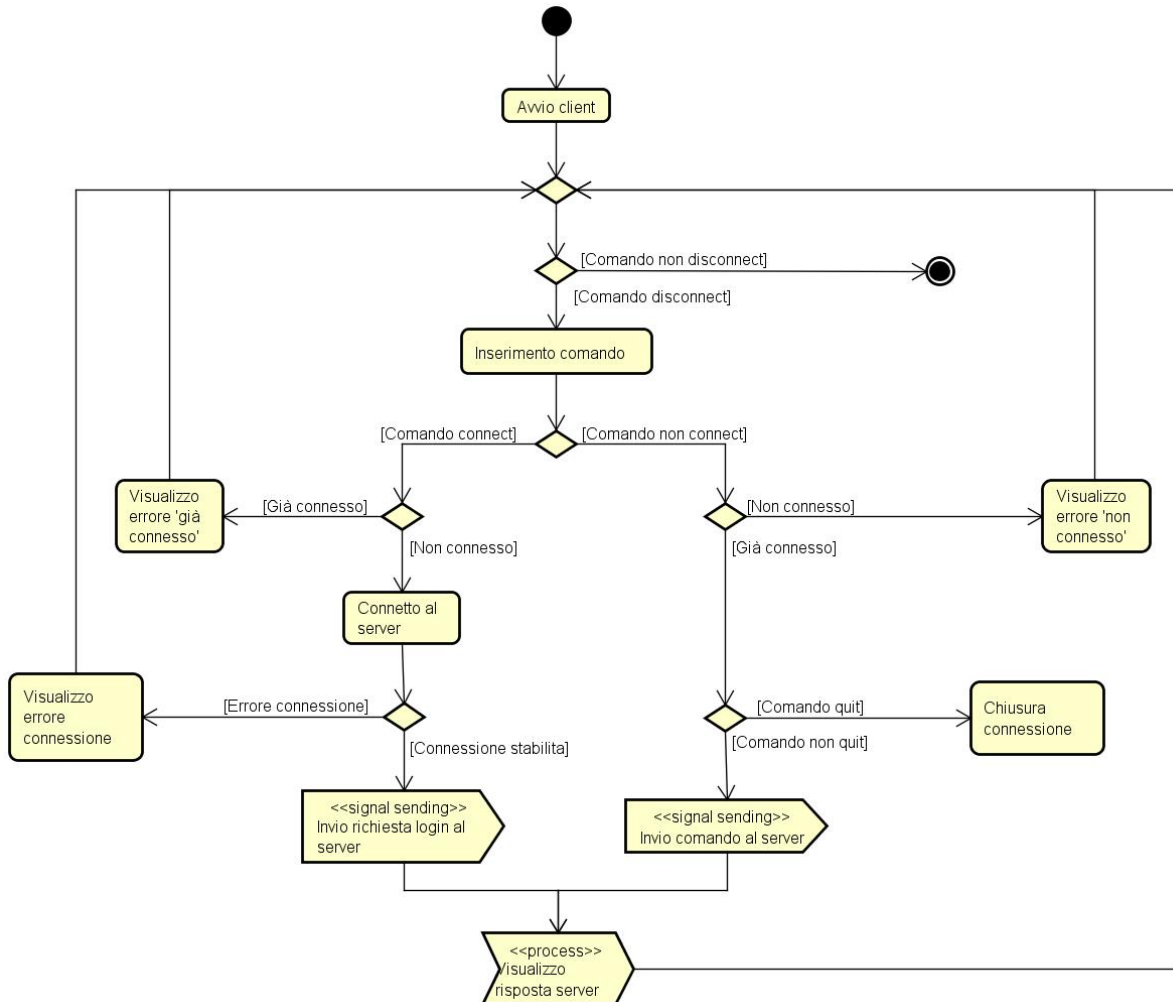


Figura 30: Diagramma attività principale

Dopo aver avviato il client l'utente può inserire dei comandi. Se il comando è di chiusura il client si spegne. Se il comando è di connect ed il client non è connesso quest'ultimo si connette ed invia un comando di login al server, altrimenti stampa un messaggio di errore. Se il comando non è di connect ed il client è connesso quest'ultimo lo controlla. Se esso è un comando di disconnessione il client si disconnette dal server, altrimenti invia la richiesta.

7 Struttura del sistema ad attori

In questa sezione viene illustrata ad alto livello l'architettura della logica ad attori del sistema lato server, vero nucleo di *Actorbase*.

7.1 Struttura principale

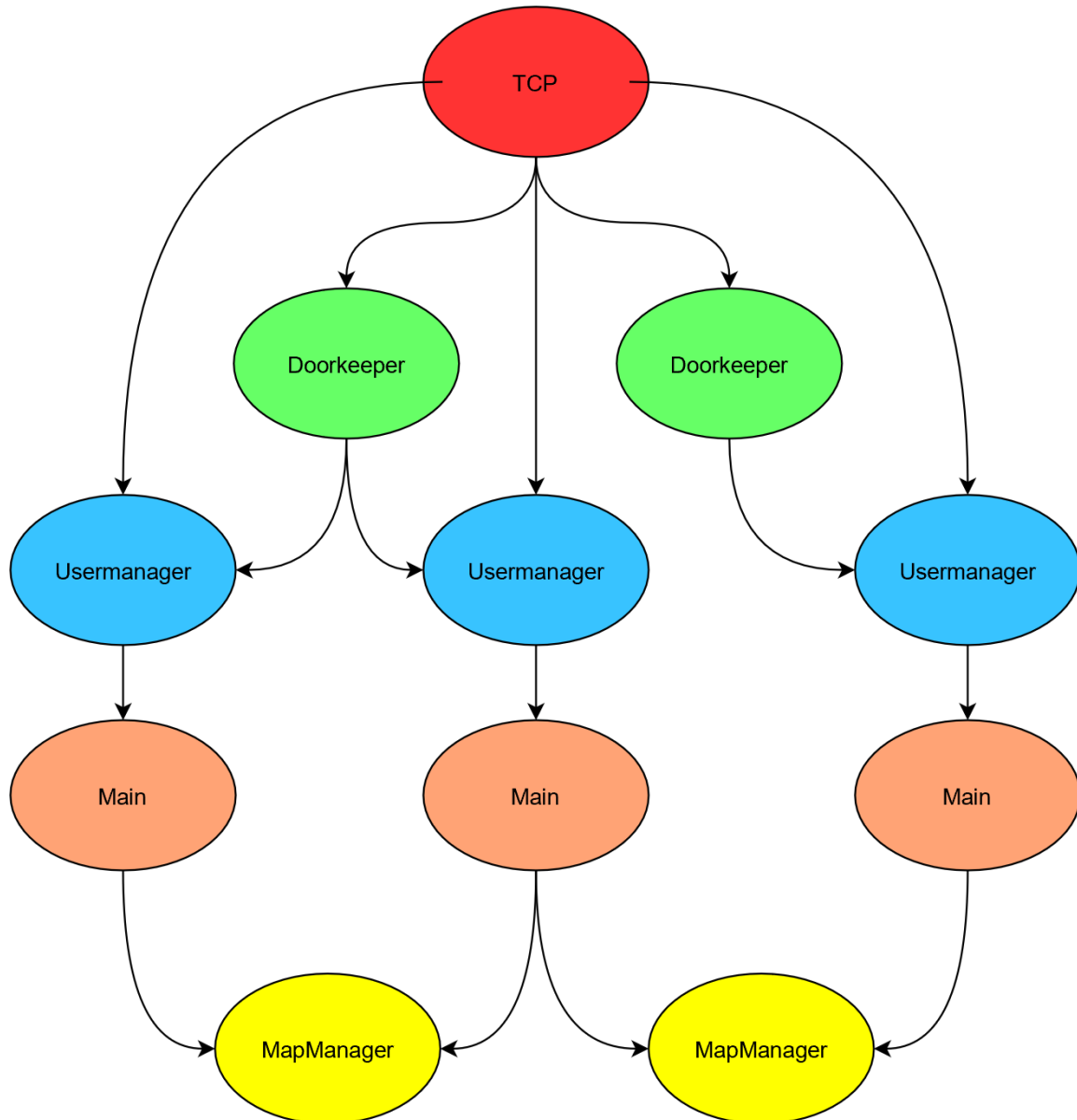


Figura 31: Il sistema ad attori di Actorbase

La struttura del server può individuare negli attori TCP (forniti da Akka) e negli attori **Doorkeeper** i nodi iniziali. Un attore **Doorkeeper** apre un socket TCP e si mette in ascolto, quando viene stabilita una nuova connessione esso crea un attore di tipo **Usermanager** a cui ne delega la gestione.

Dunque ogni client connesso al server è rappresentato e gestito da un attore **Usermanager** dedicato che si occupa di gestire i messaggi inviati dal client, trasformandoli in messaggi gestibili dal sistema ad attori. Tali messaggi vengono prima di tutto gestiti da un attore di tipo **Main**. Esso effettua una prima analisi e gestione dei messaggi. Una volta che un messaggio arriva all'attore **Main**, infatti, vie-

ne interpretato e può essere gestito direttamente dall'attore o inoltrato alle componenti più basse del sistema.

7.2 Struttura di un database

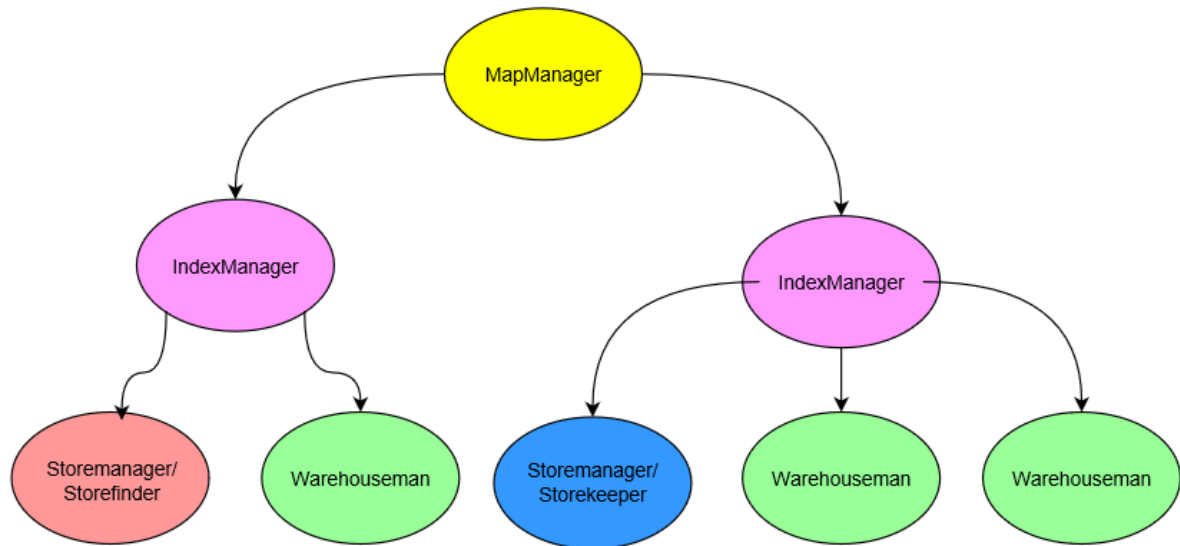


Figura 32: Il sistema ad attori di Actorbase, struttura di un singolo database

Un singolo database del sistema è rappresentato da un attore **MapManager**. Esso si occupa di gestire e indicizzare tutti gli attori di tipo **IndexManager**, rappresentanti le mappe che compongono tale database.

Ogni **IndexManager** indicizza e gestisce i dati che compongono la mappa in due modi: in memoria principale e su disco. La gestione in RAM avviene tramite l'utilizzo di attori di tipo **Storemanager** mentre della gestione dei dati su disco si occupano attori di tipo **Warehouseman**.

7.3 Struttura di una mappa in memoria principale

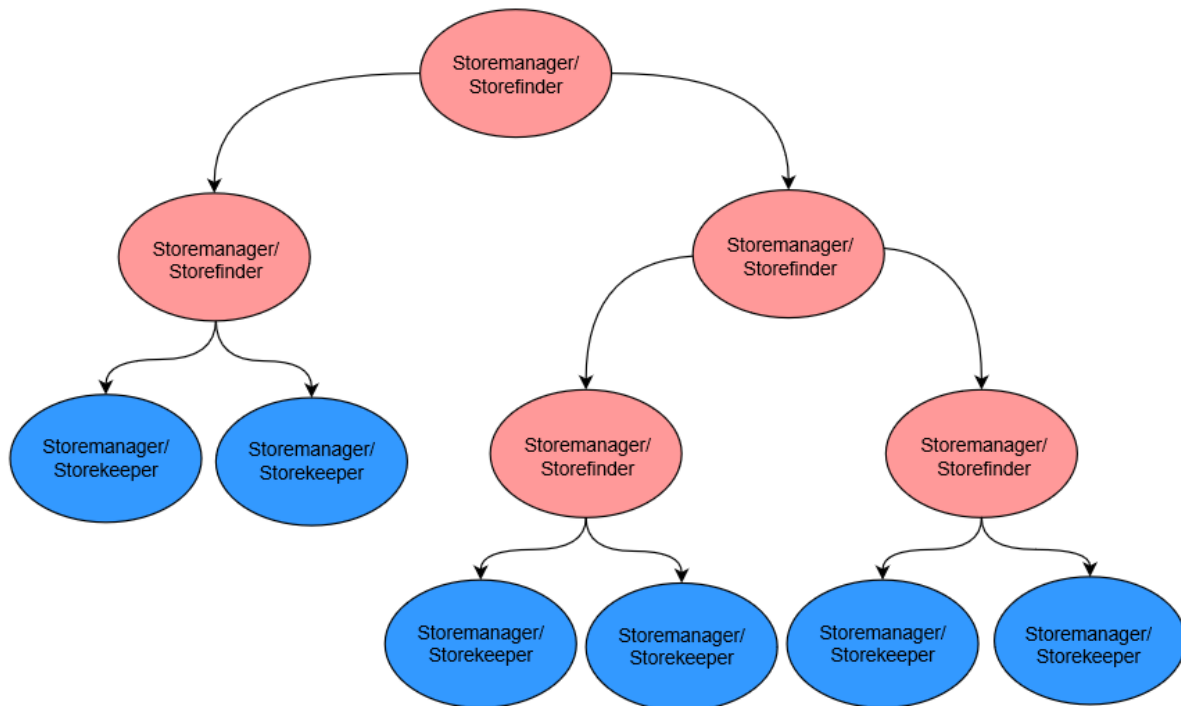


Figura 33: Il sistema ad attori di Actorbase, struttura di una mappa in memoria principale

Una mappa in memoria principale è rappresentata da un albero binario di attori **Storemanager**. Attori di tale tipo assumono un comportamento diverso nel caso in cui agiscano da indicizzatori di altri **Storemanager** o da contenitori dei dati veri e propri. Nel primo caso il comportamento adottato è quello dell'attore **Storefinder** (in rosso nel diagramma), nel secondo caso quello dell'attore **Storekeeper** (in blu nel diagramma).

Questa struttura permette di avere un sistema stabile e affidabile. Quando un attore raggiunge la sua capienza massima non si divide in due ma genera due figli e prosegue nella gestione dei messaggi che aveva in mailbox. In questo modo il padre non si accorge dello sdoppiamento del figlio, né del suo cambio di comportamento. D'altro lato il figlio non rischia di perdere messaggi nell'operazione di sdoppiamento.

7.4 Struttura degli attori di backup

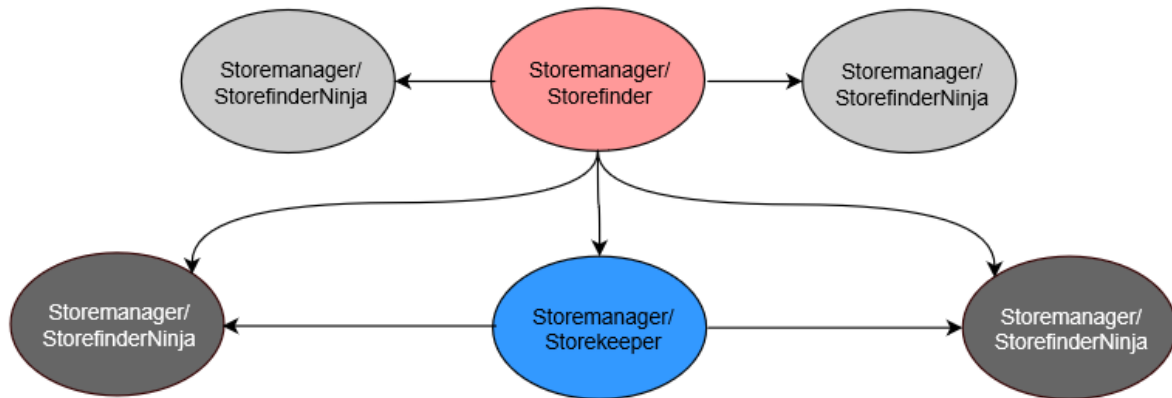


Figura 34: Il sistema ad attori di Actorbase, struttura degli attori di backup

Ogni **Storefinder** e **Storekeeper** possiede un insieme di attori **Storemanager** che agiscono da copia delle proprie informazioni, il comportamento di tali attori è detto **Ninja**. Essi permettono di avere delle copie di backup che possono essere sostituite agli attori principali in caso di necessità. Inoltre possono essere usati per incrementare le performance del sistema delegando a loro la risoluzione di alcune query, alleggerendo così l'attore principale.

8 Tracciamento

8.1 Tracciamento componenti-requisiti

Componente	Requisiti
Actorbase.server	R[1][N][F] e figli R[5][N][V]
Actorbase.server.actors	R[1.3][N][F] e figli R[1.4][N][F] e figli R[1.5][N][F] e figli R[1.6][N][F] e figli R[1.7][N][F] e figli R[5][N][V]
Actorbase.server.utils	R[1.1][N][F] e figli R[1.2][N][F] R[1.3.1][N][F] R[1.3.2][N][F] R[1.4.4.1][N][F] e figli R[1.4.5.1][N][F] e figli R[1.4.6.1][N][F] e figli R[1.4.6.2][N][F] e figli R[1.4.7.1][N][F] e figli R[1.5.2.1][D][F] e figli R[1.5.3.1][D][F] e figli R[1.5.4.1][D][F] e figli R[1.5.4.2][D][F] e figli R[1.5.5.1][D][F] e figli R[1.6.2.1][N][F] R[1.6.3.1][N][F] R[1.6.4.1][N][F] R[1.6.5.1][D][F] R[1.6.6.1][N][F] R[1.7.2][N][F] R[1.7.3][N][F] R[1.7.4][N][F] R[1.7.5][N][F] R[1.7.6][N][F] R[1.7.7][N][F]
Actorbase.client	R[2][N][F] e figli
Actorbase.driver	R[3][N][F] e figli

Tabella 2: Componenti-Requisiti

8.2 Tracciamento requisiti-componenti

Requisito	Componenti
R[1][N][F] e figli	Actorbase.server
R[1.1][N][F] e figli	Actorbase.server.utils
R[1.2][N][F]	Actorbase.server.utils
R[1.3][N][F] e figli	Actorbase.server.actors
R[1.3.1][N][F]	Actorbase.server.utils
R[1.3.2][N][F]	Actorbase.server.utils
R[1.4][N][F] e figli	Actorbase.server.actors
R[1.4.4.1][N][F] e figli	Actorbase.server.utils
R[1.4.5.1][N][F] e figli	Actorbase.server.utils
R[1.4.6.1][N][F] e figli	Actorbase.server.utils
R[1.4.6.2][N][F] e figli	Actorbase.server.utils

Requisito	Componenti
R[1.4.7.1][N][F] e figli	Actorbase.server.utils
R[1.5][N][F] e figli	Actorbase.server.actors
R[1.5.2.1][D][F] e figli	Actorbase.server.utils
R[1.5.3.1][D][F] e figli	Actorbase.server.utils
R[1.5.4.1][D][F] e figli	Actorbase.server.utils
R[1.5.4.2][D][F] e figli	Actorbase.server.utils
R[1.5.5.1][D][F] e figli	Actorbase.server.utils
R[1.6][N][F] e figli	Actorbase.server.actors
R[1.6.2.1][N][F]	Actorbase.server.utils
R[1.6.3.1][N][F]	Actorbase.server.utils
R[1.6.4.1][N][F]	Actorbase.server.utils
R[1.6.5.1][D][F]	Actorbase.server.utils
R[1.6.6.1][N][F]	Actorbase.server.utils
R[1.7][N][F] e figli	Actorbase.server.actors
R[1.7.2][N][F]	Actorbase.server.utils
R[1.7.3][N][F]	Actorbase.server.utils
R[1.7.4][N][F]	Actorbase.server.utils
R[1.7.5][N][F]	Actorbase.server.utils
R[1.7.6][N][F]	Actorbase.server.utils
R[1.7.7][N][F]	Actorbase.server.utils
R[2][N][F] e figli	Actorbase.client
R[3][N][F] e figli	Actorbase.driver
R[5][N][V]	Actorbase.server
	Actorbase.server.actors

Tabella 3: Requisito-componente

9 Appendice

9.1 Descrizione Design Pattern

Segue, per ogni Design Pattern utilizzato, la descrizione dello scopo, motivazione e applicabilità.

9.1.1 Broker event-driven

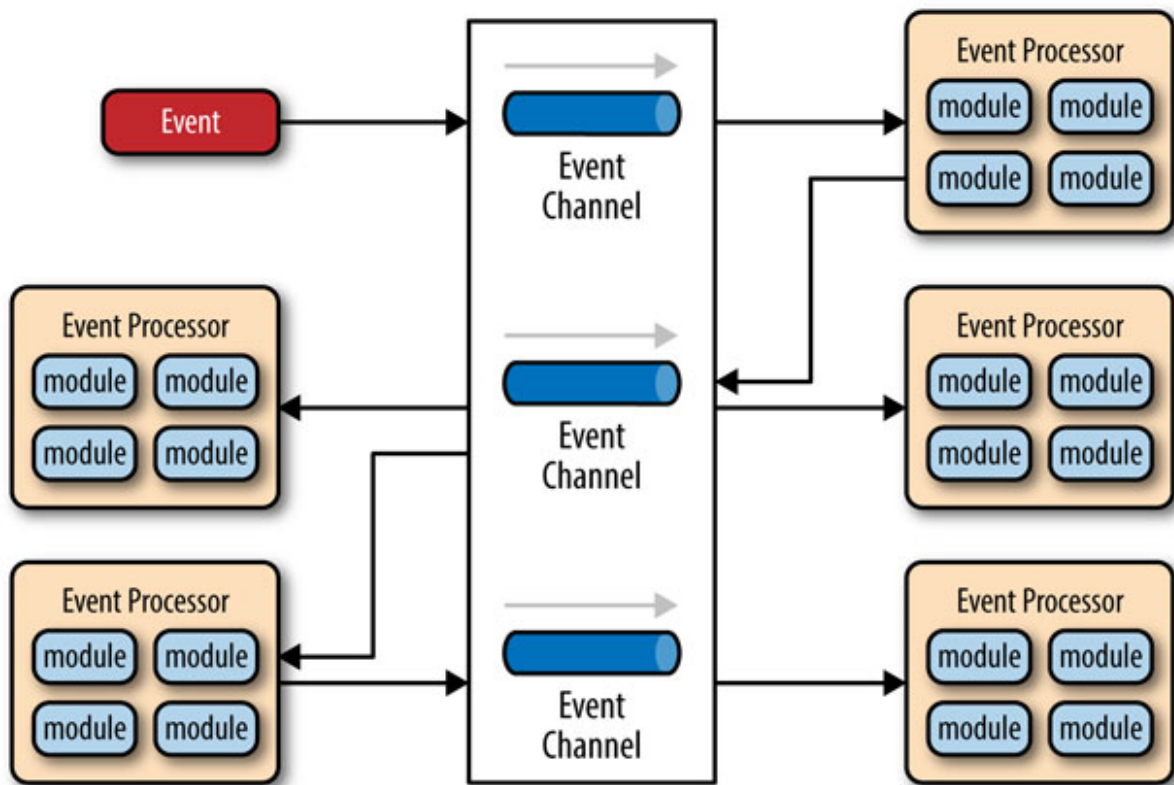


Figura 35: Diagramma del Design Pattern Event-driven

- **Scopo:** Produrre applicazioni molto scalabili e processare eventi asincroni disaccoppiati.
- **Motivazione:** Gestire le richieste che vengono volte all' applicativo tramite eventi processati in modo asincrono.
- **Applicabilità:** Gestione di eventi attraverso l'utilizzo di elaboratori di eventi e canali per passare gli eventi.

9.1.2 Singleton

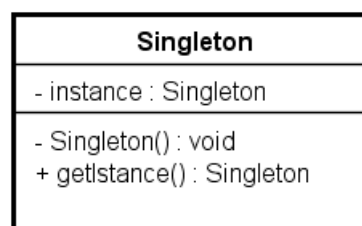


Figura 36: Diagramma del Design Pattern Command

- **Scopo:** Assicurare che una classe abbia una sola istanza con un unico punto di accesso globale.
- **Motivazione:** È necessario assicurare che esista una sola istanza di alcune classi. Una classe Singleton ha la responsabilità sulle proprie istanze, in modo che nessuna altra istanza possa essere creata, e fornisce un punto di accesso unico.
- **Applicabilità:**
 - Deve esistere una ed una sola istanza di una classe in tutta l'applicazione, accessibile dai client in modo noto.
 - L'istanza deve essere estendibile con ereditarietà, consentendo ai client di non modificare il proprio codice.

Elenco delle figure

1	Scala - logo	14
2	Akka - logo	14
3	Legenda	15
4	Architettura generale, vista Package	16
5	Componente Actorbase.server	17
6	Componente Actorbase.server.utils	19
7	Componente Actorbase.server.actors	21
8	Componente Actorbase.server.enums	26
9	Componente Actorbase.server.messages	32
10	Componente Actorbase.server.messages.internal	33
11	Componente Actorbase.server.messages.internal.AskMessages	34
12	Componente Actorbase.server.messages.internal.ScalabilityMessages	35
13	Componente Actorbase.server.messages.internal.LinkMessages	36
14	Componente Actorbase.server.messages.internal.StorageMessages	38
15	Componente Actorbase.server.messages.query	40
16	Componente Actorbase.server.messages.query.ErrorMessage	42
17	Componente Actorbase.server.messages.query.PermissionMessages	43
18	Componente Actorbase.server.messages.query.ReplyMessages	45
19	Componente Actorbase.server.messages.query.admin	47
20	Componente Actorbase.server.messages.query.admin.SettingMessages	48
21	Componente Actorbase.server.messages.query.admin.PermissionsManagementMessages	49
22	Componente Actorbase.server.messages.query.admin.UserManagementMessages	51
23	Componente Actorbase.server.messages.query.user	54
24	Componente Actorbase.server.messages.query.user.RowMessages	56
25	Componente Actorbase.server.messages.query.user.MapMessages	60
26	Componente Actorbase.server.messages.query.user.DatabaseMessages	63
27	Componente Actorbase.server.messages.query.user.HelpMessages	66
28	Componente Actorbase.client	68
29	Componente Actorbase.driver	69
30	Diagramma attività principale	72
31	Il sistema ad attori di Actorbase	73
32	Il sistema ad attori di Actorbase, struttura di un singolo database	74
33	Il sistema ad attori di Actorbase, struttura di una mappa in memoria principale	75
34	Il sistema ad attori di Actorbase, struttura degli attori di backup	76
35	Diagramma del Design Pattern Event-driven	79
36	Diagramma del Design Pattern Command	79

Elenco delle tabelle

1	Diario delle modifiche	12
2	Componenti-Requisiti	77
3	Requisito-componente	78