



Definizione di Prodotto

Informazioni sul documento

Versione	2.0.0
Redazione	Andrea Giacomo Baldan Alberto De Agostini Marco Boseggia
Verifica	Francesco Agostini Michael Munaro
Approvazione	Davide Trevisan
Uso	Esterno
Lista di Distribuzione	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
2.0.0	Davide Trevisan	Responsabile	2016-06-09	Approvazione documento
1.4.0	Francesco Agostini	Verificatore	2016-06-08	Verifica diagrammi sequenza (sezione 4)
1.3.1	Alberto De Agostini	Progettista	2016-06-07	Aggiornamento diagrammi sequenza
1.3.0	Michael Munaro	Verificatore	2016-06-06	Verifica miglioramento sezioni driver
1.2.1	Marco Boseggia	Progettista	2016-06-05	Miglioramento delle descrizioni e spiegazioni di tutte le classi delle sezioni riguardanti package driver (da sezione 3.26 a sezione 3.29)
1.2.0	Francesco Agostini	Verificatore	2016-06-05	Verifica miglioramento sezioni driver
1.1.1	Andrea Baldan	Progettista	2016-06-04	Miglioramento delle descrizioni e spiegazioni di tutte le classi delle sezioni riguardanti package driver (da sezione 3.21 a sezione 3.25)
1.1.0	Michael Munaro	Verificatore	2016-06-03	Verifica miglioramento sezioni server
1.0.2	Alberto De Agostini	Progettista	2016-06-03	Miglioramento delle descrizioni e spiegazioni di tutte le classi delle sezioni riguardanti package server (da sezione 3.1 a sezione 3.20)
1.0.1	Alberto De Agostini	Progettista	2016-05-30	Modifica degli attributi immutabili con aggiunta di metodi Getter in tutto il documento come richiesto in post RQ
1.0.0	Michael Munaro	Responsabile	2016-05-14	Approvazione documento
0.3.0	Alberto De Agostini	Verificatore	2016-05-13	Verifica incrementi
0.2.2	Marco Boseggia	Progettista	2016-05-12	Aggiunta tracciamento classi-requisiti classi
0.2.1	Davide Trevisan	Progettista	2016-05-09	Spostamento diagrammi di sequenza da Specifica Tecnica
0.2.0	Francesco Agostini	Verificatore	2016-05-04	Verifica sezioni CLI
0.1.1	Alberto De Agostini	Progettista	2016-05-03	Stesura sezioni riguardanti il package CLI (da 3.26 a 3.29)
0.1.0	Alberto De Agostini	Verificatore	2016-05-04	Verifica sezioni 2 e driver e server



0.0.4	Andrea Giacomo Baldan	Progettista	2016-05-03	Stesura sezioni riguardanti il package driver (da 3.21 a 3.25)
0.0.3	Marco Boseggia	Progettista	2016-04-29	Stesura sezioni riguardanti il package server (da 3.1 a 3.20)
0.0.2	Davide Trevisan	Progettista	2016-04-29	Stesura sezione 2
0.0.1	Andrea Giacomo Baldan	Progettista	2016-04-20	Creazione scheletro del documento

Indice

1 Sommario	1
1.1 Scopo del documento	1
1.2 Scopo del Prodotto	1
1.3 Glossario	1
1.4 Riferimenti	1
1.4.1 Normativi	1
1.4.2 Informativi	1
2 Standard di progetto	3
2.1 Standard di progettazione architetturale	3
2.2 Standard di documentazione del codice	3
2.3 Standard di programmazione	3
2.4 Strumenti di lavoro	3
3 Componenti	4
3.1 actorbase	5
3.1.1 Descrizione	6
3.1.2 Package contenuti	6
3.2 actorbase::actorsystem	7
3.2.1 Descrizione	7
3.2.2 Package contenuti	7
3.3 actorbase::actorsystem::utils	8
3.3.1 Descrizione	8
3.3.2 Interfacce	9
3.3.2.1 Permissions	9
3.3.2.1.1 Descrizione	9
3.3.2.1.2 Utilizzo	9
3.3.2.1.3 Realizzata da	9
3.3.3 Classi	9
3.3.3.1 Read	9
3.3.3.1.1 Descrizione	9
3.3.3.1.2 Utilizzo	10
3.3.3.1.3 Realizza	10
3.3.3.2 ReadWrite	10
3.3.3.2.1 Descrizione	10
3.3.3.2.2 Utilizzo	10
3.3.3.2.3 Realizza	10
3.3.3.3 CryptoUtils	11
3.3.3.3.1 Descrizione	11
3.3.3.3.2 Utilizzo	11
3.3.3.3.3 Attributi	11
3.3.3.3.4 Metodi	12
3.3.3.3.5 Parametri	12



3.3.3.4	ActorbaseCollection	13
3.3.3.4.1	Descrizione	13
3.3.3.4.2	Utilizzo	13
3.3.3.4.3	Realizza	13
3.3.3.4.4	Attributi	14
3.3.3.4.5	Metodi	14
3.3.3.4.6	Parametri	14
3.4	actorbase::actorsystem::messages	16
3.4.1	Descrizione	16
3.4.2	Package contenuti	16
3.5	actorbase::actorsystem::messages::authactormessages	17
3.5.1	Descrizione	18
3.5.2	Classi	18
3.5.2.1	actorbase::actorsystem::messages::authactormessages::AddCredentials	18
3.5.2.1.1	Descrizione	18
3.5.2.1.2	Utilizzo	18
3.5.2.1.3	Attributi	18
3.5.2.1.4	Metodi	19
3.5.2.2	actorbase::actorsystem::messages::authactormessages::UpdateCredentials	19
3.5.2.2.1	Descrizione	19
3.5.2.2.2	Utilizzo	19
3.5.2.2.3	Attributi	19
3.5.2.2.4	Metodi	20
3.5.2.3	actorbase::actorsystem::messages::authactormessages::RemoveCredentials	20
3.5.2.3.1	Descrizione	20
3.5.2.3.2	Utilizzo	20
3.5.2.3.3	Attributi	20
3.5.2.3.4	Metodi	20
3.5.2.4	actorbase::actorsystem::messages::authactormessages::Authenticate	21
3.5.2.4.1	Descrizione	21
3.5.2.4.2	Utilizzo	21
3.5.2.4.3	Attributi	21
3.5.2.4.4	Metodi	21
3.5.2.5	actorbase::actorsystem::messages::authactormessages::AddCollectionTo	22
3.5.2.5.1	Descrizione	22
3.5.2.5.2	Utilizzo	22
3.5.2.5.3	Attributi	22
3.5.2.5.4	Metodi	22
3.5.2.6	actorbase::actorsystem::messages::authactormessages::RemoveCollectionFrom	23
3.5.2.6.1	Descrizione	23
3.5.2.6.2	Utilizzo	23
3.5.2.6.3	Attributi	23
3.5.2.6.4	Metodi	24
3.5.2.7	actorbase::actorsystem::messages::authactormessages::ListCollectionsOf	24
3.5.2.7.1	Descrizione	24



3.5.2.7.2	Utilizzo	24
3.5.2.7.3	Attributi	24
3.5.2.7.4	Metodi	25
3.6	actorbase::actorsystem::messages::clientactormessages	25
3.6.1	Descrizione	25
3.6.2	Classi	26
3.6.2.1	actorbase::actorsystem::messages::clientactormessages::ListResponse	26
3.6.2.1.1	Descrizione	26
3.6.2.1.2	Utilizzo	26
3.6.2.1.3	Attributi	26
3.6.2.1.4	Metodi	26
3.6.2.2	actorbase::actorsystem::messages::clientactormessages::MapResponse	27
3.6.2.2.1	Descrizione	27
3.6.2.2.2	Utilizzo	27
3.6.2.2.3	Attributi	27
3.6.2.2.4	Metodi	27
3.6.2.3	actorbase::actorsystem::messages::clientactormessages::Response	28
3.6.2.3.1	Descrizione	28
3.6.2.3.2	Utilizzo	28
3.6.2.3.3	Attributi	28
3.6.2.3.4	Metodi	28
3.7	actorbase::actorsystem::messages::mainmessages	29
3.7.1	Descrizione	29
3.7.2	Classi	30
3.7.2.1	actorbase::actorsystem::messages::mainmessages::InsertTo	30
3.7.2.1.1	Descrizione	30
3.7.2.1.2	Utilizzo	30
3.7.2.1.3	Attributi	31
3.7.2.1.4	Metodi	31
3.7.2.2	actorbase::actorsystem::messages::mainmessages::GetFrom	31
3.7.2.2.1	Descrizione	31
3.7.2.2.2	Utilizzo	32
3.7.2.2.3	Attributi	32
3.7.2.2.4	Metodi	32
3.7.2.3	actorbase::actorsystem::messages::mainmessages::CompleteTransaction	32
3.7.2.3.1	Descrizione	32
3.7.2.3.2	Utilizzo	33
3.7.2.3.3	Attributi	33
3.7.2.3.4	Metodi	33
3.7.2.4	actorbase::actorsystem::messages::mainmessages::RemoveFrom	33
3.7.2.4.1	Descrizione	34
3.7.2.4.2	Utilizzo	34
3.7.2.4.3	Attributi	34
3.7.2.4.4	Metodi	34
3.7.2.5	actorbase::actorsystem::messages::mainmessages::AddContributor	34



3.7.2.5.1	Descrizione	35
3.7.2.5.2	Utilizzo	35
3.7.2.5.3	Attributi	35
3.7.2.5.4	Metodi	35
3.7.2.6	actorbase::actorsystem::messages::mainmessages::RemoveContributor	35
3.7.2.6.1	Descrizione	36
3.7.2.6.2	Utilizzo	36
3.7.2.6.3	Attributi	36
3.7.2.6.4	Metodi	36
3.7.2.7	actorbase::actorsystem::messages::mainmessages::CreateCollection	36
3.7.2.7.1	Descrizione	37
3.7.2.7.2	Utilizzo	37
3.7.2.7.3	Attributi	37
3.7.2.7.4	Metodi	37
3.8	actorbase::actorsystem::messages::storefindermessages	38
3.8.1	Descrizione	38
3.8.2	Classi	39
3.8.2.1	actorbase::actorsystem::messages::storefinder::GetAllItems	39
3.8.2.1.1	Descrizione	39
3.8.2.1.2	Utilizzo	39
3.8.2.2	actorbase::actorsystem::messages::storefinder::UpdateCollectionSize	39
3.8.2.2.1	Descrizione	39
3.8.2.2.2	Utilizzo	40
3.8.2.2.3	Attributi	40
3.8.2.2.4	Metodi	40
3.8.2.3	actorbase::actorsystem::messages::storefinder::Get	40
3.8.2.3.1	Descrizione	40
3.8.2.3.2	Utilizzo	40
3.8.2.3.3	Attributi	40
3.8.2.3.4	Metodi	41
3.8.2.4	actorbase::actorsystem::messages::storefinder::PartialMapTransaction	41
3.8.2.4.1	Descrizione	41
3.8.2.4.2	Utilizzo	41
3.8.2.4.3	Attributi	41
3.8.2.4.4	Metodi	42
3.8.2.5	actorbase::actorsystem::messages::storefinder::Remove	42
3.8.2.5.1	Descrizione	42
3.8.2.5.2	Utilizzo	42
3.8.2.5.3	Attributi	42
3.8.2.5.4	Metodi	42
3.8.2.6	actorbase::actorsystem::messages::storefinder::Insert	43
3.8.2.6.1	Descrizione	43
3.8.2.6.2	Utilizzo	43
3.8.2.6.3	Attributi	43
3.8.2.6.4	Metodi	44



3.9 actorbase::actorsystem::messages::storekeepermessages	44
3.9.1 Descrizione	44
3.9.2 Classi	44
3.9.2.1 actorbase::actorsystem::messages::storekeeper::Persist	44
3.9.2.1.1 Descrizione	45
3.9.2.1.2 Utilizzo	45
3.9.2.2 actorbase::actorsystem::messages::storekeeper::InitMn	45
3.9.2.2.1 Descrizione	45
3.9.2.2.2 Utilizzo	45
3.9.2.2.3 Attributi	45
3.9.2.2.4 Metodi	46
3.9.2.3 actorbase::actorsystem::messages::storekeeper::GetAll	46
3.9.2.3.1 Descrizione	46
3.9.2.3.2 Utilizzo	46
3.9.2.3.3 Attributi	46
3.9.2.3.4 Metodi	46
3.9.2.4 actorbase::actorsystem::messages::storekeeper::GetItem	47
3.9.2.4.1 Descrizione	47
3.9.2.4.2 Utilizzo	47
3.9.2.4.3 Attributi	47
3.9.2.4.4 Metodi	47
3.9.2.5 actorbase::actorsystem::messages::storekeeper::InsertItem	48
3.9.2.5.1 Descrizione	48
3.9.2.5.2 Utilizzo	48
3.9.2.5.3 Attributi	48
3.9.2.5.4 Metodi	49
3.9.2.6 actorbase::actorsystem::messages::storekeeper::RemoveItem	49
3.9.2.6.1 Descrizione	49
3.9.2.6.2 Utilizzo	49
3.9.2.6.3 Attributi	49
3.9.2.6.4 Metodi	49
3.10 actorbase::actorsystem::messages::warehousemanmessages	50
3.10.1 Descrizione	50
3.10.2 Classi	50
3.10.2.1 actorbase::actorsystem::messages::warehouseman::Clean	50
3.10.2.1.1 Descrizione	50
3.10.2.1.2 Utilizzo	51
3.10.2.2 actorbase::actorsystem::messages::warehouseman::Init	51
3.10.2.2.1 Descrizione	51
3.10.2.2.2 Utilizzo	51
3.10.2.2.3 Attributi	51
3.10.2.2.4 Metodi	52
3.10.2.3 actorbase::actorsystem::messages::warehouseman::Save	52
3.10.2.3.1 Descrizione	52
3.10.2.3.2 Utilizzo	52



3.10.2.3.3 Attributi	52
3.10.2.3.4 Metodi	52
3.10.2.4 actorbase::actorsystem::messages::warehouseman::Read	53
3.10.2.4.1 Descrizione	53
3.10.2.4.2 Utilizzo	53
3.10.2.4.3 Attributi	53
3.10.2.4.4 Metodi	53
3.11 actorbase::actorsystem::messages::managermessages	54
3.11.1 Descrizione	54
3.11.2 Classi	54
3.11.2.1 actorbase::actorsystem::messages::managermessages::OneMore	54
3.11.2.1.1 Descrizione	54
3.11.2.1.2 Utilizzo	55
3.12 actorbase::actorsystem::actors	55
3.12.1 Descrizione	55
3.12.2 Package contenuti	56
3.13 actorbase::actorsystem::actors::clientactor	56
3.13.1 Descrizione	56
3.13.2 Classi	57
3.13.2.1 actorbase::actorsystem::actors::clientactor::Authenticator	57
3.13.2.1.1 Descrizione	57
3.13.2.1.2 Utilizzo	57
3.13.2.1.3 Metodi	57
3.13.2.1.4 Parametri	57
3.13.2.2 actorbase::actorsystem::actors::clientactor::User	58
3.13.2.2.1 Descrizione	58
3.13.2.2.2 Utilizzo	58
3.13.2.2.3 Attributi	59
3.13.2.2.4 Metodi	59
3.13.2.3 actorbase::actorsystem::actors::clientactor::AuthInfo	59
3.13.2.3.1 Descrizione	59
3.13.2.3.2 Utilizzo	59
3.13.2.3.3 Attributi	59
3.13.2.3.4 Metodi	60
3.13.2.4 actorbase::actorsystem::actors::clientactor::CollectionApi	60
3.13.2.4.1 Descrizione	60
3.13.2.4.2 Utilizzo	60
3.13.2.4.3 Attributi	60
3.13.2.4.4 Metodi	60
3.13.2.4.5 Parametri	61
3.13.2.5 actorbase::actorsystem::actors::clientactor::ClientActor	61
3.13.2.5.1 Descrizione	61
3.13.2.5.2 Utilizzo	61
3.13.2.5.3 Classi ereditate	61
3.13.2.5.4 Attributi	62



3.13.2.5.5 Metodi	62
3.14 actorbase::actorsystem::actors::httpserver	63
3.14.1 Descrizione	64
3.14.2 Interfacce	64
3.14.2.1 actorbase::actorsystem::actors::httpserver::SSLConfiguration	64
3.14.2.1.1 Descrizione	64
3.14.2.1.2 Utilizzo	64
3.14.2.1.3 Realizzata da	64
3.14.2.1.4 Metodi	65
3.14.3 Classi	65
3.14.3.1 actorbase::actorsystem::actors::httpserver::HTTPServer	65
3.14.3.1.1 Descrizione	65
3.14.3.1.2 Utilizzo	65
3.14.3.1.3 Classi ereditate	65
3.14.3.1.4 Attributi	66
3.14.3.1.5 Metodi	66
3.15 actorbase::actorsystem::actors::warehouseman	67
3.15.1 Descrizione	67
3.15.2 Classi	68
3.15.2.1 actorbase::actorsystem::actors::warehouseman::Warehouseman	68
3.15.2.1.1 Descrizione	68
3.15.2.1.2 Utilizzo	68
3.15.2.1.3 Classi ereditate	68
3.15.2.1.4 Attributi	68
3.15.2.1.5 Metodi	69
3.16 actorbase::actorsystem::actors::main	69
3.16.1 Descrizione	69
3.16.2 Classi	70
3.16.2.1 actorbase::actorsystem::actors::main::Main	70
3.16.2.1.1 Descrizione	70
3.16.2.1.2 Utilizzo	70
3.16.2.1.3 Classi ereditate	70
3.16.2.1.4 Attributi	71
3.16.2.1.5 Metodi	71
3.17 actorbase::actorsystem::actors::authactor	72
3.17.1 Descrizione	72
3.17.2 Classi	73
3.17.2.1 actorbase::actorsystem::actors::authactor::AuthActor	73
3.17.2.1.1 Descrizione	73
3.17.2.1.2 Utilizzo	73
3.17.2.1.3 Classi ereditate	73
3.17.2.1.4 Attributi	73
3.17.2.1.5 Metodi	74
3.17.2.1.6 Parametri	74
3.17.2.2 actorbase::actorsystem::actors::authactor::Profile	74



3.17.2.2.1 Descrizione	75
3.17.2.2 Utilizzo	75
3.17.2.3 Attributi	75
3.17.2.4 Metodi	75
3.17.2.5 Parametri	75
3.18 actorbase::actorsystem::actors::storefinder	77
3.18.1 Descrizione	77
3.18.2 Classi	78
3.18.2.1 actorbase::actorsystem::actors::storefinder::Storefinder	78
3.18.2.1.1 Descrizione	78
3.18.2.1.2 Utilizzo	78
3.18.2.1.3 Classi ereditate	78
3.18.2.1.4 Attributi	79
3.18.2.1.5 Metodi	79
3.19 actorbase::actorsystem::actors::storekeeper	80
3.19.1 Descrizione	80
3.19.2 Classi	81
3.19.2.1 actorbase::actorsystem::actors::storekeeper::Storekeeper	81
3.19.2.1.1 Descrizione	81
3.19.2.1.2 Utilizzo	81
3.19.2.1.3 Classi ereditate	81
3.19.2.1.4 Attributi	82
3.19.2.1.5 Metodi	82
3.19.2.1.6 Parametri	82
3.20 actorbase::actorsystem::actors::manager	83
3.20.1 Descrizione	83
3.20.2 Classi	84
3.20.2.1 actorbase::actorsystem::actors::manager::Manager	84
3.20.2.1.1 Descrizione	84
3.20.2.1.2 Utilizzo	84
3.20.2.1.3 Classi ereditate	84
3.20.2.1.4 Attributi	84
3.20.2.1.5 Metodi	85
3.21 actorbase::driver	85
3.21.1 Descrizione	85
3.21.2 Package contenuti	85
3.21.3 Classi	86
3.21.3.1 ActorbaseDriver	86
3.21.3.1.1 Descrizione	86
3.21.3.1.2 Utilizzo	86
3.21.3.1.3 Eredita	86
3.21.3.1.4 Attributi	86
3.21.3.1.5 Metodi	86
3.21.3.1.6 Parametri	87
3.21.3.2 ActorbaseServices	87



3.21.3.2.1 Descrizione	87
3.21.3.2.2 Utilizzo	88
3.21.3.2.3 Eredita	88
3.21.3.2.4 Attributi	88
3.21.3.2.5 Metodi	89
3.21.3.2.6 Parametri	89
3.21.3.3 ActorbaseAdminServices (abstract)	91
3.21.3.3.1 Descrizione	91
3.21.3.3.2 Utilizzo	91
3.21.3.3.3 Eredita	92
3.21.3.3.4 Metodi	92
3.21.3.3.5 Parametri	92
3.21.3.4 Connection	93
3.21.3.4.1 Descrizione	93
3.21.3.4.2 Utilizzo	93
3.21.3.4.3 Attributi	93
3.21.3.4.4 Metodi	94
3.22 actorbase::driver::client	94
3.22.1 Descrizione	94
3.22.2 Interfacce	95
3.22.2.1 Client	95
3.22.2.1.1 Descrizione	95
3.22.2.1.2 Utilizzo	95
3.22.2.1.3 Ereditata da	95
3.22.2.1.4 Realizzata da	95
3.22.2.1.5 Metodi	95
3.22.2.1.6 Parametri	96
3.22.3 Classi	96
3.22.3.1 Serializer (abstract)	96
3.22.3.1.1 Descrizione	96
3.22.3.1.2 Utilizzo	96
3.22.3.1.3 Ereditata da	96
3.22.3.1.4 Metodi	97
3.22.3.1.5 Parametri	97
3.22.3.2 SSLClient (abstract)	98
3.22.3.2.1 Descrizione	98
3.22.3.2.2 Utilizzo	98
3.22.3.2.3 Realizzata da	98
3.22.3.2.4 Metodi	98
3.22.3.2.5 Parametri	98
3.22.3.3 ActorbaseClient	99
3.22.3.3.1 Descrizione	99
3.22.3.3.2 Utilizzo	99
3.22.3.3.3 Interfacce realizzate	99
3.22.3.3.4 Classi ereditate	99



3.22.3.3.5 Attributi	100
3.22.3.3.6 Metodi	100
3.22.3.4 Connector (abstract)	100
3.22.3.4.1 Descrizione	100
3.22.3.4.2 Utilizzo	100
3.22.3.4.3 Attributi	101
3.22.3.4.4 Eredita	101
3.22.3.4.5 Ereditata da	101
3.23 actorbase::driver::client::api	102
3.23.1 Descrizione	102
3.23.2 Classi	103
3.23.2.1 RequestMethod (abstract)	103
3.23.2.1.1 Descrizione	103
3.23.2.1.2 Utilizzo	103
3.23.2.1.3 Ereditata da	103
3.23.2.2 GET	103
3.23.2.2.1 Descrizione	103
3.23.2.2.2 Utilizzo	104
3.23.2.2.3 Eredita	104
3.23.2.3 POST	104
3.23.2.3.1 Descrizione	104
3.23.2.3.2 Utilizzo	104
3.23.2.3.3 Eredita	104
3.23.2.4 PUT	104
3.23.2.4.1 Descrizione	105
3.23.2.4.2 Utilizzo	105
3.23.2.4.3 Eredita	105
3.23.2.5 DELETE	105
3.23.2.5.1 Descrizione	105
3.23.2.5.2 Utilizzo	105
3.23.2.5.3 Eredita	105
3.23.2.6 Request	106
3.23.2.6.1 Descrizione	106
3.23.2.6.2 Utilizzo	106
3.23.2.6.3 Attributi	106
3.23.2.6.4 Metodi	107
3.23.2.7 Credentials	107
3.23.2.7.1 Descrizione	107
3.23.2.7.2 Utilizzo	107
3.23.2.7.3 Attributi	107
3.23.2.7.4 Metodi	108
3.23.2.8 Response	108
3.23.2.8.1 Descrizione	108
3.23.2.8.2 Utilizzo	108
3.23.2.8.3 Attributi	108



3.23.2.8.4 Metodi	109
3.23.2.9 Status	109
3.23.2.9.1 Descrizione	109
3.23.2.9.2 Utilizzo	109
3.23.2.9.3 Attributi	109
3.23.2.10 RequestBuilder	110
3.23.2.10.1 Descrizione	110
3.23.2.10.2 Utilizzo	110
3.23.2.10.3 Attributi	111
3.23.2.10.4 Metodi	111
3.23.2.10.5 Parametri	111
3.24 actorbase::driver::data	112
3.24.1 Descrizione	113
3.24.2 Classi	113
3.24.2.1 ActorbaseObject<T>	113
3.24.2.1.1 Descrizione	113
3.24.2.1.2 Utilizzo	113
3.24.2.1.3 Attributi	113
3.24.2.1.4 Metodi	114
3.24.2.1.5 Parametri	114
3.24.2.2 actorbase::driver::data::ActorbaseCollection	116
3.24.2.2.1 Descrizione	116
3.24.2.2.2 Utilizzo	116
3.24.2.2.3 Eredita	117
3.24.2.2.4 Attributi	117
3.24.2.2.5 Metodi	118
3.24.2.2.6 Parametri	118
3.24.2.3 actorbase::driver::data::ActorbaseCollectionMap	121
3.24.2.3.1 Descrizione	121
3.24.2.3.2 Utilizzo	121
3.24.2.3.3 Eredita	121
3.24.2.3.4 Attributi	121
3.24.2.3.5 Metodi	122
3.24.2.3.6 Parametri	122
3.25 actorbase::driver::exceptions	123
3.25.1 Descrizione	123
3.25.2 Interazione con altre componenti	123
3.25.3 Classi	123
3.25.3.1 actorbase::driver::exceptions::WrongCredentialExc	123
3.25.3.1.1 Descrizione	123
3.25.3.1.2 Utilizzo	124
3.25.3.2 actorbase::driver::exceptions::WrongPasswordExc	124
3.25.3.2.1 Descrizione	124
3.25.3.2.2 Utilizzo	124
3.25.3.3 actorbase::driver::exceptions::WrongNewPasswordExc	124



3.25.3.3.1 Descrizione	124
3.25.3.3.2 Utilizzo	124
3.25.3.4 actorbase::driver::exceptions::CollectionAlreadyExistsExc	125
3.25.3.4.1 Descrizione	125
3.25.3.4.2 Utilizzo	125
3.25.3.5 actorbase::driver::exceptions::UndefinedCollectionExc	125
3.25.3.5.1 Descrizione	125
3.25.3.5.2 Utilizzo	125
3.25.3.6 actorbase::driver::exceptions::UndefinedUsernameExc	126
3.25.3.6.1 Descrizione	126
3.25.3.6.2 Utilizzo	126
3.25.3.7 actorbase::driver::exceptions::UsernameAlreadyExistsExc	126
3.25.3.7.1 Descrizione	126
3.25.3.7.2 Utilizzo	126
3.25.3.8 actorbase::driver::exceptions::DuplicateKeyExc	127
3.25.3.8.1 Descrizione	127
3.25.3.8.2 Utilizzo	127
3.25.3.9 actorbase::driver::exceptions::UndefinedFileExc	127
3.25.3.9.1 Descrizione	127
3.25.3.9.2 Utilizzo	127
3.25.3.10 actorbase::driver::exceptions::MalformedFileExc	128
3.25.3.10.1 Descrizione	128
3.25.3.10.2 Utilizzo	128
3.26 actorbase::cli	129
3.26.1 Descrizione	130
3.26.2 Interazioni con altre componenti	130
3.26.3 Package contenuti	130
3.27 actorbase::cli::views	132
3.27.1 Descrizione	133
3.27.2 Interfacce	133
3.27.2.1 actorbase::cli::views::PromptProvider	133
3.27.2.1.1 Descrizione	133
3.27.2.1.2 Utilizzo	133
3.27.2.1.3 Realizzata da	133
3.27.2.1.4 Metodi	133
3.27.2.2 actorbase::cli::views::Observer	134
3.27.2.2.1 Descrizione	134
3.27.2.2.2 Utilizzo	134
3.27.2.2.3 Realizzata da	134
3.27.2.2.4 Metodi	134
3.27.2.2.5 Parametri	134
3.27.3 Classi	135
3.27.3.1 actorbase::cli::views::CommandLoop	135
3.27.3.1.1 Descrizione	135
3.27.3.1.2 Utilizzo	135



3.27.3.1.3 Attributi	135
3.27.3.2 actorbase::cli::views::ActorbaseBanner	136
3.27.3.2.1 Descrizione	136
3.27.3.2.2 Utilizzo	136
3.27.3.2.3 Metodi	136
3.27.3.3 actorbase::cli::views::ActorbasePrompt	136
3.27.3.3.1 Descrizione	136
3.27.3.3.2 Utilizzo	136
3.27.3.3.3 Classi ereditate	137
3.27.3.3.4 Metodi	137
3.27.3.3.5 Attributi	137
3.27.3.4 actorbase::cli::views::ResultViewer	137
3.27.3.4.1 Descrizione	137
3.27.3.4.2 Utilizzo	137
3.27.3.4.3 Realizza	137
3.27.3.4.4 Metodi	138
3.27.3.4.5 Parametri	138
3.28 actorbase::cli::controllers	139
3.28.1 Descrizione	140
3.28.2 Classi	140
3.28.2.1 actorbase::cli::controllers::GrammarParser	140
3.28.2.1.1 Descrizione	140
3.28.2.1.2 Utilizzo	140
3.28.2.1.3 Attributi	141
3.28.2.1.4 Metodi	141
3.28.2.1.5 Parametri	142
3.29 actorbase::cli::models	143
3.29.1 Descrizione	144
3.29.2 Interfacce	144
3.29.2.1 actorbase::cli::models::Command	144
3.29.2.1.1 Descrizione	144
3.29.2.1.2 Utilizzo	144
3.29.2.1.3 Realizzata da	144
3.29.2.1.4 Metodi	145
3.29.3 Classi	145
3.29.3.1 actorbase::cli::models::Observable (abstract)	145
3.29.3.1.1 Descrizione	146
3.29.3.1.2 Utilizzo	146
3.29.3.1.3 Realizzata da	146
3.29.3.1.4 Attributi	146
3.29.3.1.5 Metodi	146
3.29.3.1.6 Parametri	146
3.29.3.2 actorbase::cli::models::CommandInvoker	147
3.29.3.2.1 Descrizione	147
3.29.3.2.2 Utilizzo	147



3.29.3.2.3	Realizza	147
3.29.3.2.4	Attributi	147
3.29.3.2.5	Metodi	147
3.29.3.2.6	Parametri	148
3.29.3.3	actorbase::cli::models::ChangePasswordCommand	148
3.29.3.3.1	Descrizione	148
3.29.3.3.2	Utilizzo	148
3.29.3.3.3	Classi ereditate	148
3.29.3.3.4	Attributi	148
3.29.3.3.5	Metodi	149
3.29.3.4	actorbase::cli::models::FindCommand	149
3.29.3.4.1	Descrizione	149
3.29.3.4.2	Utilizzo	149
3.29.3.4.3	Classi ereditate	149
3.29.3.4.4	Attributi	149
3.29.3.4.5	Metodi	149
3.29.3.5	actorbase::cli::models::HelpCommand	150
3.29.3.5.1	Descrizione	150
3.29.3.5.2	Utilizzo	150
3.29.3.5.3	Classi ereditate	150
3.29.3.5.4	Attributi	150
3.29.3.5.5	Metodi	150
3.29.3.6	actorbase::cli::models::LogoutCommand	151
3.29.3.6.1	Descrizione	151
3.29.3.6.2	Utilizzo	151
3.29.3.6.3	Classi ereditate	151
3.29.3.6.4	Attributi	151
3.29.3.6.5	Metodi	151
3.29.3.7	actorbase::cli::models::ResetPasswordCommand	152
3.29.3.7.1	Descrizione	152
3.29.3.7.2	Utilizzo	152
3.29.3.7.3	Classi ereditate	152
3.29.3.7.4	Attributi	152
3.29.3.7.5	Metodi	152
3.29.3.8	actorbase::cli::models::ListCollectionsCommand	153
3.29.3.8.1	Descrizione	153
3.29.3.8.2	Utilizzo	153
3.29.3.8.3	Classi ereditate	153
3.29.3.8.4	Attributi	153
3.29.3.8.5	Metodi	153
3.29.3.9	actorbase::cli::models::AddContributorCommand	154
3.29.3.9.1	Descrizione	154
3.29.3.9.2	Utilizzo	154
3.29.3.9.3	Classi ereditate	154
3.29.3.9.4	Attributi	154



3.29.3.9.5 Metodi	154
3.29.3.10 actorbase::cli::models::LoginCommand	155
3.29.3.10.1 Descrizione	155
3.29.3.10.2 Utilizzo	155
3.29.3.10.3 Classi ereditate	155
3.29.3.10.4 Attributi	155
3.29.3.10.5 Metodi	155
3.29.3.11 actorbase::cli::models::RemoveCollectionCommand	156
3.29.3.11.1 Descrizione	156
3.29.3.11.2 Utilizzo	156
3.29.3.11.3 Classi ereditate	156
3.29.3.11.4 Attributi	156
3.29.3.11.5 Metodi	156
3.29.3.12 actorbase::cli::models::RemoveContributorCommand	157
3.29.3.12.1 Descrizione	157
3.29.3.12.2 Utilizzo	157
3.29.3.12.3 Classi ereditate	157
3.29.3.12.4 Attributi	157
3.29.3.12.5 Metodi	157
3.29.3.13 actorbase::cli::models::RenameCollectionCommand	158
3.29.3.13.1 Descrizione	158
3.29.3.13.2 Utilizzo	158
3.29.3.13.3 Classi ereditate	158
3.29.3.13.4 Attributi	158
3.29.3.13.5 Metodi	158
3.29.3.14 actorbase::cli::models::AddUserCommand	159
3.29.3.14.1 Descrizione	159
3.29.3.14.2 Utilizzo	159
3.29.3.14.3 Classi ereditate	159
3.29.3.14.4 Attributi	159
3.29.3.14.5 Metodi	159
3.29.3.15 actorbase::cli::models::RemoveItemCommand	160
3.29.3.15.1 Descrizione	160
3.29.3.15.2 Utilizzo	160
3.29.3.15.3 Classi ereditate	160
3.29.3.15.4 Attributi	160
3.29.3.15.5 Metodi	160
3.29.3.16 actorbase::cli::models::RemoveUserCommand	161
3.29.3.16.1 Descrizione	161
3.29.3.16.2 Utilizzo	161
3.29.3.16.3 Classi ereditate	161
3.29.3.16.4 Attributi	161
3.29.3.16.5 Metodi	161
3.29.3.17 actorbase::cli::models::ImportCommand	162
3.29.3.17.1 Descrizione	162



3.29.3.17.2 Utilizzo	162
3.29.3.17.3 Classi ereditate	162
3.29.3.17.4 Attributi	162
3.29.3.17.5 Metodi	162
3.29.3.18 actorbase::cli::models::InsertItemCommand	163
3.29.3.18.1 Descrizione	163
3.29.3.18.2 Utilizzo	163
3.29.3.18.3 Classi ereditate	163
3.29.3.18.4 Attributi	163
3.29.3.18.5 Metodi	163
3.29.3.19 actorbase::cli::models::CreateCollectionCommand	164
3.29.3.19.1 Descrizione	164
3.29.3.19.2 Utilizzo	164
3.29.3.19.3 Classi ereditate	164
3.29.3.19.4 Attributi	164
3.29.3.19.5 Metodi	164
3.29.3.20 actorbase::cli::models::ExportCommand	165
3.29.3.20.1 Descrizione	165
3.29.3.20.2 Utilizzo	165
3.29.3.20.3 Classi ereditate	165
3.29.3.20.4 Attributi	165
3.29.3.20.5 Metodi	165
3.29.3.21 actorbase::cli::models::CommandReceiver	166
3.29.3.21.1 Descrizione	166
3.29.3.21.2 Utilizzo	166
3.29.3.21.3 Interazioni con altre classi	167
3.29.3.21.4 Attributi	167
3.29.3.21.5 Metodi	168
4 Diagrammi di Sequenza	169
4.1 Interazioni generali tra componenti	169
4.2 Inserimento di un Item	170
4.3 Inserimento di un Item con Storekeeper pieno	170
4.4 Autenticazione	171
4.5 Salvataggio su filesystem	172
4.6 Aggiunta collaboratore (in sola lettura)	173
5 Tracciamento	175
6 Tracciamento Classi-Requisiti	175
7 Tracciamento Requisiti-Classi	191



1 Sommario

1.1 Scopo del documento

Il seguente documento ha lo scopo di definire nel dettaglio la stesura del prodotto **Actorbase** in tutte e tre le sue componenti, precedentemente identificate nella *Specifica Tecnica* e approfondendone gli aspetti implementativi in modo da fornire una struttura dettagliata ai *programmatori* per l'attività di codifica.

1.2 Scopo del Prodotto

Implementazione di un database NoSQL_G di tipo key-value_G orientato alla gestione di grandi moli di dati utilizzando il modello ad attori_G su JVM_G, comprensivo di un *Domain Specific Language* (DSL_G) da utilizzare da riga di comando per poter interagire con il database.

Il progetto dovrà essere pubblicato su *GitHub* sotto licenza *MIT*.

1.3 Glossario

Tutti i termini di carattere tecnico o frantendibile e gli acronimi sono raccolti nel file [Glossario v1.0.0](#); ogni occorrenza di parole nel *Glossario* è indicata da una "G" in pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Capitolato d'appalto C1:** *Actorbase: a NoSQL DB based on the Actor model;*
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1.pdf>
- **Norme di Progetto:** [Norme di Progetto v4.0.0](#).

1.4.2 Informativi

- **Piano di Progetto:** [Piano di Progetto v4.0.0](#)
- **Dispense fornite dall'insegnamento Ingegneria del Software mod.A:**
<http://www.math.unipd.it/~tullio/IS-1/2015/>
- **Documentazione di Akka_G su Scala_G:** <http://doc.akka.io/docs/akka/2.4.2/scala.html>
- **SBT - Scala build tool:** <http://www.scala-sbt.org/index.html>
- **Scala Pickling - Fast, Customizable, Boilerplate-Free Serialization for Scala:** Libreria di serializzazione e deserializzazione multiformato, pieno supporto JSON_G (JavaScript Object Notation) e serializzazione in binario <http://lampwww.epfl.ch/~hmiller/pickling>



- **Spray - an open-source toolkit for building REST/HTTP-based integration layers on top of Scala and Akka:** Libreria per la gestione di richieste HTTP_G(HyperText Transfert Protocol) in modo asincrono, basata sul modello ad attori offerto dalla libreria Akka. <http://spray.io/>
- **REST - Representational State Transfer:** Architettura software per sistemi di ipertesto_G distribuiti, rappresenta un sistema di localizzazione e richiesta risorse stateless_G, poggia su HTTP_G come protocollo di comunicazione https://en.wikipedia.org/wiki/Representational_state_transfer



2 Standard di progetto

2.1 Standard di progettazione architetturale

Gli standard di progettazione architetturale sono definiti nella *Specifica Tecnica v4.0.0*.

2.2 Standard di documentazione del codice

Gli standard per la scrittura della documentazione del codice sono definiti nel documento *Norme di Progetto v4.0.0*.

2.3 Standard di programmazione

Gli standard di programmazione sono definiti nel documento *Norme di Progetto v4.0.0*

2.4 Strumenti di lavoro

Gli strumenti da adottare e le procedure da seguire per il corretto utilizzo durante la realizzazione del prodotto sono definiti nel documento *Norme di Progetto v4.0.0*



3 Componenti

L'architettura principale del progetto è stata suddivisa in tre macro-componenti, rispettivamente **cli**, **driver** e **actorsystem**; contenute all'interno del package generale **actorbase**.



3.1 actorbase

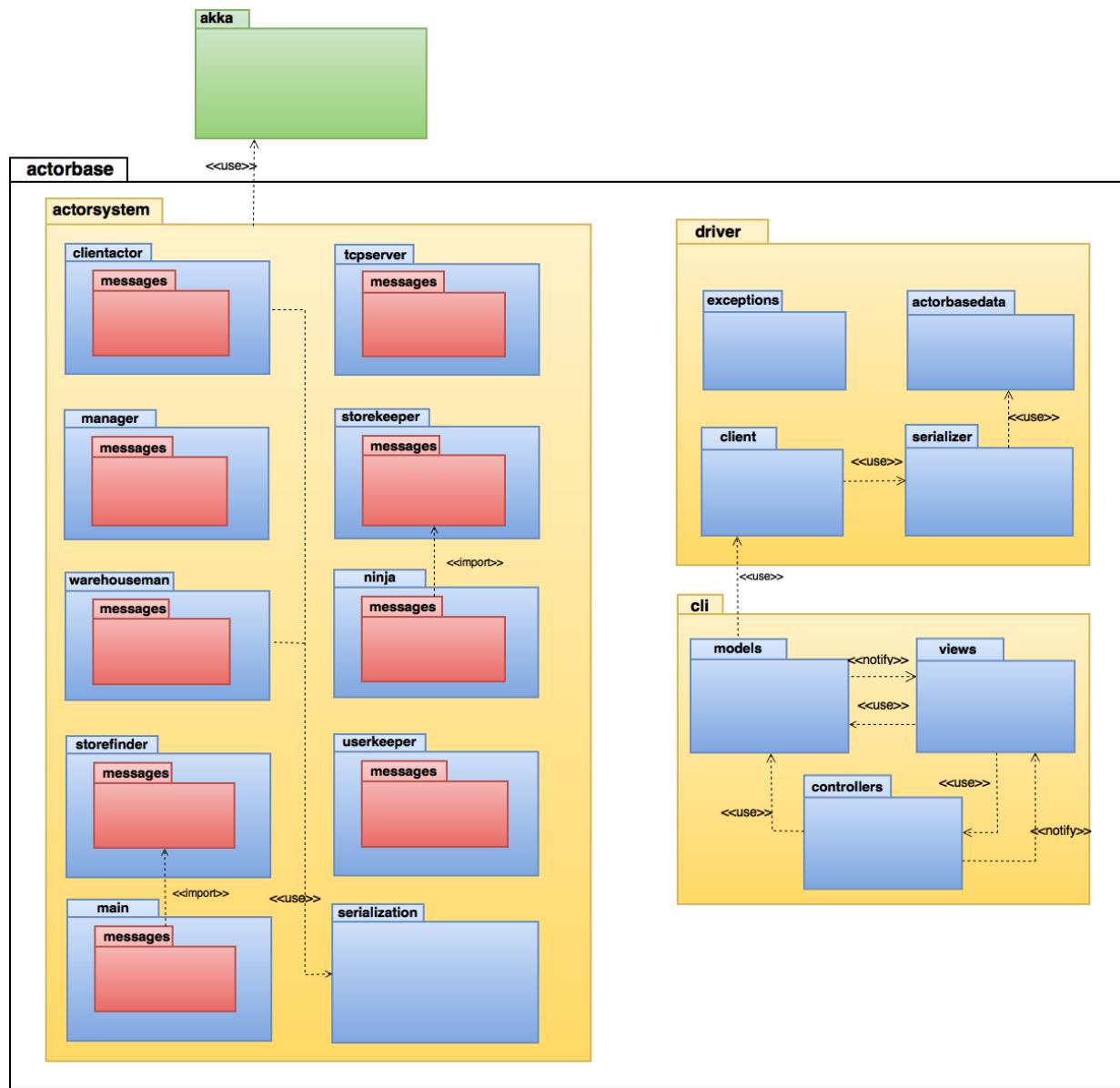


Figura 1: Diagramma componenti principali: Server, Driver e CLI



3.1.1 Descrizione

La componente principale rappresentata dal package globale **actorbase**.

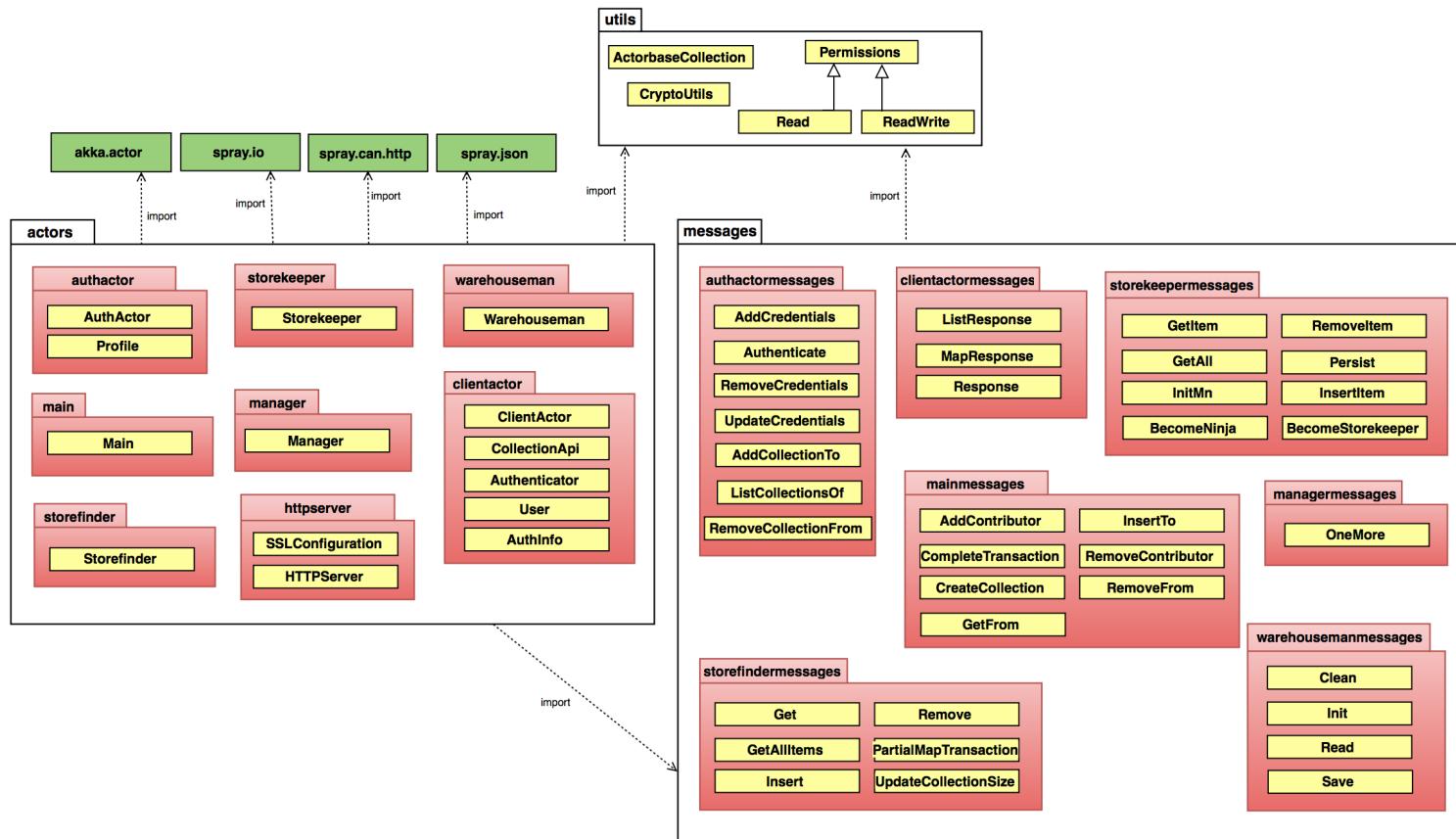
3.1.2 Package contenuti

- `actorbase::cli`;
- `actorbase::driver`;
- `actorbase::actorsystem`.

Rappresentano le tre macro-componenti del sistema.



3.2 actorbase::actorsystem

Figura 2: Package_G actorsystem, visione generale

3.2.1 Descrizione

Package_G che raggruppa tutte le componenti del sistema che rappresentano il server_G.

3.2.2 Package contenuti

- `actorbase::actorsystem::actors`
- `actorbase::actorsystem::messages`
- `actorbase::actorsystem::utils`.



3.3 actorbase::actorsystem::utils

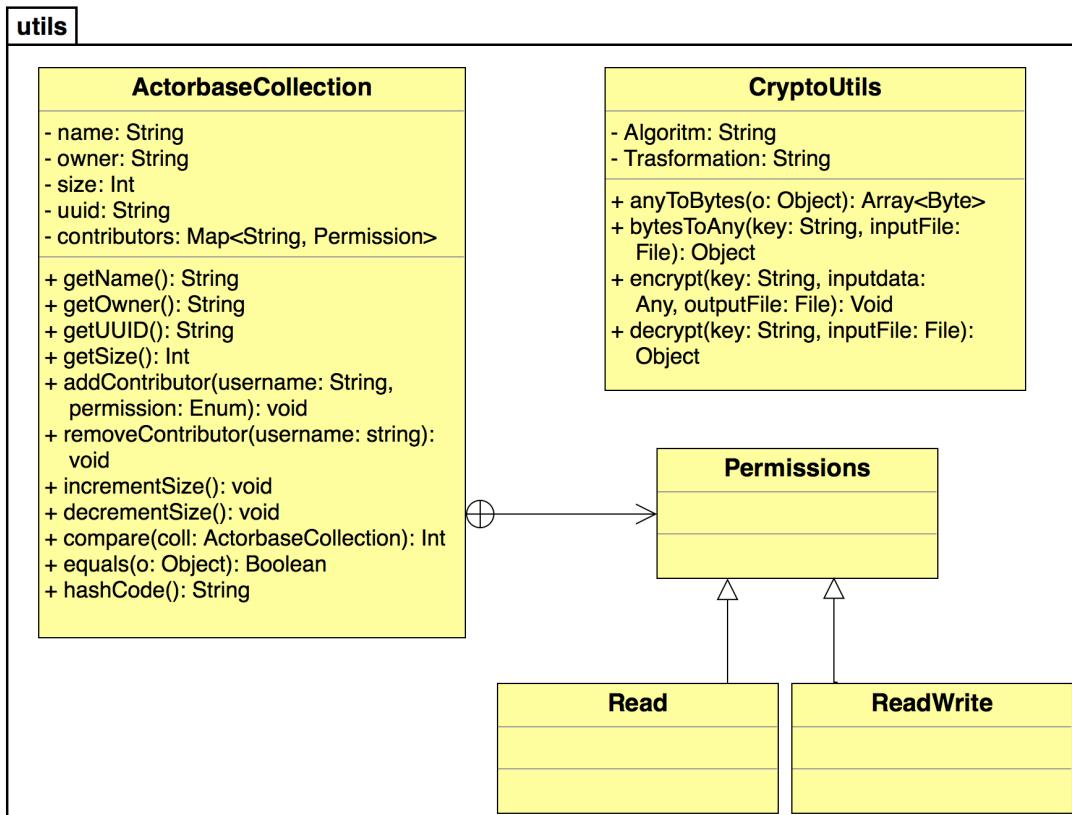


Figura 3: Package actorsystem::utils

3.3.1 Descrizione

Package_G contenente diverse classi di utilità.



3.3.2 Interfacce

3.3.2.1 Permissions

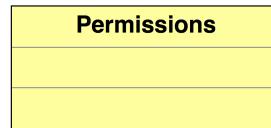


Figura 4: Classe actorsystem::utils::Permissions

3.3.2.1.1 Descrizione

Interfaccia che rappresenta i permessi sulle collezioni_G da applicare ai collaboratori_G.

3.3.2.1.2 Utilizzo

Viene utilizzata per rappresentare i tipi di permessi sulle collezioni_G da applicare ai collaboratori_G.

3.3.2.1.3 Realizzata da

- actorbase::actorsystem::utils::Read;
- actorbase::actorsystem::utils::ReadWrite.

3.3.3 Classi

3.3.3.1 Read

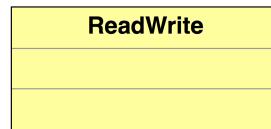


Figura 5: Classe actorsystem::utils::Read

3.3.3.1.1 Descrizione

Interfaccia che rappresenta i permessi di sola lettura sulle collezioni_G da applicare ai collaboratori_G.



3.3.3.1.2 Utilizzo

Viene utilizzata per rappresentare i tipi di permessi di sola lettura sulle collezioni_G da applicare ai collaboratori_G.

3.3.3.1.3 Realizza

- `actorbase::actorsystem::utils::Permissions`.

3.3.3.2 ReadWrite

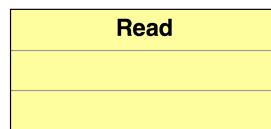


Figura 6: Classe `actorsystem::utils::ReadWrite`

3.3.3.2.1 Descrizione

Interfaccia che rappresenta i permessi di lettura e scrittura sulle collezioni_G da applicare ai collaboratori_G.

3.3.3.2.2 Utilizzo

Viene utilizzata per rappresentare i tipi di permessi di sola lettura e scrittura sulle collezioni_G da applicare ai collaboratori_G.

3.3.3.2.3 Realizza

- `actorbase::actorsystem::utils::Permissions`.



3.3.3.3 CryptoUtils

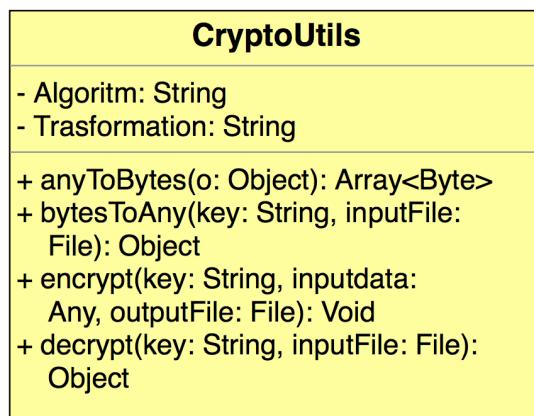


Figura 7: Classe actorsystem::utils::CryptoUtils

3.3.3.1 Descrizione

Classe atta a criptare e decriptare i dati per il salvataggio e la lettura dei dati su disco, utilizzata dagli attori di tipo [Warehouseman](#), permette di impostare il tipo di crittografia da utilizzare.

3.3.3.2 Utilizzo

Viene utilizzata dall'attore di tipo [Warehouseman_G](#) per il salvataggio e la lettura del database su disco.

3.3.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
Algorithm	privato	immutabile	String	Rappresenta l'algoritmo usato per la criptazione
Transformation	privato	immutabile	String	Rappresenta la trasformazione per ottenere il cifrario usato per la criptazione



3.3.3.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
anyToBytes	pubblico	Array _G di byte _G	Metodo utilizzato per trasformare un oggetto qualsiasi in un array _G di byte _G
bytesToAny	pubblico	Object	Metodo utilizzato per riportare allo stato originale un oggetto rappresentato in forma array _G di byte _G
encrypt	pubblico	void	Metodo che si occupa di criptare i dati
decrypt<T>	pubblico	T	Metodo che si occupa di decriptare i dati riportandoli al tipo generico richiesto

3.3.3.3.5 Parametri

anyToBytes

Nome	Tipo	Descrizione
o	Object	Riferimento all'oggetto da trasformare in array _G di byte _G

bytesToAny

Nome	Tipo	Descrizione
key	String	Stringa che rappresenta la chiave di criptazione
inputFile	File	File che contiene i dati di persistenza del sistema criptati

encrypt

Nome	Tipo	Descrizione
key	String	Stringa che rappresenta la chiave di criptazione
inputData	Map<String, Object>	Mappa che rappresenta i dati che verranno criptati
outputFile	File	File destinato a contenere i dati criptati

decrypt<T>

Nome	Tipo	Descrizione
key	String	Stringa che rappresenta la chiave di decriptazione
inputFile	File	File che contiene i dati da decriptare



3.3.3.4 ActorbaseCollection

ActorbaseCollection	
- name: String	
- owner: String	
- size: Int	
- uuid: String	
- contributors: Map<String, Permission>	
+ getName(): String	
+ getOwner(): String	
+ getUUID(): String	
+ getSize(): Int	
+ addContributor(username: String, permission: Enum): void	
+ removeContributor(username: string): void	
+ incrementSize(): void	
+ decrementSize(): void	
+ compare(coll: ActorbaseCollection): Int	
+ equals(o: Object): Boolean	
+ hashCode(): String	

Figura 8: Classe actorsystem::utils::ActorbaseCollection

3.3.3.4.1 Descrizione

Classe atta a modellare una collezione_G del sistema actorbase.

3.3.3.4.2 Utilizzo

Viene utilizzata da diversi attori del sistema. Questa classe espone metodi getter, metodi per aumentare e diminuire la dimensione della collezione e un metodo per il confronto con un altro oggetto di questo tipo. Contiene infine un identificativo univoco universale utilizzabile dall'estensione Akka_G *cluster sharding* per l'indirizzamento dei messaggi riguardanti collezioni_G.

3.3.3.4.3 Realizza

- trait scala.math.Ordered



3.3.3.4.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
name	privata	mutabile	String	Rappresenta il nome della collezione _G
owner	private	mutabile	String	Rappresenta lo username dell'utente che ha creato la collezione _G
uuid	private	immutabile	String	Rappresenta un identificativo univoco universale alla collezione _G all'interno del sistema
contributors	private	mutabile	Map<String, Permissions>	Rappresenta una mappa di username _G e permessi lettura o lettura-scrittura

3.3.3.4.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
getName	pubblico	String	Metodo che ritorna il nome della collezione _G
getOwner	pubblico	String	Metodo che ritorna lo username dell'utente che ha creato la collezione _G
getSize	pubblico	Int	Metodo che ritorna la dimensione della collezione _G rappresentata
getUUID	pubblico	String	Metodo che ritorna lo uuid della collezione. Questo rappresenta una stringa per identificare univocamente diverse collezioni _G
addContributor	pubblico	void	Metodo che aggiunge un utente alla mappa dei collaboratori _G alla collezione _G
removeContributor	pubblico	void	Metodo che rimuove un utente alla mappa dei collaboratori _G alla collezione _G
incrementSize	pubblico	void	Metodo che incrementa di una unità la size della collezione _G . Usato quando si effettua un inserimento di un item _G
decrementSize	pubblico	void	Metodo che decrementa di una unità la size della collezione _G . Usato quando si effettua una rimozione di un item _G
compare	pubblico	Int	Metodo per l'implementazione dell'interfaccia Ordered. Permette di confrontare più oggetti ActorbaseCollection per capire se rappresentano la stessa collezione _G

3.3.3.4.6 Parametri

addContributor

Nome	Tipo	Descrizione
username	String	Username _G dell'utente da aggiungere alla mappa dei collaboratori _G

**removeContributor**

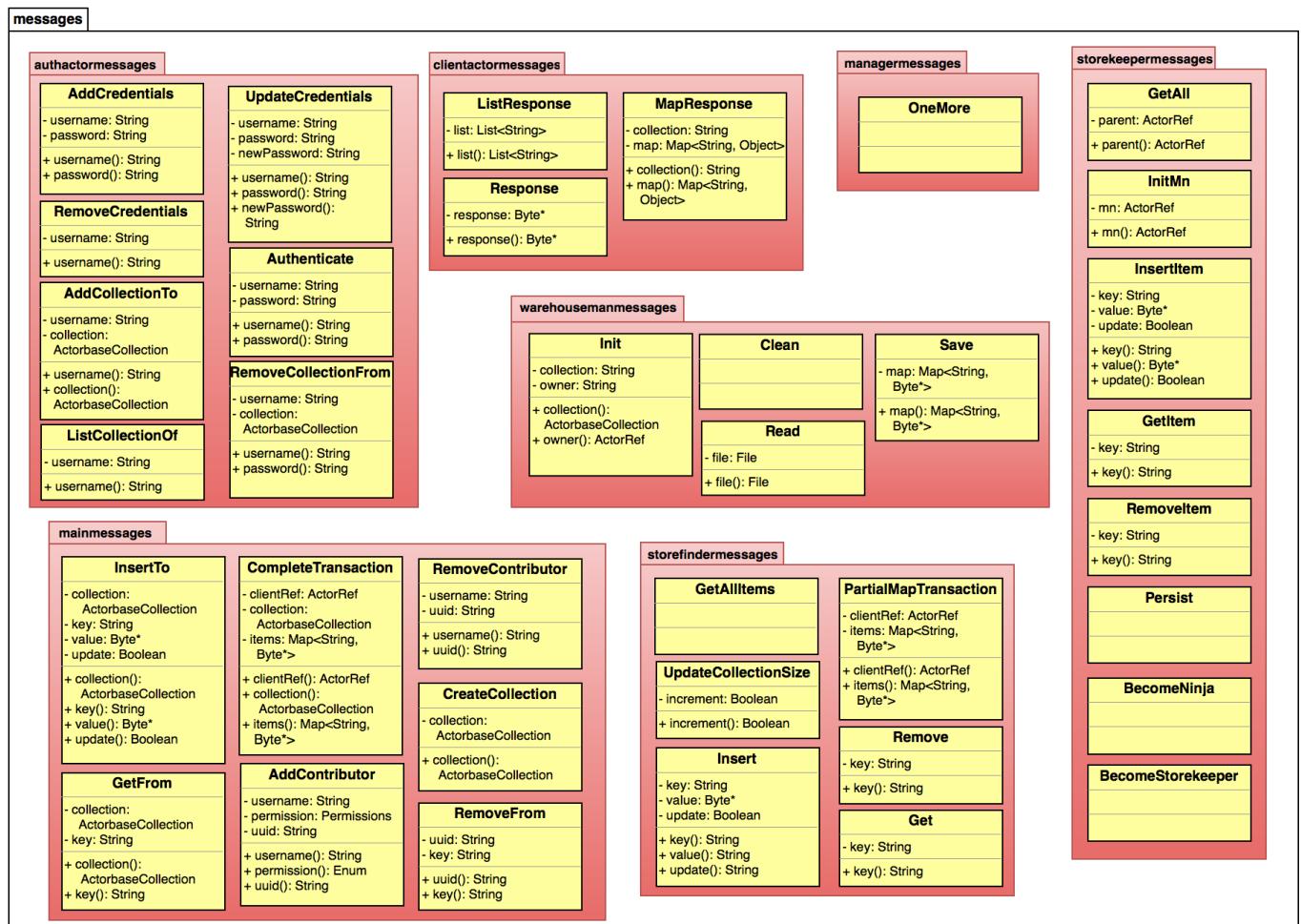
Nome	Tipo	Descrizione
username	String	Username _G dell'utente da rimuovere alla mappa dei collaboratori _G

compare

Nome	Tipo	Descrizione
that	ActorbaseCollection	ActorbaseCollection da confrontare con l'oggetto this



3.4 actorbase::actorsystem::messages

Figura 9: Package_G messages

3.4.1 Descrizione

Package_G che raggruppa tutti i messaggi del server_G.

3.4.2 Package contenuti

- `actorbase::actorsystem::messages::authactormessages`;



- actorbase::actorsystem::messages::clientactormessages;
- actorbase::actorsystem::messages::mainmessages;
- actorbase::actorsystem::messages::storefindermessages;
- actorbase::actorsystem::messages::storekeepermessages;
- actorbase::actorsystem::messages::warehousemanmessages;
- actorbase::actorsystem::messages::manangermessages.

3.5 actorbase::actorsystem::messages::authactormessages

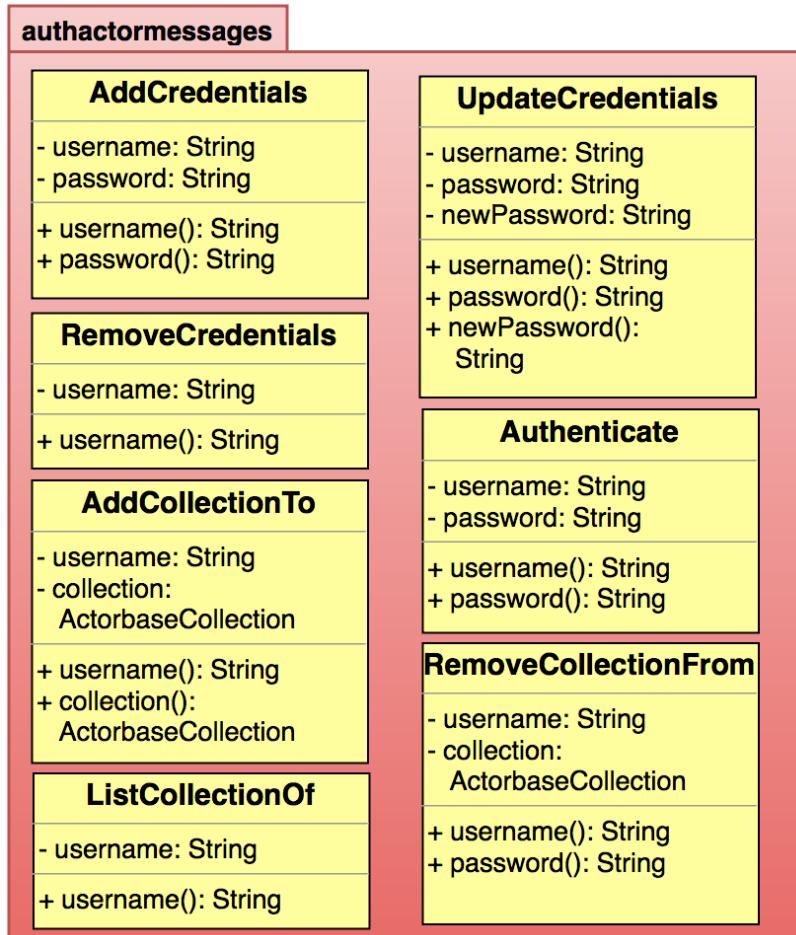


Figura 10: Package `actorsystem::messages::authactormessages`



3.5.1 Descrizione

Package_G contenente i messaggi processabili dall'attore `AuthActor`.

3.5.2 Classi

3.5.2.1 actorbase::actorsystem::messages::authactormessages::AddCredentials

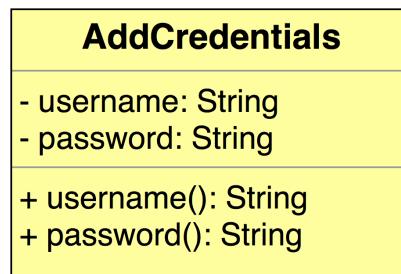


Figura 11: Classe AddCredentials

3.5.2.1.1 Descrizione

Classe che rappresenta una richiesta di aggiunta di credenziali al sistema.

Implementata utilizzando la specifica funzionalità `case class` di Scala_G, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.1.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore_G `AuthActor` di aggiunta di credenziali al sistema, all'interno del metodo `Receive` che la utilizza avviene inoltre il controllo della password, in modo che rispetti i vincoli imposti dal sistema.

3.5.2.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username da aggiungere alle credenziali del sistema
password	privato	immutabile	String	Rappresenta la password da aggiungere abbinata allo username aggiunto



3.5.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username_G
password	pubblico	String	Metodo che ritorna l'attributo password_G

3.5.2.2 actorbase::actorsystem::messages::authactormessages::UpdateCredentials

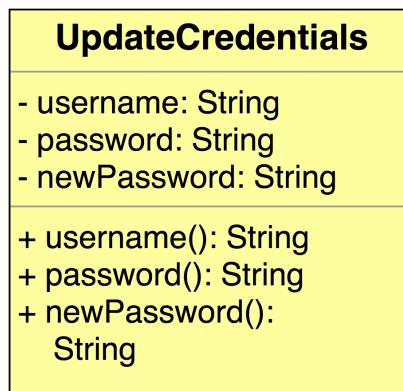


Figura 12: Classe UpdateCredentials

3.5.2.2.1 Descrizione

Classe che rappresenta una richiesta di modifica di password di un utente al sistema.

Implementata utilizzando la specifica funzionalità *case class* di Scala_G, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.2.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore AuthActor di modifica della password di un utente.

3.5.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username_G dell'utente a cui cambiare la password _G
password	privato	immutabile	String	Rappresenta la vecchia password_G da cambiare
newPassword	privato	immutabile	String	Rappresenta la nuova password da assegnare all'utente rappresentato dallo username_G immesso



3.5.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username
password	pubblico	String	Metodo che ritorna l'attributo password
newPassword	pubblico	String	Metodo che ritorna l'attributo newPassword

3.5.2.3 actorbase::actorsystem::messages::authactormessages::RemoveCredentials

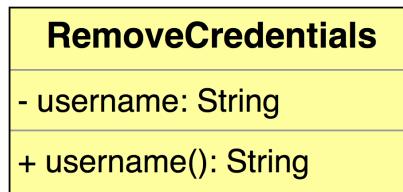


Figura 13: Classe RemoveCredentials

3.5.2.3.1 Descrizione

Classe che rappresenta una richiesta di rimozione di un utente dal sistema.

Implementata utilizzando la specifica funzionalità *case class* di Scala_G, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.3.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore_G **AuthActor** di rimozione di un utente.

3.5.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username dell'utente da rimuovere dal sistema

3.5.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username



3.5.2.4 actorbase::actorsystem::messages::authactormessages::Authenticate

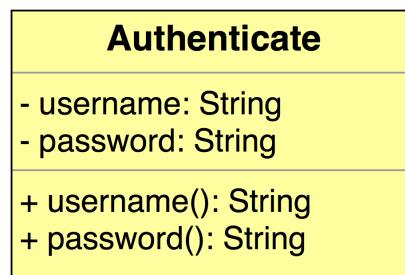


Figura 14: Classe Authenticate

3.5.2.4.1 Descrizione

Classe che rappresenta una richiesta di login di un utente al sistema.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching₆.

3.5.2.4.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore **AuthActor** di accesso al sistema. L'attore **AuthActor** dovrà controllare se le credenziali immesse dall'utente sono corrette.

3.5.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username dell'utente che vuole effettuare il login
password	privato	immutabile	String	Rappresenta la password dell'utente che vuole effettuare il login

3.5.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username
password	pubblico	String	Metodo che ritorna l'attributo password



3.5.2.5 actorbase::actorsystem::messages::authactormessages::AddCollectionTo

AddCollectionTo
- username: String - collection: ActorbaseCollection
+ username(): String + collection(): ActorbaseCollection

Figura 15: Classe AddCollectionTo

3.5.2.5.1 Descrizione

Classe che rappresenta una richiesta di inserimento di una collezione_G ad uno specifico utente del sistema. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.5.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore **AuthActor** di inserimento nuova collezione_G all'utente specificato. L'attore **AuthActor** dovrà controllare che l'utente a cui aggiungere la collezione_G sia presente nel sistema e che la collezione_G da inserire non sia già presente associata all'utente stesso.

3.5.2.5.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username dell'utente a cui aggiungere la nuova collezione _G
collection	privato	immutabile	ActorbaseCollection	Rappresenta la collezione _G da aggiungere alla lista associata all'utente specificato

3.5.2.5.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo di accesso in sola lettura che ritorna l'attributo username
collection	pubblico	ActorbaseCollection	Metodo di accesso in sola lettura che ritorna l'attributo collection



3.5.2.6 actorbase::actorsystem::messages::authactormessages::RemoveCollectionFrom

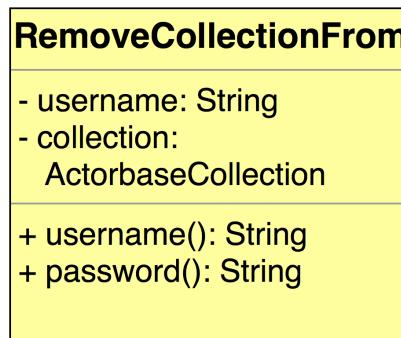


Figura 16: Classe RemoveCollectionFrom

3.5.2.6.1 Descrizione

Classe che rappresenta una richiesta di rimozione di una collezione_G associata ad uno specifico utente del sistema.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.6.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore [AuthActor](#) di rimozione di una collezione_G associata all'utente specificato. L'attore [AuthActor](#) dovrà controllare che l'utente a cui rimuovere la collezione_G sia presente nel sistema e che la collezione_G da rimuovere sia effettivamente presente e associata all'utente stesso.

3.5.2.6.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username dell'utente a cui rimuovere la collezione _G specificata
collection	privato	immutabile	ActorbaseCollection	Rappresenta la collezione _G da rimuovere dalla lista associata all'utente specificato



3.5.2.6.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo di accesso in sola lettura che ritorna l'attributo username
collection	pubblico	ActorbaseCollection	Metodo di accesso in sola lettura che ritorna l'attributo collection

3.5.2.7 actorbase::actorsystem::messages::authactormessages::ListCollectionsOf

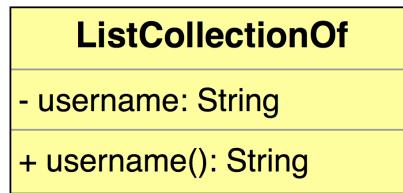


Figura 17: Classe ListCollectionsOf

3.5.2.7.1 Descrizione

Classe che rappresenta una richiesta di visualizzazione delle collezioni_G associate ad uno specifico utente del sistema.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.5.2.7.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta all'attore **AuthActor** di visualizzazione delle collezioni_G associate all'utente specificato. L'attore **AuthActor** dovrà controllare che l'utente specificato sia presente nel sistema.

3.5.2.7.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username _G dell'utente di cui si vuole visualizzare la lista collezioni _G create o a cui contribuisce



3.5.2.7.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo di accesso in sola lettura che ritorna l'attributo username _G

3.6 actorbase::actorsystem::messages::clientactormessages

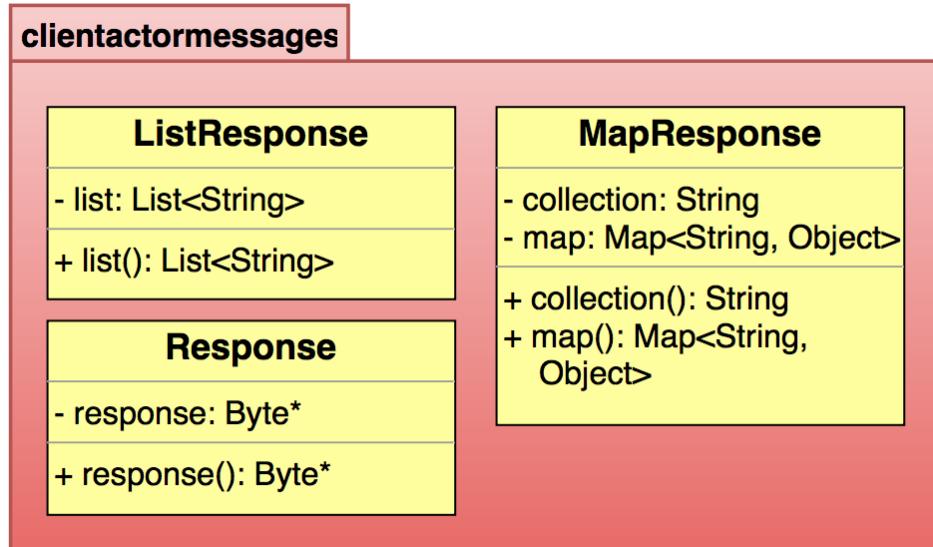


Figura 18: Package actorsystem::messages::clientactormessages

3.6.1 Descrizione

Package_G contenente i messaggi processabili dall'attore ClientActor.



3.6.2 Classi

3.6.2.1 actorbase::actorsystem::messages::clientactormessages::ListResponse



Figura 19: Classe ListResponse

3.6.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per consegnare al `ClientActor` una lista di stringhe, utilizzato per adempiere ad alcune richieste dell'utente.

Implementata utilizzando la specifica funzionalità `case class` di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.6.2.1.2 Utilizzo

Questa classe viene utilizzata per mandare una lista di stringhe all'attore `ClientActor`. Questo viene utilizzato per adempiere ad alcune richieste dell'utente come la richiesta di lista collezioni.

3.6.2.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
list	privato	immutabile	List<String>	Rappresenta una lista di stringhe da restituire all'utente a seguito di una richiesta

3.6.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
list	pubblico	List<String>	Metodo che ritorna l'attributo list



3.6.2.2 actorbase::actorsystem::messages::clientactormessages::MapResponse

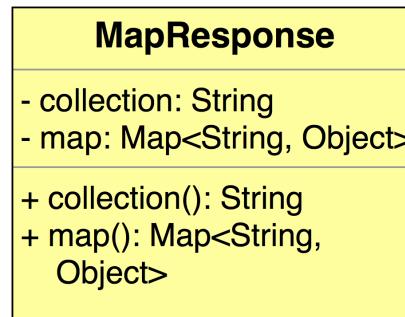


Figura 20: Classe MapResponse

3.6.2.2.1 Descrizione

Classe che rappresenta un messaggio utilizzato per consegnare al [ClientActor](#) una collezione dati, utilizzato per adempiere ad alcune richieste dell'utente.

Implementata utilizzando la specifica funzionalità `case class` di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.6.2.2.2 Utilizzo

Questa classe viene utilizzata per mandare una collezione dati all'attore [ClientActor](#). Questo viene utilizzato per adempiere ad alcune richieste dell'utente come la richiesta di restituzione di una collezione.

3.6.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	String	Rappresenta il nome di una collezione
map	privato	immutabile	Map<String, Object>	Rappresenta una struttura dati contenente le coppie chiave valore da restituire all'utente a seguito di una richiesta

3.6.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collection	pubblico	String	Metodo che ritorna l'attributo collection
map	pubblico	Map<String, Object>	Metodo che ritorna l'attributo map



3.6.2.3 actorbase::actorsystem::messages::clientactormessages::Response

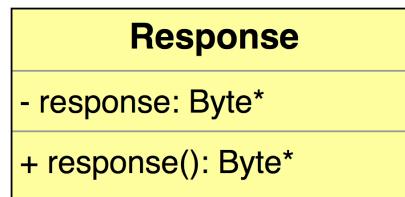


Figura 21: Classe Response

3.6.2.3.1 Descrizione

Classe che rappresenta un messaggio utilizzato per consegnare al `ClientActor` una stringa, utilizzato per adempiere ad alcune richieste dell'utente.

Implementata utilizzando la specifica funzionalità `case class` di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.6.2.3.2 Utilizzo

Questa classe viene utilizzata per mandare una stringa all'attore `ClientActor`. Questo viene utilizzato per adempiere ad alcune richieste dell'utente.

3.6.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
response	privato	immutabile	Array<Byte>	Rappresenta una stringa contenente il messaggio o il valore da restituire all'utente a seguito di una sua richiesta

3.6.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
response	pubblico	Array<Byte>	Metodo che ritorna l'attributo response



3.7 actorbase::actorsystem::messages::mainmessages

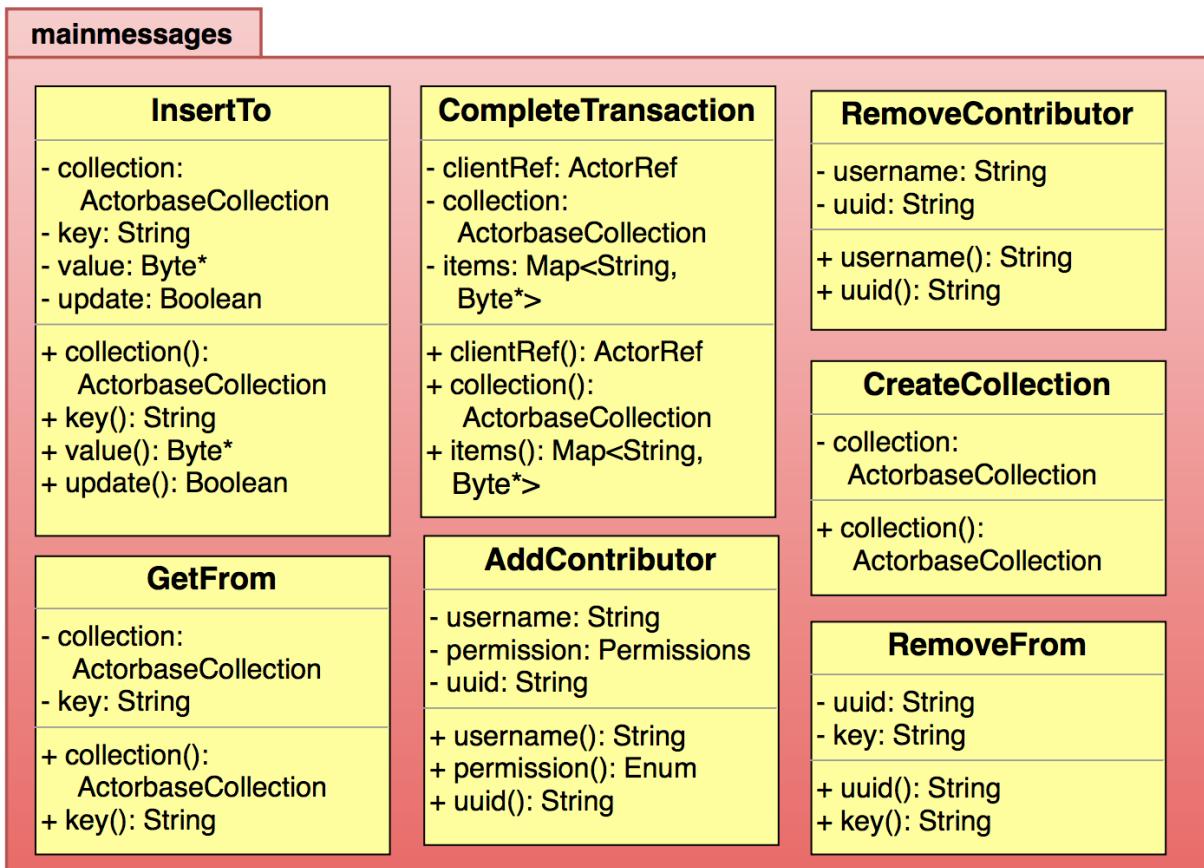


Figura 22: Package actorsystem::messages::mainmessages

3.7.1 Descrizione

Package_G contenente i messaggi processabili dall'attore `Main`.



3.7.2 Classi

3.7.2.1 actorbase::actorsystem::messages::mainmessages::InsertTo

InsertTo	
- collection:	ActorbaseCollection
- key:	String
- value:	Byte*
- update:	Boolean
+ collection():	ActorbaseCollection
+ key():	String
+ value():	Byte*
+ update():	Boolean

Figura 23: Classe InsertTo

3.7.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere al [Main](#) l'inserimento di un item. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.7.2.1.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta di inserimento di un item all'attore [Main](#). Questo attore provvederà ad inoltrare la richiesta al giusto attore [Storefinder](#) che rappresenta la collezione_G in cui inserire l'item. Se non è presente la collezione in cui inserire l'item questa verrà creata istanziando un nuovo attore [Storefinder](#), infine verrà inoltrata la richiesta di inserimento a quest'ultimo.



3.7.2.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	ActorbaseCollection	Rappresenta la collezione in cui effettuare l'inserimento
key	privato	immutabile	String	Rappresenta la chiave dell'item da inserire
value	privato	immutabile	Array<Byte>	Rappresenta il valore dell'item serializzato in array di byte
update	privato	immutabile	Boolean	Rappresenta un flag per scegliere se l'inserimento deve poter sovrascrivere un item già presente con la stessa chiave o meno

3.7.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collection	pubblico	ActorbaseCollection	Metodo che ritorna l'attributo collection
key	pubblico	String	Metodo che ritorna l'attributo key
value	pubblico	Array<Byte>	Metodo che ritorna l'attributo value
update	pubblico	Boolean	Metodo che ritorna l'attributo update

3.7.2.2 actorbase::actorsystem::messages::mainmessages::GetFrom

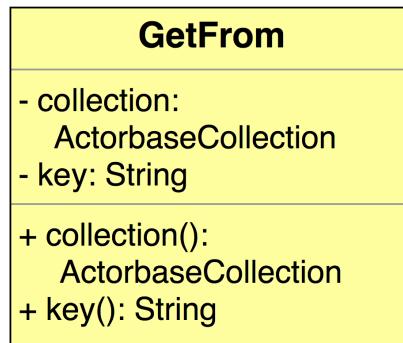


Figura 24: Classe GetFrom

3.7.2.2.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere al [Main](#) di ritornare items o collezioni. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.



3.7.2.2.2 Utilizzo

Questa classe viene utilizzata per mandare una richiesta di ritorno di un item o di una collezione all'attore [Main](#). Questo attore provvederà ad inoltrare la richiesta al giusto attore [Storefinder](#).

Il parametro in ingresso *key* determina se l'utente richiede un singolo item o una intera collezione.

3.7.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	ActorbaseCollection	Rappresenta la collezione in cui cercare o la collezione da ritornare
key	privato	immutabile	String	Rappresenta la chiave dell'item da cercare

3.7.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collection	pubblico	ActorbaseCollection	Metodo che ritorna l'attributo collection
key	pubblico	String	Metodo che ritorna l'attributo key

3.7.2.3 [actorbase::actorsystem::messages::mainmessages::CompleteTransaction](#)

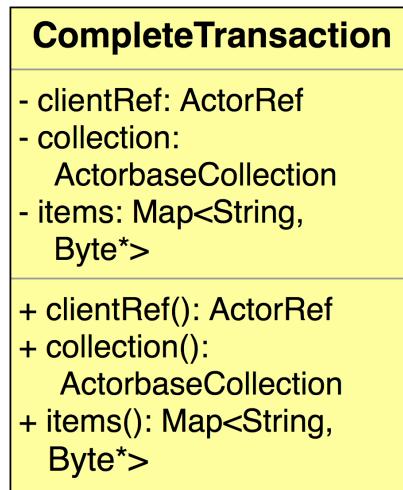


Figura 25: Classe CompleteTransaction

3.7.2.3.1 Descrizione

Classe che rappresenta un messaggio utilizzato per assemblare parti di collezioni_G provenienti dagli attori_G di tipo [Storekeeper](#) verso l'attore_G [Main](#).



Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.7.2.3.2 Utilizzo

Questa classe viene utilizzata per assemblare le parti di collezione_G provenienti dagli attori_G di tipo **Storekeeper** verso l'attore_G **Main**.

Quest'ultimo dovrà attendere tutti i messaggi **CompleteTransaction** necessari per costruire un intera collezione_G da restituire all'utente.

Questo messaggio è utilizzato quando un utente richiede un intera collezione_G.

3.7.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
clientRef	privato	immutabile	ActorRef	Rappresenta il riferimento all'attore ClientActor che ha effettuato la richiesta di ritorno collezione
collection	privato	immutabile	ActorbaseCollection	Rappresenta la collezione di cui fanno parte gli item passati in questo messaggio
items	privato	immutabile	Map<String, Array<Byte> >	Rappresenta una mappa di items, in particolare rappresentano un pezzo della collezione richiesta

3.7.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
clientRef	pubblico	ActorRef	Metodo che ritorna l'attributo clientRef
collection	pubblico	ActorbaseCollection	Metodo che ritorna l'attributo collection
items	pubblico	Map<String, Array<Byte> >	Metodo che ritorna l'attributo key

3.7.2.4 actorbase::actorsystem::messages::mainmessages::RemoveFrom

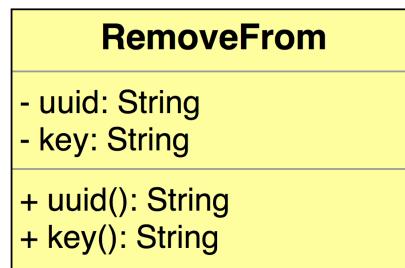


Figura 26: Classe RemoveFrom



3.7.2.4.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere la rimozione di item all'attore **Main**. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.7.2.4.2 Utilizzo

Questa classe viene utilizzata per rimuovere items dal sistema. L'attore **Main** dovrà inoltrare la richiesta al giusto attore **Storefinder**, ovvero colui che rappresenta la collezione in cui effettuare l'operazione.

3.7.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
uuid	privato	immutabile	String	Rappresenta l'id della collezione da cui eliminare l'item _G
key	privato	immutabile	String	Rappresenta la chiave dell'item da eliminare

3.7.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
uuid	pubblico	String	Metodo che ritorna l'attributo uuid
key	pubblico	String	Metodo che ritorna l'attributo key

3.7.2.5 actorbase::actorsystem::messages::mainmessages::AddContributor

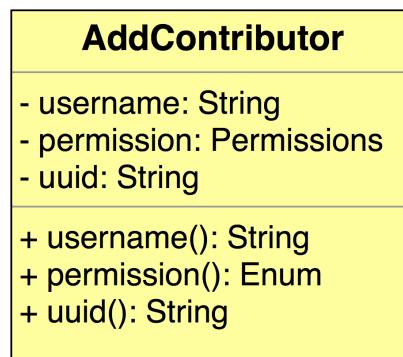


Figura 27: Classe AddContributor



3.7.2.5.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere l'aggiunta di un collaboratore ad una collezione_G all'attore **Main**.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.7.2.5.2 Utilizzo

Questa classe viene utilizzata per richiedere l'aggiunta di collaboratori ad una collezione. L'attore **Main** dovrà aggiungere lo username del collaboratore designato alla collezione richiesta.

3.7.2.5.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username _G del collaboratore _G da aggiungere alla collezione _G
permission	privato	immutabile	Enum	Rappresenta il tipo di permessi da assegnare al collaboratore _G
uuid	privato	immutabile	String	Rappresenta l'id della collezione _G a cui aggiungere il collaboratore _G

3.7.2.5.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username
permission	pubblico	Enum	Metodo che ritorna l'attributo permission
uuid	pubblico	String	Metodo che ritorna l'attributo uuid

3.7.2.6 actorbase::actorsystem::messages::mainmessages::RemoveContributor

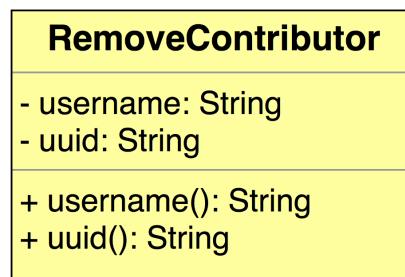


Figura 28: Classe RemoveContributor



3.7.2.6.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere la rimozione di un collaboratore ad una collezione all'attore **Main**.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.7.2.6.2 Utilizzo

Questa classe viene utilizzata per richiedere la rimozione di collaboratori ad una collezione. L'attore **Main** dovrà rimuovere lo username del collaboratore designato alla collezione richiesta.

3.7.2.6.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username del collaboratore da rimuovere dalla collezione
uuid	privato	immutabile	String	Rappresenta l'id della collezione da cui rimuovere il collaboratore

3.7.2.6.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo che ritorna l'attributo username
uuid	pubblico	String	Metodo che ritorna l'attributo uuid

3.7.2.7 actorbase::actorsystem::messages::mainmessages::CreateCollection

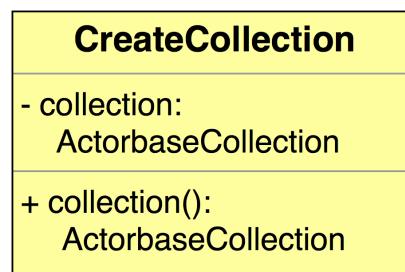


Figura 29: Classe CreateCollection



3.7.2.7.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere la creazione di una collezione all'attore [Main](#). Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.7.2.7.2 Utilizzo

Questa classe viene utilizzata per richiedere la creazione di una collezione.

L'attore [Main](#) provvederà a creare la nuova collezione e a creare un attore [Storefinder](#) adibito alla gestione di quest'ultima.

3.7.2.7.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	ActorbaseCollection	Collezione da aggiungere al sistema

3.7.2.7.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collection	pubblico	ActorbaseCollection	Metodo che ritorna l'attributo collection



3.8 actorbase::actorsystem::messages::storefindermessages

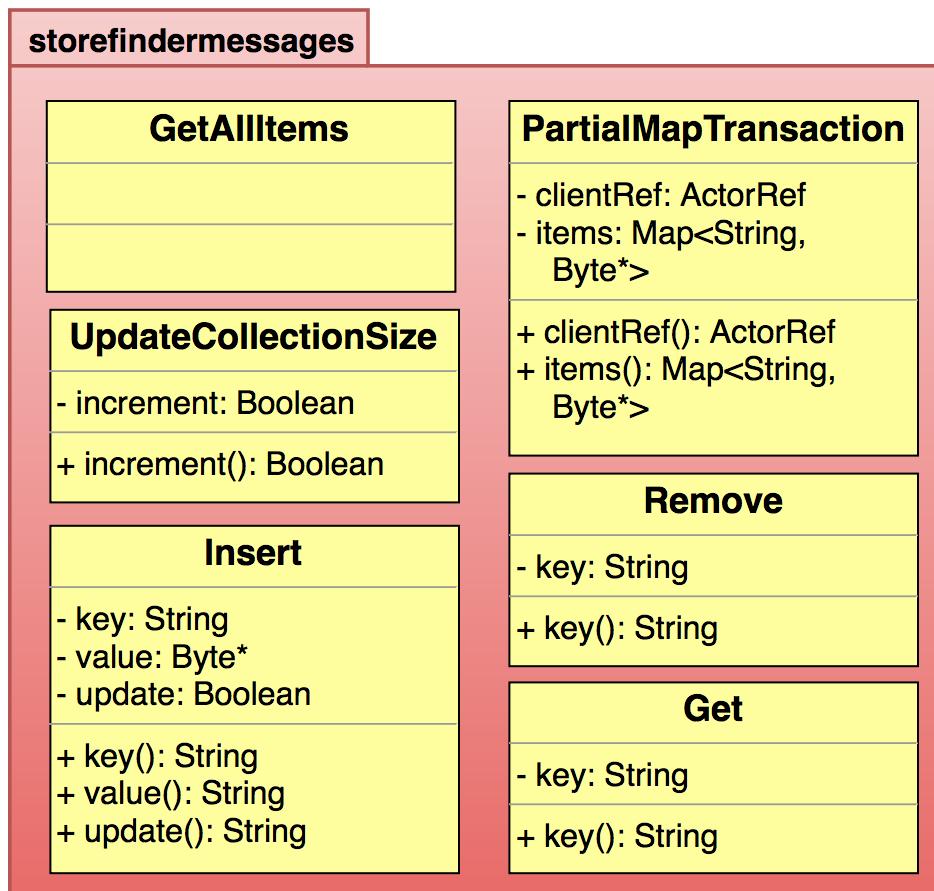


Figura 30: Package `actorsystem::messages::storefindermessages`

3.8.1 Descrizione

Package_G contenente i messaggi processabili dall'attore `Storefinder`.



3.8.2 Classi

3.8.2.1 actorbase::actorsystem::messages::storefinder::GetAllItems



Figura 31: Classe Storefinder

3.8.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere allo [Storefinder](#) di ritornare tutti gli item mappati dai propri [Storekeeper](#).

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.8.2.1.2 Utilizzo

Questa classe viene utilizzata per richiedere tutti gli items mappati dallo [Storefinder](#).

Questo attore provvederà a mandare richieste Get agli attori [Storekeeper](#).

Questo tipo di messaggio è ricevuto quando un utente intende richiedere intere collezioni.

3.8.2.2 actorbase::actorsystem::messages::storefinder::UpdateCollectionSize

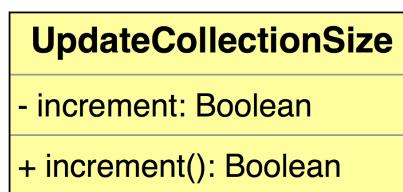


Figura 32: Classe UpdateCollectionSize

3.8.2.2.1 Descrizione

Classe che rappresenta un messaggio utilizzato per aggiornare il numero di item contenuti in una collezione. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.



3.8.2.2.2 Utilizzo

Questa classe viene utilizzata per tenere aggiornate le dimensioni delle collezioni quando si inseriscono o rimuovono items da esse.

3.8.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
increment	privato	immutabile	Boolean	flag che indica se c'è stato un incremento o un decremento nella collezione (true se incremento)

3.8.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
increment	pubblico	Boolean	Metodo che ritorna l'attributo increment

3.8.2.3 actorbase::actorsystem::messages::storefinder::Get

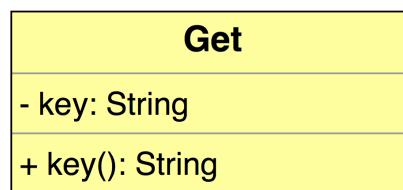


Figura 33: Classe Get

3.8.2.3.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere un item.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.8.2.3.2 Utilizzo

Questa classe viene utilizzata per richiedere un item da restituire all'utente.

Quando riceve questo messaggio l'attore provvederà a inoltrare la richiesta allo **Storekeeper** esatto, ovvero colui che contiene l'item mappato dalla chiave inserita.

3.8.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	privato	immutabile	String	Rappresenta la chiave dell'item da restituire



3.8.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key

3.8.2.4 actorbase::actorsystem::messages::storefinder::PartialMapTransaction

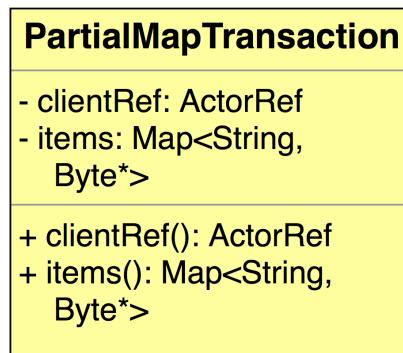


Figura 34: Classe PartialMapTransaction

3.8.2.4.1 Descrizione

Classe che rappresenta un messaggio utilizzato per ricevere parti di collezione_G da inviare all'attore parent_G. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.8.2.4.2 Utilizzo

Questa classe viene utilizzata per ricevere parti di collezione_G a seguito di una richiesta di un'intera collezione_G da parte di un utente.

Quando riceve questo messaggio l'attore_G provvederà a inoltrare la mappa di item_G all'attore_G Main, il quale provvederà ad assemblare tutta la collezione_G da restituire all'utente.

3.8.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
clientRef	privato	immutabile	ActorRef	Rappresenta il riferimento all'attore ClientActor che ha richiesto la collezione
items	privato	immutabile	Map<String, Array<Byte>>	Rappresenta una mappa contenente le coppie chiave valore degli item _G della collezione _G da restituire



3.8.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
clientRef	pubblico	ActorRef	Metodo che ritorna l'attributo clientRef
items	pubblico	Map<String, Array<Byte> >	Metodo che ritorna l'attributo items

3.8.2.5 actorbase::actorsystem::messages::storefinder::Remove

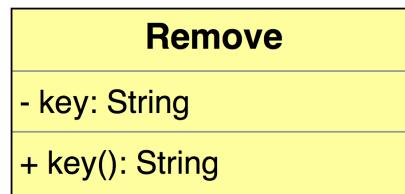


Figura 35: Classe Remove

3.8.2.5.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere la rimozione di un item_G. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.8.2.5.2 Utilizzo

Questa classe viene utilizzata per richiedere la rimozione di un item_G dal sistema. Quando riceve questo messaggio l'attore provvederà a inoltrare la richiesta allo **Storekeeper** giusto.

3.8.2.5.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	privato	immutabile	String	Rappresenta la chiave dell'elemento da rimuovere dalla collezione rappresentata da questo attore

3.8.2.5.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key



3.8.2.6 actorbase::actorsystem::messages::storefinder::Insert

Insert
- key: String - value: Byte* - update: Boolean
+ key(): String + value(): String + update(): String

Figura 36: Classe Insert

3.8.2.6.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere l'aggiunta di un item.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.8.2.6.2 Utilizzo

Questa classe viene utilizzata per richiedere l'aggiunta di un item dal sistema.

Quando riceve questo messaggio l'attore provvederà a inoltrare la richiesta allo **Storekeeper** giusto.

3.8.2.6.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	pubblico	immutabile	String	Rappresenta la chiave dell'elemento da aggiungere dalla collezione rappresentata da questo attore
value	pubblico	immutabile	Array<Byte>	Rappresenta il valore dell'item da inserire sotto forma di array di bytes
update	pubblico	immutabile	Boolean	Rappresenta un flag che serve a scegliere se l'operazione può effettuare sovrascrittura o no (true se l'operazione può sovrascrivere items con la stessa chiave)



3.8.2.6.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key
value	pubblico	Array<Byte>	Metodo che ritorna l'attributo value
update	pubblico	Boolean	Metodo che ritorna l'attributo update

3.9 actorbase::actorsystem::messages::storekeepermessages

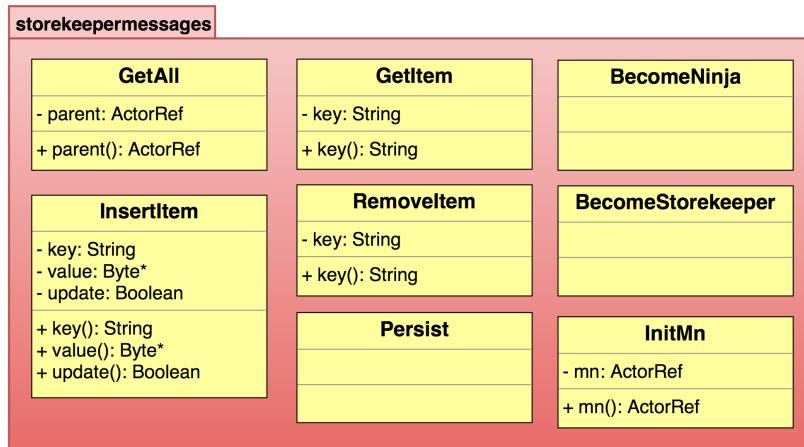


Figura 37: Package actorsystem::messages::storekeepermessages

3.9.1 Descrizione

Package_G contenente i messaggi processabili dall'attore **Storekeeper**.

3.9.2 Classi

3.9.2.1 actorbase::actorsystem::messages::storekeeper::Persist

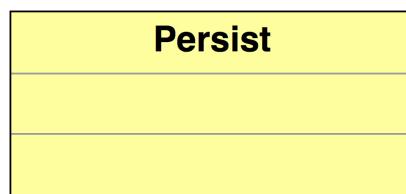


Figura 38: Classe Persist



3.9.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere allo [Storekeeper](#) di persistere i dati su disco. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.9.2.1.2 Utilizzo

Questa classe viene utilizzata per richiedere di effettuare persistenza su disco dei dati di questo attore. Questo attore provvederà a inoltrare la richiesta all'attore [Warehouseman](#).

3.9.2.2 actorbase::actorsystem::messages::storekeeper::InitMn

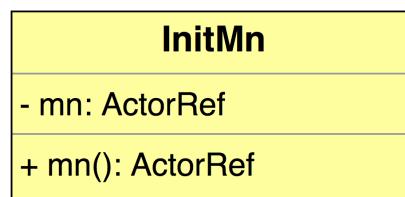


Figura 39: Classe InitMn

3.9.2.2.1 Descrizione

Classe che rappresenta un messaggio utilizzato per inizializzare un riferimento ad un attore [Manager](#). Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.9.2.2.2 Utilizzo

Questa classe viene utilizzata per ricevere un riferimento ad un attore [Manager](#). Questo messaggio verrà inviato allo [Storekeeper](#) subito dopo la sua creazione.

L'attore userà il riferimento al [Manager](#) per informarlo quando raggiunge una certa soglia di items. Il [Manager](#) creerà altri [Storekeeper](#) per distribuire il carico.

3.9.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
mn	pubblico	immutabile	ActorRef	Rappresenta un riferimento ad un attore Manager



3.9.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
man	pubblico	ActorRef	Metodo che ritorna l'attributo mn

3.9.2.3 actorbase::actorsystem::messages::storekeeper::GetAll



Figura 40: Classe GetAll

3.9.2.3.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere tutti gli items contenuti nell'attore. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.9.2.3.2 Utilizzo

Questa classe viene utilizzata per ricevere una richiesta di restituzione di tutti gli items mappati da questo attore. Quando questo attore riceve questo messaggio provvederà a inviare tutti gli items contenuti in esso al riferimento all'attore passato come parametro.

Questo messaggio serve a restituire all'utente intere collezioni.

3.9.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
parent	pubblico	immutabile	ActorRef	Rappresenta un riferimento all'attore Storefinder che rappresenta la collezione di cui lo Storekeeper fa parte

3.9.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
parent	pubblico	ActorRef	Metodo che ritorna l'attributo parent



3.9.2.4 actorbase::actorsystem::messages::storekeeper::GetItem



Figura 41: Classe GetItem

3.9.2.4.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere un item contenuto nell'attore. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching₆.

3.9.2.4.2 Utilizzo

Questa classe viene utilizzata per richiedere la restituzione di un item mappato da questo attore. Quando questo attore riceve questo messaggio provvederà a inviare l'item richiesto all'attore **ClientActor** che ha effettuato la richiesta.

3.9.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	pubblico	immutabile	String	Rappresenta la chiave dell'item da restituire

3.9.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key



3.9.2.5 actorbase::actorsystem::messages::storekeeper::InsertItem

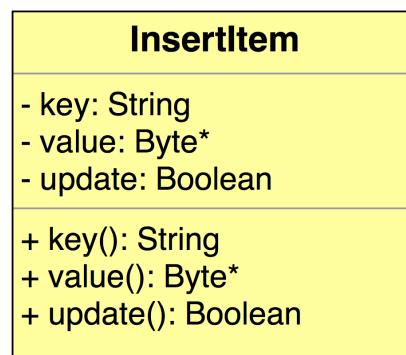


Figura 42: Classe InsertItem

3.9.2.5.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere l'inserimento di un item.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.9.2.5.2 Utilizzo

Questa classe viene utilizzata per richiedere l'inserimento di un item.

Quando questo attore riceve questo messaggio provvederà ad aggiungere l'item in questione nella propria struttura dati.

L'inserimento può avvenire con sovrascrittura o senza a seconda del valore dell'attributo `update` passato nel messaggio.

3.9.2.5.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	pubblico	immutabile	String	Rappresenta la chiave dell'item da inserire
value	pubblico	immutabile	Array<Byte>	Rappresenta il valore dell'item in formato array di bytes
update	pubblico	immutabile	Boolean	Flag per scegliere se l'inserimento può effettuare sovrascrittura o meno (true se può effettuare sovrascrittura)



3.9.2.5.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key
value	pubblico	Array<Byte>	Metodo che ritorna l'attributo value
update	pubblico	Boolean	Metodo che ritorna l'attributo update

3.9.2.6 actorbase::actorsystem::messages::storekeeper::RemoveItem



Figura 43: Classe RemoveItem

3.9.2.6.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere la rimozione di un item contenuto nell'attore. Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.9.2.6.2 Utilizzo

Questa classe viene utilizzata per richiedere la rimozione di un item mappato da questo attore. Quando questo attore riceve questo messaggio provvederà a rimuovere dalla propria struttura dati l'item con la chiave passata come parametro.

3.9.2.6.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
key	privato	immutabile	String	Rappresenta la chiave dell'item da rimuovere

3.9.2.6.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
key	pubblico	String	Metodo che ritorna l'attributo key



3.10 actorbase::actorsystem::messages::warehousemanmessages

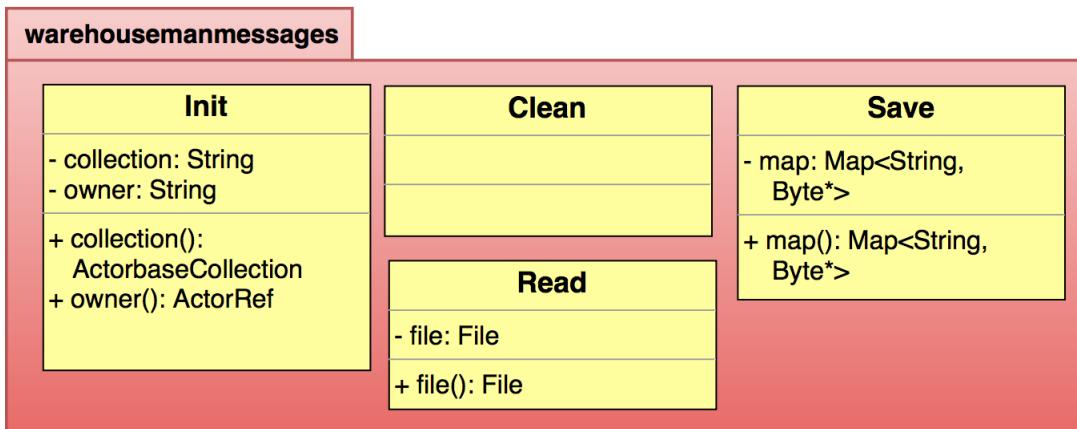


Figura 44: Package actorsystem::messages::warehousemanmessages

3.10.1 Descrizione

Package_G contenente i messaggi processabili dall'attore [Warehouseman](#).

3.10.2 Classi

3.10.2.1 actorbase::actorsystem::messages::warehouseman::Clean

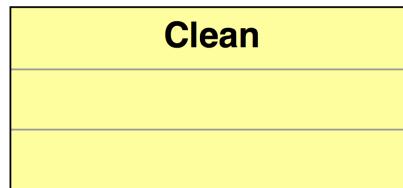


Figura 45: Classe Clean

3.10.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per richiedere allo [Warehouseman](#) di rimuovere i dati rappresentati da esso dal disco.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.



3.10.2.1.2 Utilizzo

Questa classe viene utilizzata per richiedere di rimuovere dal disco tutti i file contenenti dati rappresentati dall'attore in questione.

Questo messaggio è utilizzato ad esempio quando si rimuove una collezione dal sistema.

3.10.2.2 actorbase::actorsystem::messages::warehouseman::Init

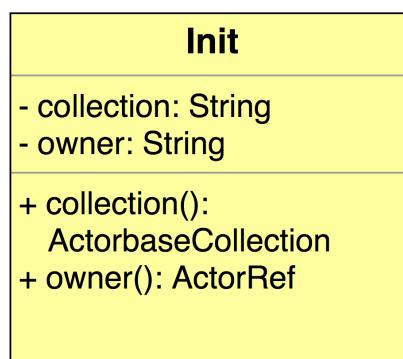


Figura 46: Classe Init

3.10.2.2.1 Descrizione

Classe che rappresenta un messaggio utilizzato per inizializzare l'attore.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching.

3.10.2.2.2 Utilizzo

Questa classe viene utilizzata per inizializzare l'attore con le informazioni necessarie come la collezione che rappresenta.

Questo messaggio viene ricevuto subito dopo la propria creazione in modo da inizializzarlo immediatamente.

3.10.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	String	Rappresenta il nome della collezione che questo attore rappresenta
owner	privato	immutable	String	Rappresenta lo username dell'utente che ha creato la collezione che questo attore rappresenta



3.10.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collection	pubblico	String	Metodo che ritorna l'attributo collection
owner	pubblico	String	Metodo che ritorna l'attributo owner

3.10.2.3 actorbase::actorsystem::messages::warehouseman::Save

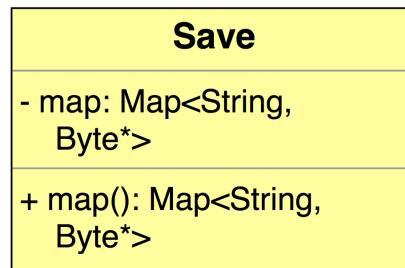


Figura 47: Classe Save

3.10.2.3.1 Descrizione

Classe che rappresenta un messaggio utilizzato per persistere i dati su disco.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.10.2.3.2 Utilizzo

Questa classe viene utilizzata per richiedere all'attore di persistere dati su disco. Quando l'attore riceve questo messaggio provvederà a salvare su filesystem i dati passati come parametri in file che seguono una gerarchia simile alla gerarchia delle collezioni del sistema.

I dati salvati vengono prima criptati tramite l'ausilio della classe [CryptoUtils](#)

3.10.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
map	privato	immutable	Map<String, Array<Byte>>	Rappresenta una mappa di items da salvare su disco

3.10.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
map	pubblico	Map<String, Array<Byte>>	Metodo che ritorna l'attributo map



3.10.2.4 actorbase::actorsystem::messages::warehouseman::Read

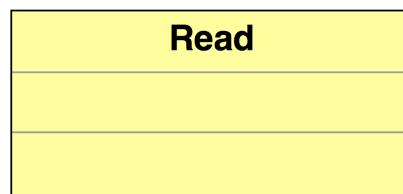


Figura 48: Classe Read

3.10.2.4.1 Descrizione

Classe che rappresenta un messaggio utilizzato per leggere i dati da disco.

Implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.10.2.4.2 Utilizzo

Questa classe viene utilizzata per richiedere all'attore di leggere dati da disco. Quando l'attore riceve questo messaggio provvederà a decriptare, leggere e restituire i dati all'attore richiedente.

Questo messaggio è usato ad esempio quando il sistema subisce un riavvio per ripopolare la struttura coi dati scritti su disco.

3.10.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
file	privato	immutable	File	Rappresenta un file da cui leggere gli items

3.10.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
file	pubblico	File	Metodo che ritorna l'attributo file



3.11 actorbase::actorsystem::messages::managermessages

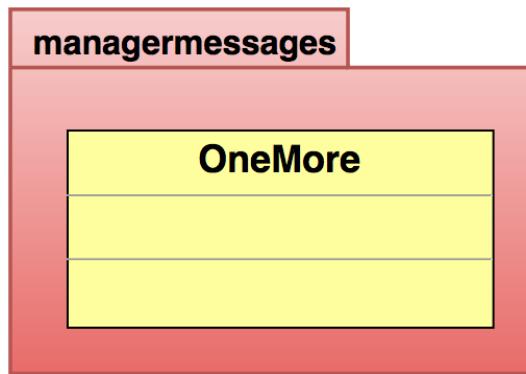


Figura 49: Package actorsystem::messages::managermessages

3.11.1 Descrizione

Package_G contenente i messaggi processabili dall'attore Manager.

3.11.2 Classi

3.11.2.1 actorbase::actorsystem::messages::managermessages::OneMore

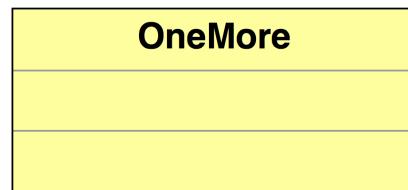


Figura 50: Classe OneMore

3.11.2.1.1 Descrizione

Classe che rappresenta un messaggio utilizzato per alleggerire il carico in ingresso degli attori_G di tipo Storekeeper, creandone di nuovi e aggiungendoli alla pool_G dell'attore_G di tipo Storefinder che li coordina.



3.11.2.1.2 Utilizzo

Viene utilizzato dagli attori_G di tipo **Storekeeper** per notificare il superamento della soglia di coppie chiave-valore impostata da configurazione, che determina quindi un sovraccarico dell'attore_G in questione.

Il messaggio si occupa dunque di generare nuovi attori_G di tipo **Storekeeper** in modo da redistribuire il carico in ingresso.

3.12 actorbase::actorsystem::actors

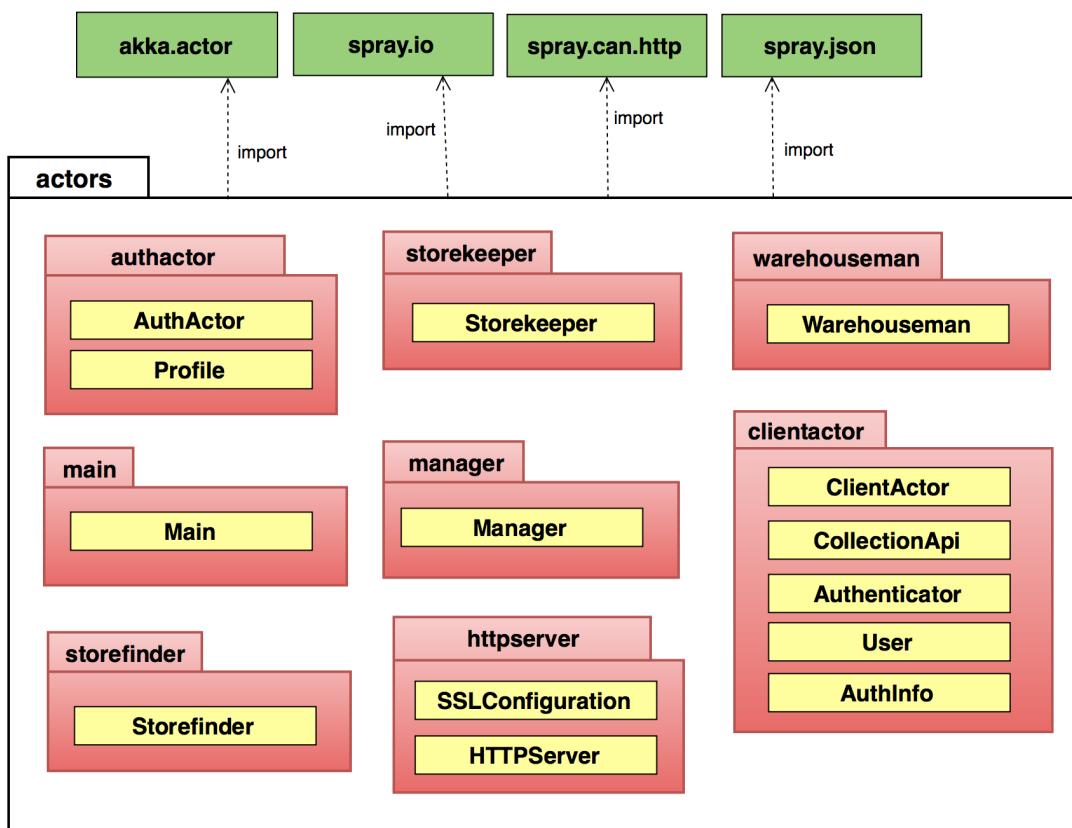


Figura 51: Package_G actors, visione generale

3.12.1 Descrizione

Package_G che raggruppa tutti gli attori necessari al funzionamento del database.



3.12.2 Package contenuti

- actorbase::actorsystem::actors::clientactor;
- actorbase::actorsystem::actors::httpserver;
- actorbase::actorsystem::actors::storefinder;
- actorbase::actorsystem::actors::storekeeper;
- actorbase::actorsystem::actors::warehouseman;
- actorbase::actorsystem::actors::manager;
- actorbase::actorsystem::actors::main;
- actorbase::actorsystem::actors::authactor.

3.13 actorbase::actorsystem::actors::clientactor

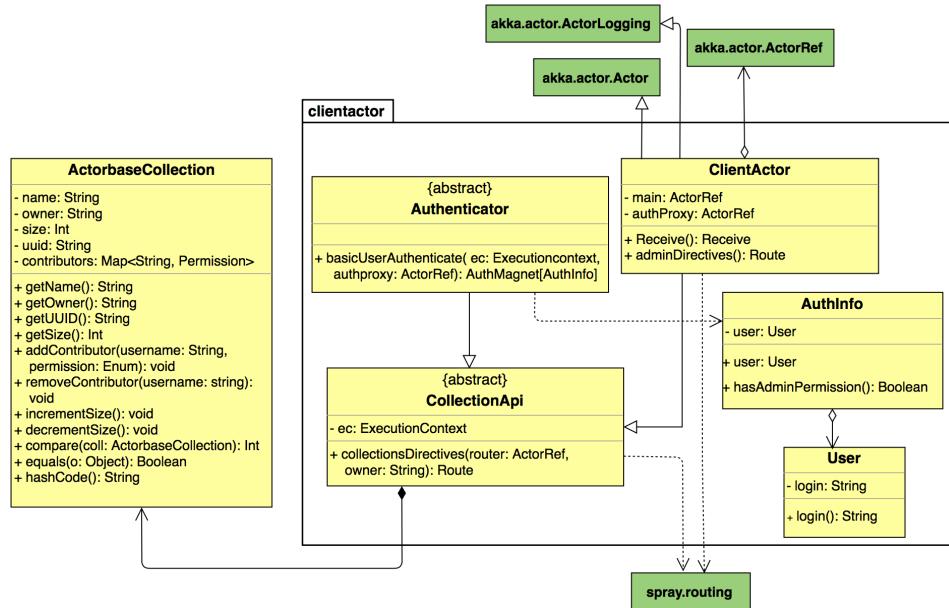


Figura 52: Package_G Clientactor e interazioni con Main

3.13.1 Descrizione

Package_G per l'attore_G con cui si interfacerà il driver_G, fornisce un'interfaccia HTTP_G verso l'esterno.



3.13.2 Classi

3.13.2.1 actorbase::actorsystem::actors::clientactor::Authenticator

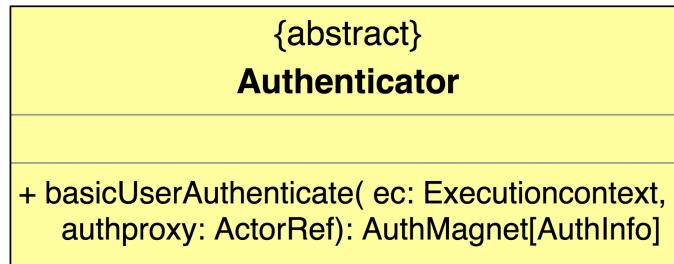


Figura 53: Classe actorsystem::actors::clientactor::Authenticator

3.13.2.1.1 Descrizione

Classe astratta usata per effettuare la validazione di un login.

3.13.2.1.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare il controllo delle credenziali di login mediante richiesta all'attore_c di tipo **AuthActor**.

3.13.2.1.3 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
basicUserAuthenticator	pubblico	AuthMagnet[AuthInfo]	Metodo che riceve e controlla tramite metodi interni le credenziali
validateUser	pubblico	Option[AuthInfo]	Metodo che controlla che le credenziali siano corrette
authenticator	pubblico	Future[Option[AuthInfo]]	Metodo che restituisce un future _c che punta alle credenziali dell'utente autenticato

3.13.2.1.4 Parametri

basicUserAuthenticator



Nome	Tipo	Descrizione
ec	ExecutionContext	Contesto di esecuzione all'interno del quale è stato chiamato questo metodo
authProxy	ActorRef	Riferimento dell'attore _G di tipo AuthActor che si occuperà di validare le credenziali di accesso al sistema

validateUser

Nome	Tipo	Descrizione
userPass	Option[UserPass]	Credenziali di accesso dell'utente che vuole autenticarsi

authenticator

Nome	Tipo	Descrizione
userPass	Option[UserPass]	Credenziali di accesso dell'utente che vuole autenticarsi

3.13.2.2 actorbase::actorsystem::actors::clientactor::User

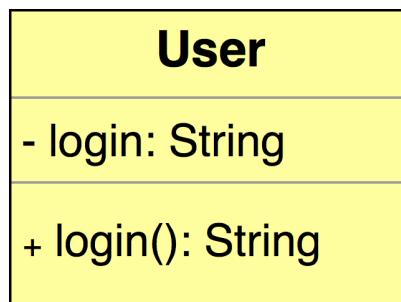


Figura 54: Classe actorsystem::actors::clientactor::User

3.13.2.2.1 Descrizione

Classe che rappresenta un utente collegato e autenticato dall'esterno al sistema mediante richiesta di login_G.

3.13.2.2.2 Utilizzo

Questa classe viene usata per fornire le informazioni di autenticazione all'attore_G di tipo [ClientActor](#) una volta ottenuta la validazione da parte dell'Authenticator, mediante oggetto di tipo [AuthInfo](#).



3.13.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
login	pubblico	immutabile	String	Rappresenta lo username _G dell'utente autenticato all'interno del sistema

3.13.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
login	pubblico	String	Metodo di accesso in sola lettura all'attributo login

3.13.2.3 actorbase::actorsystem::actors::clientactor::AuthInfo

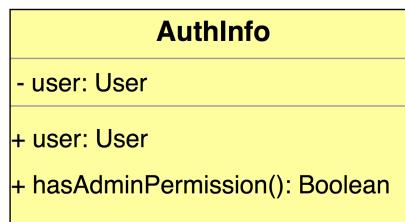


Figura 55: Classe actorsystem::actors::clientactor::AuthInfo

3.13.2.3.1 Descrizione

Classe che rappresenta un utente collegato e autenticato dall'esterno al sistema mediante richiesta di login_G, mantiene inoltre traccia dei privilegi dell'utente.

3.13.2.3.2 Utilizzo

Questa classe viene usata per fornire le informazioni di autenticazione e privilegi sulle operazioni (amministrativi o non) all'attore_G di tipo **ClientActor** una volta ottenuta la validazione da parte dell'**Authenticator**.

3.13.2.3.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
user	privato	immutabile	User	Rappresenta l'oggetto contenente le informazioni dell'utente autenticato



3.13.2.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
user	pubblico	User	Metodo di accesso in sola lettura all'attributo user
hasAdmin Permissions	pubblico	Boolean	Metodo di verifica dei privilegi amministrativi dell'utente autenticato

3.13.2.4 actorbase::actorsystem::actors::clientactor::CollectionApi

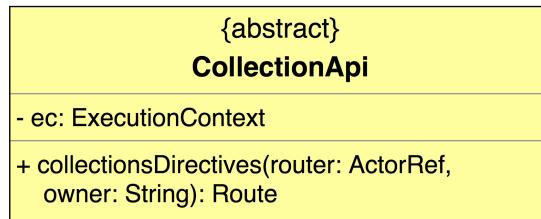


Figura 56: Classe actorsystem::actors::clientactor::CollectionApi

3.13.2.4.1 Descrizione

Classe astratta usata per analizzare le route che richiedono operazioni su collezioni_G. Rappresenta la codifica dell'interfaccia REST_G da esporre all'esterno.

3.13.2.4.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare l'analisi delle route che richiedono operazioni su collezioni_G, implementata utilizzando il package_G directives della libreria Spray.

3.13.2.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
ec	pubblico	immutabile	ExecutionContext	Rappresenta il Contesto di esecuzione all'interno del quale è stata creata questa classe

3.13.2.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collections	pubblico	Route	Metodo che interagisce con il server tramite un router basandosi sulla route immessa dall'utente



3.13.2.4.5 Parametri

collections

Nome	Tipo	Descrizione
owner	String	Proprietario della collezione _G su cui si vuole operare
router	ActorRef	Riferimento dell'attore che si occuperà di comunicare con un attore main _G per effettuare le operazioni richieste

3.13.2.5 actorbase::actorsystem::actors::clientactor::ClientActor

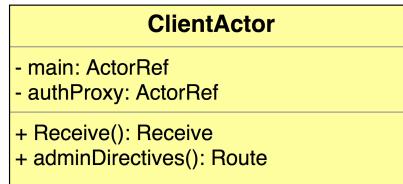


Figura 57: Classe actorsystem::actors::clientactor::ClientActor

3.13.2.5.1 Descrizione

Classe che rappresenta l'attore_G con cui si interfaccia il driver_G dopo la connessione.

3.13.2.5.2 Utilizzo

Questo attore_G riceve le richieste da actorbase::driver::client::Connection e si occupa di inviare messaggi a actorbase::actorsystem::actors::main::Main.

Questa classe inoltre si occupa di capire che tipo di richiesta è stata fatta dal driver_G basandosi sulla route di quest'ultima.

3.13.2.5.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging;
- com::actorbase::actorsystem::actors::clientactor::Authenticator;
- com::actorbase::actorsystem::actors::clientactor::CollectionApi.



3.13.2.5.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
router	pubblico	immutabile	ActorRef	Rappresenta il riferimento di un attore che si occuperà di gestire le richieste verso i vari main _G
authProxy	pubblico	immutabile	ActorRef	Rappresenta il riferimento dell' attore _G di tipo AuthActor, utilizzato per effettuare validazione delle credenziali di accesso e autenticazione
timeout	pubblico	immutabile	Timeout	Rappresenta il tempo limite oltre il quale se un'operazione non è conclusa viene lanciato un errore di timeout

3.13.2.5.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
adminDirectives	pubblico	Route	Metodo utilizzato per definire le direttive di amministrazione
receive	pubblico	Actor.Receive	Effettua operazioni diverse a seconda del tipo di messaggio ricevuto



3.14 actorbase::actorsystem::actors::httpserver

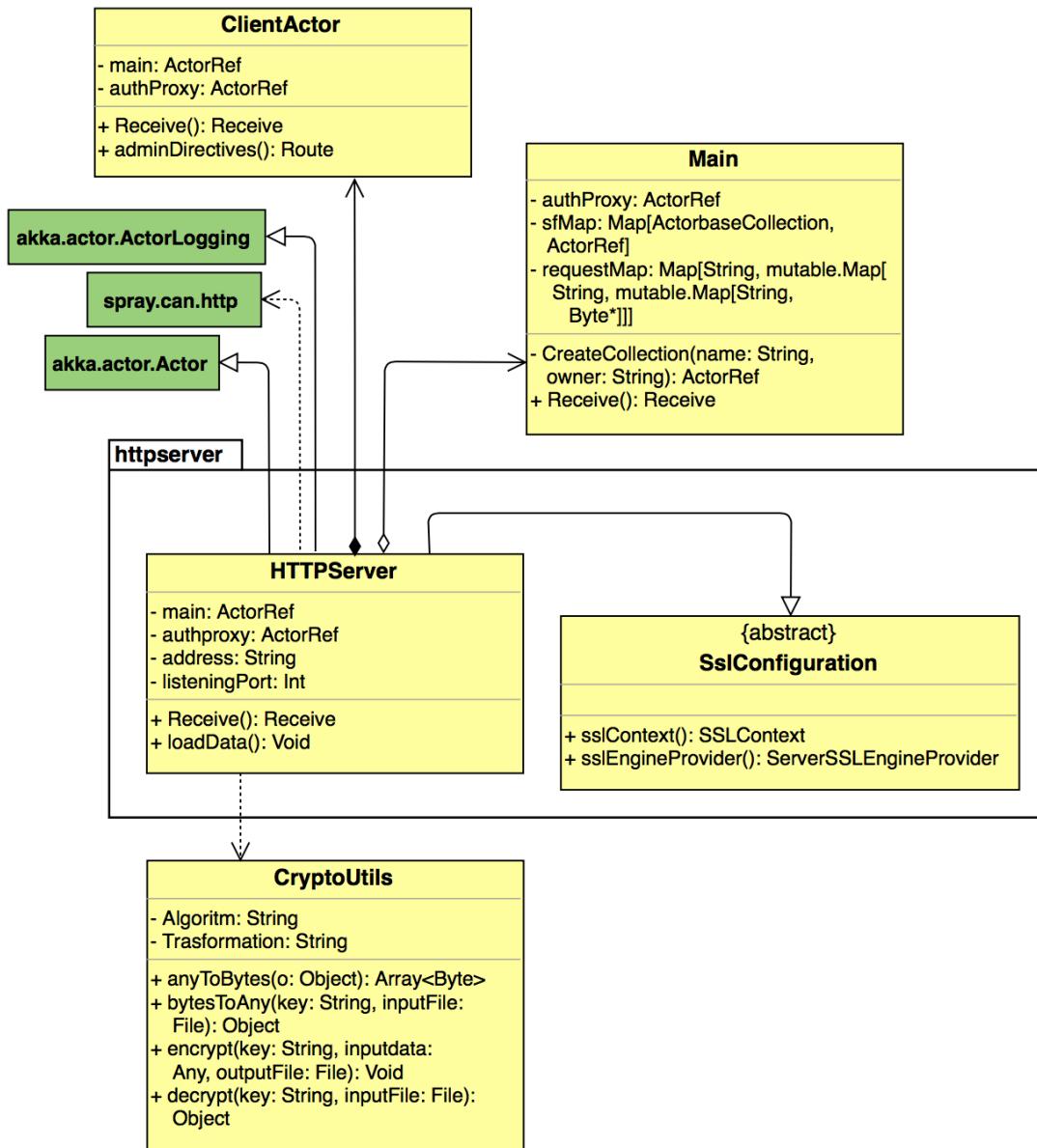


Figura 58: httpserver, visione generale del package



3.14.1 Descrizione

Package_G per l'attore_G con cui si interfacerà il driver_G per la connessione iniziale.

3.14.2 Interfacce

3.14.2.1 actorbase::actorsystem::actors::httpserver::SSLConfiguration



Figura 59: Classe actorsystem::actors::httpserver::SslConfiguration

3.14.2.1.1 Descrizione

Interfaccia che delinea il comportamento del motore di crittografia da utilizzare in caso di avvio del sistema con crittografia SSL/TLS_G attivata, mediante la gestione di protocolli di cifratura e certificati.

3.14.2.1.2 Utilizzo

Definisce il comportamento del motore di crittografia da applicare alla classe che si occupa di ricezione e invio dati, da e verso l'esterno, trattandosi in **Actorbase** di un server_G HTTP_G, la classe designata alla realizzazione di questa interfaccia è **HTTPServer**.

L'interfaccia dà una definizione default_G di due metodi impliciti, sslContext e sslEngineProvider i quali si occupano rispettivamente del caricamento dei certificati forniti e della lista di protocolli di crittografia (SSL, TLSv1, TLSv1.1..etc) e cifrari da utilizzare nella comunicazione.

3.14.2.1.3 Realizzata da

- actorbase::actorsystem::httpserver::HTTPServer.



3.14.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
sslContext	pubblico	SSLContext	metodo implicito per il caricamento dei certificati TLS/SSL _G
sslEngineProvider	pubblico	ServerSSLEngineProvider	metodo implicito per impostare una cipher suite _G e una lista di protocolli di crittografia da supportare

3.14.3 Classi

3.14.3.1 actorbase::actorsystem::actors::httpserver::HTTPServer

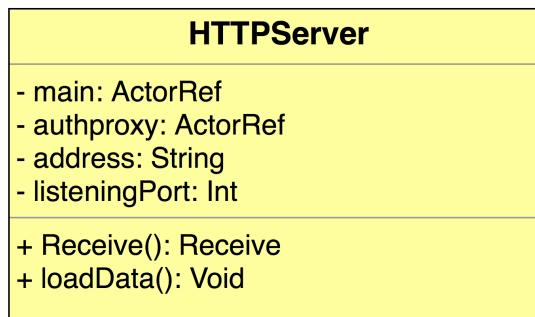


Figura 60: Classe actorsystem::actors::httpserver::HTTPServer

3.14.3.1.1 Descrizione

Classe che rappresenta l'attore_G con cui si interfaccia il driver_G per istanziare la connessione. Si occupa di accettare richieste di tipo HTTP_G e rispondere in formato JSON_G alle richieste esterne.

3.14.3.1.2 Utilizzo

Questo attore_G riceve la richiesta di connessione da [ActorbaseClient](#) e si occupa di associare al client_G un attore_G di tipo [ClientActor](#) per continuare le comunicazioni.

3.14.3.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging;
- actorbase::actorsystem::actors::clientactor::SSLConfiguration.



3.14.3.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
router	privato	immutabile	ActorRef	Rappresenta un riferimento all'attore che si occupa del routing delle richieste agli attori di tipo Main
address	privato	immutabile	String	Rappresenta l'indirizzo del server _g su cui aprire il socket _g di ascolto
listenPort	privato	immutabile	Int	Rappresenta la porta in cui il server si mette in ascolto
authProxy	privato	immutabile	ActorRef	Rappresenta un riferimento all'attore di tipo AuthActor

3.14.3.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
loadData	pubblico	void	Effettua la lettura della persistenza criptata su disco e ripopola il sistema all'atto dell'accensione
receive	pubblico	Actor.Receive	Effettua operazioni diverse a seconda del tipo di messaggio ricevuto



3.15 actorbase::actorsystem::actors::warehouseman

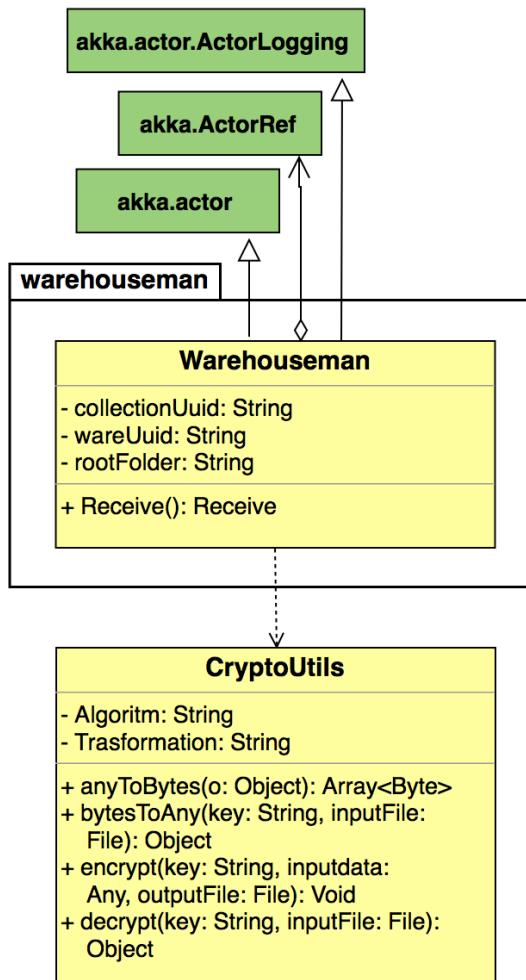


Figura 61: Package `actorsystem::actors::warehouseman`

3.15.1 Descrizione

Package_G che rappresenta l'attore_G che si occuperà della persistenza_G su disco dei dati.



3.15.2 Classi

3.15.2.1 actorbase::actorsystem::actors::warehouseman::Warehouseman

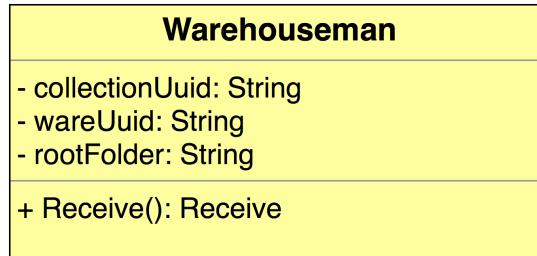


Figura 62: Classe actorsystem::actors::warehouseman::Warehouseman

3.15.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo Warehouseman_G, si occupa di persistere i dati su disco, criptandoli mediante i metodi della classe [CryptoUtils](#).

3.15.2.1.2 Utilizzo

Questa classe viene utilizzata per effettuare il salvataggio su filesystem del database_G criptandoli mediante i metodi della classe [CryptoUtils](#), ogni attore_G di tipo [Storekeeper](#) contiene un riferimento a questa classe, periodicamente invia un messaggio di salvataggio del proprio contenuto.

3.15.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.

3.15.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collectionShard	privato	immutable	String	Rappresenta il nome della cartella dentro cui verranno salvati i dati
wareUUID	privato	immutable	String	Rappresenta un identificativo univoco universale
rootFolder	privato	immutable	String	Rappresenta il percorso della directory _G su cui persistere i dati



3.15.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
collectionShard	pubblico	String	Metodo di accesso in sola lettura all'attributo collection-Shard
receive	pubblico	void	Effettua operazioni diverse a seconda del tipo di messaggio ricevuto

3.16 actorbase::actorsystem::actors::main

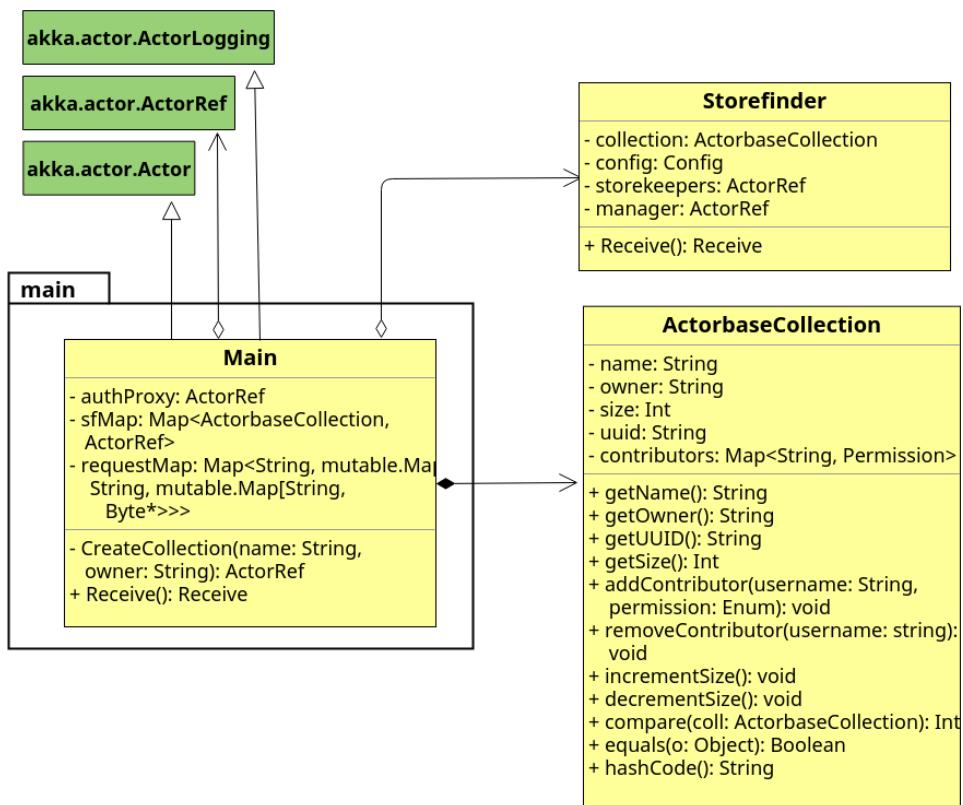


Figura 63: Package dell'attore di tipo Main

3.16.1 Descrizione

Package_G che rappresenta l'attore_G che si occuperà di gestire le richieste al server_G, inoltrandole agli attori_G di tipo `Storefinder`.



3.16.2 Classi

3.16.2.1 actorbase::actorsystem::actors::main::Main

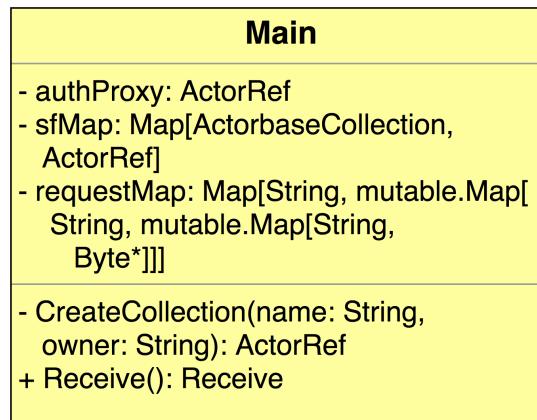


Figura 64: Classe actorsystem::actors::main::Main

3.16.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo Main_G. Questo attore_G si occupa di gestire le richieste di lettura e scrittura provenienti dal ClientActor, inoltrandole agli attori_G di tipo Storefinder. Mantiene inoltre una mappa che associa le richieste di lettura agli utenti che le effettuano, in modo da fornire le risposte corrette al giusto richiedente una volta completato il processo di lettura.

3.16.2.1.2 Utilizzo

Questa classe viene utilizzata per gestire le richieste ricevute dall'attore_G di tipo ClientActor. Rappresenta la classe principale che fa uso dell'estensione akka_G cluster sharding, viene infatti "frammentata" in regioni che manterranno ognuna un insieme di identificativi scelti nell'algoritmo di instradamento dei messaggi, permettendo quindi una distribuzione degli attori figli (Storefinder e Storekeeper) equa all'interno del cluster_G e un sistema di invio messaggi trasparente all'utilizzatore.

3.16.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.



3.16.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
sfMap	privato	mutabile	Map< ActorbaseCollection , ActorRef>	Rappresenta una mappa contenente come chiavi degli oggetti ActorbaseCollection e come valore i riferimenti degli Storefinder che mappano quelle chiavi
requestMap	privato	mutabile	Map<ActorRef, Map< ActorbaseCollection , Map<String, Object> >	Struttura dati atta a restituire intere collezioni

3.16.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
receive	pubblico		Effettua operazioni diverse a seconda del tipo di messaggio ricevuto
extractShardId	pubblico	ExtractShardId	Metodo di servizio utilizzato dall'estensione Akka _G cluster sharding, permette di ottenere l'identificativo della regione _G di riferimento
extractEntityId	pubblico	ExtractEntityId	Metodo di servizio utilizzato dall'estensione Akka _G cluster sharding, permette di ottenere l'identificativo dell'attore _G all'interno della regione _G di riferimento



3.17 actorbase::actorsystem::actors::authactor

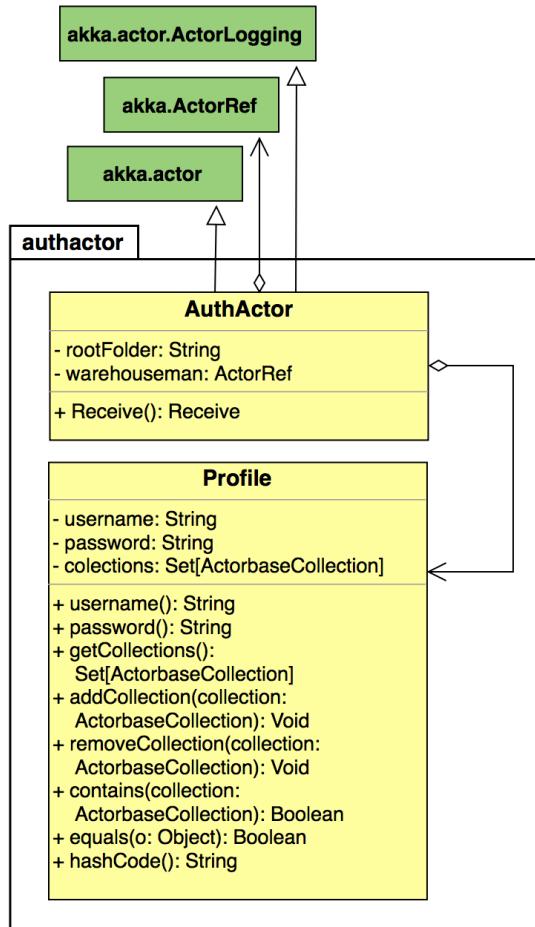


Figura 65: AuthActor, visione generale del package

3.17.1 Descrizione

Package_G che rappresenta un attore_G di tipo AuthActor_G, contiene tutte le classi di utilità da utilizzare nei processi di autenticazione e gestione degli utenti del sistema.



3.17.2 Classi

3.17.2.1 actorbase::actorsystem::actors::authactor::AuthActor

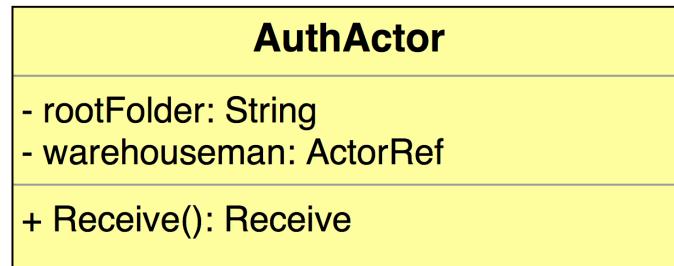


Figura 66: Classe actorsystem::actors::authactor::AuthActor

3.17.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo AuthActor_G, permette di processare le operazioni di gestione degli utenti, sia dal punto di vista amministrativo sia comune. Permette quindi di inserire, rimuovere e modificare dati inerenti l'utenza del sistema, incluse collezioni_G possedute o di contribuzione.

3.17.2.1.2 Utilizzo

Questa classe viene utilizzata per istanziare un attore_G di tipo AuthActor_G, utilizzato nella gestione degli utenti del sistema, dalle credenziali alle collezioni_G di appartenenza o di contribuzione dell'utente stesso.

3.17.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.

3.17.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
profiles	private	immutabile	Set _G di Pro-file	Rappresenta un insieme di profili utente
rootFolder	private	immutabile	String	Rappresenta il percorso dei file da persistere nel filesystem _G utilizzando algoritmi di criptazione



3.17.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
receive	pubblico	Actor.Receive	Effettua operazioni diverse a seconda del tipo di messaggio ricevuto
persist	pubblico	void	Effettua la scrittura dei dati su disco utilizzando algoritmi di criptazione predefiniti mediante configurazione

3.17.2.1.6 Parametri

persist

Nome	Tipo	Descrizione
profiles	Set _G di Profile	Rappresenta un insieme di profili utente da persistere su filesystem _G

3.17.2.2 actorbase::actorsystem::actors::authactor::Profile

Profile
<ul style="list-style-type: none">- username: String- password: String- collections: Set[ActorbaseCollection] + username(): String+ password(): String+ getCollections(): Set[ActorbaseCollection]+ addCollection(collection: ActorbaseCollection): Void+ removeCollection(collection: ActorbaseCollection): Void+ contains(collection: ActorbaseCollection): Boolean+ equals(o: Object): Boolean+ hashCode(): String

Figura 67: Classe actorsystem::actors::authactor::AuthActor



3.17.2.2.1 Descrizione

Classe che rappresenta un profilo_G utente, incapsula le credenziali di accesso al sistema e l'insieme delle collezioni_G a cui l'utente ha accesso, sia in lettura che in scrittura.

3.17.2.2.2 Utilizzo

Questa classe viene utilizzata per istanziare un profilo_G utente, utilizzato nella gestione degli utenti del sistema, dalle credenziali alle collezioni_G di appartenenza o di contribuzione dell'utente stesso. Contiene metodi di accesso in sola lettura agli attributi che rappresentano le credenziali di accesso al sistema e all'insieme delle collezioni_G accedibili all'utente.

3.17.2.2.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutable	String	Rappresenta lo username _G dell'utente
password	privato	immutable	String	Rappresenta la password _G associata all'utente
collections	privato	mutabile	Set _G di ActorbaseCollection	Rappresenta un insieme di collezioni _G accedibili all'utente, senza distinzione tra permessi di sola lettura o lettura e scrittura

3.17.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo di accesso in sola lettura all'attributo username _G
password	pubblico	String	Metodo di accesso in sola lettura all'attributo password _G
getCollections	pubblico	Set _G di ActorbaseCollection	
addCollection	pubblico	void	Aggiunge un oggetto di tipo ActorbaseCollection al Set _G dell'utente
removeCollection	pubblico	void	Rimuove un oggetto di tipo ActorbaseCollection dal Set _G dell'utente
contains	pubblico	Boolean	Controlla l'appartenenza di un oggetto di tipo ActorbaseCollection al Set _G dell'utente
equals	pubblico	Boolean	Confronta due oggetti di tipo Profile
hashCode	pubblico	int	Ritorna l'hashcode _G dell'oggetto Profile

3.17.2.2.5 Parametri

addCollection

Nome	Tipo	Descrizione
collection	ActorbaseCollection	Rappresenta un oggetto di tipo ActorbaseCollection da aggiungere al Set _G dell'utente

**removeCollection**

Nome	Tipo	Descrizione
collection	ActorbaseCollection	Rappresenta un oggetto di tipo ActorbaseCollection da rimuovere dal Set _G dell'utente

contains

Nome	Tipo	Descrizione
collection	ActorbaseCollection	Rappresenta un oggetto di tipo ActorbaseCollection di cui controllare l'esistenza nel Set _G

equals

Nome	Tipo	Descrizione
o	Object	Rappresenta un oggetto generico da confrontare con l'oggetto di tipo Profile



3.18 actorbase::actorsystem::actors::storefinder

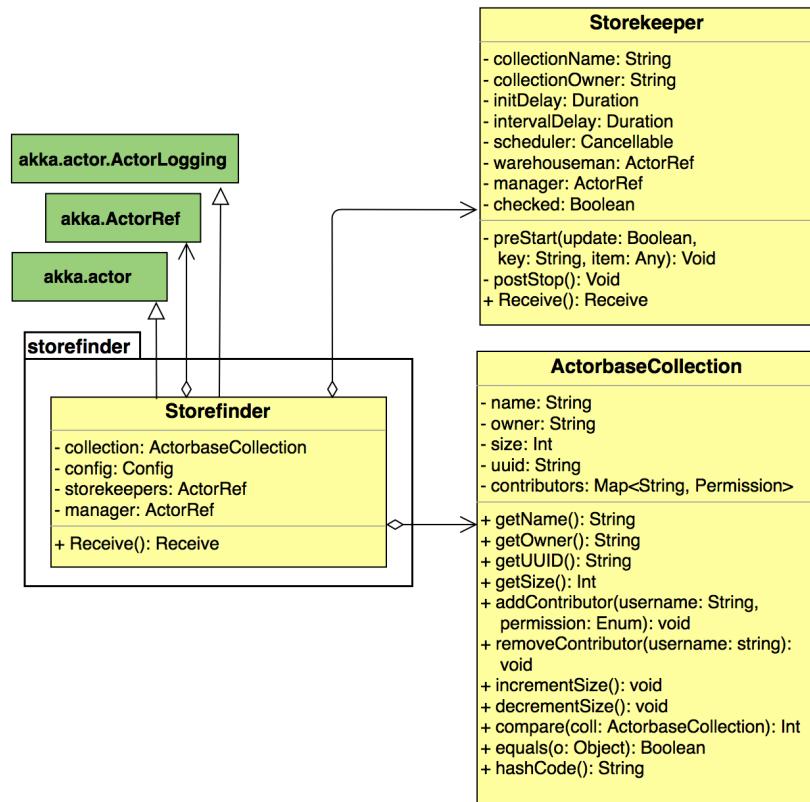


Figura 68: Storefinder, visione generale del package

3.18.1 Descrizione

Package_G che rappresenta un attore_G di tipo Storefinder_G.



3.18.2 Classi

3.18.2.1 actorbase::actorsystem::actors::storefinder::Storefinder

Storefinder	
- collection:	ActorbaseCollection
- config:	Config
- storekeepers:	ActorRef
- manager:	ActorRef
+ Receive()	: Receive

Figura 69: Classe actorsystem::actors::storefinder::Storefinder

3.18.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo Storefinder_G, si occupa di ricevere richieste di interrogazione e modifica dall'esterno, ossia dall'attore di tipo Main e rappresenta idealmente un indice sulle collezioni_G del sistema.

3.18.2.1.2 Utilizzo

Questa classe viene utilizzata per tener traccia degli attori di tipo actorbase::actorsystem::actors::storekeeper::Storekeeper e il soddisfacimento delle richieste proveniente dall'esterno, le quali verranno inoltrate agli attori_G di tipo Sto-rekeeper. Rappresenta una sorta di indice sulle collezioni_G contenute negli Storekeeper, e si occupa inoltre di distribuire il carico di questi ultimi su più nodi_G del cluster_G.

3.18.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.



3.18.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	mutabile	actorbaseCollection	Rappresenta il nome della collezione _G che va a rappresentare
config	privato	immutabile	Config	Rappresenta il contenuto della configurazione statica da attribuire agli attori di tipo Storekeeper , e impostazioni generali
storekeepers	privato	immutabile	ActorRef	Rappresenta l'insieme degli attori _G di tipo Storekeeper da distribuire all'interno del cluster _G
manager	privato	immutabile	int	Rappresenta l'attore _G di tipo Manager dedicato al bilanciamento degli Storekeeper figli

3.18.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
receive	pubblico		Effettua operazioni diverse a seconda del tipo di messaggio ricevuto



3.19 actorbase::actorsystem::actors::storekeeper

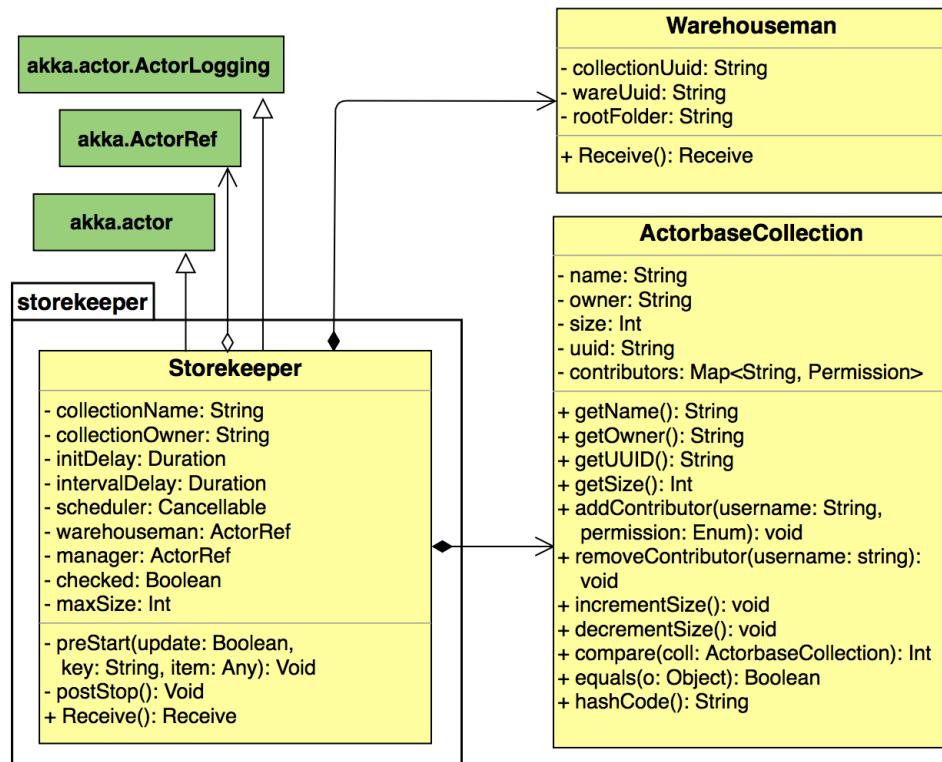


Figura 70: Storekeeper, visione generale del package

3.19.1 Descrizione

Package_G che rappresenta l'attore_G di tipo Storekeeper_G.



3.19.2 Classi

3.19.2.1 actorbase::actorsystem::actors::storekeeper::Storekeeper

Storekeeper	
- collectionName: String	
- collectionOwner: String	
- initDelay: Duration	
- intervalDelay: Duration	
- scheduler: Cancellable	
- warehouseman: ActorRef	
- manager: ActorRef	
- checked: Boolean	
- maxSize: Int	
- preStart(update: Boolean, key: String, item: Any): Void	
- postStop(): Void	
+ Receive(): Receive	

Figura 71: Classe actorsystem::actors::storekeeper::Storekeeper

3.19.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo Storekeeper_G, contiene un frammento di un'intera collezione_G, il numero di attori di questo tipo è decidibile mediante configurazione statica, tuttavia è variabile dinamicamente mediante euristiche basate sul carico effettuate dall'attore_G di tipo Manager.

3.19.2.1.2 Utilizzo

Questa classe viene utilizzata per salvare nel database_G gli item_G, contiene infatti un frammento di un'intera collezione_G, il numero di attori di questo tipo è decidibile mediante configurazione statica, tuttavia è variabile dinamicamente mediante euristiche basate sul carico effettuate dall'attore_G di tipo Manager, inoltre è possibile decidere il numero di coppie chiave-valore contenute all'interno di questi attori_G, in modo da avere una metrica di decisione sull'aggiunta di eventuali Storekeeper_G extra da parte del Manager.

3.19.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.



3.19.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
data	privato	mutabile	Map <String, Array<Byte>>	Rappresenta l'insieme di item _G contenuta nell'attore
collectionName	privato	immutabile	String	Rappresenta il nome della collezione _G a cui lo Storekeeper _G appartiene
collectionOwner	privato	immutabile	String	Rappresenta il nome dell'autore della collezione _G a cui lo Storekeeper _G appartiene
maxSize	privato	immutabile	Int	Rappresenta il numero massimo di item _G che possono essere contenuti
warehouseman	privato	immutabile	ActorRef	Rappresenta il Warehouseman _G responsabile del salvataggio dei dati dello Storekeeper _G
manager	privato	mutabile	Option<ActorRef>	Rappresenta il riferimento all'attore _G di tipo Manager di riferimento responsabile dell'aggiunta di nuovi Storekeeper _G in caso di forte carico
initDelay	privato	immutabile	Duration	Rappresenta l'intervallo di tempo dopo il primo inserimento dopo cui iniziare il ciclo di persistenza dei dati
intervalDelay	privato	immutabile	Duration	Rappresenta l'intervallo di tempo dei cicli di persistenza dei dati
scheduler	privato	mutabile	Cancellable	Rappresenta lo scheduler offerto dalla libreria Akka _G , tiene traccia dei timer _G di persistenza dei dati
checked	privato	mutabile	Boolean	Rappresenta una flag _G di controllo sul carico in ingresso, permette di avvisare il Manager in caso di carico pesante

3.19.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
receive	pubblico	Actor.Receive	Effettua operazioni diverse a seconda del tipo di messaggio ricevuto
insertOrUpdate	pubblico	Boolean	Inserisce un item nella collezione _G o lo aggiorna in caso di item _G già presente
preStart	pubblico	void	Metodo di definizione del comportamento dell'attore _G durante la fase di inizializzazione
postStop	pubblico	void	Metodo di definizione del comportamento dell'attore _G durante la fase di spegnimento

3.19.2.1.6 Parametri

insertOrUpdate



Nome	Tipo	Descrizione
update	Boolean	Variabile che, se uguale a true, indica che l'inserimento da effettuare è con sovrascrittura
key	String	Variabile che indica la chiave dell'item _G da inserire
value	Object	Variabile che indica il valore dell'item _G da inserire

3.20 actorbase::actorsystem::actors::manager

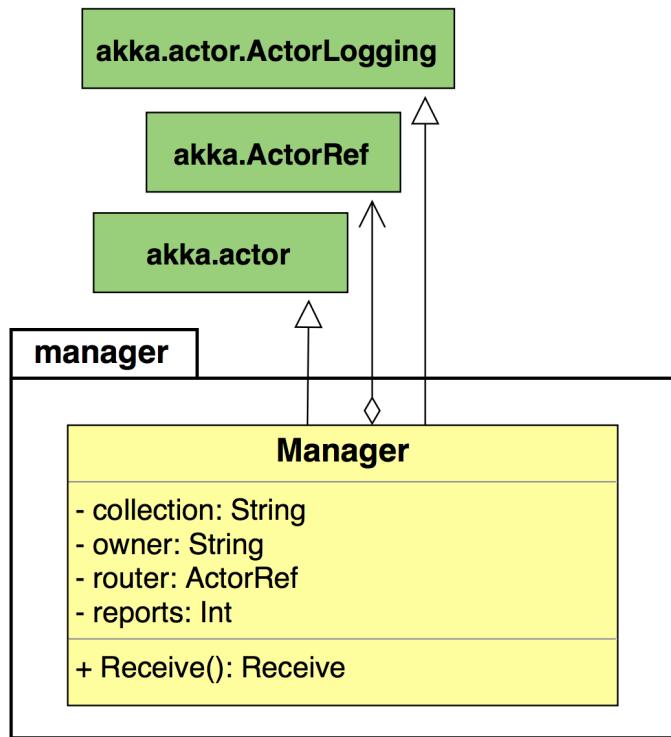


Figura 72: Manager, visione generale del package

3.20.1 Descrizione

Package_G che rappresenta l'attore_G di tipo Manager_G.



3.20.2 Classi

3.20.2.1 actorbase::actorsystem::actors::manager::Manager

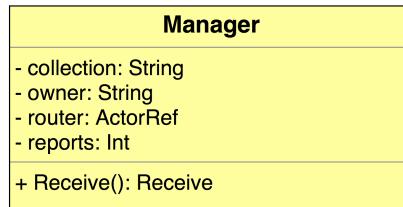


Figura 73: Classe actorsystem::actors::manager::Manager

3.20.2.1.1 Descrizione

Classe che rappresenta un attore_G di tipo Manager_G, dedicato al tracciamento delle coppie-chiave valore dei vari Storekeeper e all'eventuale redistribuzione del carico degli attori_G sotto sforzo.

3.20.2.1.2 Utilizzo

Questa classe viene utilizzata per gestire le richieste di divisione di attori di tipo Storekeeper, ogni volta che uno di questi attori_G supera una soglia predefinita di ingressi, invia un messaggio a questo attore_G, e secondo euristiche predefinite viene deciso se creare un nuovo Storekeeper da aggiungere al router_G dedicato.

3.20.2.1.3 Classi ereditate

- akka::actor::Actor;
- akka::actor::ActorLogging.

3.20.2.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
collection	privato	immutabile	String	Nome della collezione di riferimento la creazione di eventuali nuovi Storekeeper
owner	privato	immutabile	String	Nome dell'autore della collezione di riferimento per la creazione di eventuali nuovi Storekeeper
router	privato	immutabile	ActorRef	Riferimento all'attore di tipo Storekeeper responsabile della gestione e dell'instradamento dei messaggi verso la pool _G di attori Storekeeper dedicati al salvataggio delle coppie chiave-valore
reports	privato	immutabile	int	Indica il numero di richieste da parte di attori _G di tipo Storekeeper sotto sforzo



3.20.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
receive	pubblico		Effettua operazioni diverse a seconda del tipo di messaggio ricevuto

3.21 actorbase::driver

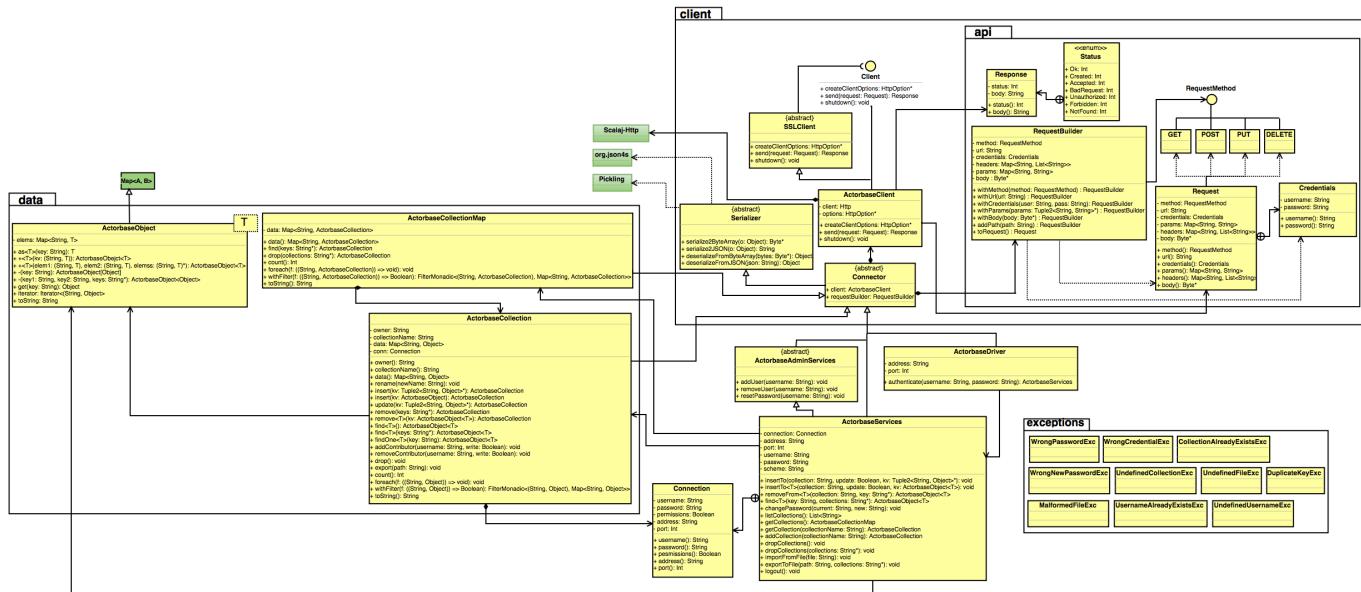


Figura 74: Driver: Architettura generale

3.21.1 Descrizione

Package_G per la componente driver_G da utilizzare in un programma Scalate_G, contiene le classi e i package_G necessari all'interrogazione e scrittura del contenuto del database_G.

3.21.2 Package contenuti

- actorbase::driver::client;
- actorbase::driver::client::api;
- actorbase::driver::data;
- actorbase::driver::exceptions.



3.21.3 Classi

3.21.3.1 ActorbaseDriver

ActorbaseDriver	
- address:	String
- port:	Int
+ authenticate(username: String, password: String):	ActorbaseServices

Figura 75: Driver: Classe ActorbaseDriver

3.21.3.1.1 Descrizione

Classe principale della componente driver_G , rappresenta l'interfaccia di connessione e autenticazione con il server $_G$. Funge da oggetto factory $_G$: è possibile infatti ottenere un' istanza della classe **ActorbaseServices** utilizzabile per interrogare e inviare comandi al sistema lato server $_G$.

3.21.3.1.2 Utilizzo

Espone i metodi di connessione e autenticazione al server $_G$ **Actorbase**, e restituisce un oggetto di tipo ActorbaseServices in base ai privilegi del proprio profilo nel sistema.

3.21.3.1.3 Eredità

- actorbase::driver::client::Connector.

3.21.3.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
address	privato	immutabile	String	Rappresenta l'indirizzo del server $_G$ Actorbase
port	privato	immutabile	Int	Rappresenta la porta esposta dal server $_G$

3.21.3.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
authenticate	pubblico	ActorbaseServices	Metodo di autenticazione sul server mediante credenziali



3.21.3.1.6 Parametri

authenticate

Nome	Tipo	Descrizione
username	String	Rappresenta lo username _G dell'utente
password	String	Rappresenta la password dell'utente

3.21.3.2 ActorbaseServices

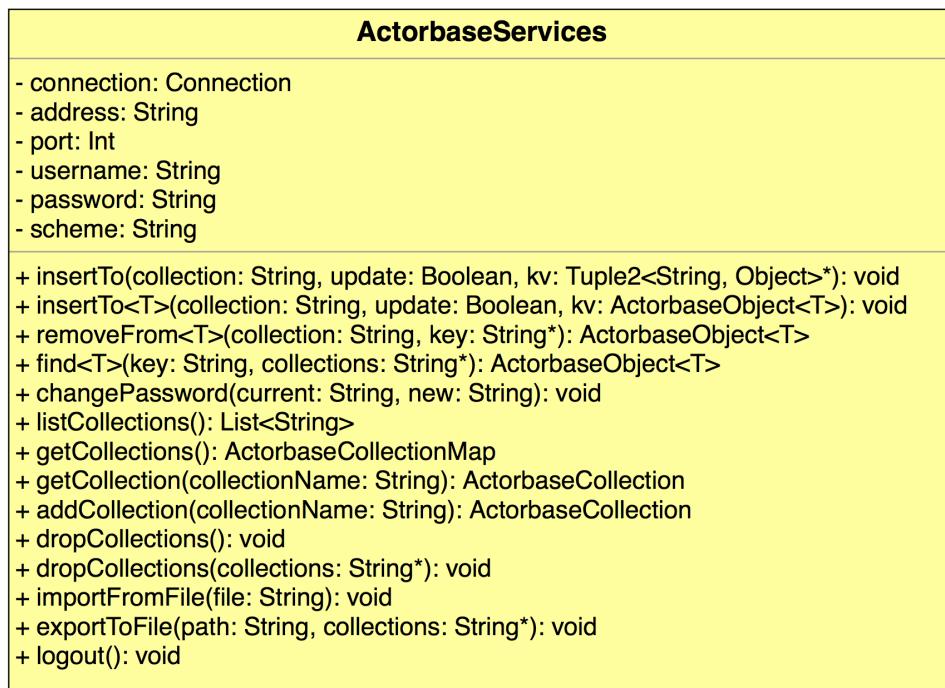


Figura 76: Driver: Classe ActorbaseServices

3.21.3.2.1 Descrizione

Classe principale della componente driver_G, rappresenta l'interfaccia contenente i metodi designati all'utilizzo di **Actorbase** in un programma Scala_G. Attraverso questa classe è possibile eseguire operazioni di lettura e scrittura sulle collezioni_G del sistema **Actorbase** e ottenere degli oggetti che facilitino l'interrogazione o la modifica dello stato presente del sistema.



3.21.3.2.2 Utilizzo

Espone i metodi di interrogazione, ricerca e gestione delle collezioni_G remote all'interno del database_G fornendo una struttura che faciliti le operazioni di lettura e scrittura di queste ultime.

3.21.3.2.3 Eredità

- actorbase::driver::client::Connector.
- actorbase::driver::ActorbaseAdminServices.

3.21.3.2.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username _G dell'utilizzatore del sistema Actorbase
password	privato	immutabile	String	Rappresenta la password associata allo username dell'utilizzatore del sistema Actorbase
address	privato	immutabile	String	Rappresenta l'indirizzo del server _G Actorbase
port	privato	immutabile	Int	Rappresenta la porta esposta dal server _G
scheme	privato	immutabile	String	Attributo隐式的, rappresenta il prefisso HTTP _G da utilizzare nella connessione e invio comandi al server _G
connection	privato	immutabile	Connection	Attributo隐式的, rappresenta un oggetto di tipo Connection da utilizzare durante l'invio di comandi al server _G



3.21.3.2.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
insertTo	pubblico	void	Metodo utilizzato per l'inserimento di nuovi item _G direttamente alla collezione _G specificata senza doverla preventivamente scaricare
insertTo<T> (overload)	pubblico	void	Metodo utilizzato per l'inserimento di nuovi item _G direttamente alla collezione _G specificata senza doverla preventivamente scaricare
removeFrom<T>	pubblico	ActorbaseObject	Metodo utilizzato per la rimozione di item _G direttamente dalla collezione _G specificata senza doverla preventivamente scaricare
find<T>	pubblico	ActorbaseObject	Metodo utilizzato per la ricerca di item _G direttamente dalle collezioni specificate senza doverle preventivamente scaricare
changePassword	pubblico	void	Metodo utilizzato per la modifica della propria password di profilo Actorbase
listCollections	pubblico	List[String]	Ritorna una lista di stringhe, rappresentano i nomi delle collezioni
getCollections	pubblico	ActorbaseCollection Map	Ritorna un oggetto di tipo ActorbaseCollectionMap
getCollection	pubblico	ActorbaseCollection	Ritorna un oggetto di tipo ActorbaseCollection
addCollection	pubblico	ActorbaseCollection	Ritorna un oggetto di tipo ActorbaseCollection
dropCollection	pubblico	ActorbaseCollection	Ritorna un oggetto di tipo ActorbaseCollection
importFromFile	pubblico	void	Metodo da utilizzare per importare collezioni _G sul server _G da file _G
exportToFile	pubblico	void	Metodo da utilizzare per esportare collezioni _G dal server _G su filesystem _G
logout	pubblico	void	Effettua il logout dal server

3.21.3.2.6 Parametri

insertTo

Nome	Tipo	Descrizione
collection	String	Rappresenta la collezione _G a cui effettuare l'inserimento del nuovo item _G
kv	vararg _G di Tuple2<String, Object>	Rappresenta la coppia chiave-valore da inserire nella collezione _G specificata, insieme formano il nuovo item _G da inserire
update	Boolean	Rappresenta un flag di sovrascrittura sull' item _G da inserire nella collezione _G specificata

**insertTo<T> (overload)**

Nome	Tipo	Descrizione
collection	String	Rappresenta la collezione _G a cui effettuare l'inserimento del nuovo item _G
kv	ActorbaseObject	Rappresenta un insieme di coppie chiave-valore da inserire nella collezione _G specificata, insieme formano una lista di nuovi item _G da inserire
update	Boolean	Rappresenta un flag di sovrascrittura sull' item _G da inserire nella collezione _G specificata

removeFrom<T>

Nome	Tipo	Descrizione
collection	String	Rappresenta la collezione _G a cui effettuare l'inserimento del nuovo item _G
key	String	Rappresenta la chiave associata ad un valore da rimuovere dalla collezione _G specificata

find<T>

Nome	Tipo	Descrizione
key	String	Rappresenta la chiave associata ad un valore da cercare nella collezione _G specificata
collections	vararg _G di String	Rappresenta una sequenza di collezioni _G su cui effettuare la ricerca della chiave specificata

changePassword

Nome	Tipo	Descrizione
current	String	Rappresenta la password attuale sul profilo Actorbase
new	String	Rappresenta la nuova password di profilo Actorbase

getCollection

Nome	Tipo	Descrizione
collectionName	String	Rappresenta il nome della collezione _G da richiedere al server _G

addCollection

Nome	Tipo	Descrizione
collectionName	String	Rappresenta il nome della collezione _G da richiedere al server _G



importFromFile

Nome	Tipo	Descrizione
path	String	Rappresenta il percorso su filesystem _G del file _G contenente le collezioni _G da importare sul server _G

exportToFile

Nome	Tipo	Descrizione
path	String	Rappresenta il percorso su filesystem _G del file _G da utilizzare per l'esportazione delle collezioni _G dal server _G
collections	vararg _G di String	Rappresenta una sequenza eventualmente vuota di nomi di collezioni _G da esportare su filesystem _G .

3.21.3.3 ActorbaseAdminServices (abstract)

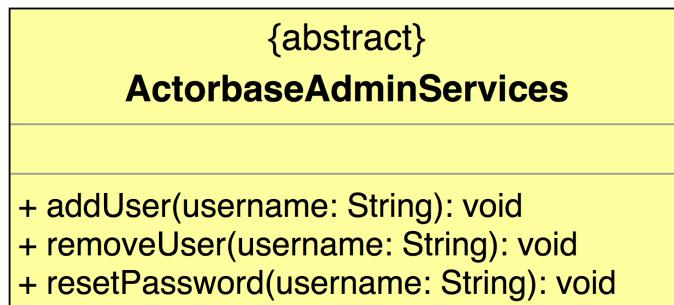


Figura 77: Driver: Classe ActorbaseAdminServices

3.21.3.3.1 Descrizione

Classe astratta contenente alcuni metodi di utilità esclusivamente amministrativa, utilizzata come dipendenza da iniettare alla classe [ActorbaseServices](#) qualora avvenisse un'autenticazione mediante la classe [ActorbaseDriver](#) utilizzando un profilo amministrativo.

3.21.3.3.2 Utilizzo

Aggiunge funzionalità amministrative all'oggetto [ActorbaseServices](#), ossia operazioni di gestione degli utenti oltre alle comuni operazioni di lettura e scrittura delle collezioni_G. Funge da iniezione di dipendenze mediante cake pattern_G.



3.21.3.3.3 Eredità

- `actorbase::driver::client::Connector`.

3.21.3.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
<code>addUser</code>	pubblico	<code>void</code>	Permette di aggiungere un nuovo utente al sistema
<code>removeUser</code>	pubblico	<code>void</code>	Permette la rimozione di un utente dal sistema
<code>resetPassword</code>	pubblico	<code>void</code>	Effettua il reset_G della password di un utente

3.21.3.3.5 Parametri

addUser

Nome	Tipo	Descrizione
<code>username</code>	<code>String</code>	Rappresenta lo username_G dell'utente da aggiungere al sistema

removeUser

Nome	Tipo	Descrizione
<code>username</code>	<code>String</code>	Rappresenta lo username_G dell'utente da rimuovere dal sistema

resetPassword

Nome	Tipo	Descrizione
<code>username</code>	<code>String</code>	Rappresenta lo username_G dell'utente a cui effettuare il reset_G della password



3.21.3.4 Connection

Connection	
- username: String	
- password: String	
- permissions: Boolean	
- address: String	
- port: Int	
+ username(): String	
+ password(): String	
+ permissions(): Boolean	
+ address(): String	
+ port(): Int	

Figura 78: Driver: Classe Connection

3.21.3.4.1 Descrizione

Classe interna all'oggetto `ActorbaseServices` contenente i parametri di connessione al server_G e le credenziali del profilo **Actorbase**.

3.21.3.4.2 Utilizzo

Viene utilizzata come oggetto contenente i parametri di connessione, da utilizzare nelle classi che necessitano di collegamenti al server_G.

3.21.3.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	pubblico	immutable	String	Rappresenta lo username _G dell'utente
password	pubblico	immutable	String	Rappresenta la password dell'utente
permissions	pubblico	immutable	Boolean	Indica il livello di privilegio del profilo Actorbase , true _G se amministrativo, altrimenti false _G
address	pubblico	immutable	String	Rappresenta l'indirizzo del server _G Actorbase
port	pubblico	immutable	Int	Rappresenta la porta esposta dal server _G



3.21.3.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo username_G
password	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo password
permissions	pubblico	Boolean	Metodo pubblico di accesso in sola lettura all'attributo permissions
address	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo address
port	pubblico	Int	Metodo pubblico di accesso in sola lettura all'attributo port

3.22 actorbase::driver::client

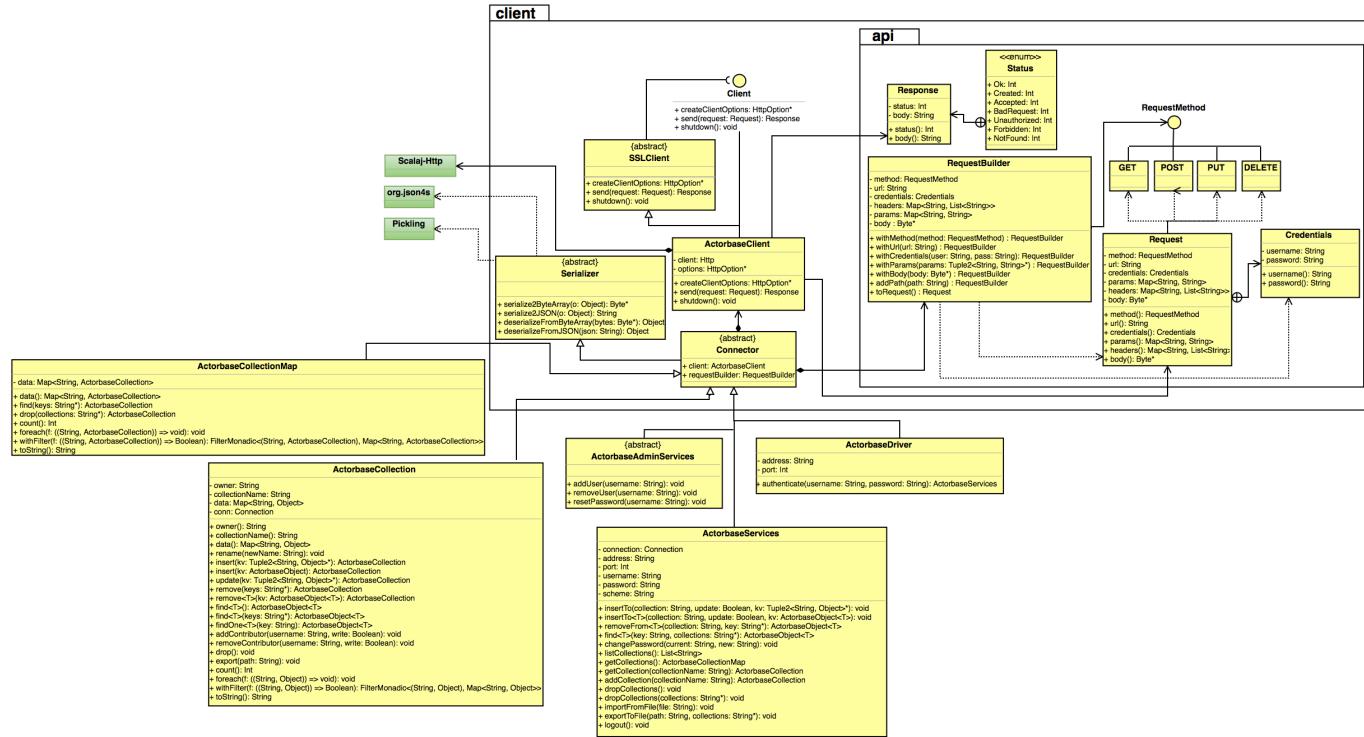


Figura 79: Driver: Client

3.22.1 Descrizione

Package_G che si occupa di comunicare sia con la componente `cliG` che con la componente `serverG` del sistema.



3.22.2 Interfacce

3.22.2.1 Client

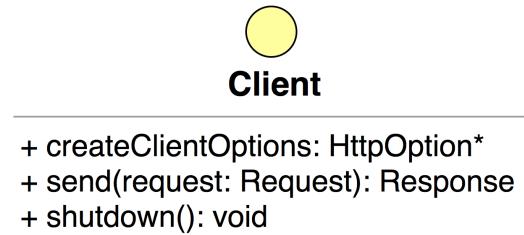


Figura 80: Driver: Client interface

3.22.2.1.1 Descrizione

Interfaccia che rappresenta il comportamento di un client attivo alla connessione con un server.

3.22.2.1.2 Utilizzo

Viene utilizzata per delineare il comportamento generale di un generico client, le classi che la realizzano hanno libertà di implementazione specifica dei metodi.

3.22.2.1.3 Ereditata da

- `actorbase::driver::client::SSLClient`

3.22.2.1.4 Realizzata da

- `actorbase::driver::client::ActorbaseClient`

3.22.2.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
<code>createClientOptions</code>	pubblico	array _G di <code>HttpOption</code>	Crea le opzioni di utilizzo del client _G <code>Http_G</code>
<code>send</code>	pubblico	<code>Response</code>	Invia la richiesta al server e riceve la risposta sincronamente
<code>shutdown</code>	pubblico	<code>void</code>	Chiude la connessione con il server



3.22.2.1.6 Parametri

send

Nome	Tipo	Descrizione
request	Request	Rappresenta una richiesta da inviare al server, comprensiva di payload _G dati

3.22.3 Classi

3.22.3.1 Serializer (abstract)

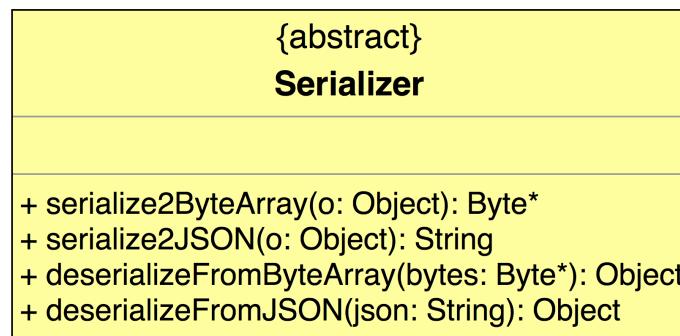


Figura 81: Driver: Classe serializer

3.22.3.1.1 Descrizione

Classe astratta atta a fornire metodi di serializzazione e deserializzazione nelle varianti JSON_G e array_G di byte_G.

3.22.3.1.2 Utilizzo

Viene utilizzata mediante estensione dalle classi che necessitano di operazioni di serializzazione e deserializzazione in formato JSON_G o array_G di byte o entrambi.

3.22.3.1.3 Ereditata da

- actorbase::driver::client::Connector.



3.22.3.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
serialize2ByteArray	pubblico	array _G di Byte _G	Serializza l'oggetto parametro in formato array _G di Byte _G
serialize2JSON	pubblico	String	Serializza l'oggetto parametro in formato JSON _G
deserializeFromByteArray	pubblico	Object _G	Deserializza l'array _G di Byte _G parametro riportandolo al formato originale
deserializeFromJSON	pubblico	Object _G	Deserializza la stringa JSON _G parametro riportandola al formato originale

3.22.3.1.5 Parametri

serialize2ByteArray

Nome	Tipo	Descrizione
o	Object _G	L'oggetto generico da serializzare in formato array _G di byte _G

serialize2JSON

Nome	Tipo	Descrizione
o	Object _G	L'oggetto generico da serializzare in formato JSON _G

deserializeFromByteArray

Nome	Tipo	Descrizione
bytes	array _G di Byte _G	Rappresentazione di un' oggetto serializzato in formato array _G di byte _G

deserializeFromJSON

Nome	Tipo	Descrizione
json	String	Rappresentazione di un' oggetto serializzato in formato JSON _G



3.22.3.2 SSLClient (abstract)

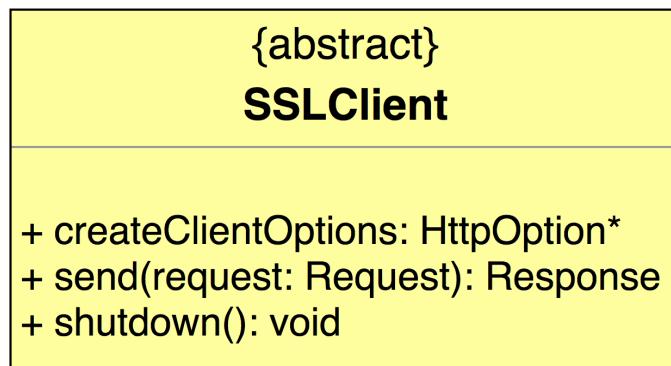


Figura 82: Driver: Classe SSLClient

3.22.3.2.1 Descrizione

Classe astratta che rappresenta il comportamento di un client attivo alla connessione protetta con un server.

3.22.3.2.2 Utilizzo

Viene utilizzata per delineare il comportamento generale di un generico client, funge da Decorator_G in quanto aggiunge funzionalità di protezione della connessione mediante protocolli TLS/SSL_G, le classi che la realizzano hanno libertà di implementazione specifica dei metodi.

3.22.3.2.3 Realizzata da

- actorbase::driver::client::ActorbaseClient

3.22.3.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
createClientOptions	pubblico	array _G di HttpOption	Crea le opzioni di utilizzo del client _G Http _G specificandone il protocollo di protezione
send	pubblico	Response	Invia la richiesta al server e riceve la risposta sincronamente
shutdown	pubblico	void	Chiude la connessione con il server

3.22.3.2.5 Parametri

send



Nome	Tipo	Descrizione
request	Request	Rappresenta una richiesta da inviare al server, comprensiva di payload _G dati

3.22.3.3 ActorbaseClient

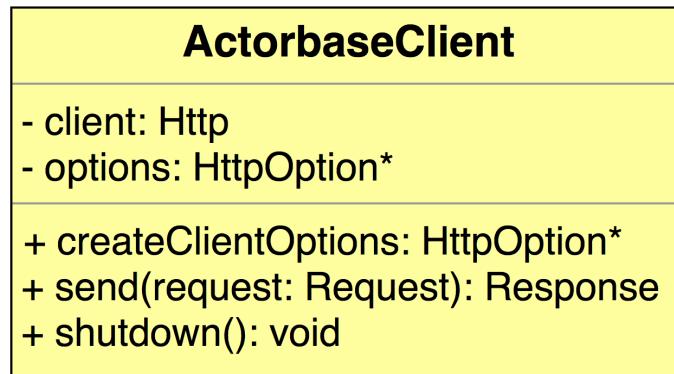


Figura 83: Driver: Classe ActorbaseClient

3.22.3.1 Descrizione

Classe che rappresenta il client utilizzato per la connessione con la componente server_G del sistema.

3.22.3.2 Utilizzo

Viene utilizzata dalla componente Connector per fornire la comunicazione con la componente server_G del sistema alle classi che lo necessitano.

3.22.3.3 Interfacce realizzate

- `actorbase::driver::client::Client`.

3.22.3.4 Classi ereditate

- `actorbase::driver::client::SSLClient`.



3.22.3.3.5 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
client	privato	immutabile	Http	Oggetto istanziato utilizzando la libreria esterna Scalaj _G dedicato alla comunicazione HTTP _G con il server
options	privato	immutabile	array di Htt-pOption	Array di opzioni da utilizzare con l'oggetto client, responsabile di settaggi quali livello di protezione della comunicazione e timeout _G richieste

3.22.3.3.6 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
createClientOptions	pubblico	array _G di Htt-pOption	Crea le opzioni di utilizzo del client _G Http _G
send	pubblico	Response	Invia la richiesta al server e riceve la risposta sincronamente
shutdown	pubblico	void	Chiude la connessione con il server

3.22.3.4 Connector (abstract)

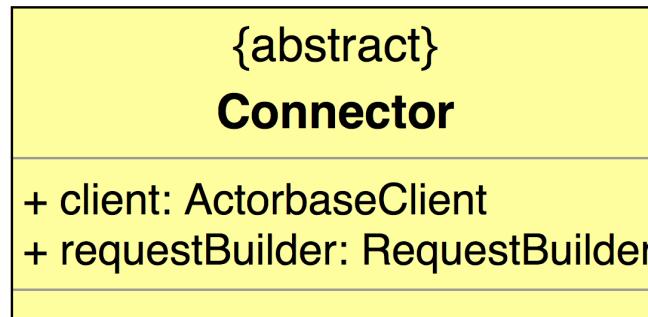


Figura 84: Driver: Classe astratta Connector

3.22.3.4.1 Descrizione

Classe astratta che si occupa di istanziare un oggetto di tipo **ActorbaseClient** e uno di tipo **RequestBuilder** in modo da fornire possibilità di comunicazione con il server alle classi che la estendono.

3.22.3.4.2 Utilizzo

Viene estesa dalle classi che necessitano di comunicare con il server.



3.22.3.4.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
client	pubblico	immutabile	ActorbaseClient	Ha lo scopo di fornire comunicazione con il server, mediante protocollo HTTP_G
requestBuilder	pubblico	immutabile	RequestBuilder	E' utilizzato per costruire le richieste REST_G da inviare al server mediante oggetto di tipo Request

3.22.3.4.4 Eredita

- `actorbase::driver::client::Serializer`.

3.22.3.4.5 Ereditata da

- `actorbase::driver::client::Connector`.



3.23 actorbase::driver::client::api

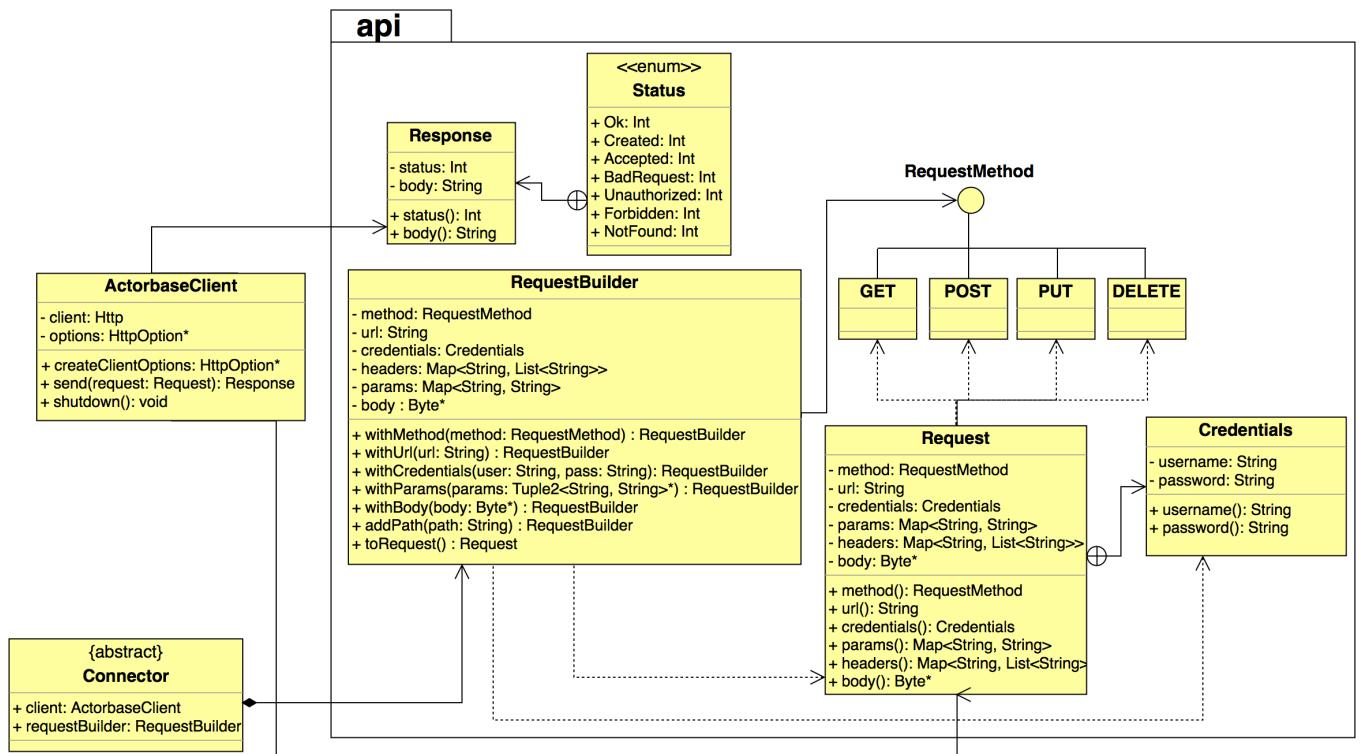


Figura 85: Driver: Package api

3.23.1 Descrizione

Package_G che offre le strutture dati necessarie per la comunicazione con il server_G, con la cli_G o eventuali programmi esterni. Si occupa inoltre della gestione delle richieste REST_G da inviare utilizzando il protocollo HTTP_G.



3.23.2 Classi

3.23.2.1 RequestMethod (abstract)

RequestMethod



Figura 86: Driver: RequestMethod interface

3.23.2.1.1 Descrizione

Classe astratta che rappresenta una generica richiesta HTTP_G , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.23.2.1.2 Utilizzo

Viene utilizzata come classe padre per l'implementazione delle richieste HTTP_G da utilizzare nella comunicazione con il server.

3.23.2.1.3 Ereditata da

- actorbase::driver::client::api::GET
- actorbase::driver::client::api::POST
- actorbase::driver::client::api::PUT
- actorbase::driver::client::api::DELETE

3.23.2.2 GET



Figura 87: Driver: Oggetto rappresentante richiesta get

3.23.2.2.1 Descrizione

Classe che rappresenta una generica richiesta HTTP_G di tipo GET_G, implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.



3.23.2.2.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste HTTP_G di tipo GET_G da utilizzare nella comunicazione con il server.

3.23.2.2.3 Eredità

- `actorbase::driver::client::api::RequestMethod`

3.23.2.3 POST



Figura 88: Driver: Oggetto rappresentante richiesta post

3.23.2.3.1 Descrizione

Classe che rappresenta una generica richiesta HTTP_G di tipo POST_G , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.23.2.3.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste HTTP_G di tipo POST_G da utilizzare nella comunicazione con il server.

3.23.2.3.3 Eredità

- `actorbase::driver::client::api::RequestMethod`

3.23.2.4 PUT



Figura 89: Driver: Oggetto rappresentante richiesta Put



3.23.2.4.1 Descrizione

Classe che rappresenta una generica richiesta HTTP_G di tipo PUT_G , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.23.2.4.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste HTTP_G di tipo PUT_G da utilizzare nella comunicazione con il server.

3.23.2.4.3 Eredità

- `actorbase::driver::client::api::RequestMethod`

3.23.2.5 DELETE



Figura 90: Driver: Oggetto rappresentante richiesta Delete

3.23.2.5.1 Descrizione

Classe che rappresenta una generica richiesta HTTP_G di tipo DELETE_G , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G.

3.23.2.5.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste HTTP_G di tipo DELETE_G da utilizzare nella comunicazione con il server.

3.23.2.5.3 Eredità

- `actorbase::driver::client::api::RequestMethod`



3.23.2.6 Request

Request	
- method:	RequestMethod
- url:	String
- credentials:	Credentials
- params:	Map<String, String>
- headers:	Map<String, List<String>>
- body:	Byte*
+ method():	RequestMethod
+ url():	String
+ credentials():	Credentials
+ params():	Map<String, String>
+ headers():	Map<String, List<String>>
+ body():	Byte*

Figura 91: Driver: Classe request

3.23.2.6.1 Descrizione

Classe che rappresenta una generica richiesta HTTP_G , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching $_G$, racchiude al suo interno i parametri di richiesta, quali il metodo, di tipo `RequestMethod`, l'URL $_G$ a cui inviare la richiesta, le credenziali di accesso al sistema, gli header $_G$ HTTP $_G$ associati, i parametri e il payload $_G$.

3.23.2.6.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste HTTP $_G$ di tipo DELETE $_G$ da utilizzare nella comunicazione con il server. Viene creata mediante il design pattern Builder $_G$ nella classe RequestBuilder.

3.23.2.6.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
method	privato	immutabile	RequestMethod	Indica il tipo di richiesta HTTP $_G$
url	privato	immutabile	String	Rappresenta l'indirizzo dell'API $_G$ esposta sul server a cui inviare la richiesta
credentials	privato	immutabile	Credentials	Oggetto rappresentante le credenziali di accesso al sistema Actorbase
headers	privato	immutabile	Map<String, List<String> >	Rappresenta gli header $_G$ HTTP $_G$ da associare alla richiesta
params	privato	immutabile	Map<String, String>	Rappresenta i parametri da allegare alla richiesta HTTP $_G$
body	privato	immutabile	Array di Byte	Rappresenta i dati di payload $_G$ da inviare sul server



3.23.2.6.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
method	pubblico	RequestMethod	Metodo pubblico di accesso in sola lettura all'attributo method
url	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo url
credentials	pubblico	Credentials	Metodo pubblico di accesso in sola lettura all'attributo credentials
params	pubblico	Map<String, String>	Metodo pubblico di accesso in sola lettura all'attributo params
headers	pubblico	Map<String, List<String>>	Metodo pubblico di accesso in sola lettura all'attributo headers
body	pubblico	Array di byte	Metodo pubblico di accesso in sola lettura all'attributo body

3.23.2.7 Credentials

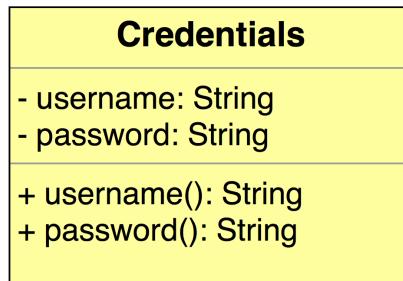


Figura 92: Driver: Classe request

3.23.2.7.1 Descrizione

Rappresenta delle credenziali di accesso al sistema **Actorbase**, composte da username_G e password.

3.23.2.7.2 Utilizzo

Classe interna alla classe **Request**, viene utilizzata per incapsulare le credenziali di accesso al sistema **Actorbase**, contiene due campi dati stringa, username_G e password.

3.23.2.7.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
username	privato	immutabile	String	Rappresenta lo username_G associato all'utente utilizzatore del sistema
password	privato	immutabile	String	Rappresenta la password associata allo username_G dell'utente utilizzatore del sistema



3.23.2.7.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
username	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo username_G
password	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo password

3.23.2.8 Response

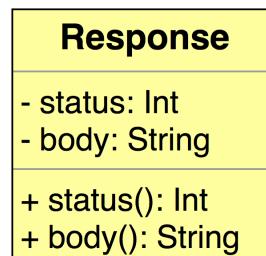


Figura 93: Driver: Classe response

3.23.2.8.1 Descrizione

Classe che rappresenta una generica risposta HTTP_G, implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching_G, racchiude al suo interno i parametri di risposta, quali il codice di stato, rappresentato da una enum interna **Status** e il payload_G di risposta in formato stringa.

3.23.2.8.2 Utilizzo

Viene utilizzata come classe per l'incapsulamento delle risposte HTTP_G da utilizzare nella comunicazione con il server.

3.23.2.8.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
status	privato	immutabile	Int	Indica il codice di risposta HTTP _G
body	privato	immutabile	String	Rappresenta i dati di payload _G ricevuti dal server



3.23.2.8.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
status	pubblico	Int	Metodo pubblico di accesso in sola lettura all'attributo status
body	pubblico	String	Metodo pubblico di accesso in sola lettura all'attributo body

3.23.2.9 Status

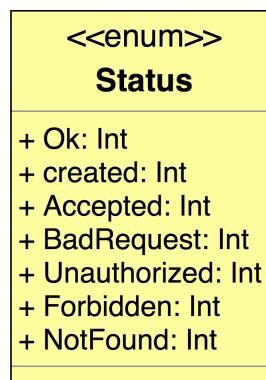


Figura 94: Driver: Classe response

3.23.2.9.1 Descrizione

Enum interna a `hyperref[sec:actorbase::driver::client::api::Response]Response`, rappresenta il codice di ritorno della richiesta $HTTP_G$ inviata al server.

3.23.2.9.2 Utilizzo

Viene utilizzata come classe per l'incapsulamento dei codici di ritorno $HTTP_G$ da utilizzare nella comunicazione con il server.

3.23.2.9.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
OK	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 200
Created	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 201
Accepted	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 202
BadRequest	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 400
Unauthorized	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 401
Forbidden	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 403
NotFound	privato	immutable	Int	Indica il codice di risposta $HTTP_G$ 404



3.23.2.10 RequestBuilder

RequestBuilder	
- method: RequestMethod	
- url: String	
- credentials: Credentials	
- headers: Map<String, List<String>>	
- params: Map<String, String>	
- body : Byte*	
+ withMethod(method: RequestMethod) : RequestBuilder	
+ withUrl(url: String) : RequestBuilder	
+ withCredentials(user: String, pass: String): RequestBuilder	
+ withParams(params: Tuple2<String, String>*) : RequestBuilder	
+ withBody(body: Byte*) : RequestBuilder	
+ addPath(path: String) : RequestBuilder	
+ toRequest() : Request	

Figura 95: Driver: Classe RequestBuilder

3.23.2.10.1 Descrizione

Rappresenta una classe builder per la creazione di richieste $HTTP_G$, racchiude al suo interno i parametri di richiesta, quali il metodo, di tipo [RequestMethod](#), l'URL $_G$ a cui inviare la richiesta, gli header $_G$ $HTTP_G$ associati, i parametri e il payload $_G$ e i metodi with per aggiungere attributi alla richiesta di tipo [Request](#) da generare.

3.23.2.10.2 Utilizzo

Viene utilizzata come classe buider per la costruzione delle richieste $HTTP_G$ di tipo [Request](#) da utilizzare nella comunicazione con il server.



3.23.2.10.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
method	privato	immutabile	RequestMethod	Indica il tipo di richiesta HTTP_G
url	privato	immutabile	String	Rappresenta l'indirizzo dell' API_G esposta sul server a cui inviare la richiesta
credentials	privato	immutabile	Credentials	Rappresenta le credenziali di accesso al sistema Actorbase
headers	privato	immutabile	Map<String, List<String> >	Rappresenta gli header $_G$ HTTP_G da associare alla richiesta
params	privato	immutabile	Map<String, String>	Rappresenta i parametri da allegare alla richiesta HTTP_G
body	privato	immutabile	Array di Byte	Rappresenta i dati di payload $_G$ da inviare sul server

3.23.2.10.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
withMethod	pubblico	RequestMethod	Imposta il metodo di richiesta HTTP_G
withUrl	pubblico	RequestMethod	Imposta l'indirizzo dell' API_G esposta sul server
withCredentials	pubblico	RequestMethod	Imposta le credenziali di accesso al sistema Actorbase da utilizzare nelle richieste HTTP_G
withParams	pubblico	RequestMethod	Imposta i parametri da inviare allegati alla richiesta HTTP_G
withBody	pubblico	RequestMethod	Imposta il payload $_G$ da inviare allegato alla richiesta HTTP_G
addPath	pubblico	RequestMethod	Aggiunge eventuali indirizzi aggiuntivi all'url $_G$ della richiesta HTTP_G
toRequest	pubblico	Request	Trasforma l'oggetto riferito da this $_G$, in un oggetto di tipo Request

3.23.2.10.5 Parametri

withMethod

Nome	Tipo	Descrizione
method	RequestMethod	Rappresenta il tipo di richiesta HTTP_G da inviare al server

withUrl

Nome	Tipo	Descrizione
url	String	Rappresenta il la route $_G$ dell' API_G esposta sul server a cui inviare le richieste

withParams



Nome	Tipo	Descrizione
params	array di Tuple2<String, String>	Rappresenta i parametri da allegare alle richieste HTTP _G

withBody

Nome	Tipo	Descrizione
body	array di Byte	Rappresenta il payload _G da allegare alle richieste HTTP _G

addPath

Nome	Tipo	Descrizione
path	String	Rappresenta l'eventuale indirizzo parziale da aggiungere all'url _G da allegare alle richieste HTTP _G

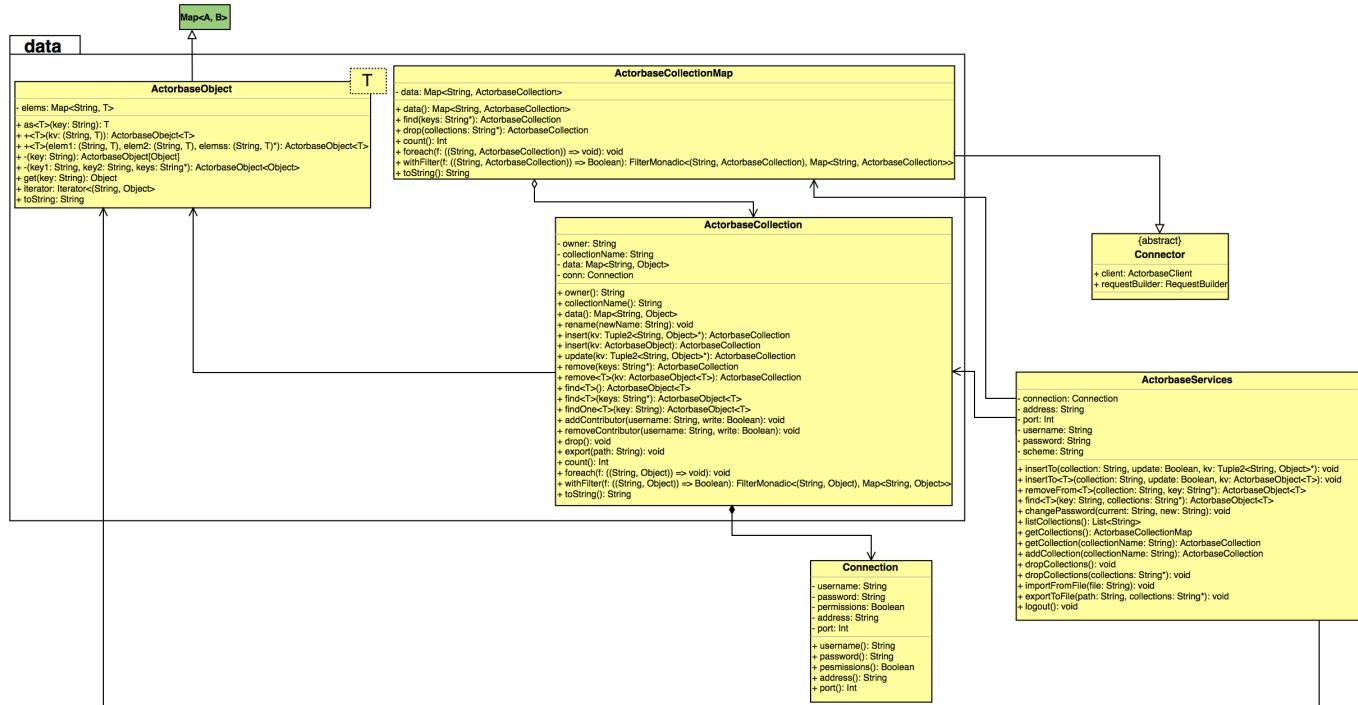
3.24 actorbase::driver::data

Figura 96: Driver: Package data



3.24.1 Descrizione

Package_G che offre le strutture dati utili alla gestione dei contenuti della base di dati in locale, permettendo di eseguire le comuni operazioni di CRUD_G e ricerche più accurate in maniera semplificata, riflettendo le eventuali modifiche locali sul server remoto che contiene i dati originali.

3.24.2 Classi

3.24.2.1 ActorbaseObject<T>

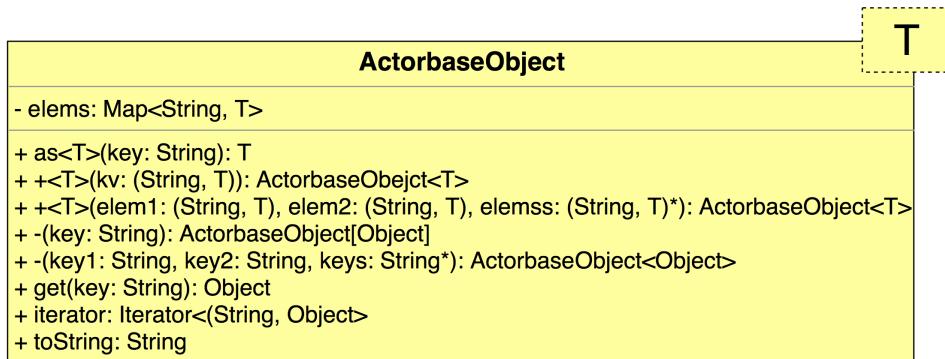


Figura 97: Driver: Classe ActorbaseObject

3.24.2.1.1 Descrizione

Classe istanziata con tipo generico, rappresenta un item_G contenente una o più coppie chiave-valore, estende l'interfaccia Map di Scala_G e ne eredita quindi tutti i metodi.

3.24.2.1.2 Utilizzo

Classe utilizzata per agevolare gli inserimenti e le ricerche specifiche all'interno delle collezioni_G del database_G, è possibile utilizzarla per effettuare inserimenti multipli in un'unica chiamata del metodo di inserimento delle classi **ActorbaseServices** o . Rende inoltre più agevole la navigazione e modifica di item_G ricevuti in seguito ad una ricerca, sia generale, sia interna a specifiche collezioni_G.

3.24.2.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
elems	privato	immutabile	Map<String, Object> _G	Mappa contenente una o più coppie chiave-valore, utile a rappresentare una frazione di una collezione _G



3.24.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
as<T>	pubblico	T	Metodo con tipo generico, permette di ottenere il valore associato alla chiave parametro del tipo specificato come tipo generico T
+<T>	pubblico	ActorbaseObject<T>	Metodo ereditato dall'interfaccia Map di Scala _G , permette di aggiungere una coppia chiave-valore all'oggetto ActorbaseObject<T> di tipo String-T generico
+<T> (overload)	pubblico	ActorbaseObject<T>	Metodo ereditato dall'interfaccia Map di Scala _G , permette di aggiungere più coppie chiave-valore all'oggetto ActorbaseObject<T> di tipo String-T generico
-	pubblico	ActorbaseObject<Object>	Metodo ereditato dall'interfaccia Map di Scala _G , permette di rimuovere una coppia chiave-valore dall'oggetto ActorbaseObject<T> di tipo String-T generico
- (overload)	pubblico	ActorbaseObject<Object>	Metodo ereditato dall'interfaccia Map di Scala _G , permette di rimuovere più coppie chiave-valore dall'oggetto ActorbaseObject<T> di tipo String-T generico
get	pubblico	Object	Metodo ereditato dall'interfaccia Map di Scala _G , permette di ottenere il valore associato alla chiave specificata nell'oggetto ActorbaseObject<T> di tipo String-T generico
iterator	pubblico	Iterator<(String, Object)>	Metodo ereditato dall'interfaccia Map di Scala _G , permette di ottenere un iteratore all'oggetto ActorbaseObject<T> di tipo String-Object
toString	pubblico	String	Ritorna l'oggetto di tipo ActorbaseObject in formato String

3.24.2.1.5 Parametri

as<T>

Nome	Tipo	Descrizione
key	String	Rappresenta una chiave associata ad un valore, insieme formano un item _G Actorbase

+<T>

Nome	Tipo	Descrizione
kv	Tuple2<String, T> _G	Coppia chiave-valore, rappresenta un generico oggetto del database _G

+<T> (overload)



Nome	Tipo	Descrizione
elem1	$\text{Tuple2} < \text{String}, T \rangle_G$	Coppia chiave-valore, rappresenta un generico oggetto del database _G
elem2	$\text{Tuple2} < \text{String}, T \rangle_G$	Coppia chiave-valore, rappresenta un generico oggetto del database _G
elemss	vararg _G di $\text{Tuple2} < \text{String}, T \rangle_G$	Rappresenta una o più coppie chiave-valore, formano item _G del database _G

-<T>

Nome	Tipo	Descrizione
key	String	Oggetto di tipo stringa, rappresenta una chiave associata ad un valore, insieme formano un item _G del database _G

-<T> (overload)

Nome	Tipo	Descrizione
key1	String	Oggetto di tipo stringa, rappresenta una chiave associata ad un valore, insieme formano un item _G del database _G
key2	String	Oggetto di tipo stringa, rappresenta una chiave associata ad un valore, insieme formano un item _G del database _G
keys	vararg _G di String	Sequenza di stringhe, rappresenta una sequenza di chiavi associate a un valori, insieme formano degli item _G del database _G

get

Nome	Tipo	Descrizione
key	String	Oggetto di tipo stringa, rappresenta una chiave associata ad un valore, insieme formano un item _G del database _G



3.24.2.2 actorbase::driver::data::ActorbaseCollection

ActorbaseCollection	
- owner: String	
- collectionName: String	
- data: Map<String, Object>	
- conn: Connection	
+ owner(): String	
+ collectionName(): String	
+ data(): Map<String, Object>	
+ rename(newName: String): void	
+ insert(kv: Tuple2<String, Object>*): ActorbaseCollection	
+ insert(kv: ActorbaseObject): ActorbaseCollection	
+ update(kv: Tuple2<String, Object>*): ActorbaseCollection	
+ remove(keys: String*): ActorbaseCollection	
+ remove<T>(kv: ActorbaseObject<T>): ActorbaseCollection	
+ find<T>(): ActorbaseObject<T>	
+ find<T>(keys: String*): ActorbaseObject<T>	
+ findOne<T>(key: String): ActorbaseObject<T>	
+ addContributor(username: String, write: Boolean): void	
+ removeContributor(username: String, write: Boolean): void	
+ drop(): void	
+ export(path: String): void	
+ count(): Int	
+ foreach(f: ((String, Object)) => void): void	
+ withFilter(f: ((String, Object)) => Boolean): FilterMonadic<(String, Object), Map<String, Object>>	
+ toString(): String	

Figura 98: Driver: Classe ActorbaseCollection

3.24.2.2.1 Descrizione

Classe che rappresenta una singola collezione_G del database_G allo stato corrente, al momento cioè dello scaricamento in locale della stessa. Contiene le coppie chiave-valore salvate e permette di eseguire le operazioni più comuni quali inserimenti, rimozioni e ricerche, riflettendo eventuali modifiche direttamente nella componente server_G del sistema.

3.24.2.2.2 Utilizzo

Questa classe viene utilizzata per la rappresentazione di una singola collezione_G vista come insieme di oggetti di tipo coppia chiave-valore. Il valore può essere qualsiasi tipo, inclusi oggetti arbitrari creati all'interno di un programma Scala_G. Offre metodi di utilità e navigabilità dei contenuti.



3.24.2.2.3 Eredita

- `actorbase::driver::client::Connector`.

3.24.2.2.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
owner	privato	immutabile	String	Creatore della collezione _G
collectionName	privato	immutabile	String	Nome della collezione _G
data	privato	immutabile	Map<String, Object>	Mappa contenente le coppie chiave-valore della collezione _G
conn	privato	immutabile	Connection	Oggetto che rappresenta i parametri di connessione al server _G



3.24.2.2.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
owner	pubblico	String	Metodo pubblico di accesso in sola lettura dell'attributo owner
collectionName	pubblico	String	Metodo pubblico di accesso in sola lettura dell'attributo collectionName
data	pubblico	String	Metodo pubblico di accesso in sola lettura dell'attributo data
conn	pubblico	Connection	Metodo pubblico di accesso in sola lettura dell'attributo conn
rename	pubblico	void	Rinomina la collezione _G
insert	pubblico	ActorbaseCollection	Inserisce una o più coppie chiave-valore all'interno di Actorbase
insert (overload)	pubblico	ActorbaseCollection	Inserisce un oggetto di tipo ActorbaseObject<T> all'interno della collezione _G
update	pubblico	ActorbaseCollection	Aggiorna la coppia chiave-valore all'interno della collezione _G
remove	pubblico	ActorbaseCollection	Rimuove una o più coppie chiave-valore dalla collezione _G
remove<T> (overload)	pubblico	ActorbaseObject	Rimuove una coppia chiave-valore dalla collezione _G
find<T>	pubblico	ActorbaseObject<T>	Cerca tutte le coppie-chiave valore nella collezione _G restituendo un'istanza ActorbaseObject<T>
find<T> (overload)	pubblico	ActorbaseObject<T>	Cerca uno o più coppie-chiave valore nella collezione _G
findOne<T>	pubblico	ActorbaseObject<T>	Cerca una coppia chiave-valore nella collezione _G
addContributor	pubblico	void	Aggiunge un collaboratore _G alla collezione _G
removeContributor	pubblico	void	Rimuove un collaboratore _G dalla collezione _G
drop	pubblico	void	Cancella la collezione _G
export	pubblico	void	Esporta la collezione _G su filesystem _G al percorso specificato
count	pubblico	Int	Conta il numero di coppie chiave-valore che formano la collezione _G
foreach	pubblico	void	Permette di visitare l'intera collezione _G applicando una funzione ad ogni item _G
withFilter	pubblico	FilterMonadic<(String, Object), Map<String, Object> >	Permette di visitare l'intera collezione _G utilizzando il costrutto for Scala _G
toString	pubblico	String	Ritorna la collezione _G in formato String JSON _G

3.24.2.2.6 Parametri

rename



Nome	Tipo	Descrizione
newName	String	Rappresenta il nuovo nome della collezione _G da rinominare

insert

Nome	Tipo	Descrizione
kv	vararg _G di Tuple2<String, Object>	Rappresenta una sequenza di Tuple2<String, Object> da inserire nella collezione _G

insert (overload)

Nome	Tipo	Descrizione
kv	ActorbaseObject<T>	Rappresenta un oggetto del sistema Actorbase da inserire nel sistema

update

Nome	Tipo	Descrizione
kv	vararg _G di Tuple2<String, Object>	Rappresenta una sequenza di Tuple2<String, Object> da aggiornare nella collezione _G

remove

Nome	Tipo	Descrizione
keys	vararg _G di String	Rappresenta una sequenza di String, rappresentano le chiavi da rimuovere dal sistema

remove (overload)

Nome	Tipo	Descrizione
kv	ActorbaseObject<T>	Rappresenta un oggetto di tipo chiave-valore del sistema, rappresentano le chiavi da rimuovere dal sistema

find

Nome	Tipo	Descrizione
keys	vararg _G di String	Rappresenta una sequenza di String, sono chiavi da ricercare nel sistema

findOne



Nome	Tipo	Descrizione
key	String	Rappresenta una String da ricercare nel sistema

addContributor

Nome	Tipo	Descrizione
username	String	Rappresenta lo username _G dell'utente da aggiungere ai collaboratori della collezione _G
write	Boolean	Rappresenta il flag _G dei permessi in scrittura

removeContributor

Nome	Tipo	Descrizione
username	String	Rappresenta lo username _G dell'utente da rimuovere ai collaboratori _G della collezione _G
write	Boolean	Rappresenta il flag _G dei permessi in scrittura

export

Nome	Tipo	Descrizione
path	String	Rappresenta il percorso su filesystem _G su cui esportare la collezione _G in formato JSON

foreach

Nome	Tipo	Descrizione
f: ((String, Any)) => Unit	Funzione anonima	Rappresenta la funzione da applicare a tutti gli item _G della collezione _G

withFilter

Nome	Tipo	Descrizione
f: ((String, Any)) => Boolean	Funzione anonima	Rappresenta la funzione da applicare a tutti gli item _G della collezione _G



3.24.2.3 actorbase::driver::data::ActorbaseCollectionMap

ActorbaseCollectionMap
- data: Map<String, ActorbaseCollection>
+ data(): Map<String, ActorbaseCollection>
+ find(keys: String*): ActorbaseCollection
+ drop(collections: String*): ActorbaseCollection
+ count(): Int
+ foreach(f: ((String, ActorbaseCollection)) => void): void
+ withFilter(f: ((String, ActorbaseCollection)) => Boolean): FilterMonadic<(String, ActorbaseCollection), Map<String, ActorbaseCollection>>
+ toString(): String

Figura 99: Driver: Classe ActorbaseCollectionMap

3.24.2.3.1 Descrizione

Classe che rappresenta una collezione_G di collezione_G del sistema.

3.24.2.3.2 Utilizzo

Questa classe viene utilizzata per la rappresentazione dell'aggregazione di collezioni_G, è formata da una mappa di coppie nome collezione_G - ActorbaseCollection.

3.24.2.3.3 Eredità

- actorbase::driver::client::Connector.

3.24.2.3.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
data	privato	mutabile	Map<String, ActorbaseCollection>	Mappa che rappresenta un insieme di ActorbaseCollection



3.24.2.3.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
find	pubblico	ActorbaseCollection	Cerca uno o più coppie-chiave valore nelle collezioni _G
drop	pubblico	ActorbaseCollection	Cancella una o più collezioni _G
count	pubblico	Int	Ritorna il numero di collezioni _G contenute nell'oggetto ActorbaseCollectionMap
foreach	pubblico	void	Permette di visitare l'intera collezione _G di collezioni _G applicando una funzione ad ogni item _G
withFilter	pubblico	FilterMonadic<(String, ActorbaseCollection), Map<String, ActorbaseCollection>	Permette di visitare l'intero oggetto e applicare funzioni alle coppie chiave-valore
toString	pubblico	String	Ritorna la collezione _G in formato Stringa JSON _G

3.24.2.3.6 Parametri

find

Nome	Tipo	Descrizione
keys	vararg _G di String	Rappresenta una sequenza di String, sono chiavi da ricercare nella collezione _G

drop

Nome	Tipo	Descrizione
collections	vararg _G di String	Rappresenta una sequenza di String, sono nomi di collezioni _G da rimuovere

foreach

Nome	Tipo	Descrizione
f: ((A, B)) => Unit	Funzione anonima	Rappresenta la funzione da applicare a tutti gli item _G della collezione _G



3.25 actorbase::driver::exceptions

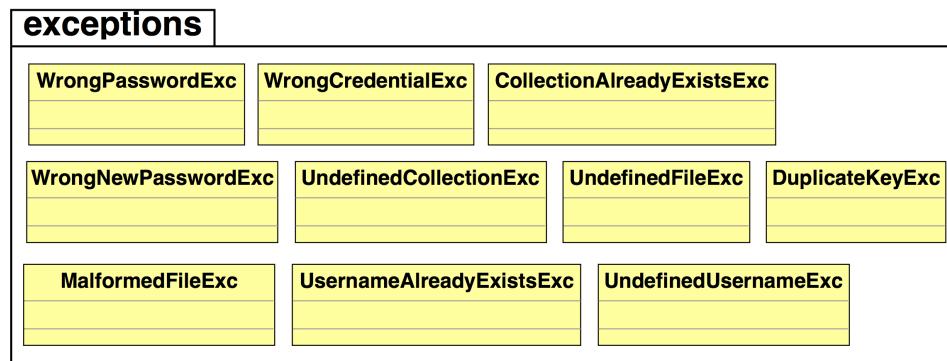


Figura 100: Driver:Package exceptions

3.25.1 Descrizione

Package_G che contiene le possibili eccezioni che si possono verificare nel driver_G

3.25.2 Interazione con altre componenti

- actorbase::driver::client;

3.25.3 Classi

3.25.3.1 actorbase::driver::exceptions::WrongCredentialExc



Figura 101: Driver: class WrongCredentialExc

3.25.3.1.1 Descrizione

Classe che rappresenta l'eccezione del tentativo di accesso con username o password sbagliati.



3.25.3.1.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire errori nelle credenziali di accesso.

3.25.3.2 actorbase::driver::exceptions::WrongPasswordExc

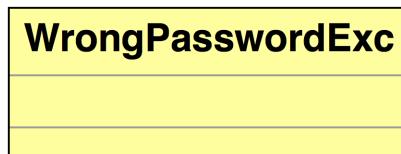


Figura 102: Driver: class WrongPasswordExc

3.25.3.2.1 Descrizione

Classe che rappresenta l'eccezione dell'inserimento sbagliato della vecchia password.

3.25.3.2.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla modifica della propria password, quando dovrei inserire la vecchia password questa non corrisponda.

3.25.3.3 actorbase::driver::exceptions::WrongNewPasswordExc

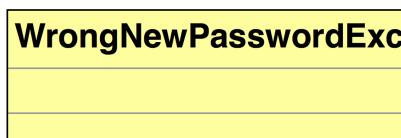


Figura 103: Driver: class WrongNewPasswordExc

3.25.3.3.1 Descrizione

Classe che rappresenta l'eccezione dell'inserimento sbagliato della nuova password.

3.25.3.3.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla modifica della propria password, quando dovrei inserire la nuova password questa non soddisfi i requisiti richiesti.



3.25.3.4 **actorbase::driver::exceptions::CollectionAlreadyExistsExc**

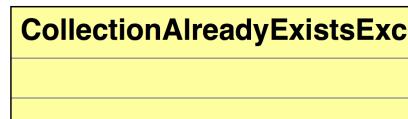


Figura 104: Driver: class CollectionAlreadyExistsExc

3.25.3.4.1 Descrizione

Classe che rappresenta l'eccezione del conflitto di nomi tra una collezione_G esistente e una nuova.

3.25.3.4.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla creazione di una nuova collezione_G, il nome della nuova collezione_G corrisponda al nome di una collezione_G già esistente.

3.25.3.5 **actorbase::driver::exceptions::UndefinedCollectionExc**



Figura 105: Driver: class UndefinedCollectionExc

3.25.3.5.1 Descrizione

Classe che rappresenta l'eccezione di mancato inserimento del nome di una nuova collezione_G.

3.25.3.5.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla creazione di una nuova collezione_G, venga omesso il nome di quest'ultima.



3.25.3.6 `actorbase::driver::exceptions::UndefinedUsernameExc`



Figura 106: Driver: class UndefinedUsernameExc

3.25.3.6.1 Descrizione

Classe che rappresenta l'eccezione di mancato inserimento dell'username utente.

3.25.3.6.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla registrazione di un nuovo utente, venga omesso l'username.

3.25.3.7 `actorbase::driver::exceptions::UsernameAlreadyExistsExc`

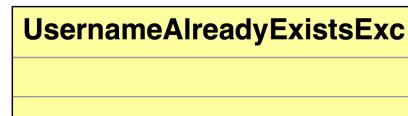


Figura 107: Driver: class UsernameAlreadyExistsExc

3.25.3.7.1 Descrizione

Classe che rappresenta l'eccezione del conflitto di username tra un utente già esistente e uno nuovo.

3.25.3.7.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui alla registrazione di un nuovo utente, l'username di quest'ultimo corrisponda all'username di un utente già esistente.



3.25.3.8 `actorbase::driver::exceptions::DuplicateKeyExc`



Figura 108: Driver: class DuplicateKeyExc

3.25.3.8.1 Descrizione

Classe che rappresenta l'eccezione del conflitto tra una chiave già esistente e una nuova chiave.

3.25.3.8.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui all'inserimento di una nuova chiave, questa corrisponda ad una chiave già esistente nella stessa collezione_G.

3.25.3.9 `actorbase::driver::exceptions::UndefinedFileExc`



Figura 109: Driver: class UndefinedFileExc

3.25.3.9.1 Descrizione

Classe che rappresenta l'eccezione di inserimento di un path sbagliato di un file o del mancato inserimento del path.

3.25.3.9.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui all'inserimento di un nuovo item o di una serie di item da file, si sbagli il path del file o si ometta di inserirlo.



3.25.3.10 `actorbase::driver::exceptions::MalformedFileExc`

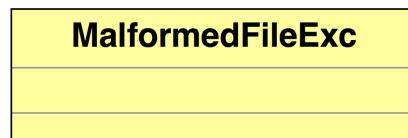


Figura 110: Driver: class MalformedFileExc

3.25.3.10.1 Descrizione

Classe che rappresenta l'eccezione di inserimento di un path di un file non conforme.

3.25.3.10.2 Utilizzo

Questa classe viene utilizzata dal driver_G per gestire il caso in cui all'inserimento di un nuovo item o di una serie di item da file, si inserisca il path di un file in un formato sbagliato o il cui testo non rispetta le regole di formattazione.



3.26 actorbase::cli

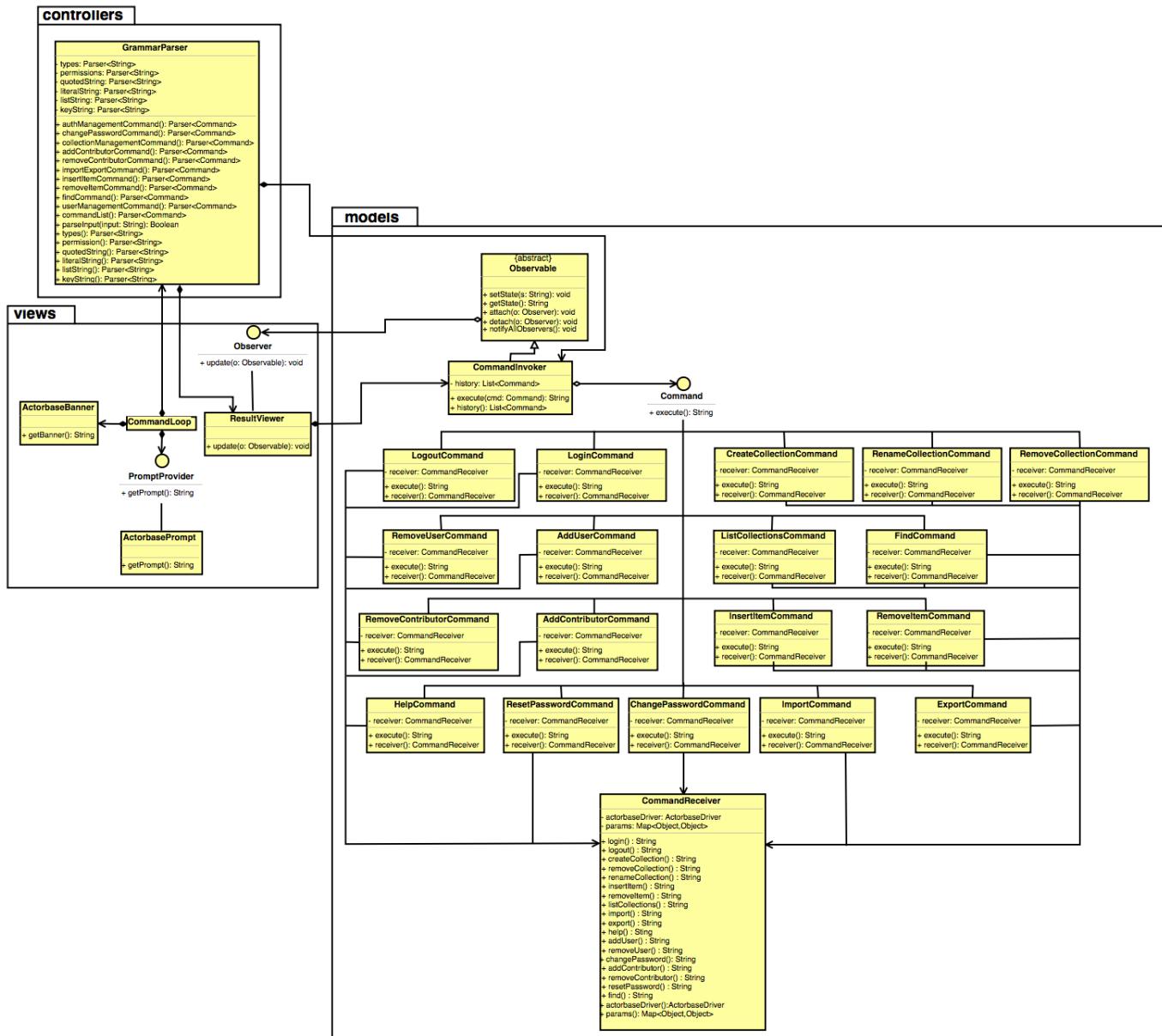


Figura 111: CLI: architettura MVC variante push model



3.26.1 Descrizione

Package_G per la parte di client_G rappresentata dalla cli_G. Questa componente viene rappresentata usando il design pattern_G MVC_G in versione push model_G.

3.26.2 Interazioni con altre componenti

- actorbase::driver.

3.26.3 Package contenuti

- actorbase::cli::views;
- actorbase::cli::controllers;
- actorbase::cli::models.



ScalateKids

3 COMPONENTI



3.27 actorbase::cli::views

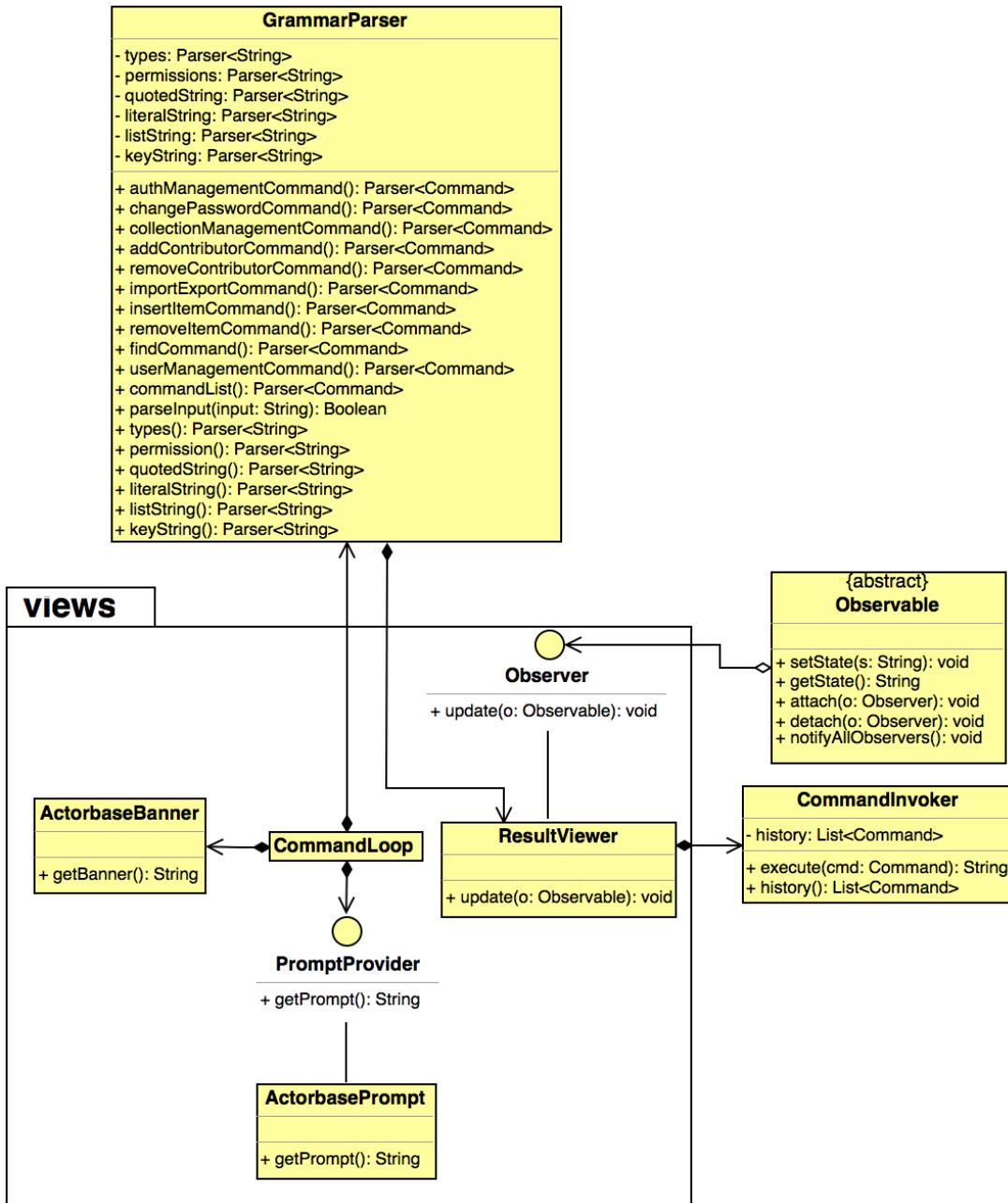


Figura 112: CLI: views package, interazioni con controllers e models



3.27.1 Descrizione

Package_G per la parte view_G della cli_G. Questa componente gestisce l'output ricevuto dal driver e offre l'interfaccia tramite cui l'utente può inserire i comandi.

3.27.2 Interfacce

3.27.2.1 actorbase::cli::views::PromptProvider

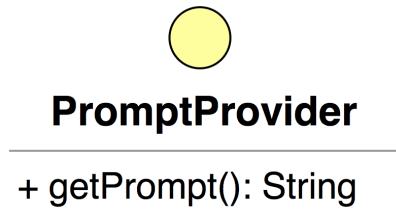


Figura 113: CLI: interfaccia PromptProvider

3.27.2.1.1 Descrizione

Interfaccia per la creazione del prompt_G dei comandi.

3.27.2.1.2 Utilizzo

Offre un'interfaccia per la creazione di un prompt_G generico.

3.27.2.1.3 Realizzata da

- actorbase::cli::views::ActorbasePrompt

3.27.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
getPrompt	pubblico	String	Ritorna la stringa che rappresenta il prompt _G .



3.27.2.2 actorbase::cli::views::Observer

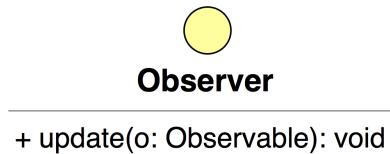


Figura 114: CLI: interfaccia Observer

3.27.2.2.1 Descrizione

Interfaccia per l'implementazione del design pattern_G Observer_G che permette di aggiornare la view_G ad ogni risposta della componente driver_G.

3.27.2.2.2 Utilizzo

Offre un'interfaccia per permettere di aggiornare l'output della componente view_G ad ogni risposta ricevuta dal driver_G.

3.27.2.2.3 Realizzata da

- actorbase::cli::views::ResultViewer

3.27.2.2.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
update	pubblico	void	Aggiorna lo stato dell'oggetto di tipo Observer

3.27.2.2.5 Parametri

update

Nome	Tipo	Descrizione
o	Observable	Rappresenta l'oggetto osservato



3.27.3 Classi

3.27.3.1 actorbase::cli::views::CommandLoop

CommandLoop

Figura 115: CLI: classe CommandLoop

3.27.3.1.1 Descrizione

Questa classe verrà utilizzata per leggere i comandi inseriti dall'utente.

3.27.3.1.2 Utilizzo

Viene utilizzata per la lettura dei comandi inseriti dall'utente. Una volta inserito il comando da parte dell'utente quest'ultimo viene inviato a [GrammarParser](#) che si occuperà del parsing e dell'analisi del comando.

3.27.3.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
loop	privato	immutabile	Boolean	Rappresenta la condizione di uscita dal ciclo di lettura dei comandi
commandInvoker	privato	immutabile	CommandInvoker	Rappresenta il riferimento al model
view	privato	immutabile	ResultViewer	Rappresenta il riferimento alla view
grammarParser	privato	immutabile	GrammarParser	Rappresenta il riferimento al controller
reader	privato	immutabile	ConsoleReader	Rappresenta il reader per console application
history	privato	immutabile	FileHistory	Rappresenta il file contenente la storia dei comandi
prompt	privato	immutabile	ActorbasePrompt	Rappresenta il prompt di Actorbase
banner	privato	immutabile	ActorbaseBanner	Rappresenta il banner di Actorbase
completers	privato	mutabile	List di String	Lista di stringhe che serve per suggerire i completamenti
out	privato	immutabile	PrintWriter	Oggetto di tipo PrintWriter



3.27.3.2 `actorbase::cli::views::ActorbaseBanner`

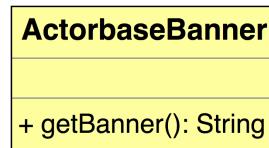


Figura 116: CLI: classe ActorbaseBanner

3.27.3.2.1 Descrizione

Classe per la creazione del banner_G da mostrare all'avvio della CLI_G.

3.27.3.2.2 Utilizzo

Viene utilizzata per la creazione di un banner_G di presentazione all'avvio della CLI_G.

3.27.3.2.3 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
getBanner	pubblico	String	Crea il banner e lo ritorna come stringa

3.27.3.3 `actorbase::cli::views::ActorbasePrompt`

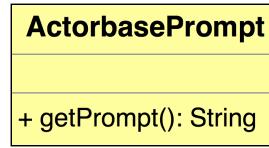


Figura 117: CLI: classe ActorbasePrompt

3.27.3.3.1 Descrizione

Classe che implementa l'interfaccia `actorbase::cli::views::PromptProvider` per la creazione di un prompt_G dei comandi.

3.27.3.3.2 Utilizzo

Viene utilizzata per la creazione di un prompt_G per l'inserimento dei comandi e la visualizzazione delle informazioni relative alla connessione effettuata.



3.27.3.3.3 Classi ereditate

- `actorbase::cli::views::PromptProvider`.

3.27.3.3.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
<code>getPrompt</code>	pubblico	String	Ritorna la stringa che rappresenta il prompt_G di <code>actorbase</code>

3.27.3.3.5 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
<code>os</code>	privato	immutabile	String	Oggetto che rappresenta il nome del sistema

3.27.3.4 `actorbase::cli::views::ResultViewer`

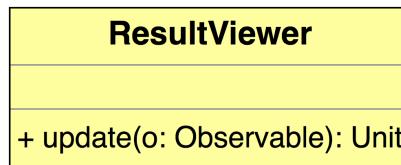


Figura 118: CLI: classe `ResultViewer`

3.27.3.4.1 Descrizione

Classe per la gestione e la formattazione in output dei risultati ottenuti dall'esecuzione del comando ricevuto in input tramite il design pattern_G Observer_G.

3.27.3.4.2 Utilizzo

Viene utilizzata per la gestione e la formattazione in output dei risultati ottenuti dal server mediante l'aggiornamento ricevuto dalla componente model tramite il design pattern_G Observer_G. Essa quindi si occupa di gestire la visualizzazione dei dati ricevuti dalla classe `CommandInvoker`.

3.27.3.4.3 Realizza

- `actorbase::cli::views::Observer`.



3.27.3.4.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
update	pubblico	void	Aggiorna lo stato della view _G

3.27.3.4.5 Parametri

update

Nome	Tipo	Descrizione
o	Observable	Rappresenta l'oggetto osservato



3.28 actorbase::cli::controllers

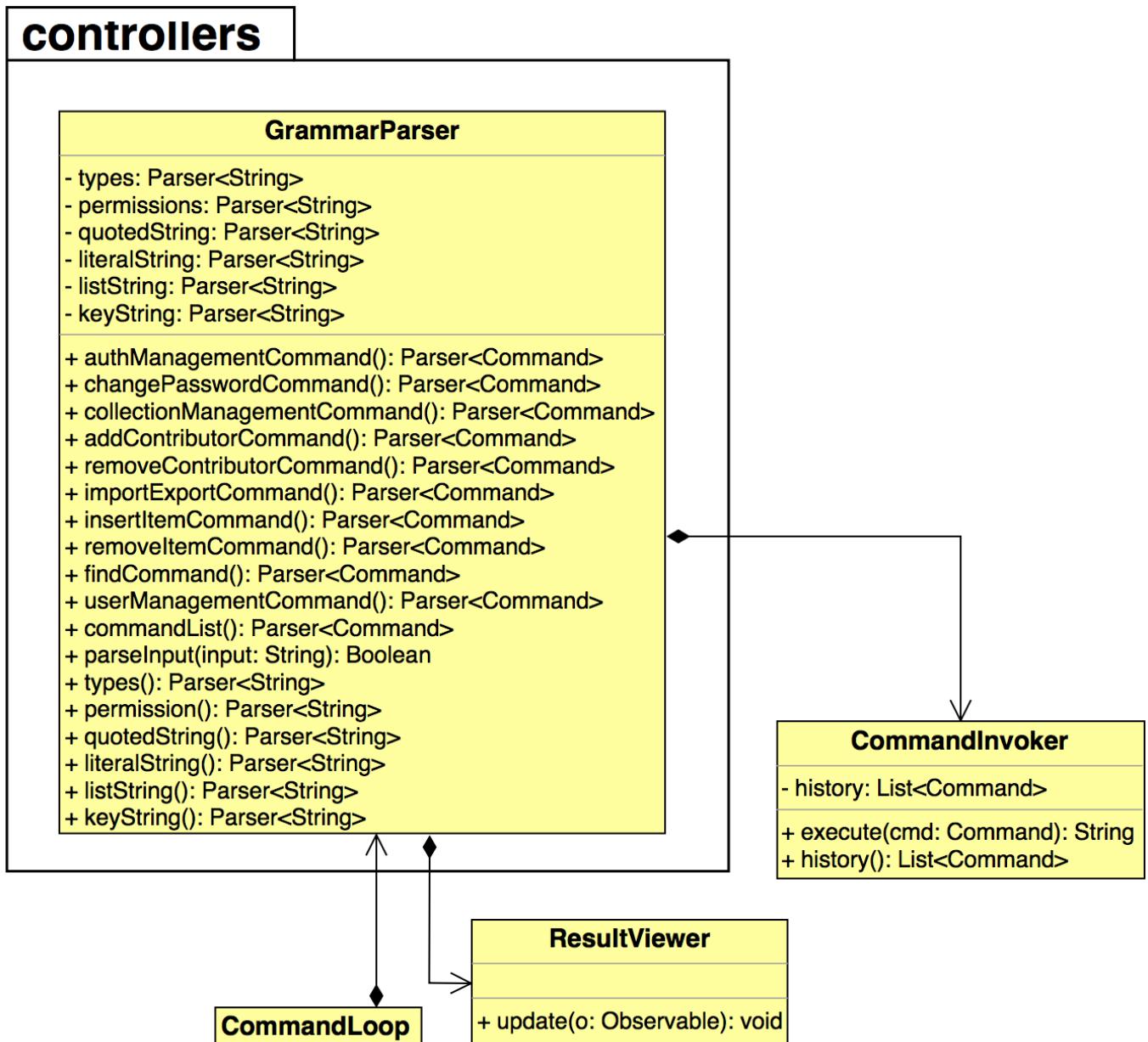


Figura 119: CLI: package Controllers



3.28.1 Descrizione

Package_G per la parte controller_G della cli_G. Questa componente si occupa di passare il comando ricevuto dalla view_G al model_G.

3.28.2 Classi

3.28.2.1 actorbase::cli::controllers::GrammarParser

GrammarParser
- types: Parser[String] - permissions: Parser[String] - quotedString: Parser[String] - literalString: Parser[String] - listString: Parser[String] - keyString: Parser[String]
+ authManagementCommand(): Parser[Command] + changePasswordCommand(): Parser[Command] + collectionManagementCommand(): Parser[Command] + addContributorCommand(): Parser[Command] + removeContributorCommand(): Parser[Command] + importExportCommand(): Parser[Command] + insertItemCommand(): Parser[Command] + removeItemCommand(): Parser[Command] + findCommand(): Parser[Command] + userManagementCommand(): Parser[Command] + commandList(): Parser[Command] + parseInput(input: String): Boolean

Figura 120: CLI: package Controllers

3.28.2.1.1 Descrizione

Classe che si occupa di effettuare il parsing_G del comando ricevuto dalla componente view_G e di chiamare la componente model_G con i parametri opportuni.

3.28.2.1.2 Utilizzo

Viene utilizzata per effettuare il parsing_G del comando ricevuto da [actorbase::cli::views::CommandLoop](#). Durante questa operazione controlla anche che il comando inserito sia effettivamente accettato dal DSL (Domain-Specific Language). In seguito chiama la classe [CommandInvoker](#) con i parametri corretti per consentire l'esecuzione del comando.



3.28.2.1.3 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
types	privato	immutabile	Parser<String>	Rappresenta i tipi accettati dalla shell
permissions	privato	immutabile	Parser<String>	Rappresenta i permessi accettati dalla shell
quotedString	privato	immutabile	Parser<String>	Rappresenta le Quoted String accettate dalla shell
literalString	privato	immutabile	Parser<String>	Rappresenta le Stringhe letterali accettate dalla shell
listString	privato	immutabile	Parser<String>	Rappresenta le ListString accettate dalla shell
keyString	privato	immutabile	Parser<String>	Rappresenta le keyString accettate dalla shell

3.28.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
authManagementCommand	pubblico	Parser di Command	Fa il parsing del comando di autenticazione e istanzia il LoginCommand o il LogoutCommand dedicato a seconda del caso
changePasswordCommand	pubblico	Parser di Command	Fa il parsing del comando di cambio password e istanzia il ChangePasswordCommand
helpCommand	pubblico	Parser di Command	Fa il parsing del comando di aiuto e istanzia l' HelpCommand
collectionManagementCommand	pubblico	Parser di Command	Fa il parsing del comando di gestione delle collezioni e istanzia il CreateCollectionCommand , il RemoveCollectionCommand , il ListCollectionsCommand o il RenameCollectionCommand a seconda del caso
addContributorCommand	pubblico	Parser di Command	Fa il parsing del comando di aggiunta collaboratore e istanzia l' AddContributorCommand
removeContributorCommand	pubblico	Parser di Command	Fa il parsing del comando di rimozione collaboratore e istanzia il RemoveContributorCommand
importExportCommand	pubblico	Parser di Command	Fa il parsing del comando di import/export e istanzia l' ImportCommand o l' ExportCommand a seconda del caso
insertItemCommand	pubblico	Parser di Command	Fa il parsing del comando di inserimento item e istanzia l' InsertItemCommand
removeItemCommand	pubblico	Parser di Command	Fa il parsing del comando di rimozione item e istanzia il RemoveItemCommand
findCommand	pubblico	Parser di Command	Fa il parsing del comando di ricerca e istanzia il FindCommand
userManagerCommand	pubblico	Parser di Command	Fa il parsing del comando di gestione utente e istanzia l' AddUserCommand , il RemoveUserCommand o il ResetPasswordCommand a seconda del caso



commandList	pubblico	Parser di Command	Racchiude tutti i possibili metodi di parsing
parseInput	pubblico	Boolean	Metodo dedicato alla cattura dell'input a cui fare il parsing
types	pubblico	Parser di Stringa	Ritorna l'attributo types
permissions	pubblico	Parser di Stringa	Ritorna l'attributo permissions
quotedString	pubblico	Parser di Stringa	Ritorna l'attributo quotedString
literalString	pubblico	Parser di Stringa	Ritorna l'attributo literalString
listString	pubblico	Parser di Stringa	Ritorna l'attributo listString
keyString	pubblico	Parser di Stringa	Ritorna l'attributo keyString

3.28.2.1.5 Parametri

parserInput

Nome	Tipo	Descrizione
input	String	Stringa immessa dall'utente



3.29 actorbase::cli::models

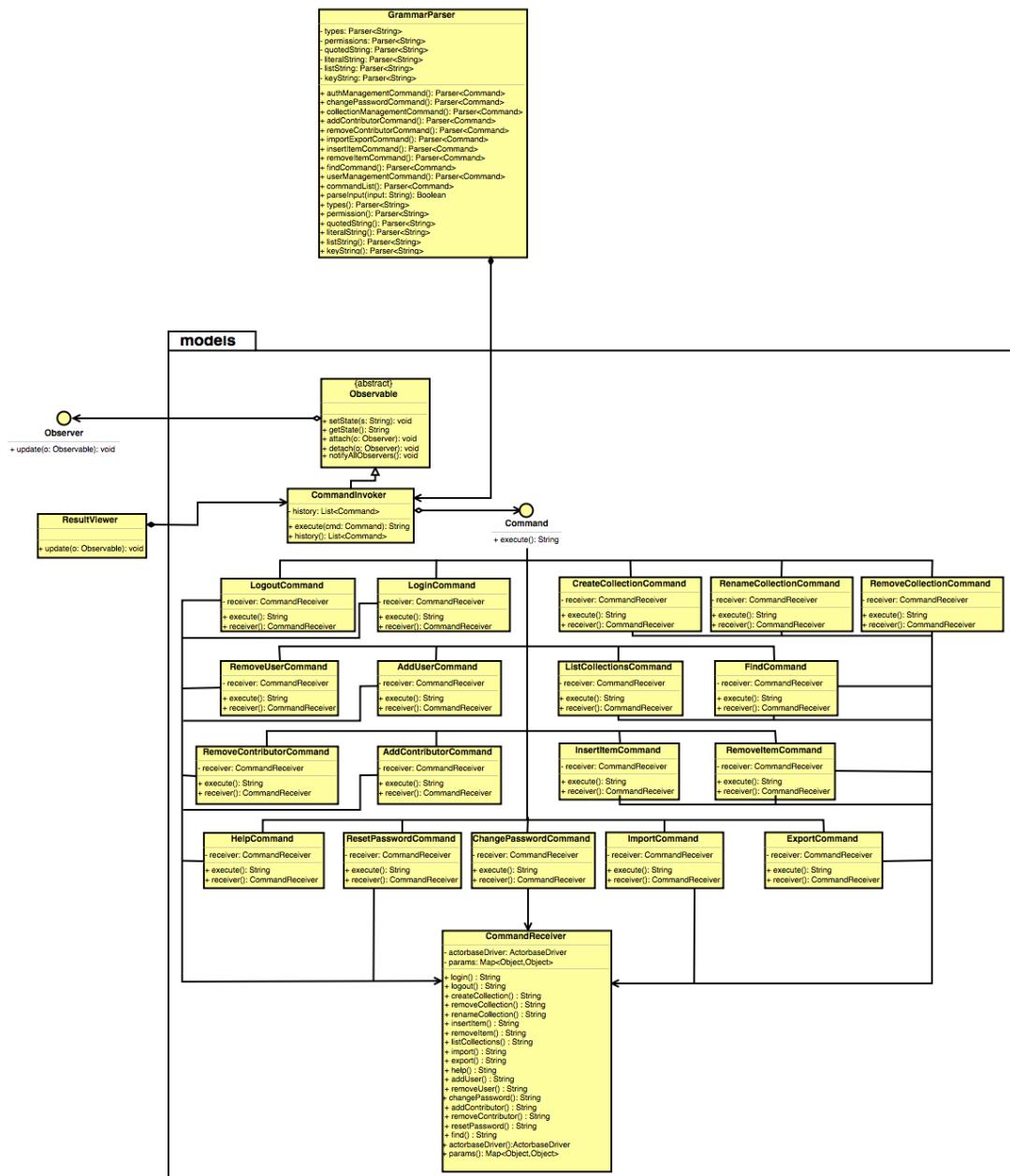


Figura 121: CLI: package Models



3.29.1 Descrizione

Package_G per la parte model_G della cli_G. Questa componente si occupa della comunicazione con la componente driver_G e con la componente view_G tramite il push model_G del design pattern_G observer_G. Per la gestione dei comandi si è scelto di utilizzare il command pattern_G.

3.29.2 Interfacce

3.29.2.1 actorbase::cli::models::Command

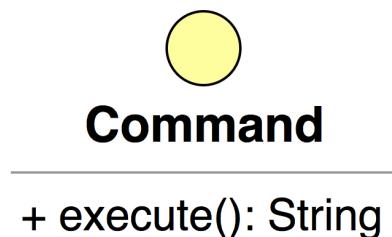


Figura 122: CLI: interfaccia Command

3.29.2.1.1 Descrizione

Interfaccia generica che rappresenta un comando.

3.29.2.1.2 Utilizzo

Viene utilizzata per offrire un'interfaccia comune a tutti i comandi.
Ha un riferimento a [CommandReceiver](#) per l'esecuzione del comando.

3.29.2.1.3 Realizzata da

- [actorbase::cli::models::AddContributorCommand](#);
- [actorbase::cli::models::AddUserCommand](#);
- [actorbase::cli::models::ChangePasswordCommand](#);
- [actorbase::cli::models::CreateCollectionCommand](#);
- [actorbase::cli::models::RemoveCollectionCommand](#);
- [actorbase::cli::models::ExportCommand](#);
- [actorbase::cli::models::FindCommand](#);



- actorbase::cli::models::HelpCommand;
- actorbase::cli::models::InsertItemCommand;
- actorbase::cli::models::ListCollectionsCommand;
- actorbase::cli::models::LoginCommand;
- actorbase::cli::models::LogoutCommand;
- actorbase::cli::models::RemoveContributorCommand;
- actorbase::cli::models::RemoveItemCommand;
- actorbase::cli::models::RemoveUserCommand;
- actorbase::cli::models::RenameCollectionCommand;
- actorbase::cli::models::RemoveUserCommand;
- actorbase::cli::models::ResetPasswordCommand;

3.29.2.1.4 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Esegue il comando

3.29.3 Classi

3.29.3.1 actorbase::cli::models::Observable (abstract)

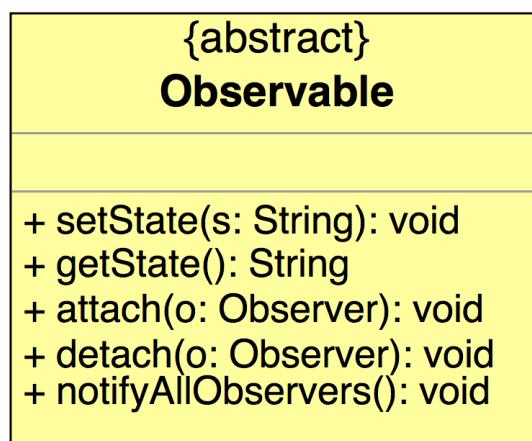


Figura 123: CLI: classe astratta Observable



3.29.3.1.1 Descrizione

Classe astratta che rappresenta l'oggetto osservato nell'ambito del design pattern_G observer_G.

3.29.3.1.2 Utilizzo

Viene utilizzata per notificare la classe che realizza l'interfaccia Observer quando avviene una modifica al proprio stato, ossia quando il driver_G ha restituito un risultato.

3.29.3.1.3 Realizzata da

- actorbase::cli::models::CommandInvoker
- actorbase::cli::controllers::GrammarParser

3.29.3.1.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
observers	privato	mutabile	ListBuffer[Observer]	Lista di oggetti che osservano questa classe
state	privato	mutabile	String	Stato dell'oggetto

3.29.3.1.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
setState	pubblico	void	Cambia lo stato con quello in input
getState	pubblico	String	Ritorna lo stato
attach	pubblico	void	Aggiunge un observer
detach	pubblico	void	Rimuove un observer
notifyAllObservers	pubblico	void	Notifica tutti gli observers col proprio stato

3.29.3.1.6 Parametri

setState

Nome	Tipo	Descrizione
s	String	Stato da assegnare

attach

Nome	Tipo	Descrizione
observer	Observer	Observer da aggiungere all'oggetto

detach



Nome	Tipo	Descrizione
observer	Observer	Observer da rimuovere dall'oggetto

3.29.3.2 actorbase::cli::models::CommandInvoker



Figura 124: CLI: classe CommandInvoker

3.29.3.2.1 Descrizione

Questa classe rappresenta l'Invoker_G del command pattern_G. Essa si occupa di eseguire il comando ricevuto dalla componente controller_G.

3.29.3.2.2 Utilizzo

Viene utilizzata per eseguire il comando ricevuto da [GrammarParser](#) chiamando il metodo opportuno che implementa l'interfaccia [Command](#). Quando riceve una risposta si occupa di notificare la classe [ResultViewer](#) affinché quest'ultima possa procedere con la visualizzazione del risultato in output.

3.29.3.2.3 Realizza

- [actorbase::cli::models::Observable](#).

3.29.3.2.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
history	privato	mutabile	List<Command>	Una lista contenente la history dei comandi inseriti

3.29.3.2.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
storeAndExecute	pubblico	String	Immagazzina il comando nella history e provvede alla sua esecuzione
history	pubblico	List<Command>	Ritorna l'attributo history



3.29.3.2.6 Parametri

setState

Nome	Tipo	Descrizione
cmd	Command	Comando da eseguire

3.29.3.3 actorbase::cli::models::ChangePasswordCommand

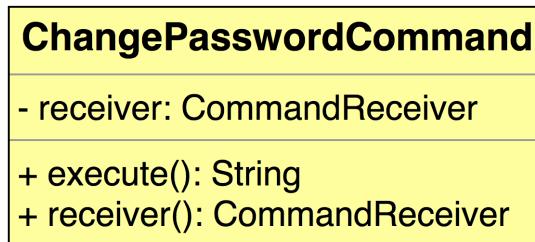


Figura 125: CLI: classe ChangePasswordCommand

3.29.3.3.1 Descrizione

Classe per il comando di cambiamento password.

3.29.3.3.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per il cambiamento della password.

3.29.3.3.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.3.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando



3.29.3.3.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver

3.29.3.4 actorbase::cli::models::FindCommand



Figura 126: CLI: classe ChangePasswordCommand

3.29.3.4.1 Descrizione

Classe per il comando di ricerca.

3.29.3.4.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la ricerca con i parametri immessi dall'utente.

3.29.3.4.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.4.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	CommandReceiver	Oggetto per l'esecuzione del comando

3.29.3.4.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver



3.29.3.5 actorbase::cli::models::HelpCommand

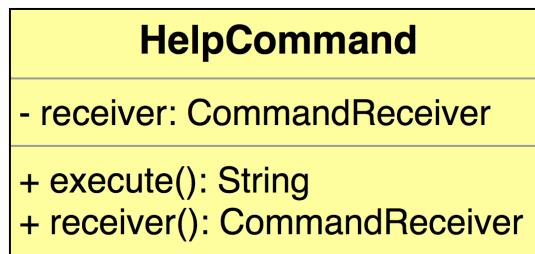


Figura 127: CLI: classe HelpCommand

3.29.3.5.1 Descrizione

Classe per il comando di aiuto.

3.29.3.5.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per richiedere aiuto sulle possibili operazioni da fare.

3.29.3.5.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.5.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.5.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.6 actorbase::cli::models::LogoutCommand

LogoutCommand
- receiver: CommandReceiver
+ execute(): String
+ receiver(): CommandReceiver

Figura 128: CLI: classe LogoutCommand

3.29.3.6.1 Descrizione

Classe per il comando di logout.

3.29.3.6.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per il logout.

3.29.3.6.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.6.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.6.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.7 `actorbase::cli::models::ResetPasswordCommand`

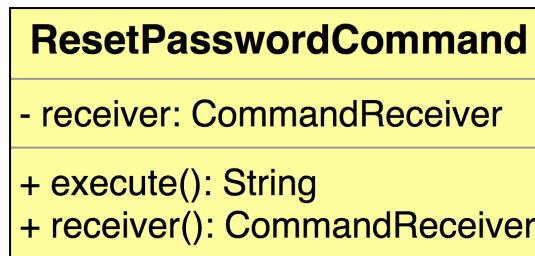


Figura 129: CLI: classe ResetPasswordCommand

3.29.3.7.1 Descrizione

Classe per il comando di reset della password.

3.29.3.7.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per il reset della password dell'utente specificato in input.

3.29.3.7.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.7.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
<code>receiver</code>	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.7.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
<code>execute</code>	pubblico	<code>String</code>	Provvede all'esecuzione del comando
<code>receiver</code>	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.8 `actorbase::cli::models::ListCollectionsCommand`



Figura 130: CLI: classe ListCollectionsCommand

3.29.3.8.1 Descrizione

Classe per il comando di visualizzazione dell'elenco dei nomi di tutte le collezioni presenti nel database_G.

3.29.3.8.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la visualizzazione dell'elenco dei nomi di tutte le collezioni presenti nel database_G.

3.29.3.8.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.8.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.8.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.9 `actorbase::cli::models::AddContributorCommand`

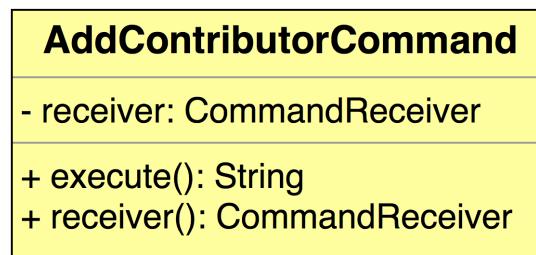


Figura 131: CLI: classe AddContributorCommand

3.29.3.9.1 Descrizione

Classe per il comando di aggiunta collaboratore a una collezione_G.

3.29.3.9.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l'aggiunta di un collaboratore a una collezione_G con il nome della collezione_G e lo username specificati in input.

3.29.3.9.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.9.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.9.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.10 actorbase::cli::models::LoginCommand

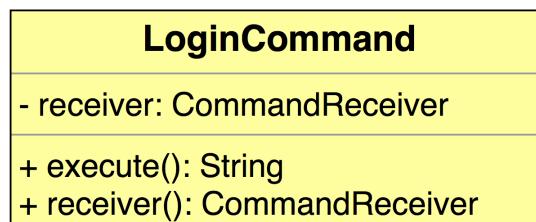


Figura 132: CLI: classe LoginCommand

3.29.3.10.1 Descrizione

Classe per il comando di login.

3.29.3.10.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per il login con le credenziali immesse dall'utente.

3.29.3.10.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.10.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.10.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.11 actorbase::cli::models::RemoveCollectionCommand

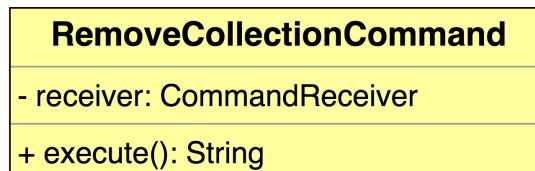


Figura 133: CLI: classe RemoveCollectionCommand

3.29.3.11.1 Descrizione

Classe per il comando di rimozione una collezione_G.

3.29.3.11.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rimozione di una collezione_G con i parametri specificati in input.

3.29.3.11.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.11.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	CommandReceiver	Oggetto per l'esecuzione del comando

3.29.3.11.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver



3.29.3.12 actorbase::cli::models::RemoveContributorCommand

RemoveContributorCommand	
- receiver:	CommandReceiver
+ execute():	String
+ receiver():	CommandReceiver

Figura 134: CLI: classe RemoveContributorCommand

3.29.3.12.1 Descrizione

Classe per il comando di rimozione di un collaboratore da una collezione_G.

3.29.3.12.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rimozione di un collaboratore da una collezione_G con con il nome della collezione_G e lo username specificati in input.

3.29.3.12.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.12.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.12.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.13 actorbase::cli::models::RenameCollectionCommand

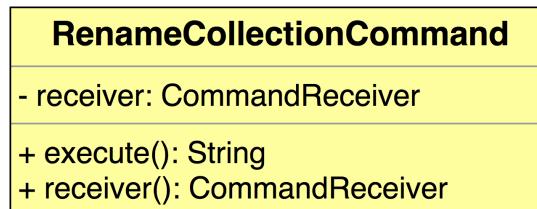


Figura 135: CLI: classe RenameCollectionCommand

3.29.3.13.1 Descrizione

Classe per il comando rinominazione di una collezione_G.

3.29.3.13.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rinominazione di una collezione_G con i parametri specificati in input.

3.29.3.13.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.13.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	CommandReceiver	Oggetto per l'esecuzione del comando

3.29.3.13.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver



3.29.3.14 actorbase::cli::models::AddUserCommand

AddUserCommand
- receiver: CommandReceiver
+ execute(): String
+ receiver(): CommandReceiver

Figura 136: CLI: classe AddUserCommand

3.29.3.14.1 Descrizione

Classe per il comando di aggiunta di un utente.

3.29.3.14.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l'aggiunta di un utente con i parametri specificati in input.

3.29.3.14.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.14.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	CommandReceiver	Oggetto per l'esecuzione del comando

3.29.3.14.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver



3.29.3.15 actorbase::cli::models::RemoveItemCommand

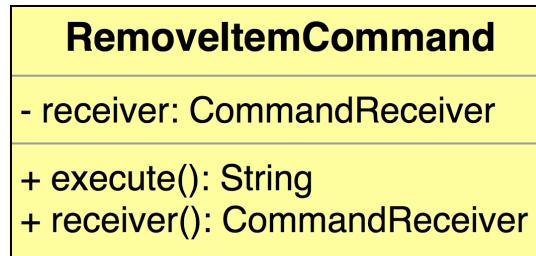


Figura 137: CLI: classe RemoveItemCommand

3.29.3.15.1 Descrizione

Classe per il comando di rimozione di un item_G.

3.29.3.15.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rimozione di un item_G con i parametri immessi dall'utente.

3.29.3.15.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.15.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.15.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.16 actorbase::cli::models::RemoveUserCommand

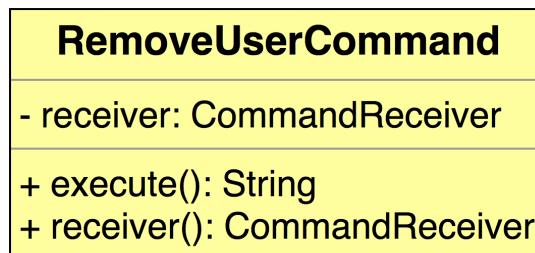


Figura 138: CLI: classe RemoveUserCommand

3.29.3.16.1 Descrizione

Classe per il comando di rimozione di un utente.

3.29.3.16.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rimozione di un utente con i parametri immessi dall'utente.

3.29.3.16.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.16.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.16.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.17 actorbase::cli::models::ImportCommand

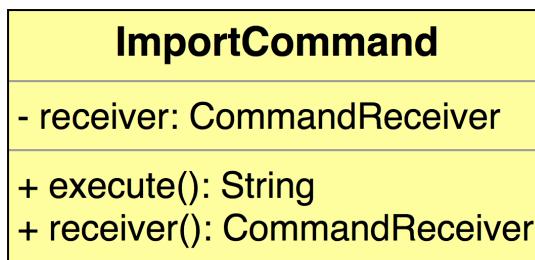


Figura 139: CLI: classe ImportCommand

3.29.3.17.1 Descrizione

Classe per il comando di importazione da un file JSON_c.

3.29.3.17.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l'importazione da un file JSON_c con i parametri immessi dall'utente.

3.29.3.17.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.17.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.17.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.18 actorbase::cli::models::InsertItemCommand

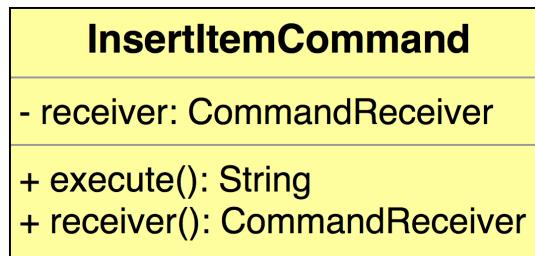


Figura 140: CLI: classe InsertItemCommand

3.29.3.18.1 Descrizione

Classe per il comando di inserimento di un item_G in una collezione_G.

3.29.3.18.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l' inserimento di un item_G in una collezione_G con i parametri immessi dall'utente.

3.29.3.18.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.18.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	CommandReceiver	Oggetto per l'esecuzione del comando

3.29.3.18.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	String	Provvede all'esecuzione del comando
receiver	pubblico	CommandReceiver	Ritorna l'attributo receiver



3.29.3.19 actorbase::cli::models::CreateCollectionCommand

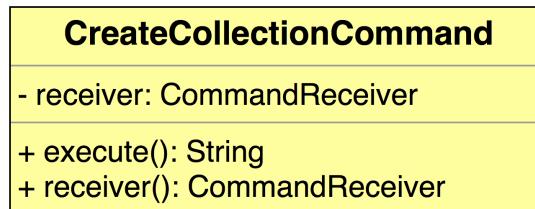


Figura 141: CLI: classe CreateCollectionCommand

3.29.3.19.1 Descrizione

Classe per il comando di creazione di una nuova collezione_G.

3.29.3.19.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la creazione di una nuova collezione_G con i parametri immessi dall'utente.

3.29.3.19.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.19.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.19.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.20 actorbase::cli::models::ExportCommand



Figura 142: CLI: classe ExportCommand

3.29.3.20.1 Descrizione

Classe per il comando di esportazione in file JSON_G.

3.29.3.20.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l'esportazione in file JSON_G con i parametri immessi dall'utente.

3.29.3.20.3 Classi ereditate

- `actorbase::cli::models::Command`.

3.29.3.20.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
receiver	privato	mutabile	<code>CommandReceiver</code>	Oggetto per l'esecuzione del comando

3.29.3.20.5 Metodi

Nome	Accesso	Tipo di ritorno	Descrizione
execute	pubblico	<code>String</code>	Provvede all'esecuzione del comando
receiver	pubblico	<code>CommandReceiver</code>	Ritorna l'attributo receiver



3.29.3.21 actorbase::cli::models::CommandReceiver

CommandReceiver	
- actorbaseDriver: ActorbaseDriver	
- params: Map<Object, Object>	

+ login(): String
+ logout(): String
+ createCollection(): String
+ removeCollection(): String
+ renameCollection(): String
+ insertItem(): String
+ removeItem(): String
+ listCollections(): String
+ import(): String
+ export(): String
+ help(): String
+ addUser(): String
+ removeUser(): String
+ changePassword(): String
+ addContributor(): String
+ removeContributor(): String
+ resetPassword(): String
+ find(): String
+ actorbaseDriver(): ActorbaseDriver
+ params(): Map<Object, Object>

Figura 143: CLI: classe CommandReceiver

3.29.3.21.1 Descrizione

Classe contenente i metodi per richiamare la componente driver_G per l'esecuzione dei comandi o per restituire l'aiuto se richiesto.

3.29.3.21.2 Utilizzo

Viene utilizzata da [actorbase::cli::models::Command](#) per chiamare [actorbase::driver::client::ActorbaseClient](#) o per ricevere un aiuto da visualizzare se richiesto dall'utente. Quando riceve una risposta dal driver_G, o dopo aver elaborato la richiesta di aiuto, si occupa di mandare il risultato delle operazioni a [actorbase::cli::models::CommandInvoker](#) che si occuperà di mandarlo alla parte view sfruttando il design pattern_G Observer_G.



3.29.3.21.3 Interazioni con altre classi

- `actorbase::cli::models::Command`;
- `actorbase::driver::client::ActorbaseClient`.

3.29.3.21.4 Attributi

Nome	Accesso	Mutabilità	Tipo	Descrizione
<code>actorbaseDriver</code>	privato	mutabile	<code>ActorbaseDriver</code>	Istanza di ActorbaseDriver usata per l'esecuzione dei comandi
<code>params</code>	privato	mutabile	<code>Map<Object, Object></code>	Parametri inseriti dall'utente per l'esecuzione dei comandi



3.29.3.21.5 Metodi

Nome	Accesso	Ritorno	Descrizione
insert	pubblico	String	Metodo per richiamare il comando di inserimento del driver
removeItem	pubblico	String	Metodo per richiamare il comando di rimozione Item del driver
export	pubblico	String	Metodo per richiamare il comando di esportazione su file del driver
login	pubblico	String	Metodo per richiamare il comando di login del driver
logout	pubblico	String	Metodo per richiamare il comando di logout del driver
find	pubblico	String	Metodo per richiamare il comando di ricerca di items del driver
help	pubblico	String	Metodo per il comando di aiuto
createCollection	pubblico	String	Metodo per richiamare il comando di creazione collezione _g del driver
listCollections	pubblico	String	Metodo per richiamare il metodo di lista collezioni del driver
renameCollection	pubblico	String	Metodo per richiamare il metodo di rinomina collezione _g del driver
deleteCollection	pubblico	String	Metodo per richiamare il metodo di rimozione collezione _g del driver
addContributor	pubblico	String	Metodo per richiamare il metodo di aggiunta collaboratore del driver
removeContributor	pubblico	String	Metodo per richiamare il metodo di rimozione collaboratore del driver
changePassword	pubblico	String	Metodo per richiamare il metodo di cambio password del driver
addUser	pubblico	String	Metodo per richiamare il metodo di aggiunta utente del driver
removeUser	pubblico	String	Metodo per richiamare il metodo di rimozione utente del driver
resetPassword	pubblico	String	Metodo per richiamare il metodo di reset password del driver
actorbaseDriver	pubblico	ActorbaseDriver	Ritorna l'attributo actorbaseDriver
params	pubblico	Map<Object, Object>	Ritorna l'attributo params



4 Diagrammi di Sequenza

In questa sezione, vengono illustrati i diagrammi di sequenza che descrivono le interazioni tra le varie componenti del sistema e in particolare lo scambio di messaggi che avviene tra gli attori nella componente server di actorbase.

Sono stati creati dai *Progettisti* diversi diagrammi con diversi livelli di astrazione.

Per semplificare i diagrammi, evitare ripetitività e migliorare la leggibilità di questi, nella maggior parte dei seguenti l'input sarà dato dalla componente Driver_G. In un utilizzo reale sarà un utente che utilizza il Driver_G o la CLI_G che darà l'input iniziale.

4.1 Interazioni generali tra componenti

Questo diagramma rappresenta l'interazione generale tra le tre componenti che compongono il sistema actorbase.

L'utente è connesso alla CLI_G e inserisce un comando. A questo punto la componente CLI_G (se non è instanziato un Driver_G lo crea per poi utilizzarlo) utilizza il Driver_G per richiamare il metodo relativo al comando inserito dall'utente.

Il Driver_G provvederà a tradurre il comando coi relativi parametri immessi dall'utente in una richiesta HTTP e la invierà al server. Il server provvederà a fare le elaborazioni richieste attraverso l'uso degli attori e quando queste saranno finite risponderà al Driver_G tramite risposta HTTP il quale provvederà a trasformarla in una Response e lo restituirà al suo utilizzatore.

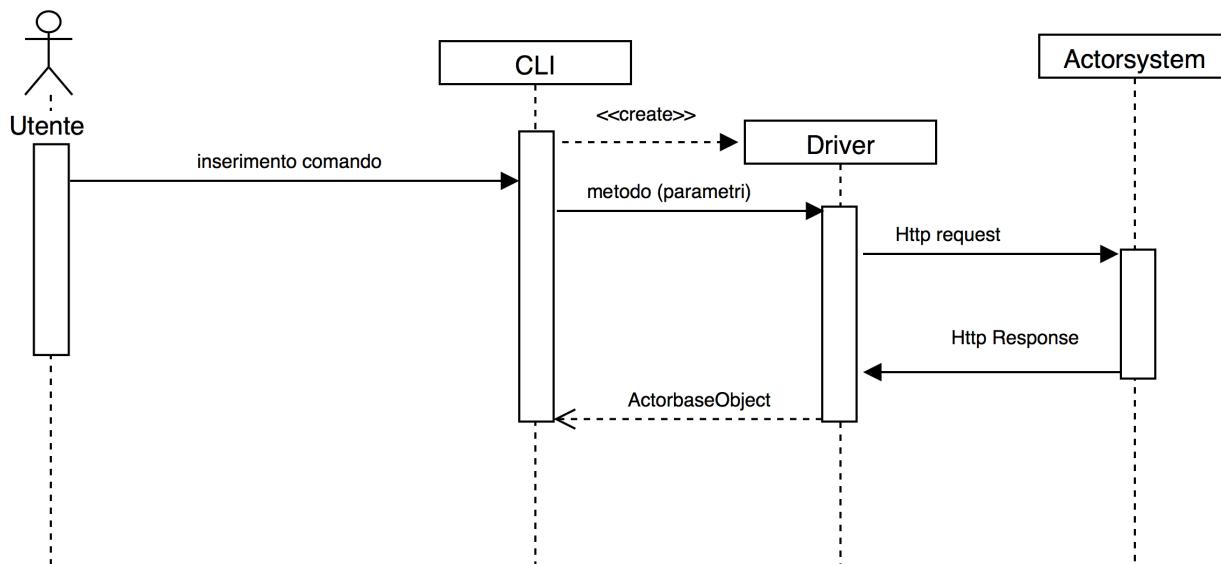


Figura 144: Diagramma di sequenza - Interazioni generali tra componenti



4.2 Inserimento di un Item

Questo diagramma rappresenta l'interazione tra le componenti nel caso di una richiesta di inserimento di un item_G.

La componente `ActorbaseDriver` manda tramite richiesta HTTP al `ClientActor` un payload contenente i dati per eseguire l'operazione scelta dalla route_G.

Da questo componente parte il messaggio di `Insert(key, value, update)` ad un `Main` scelto in base ad una politica di routing_G. Il `Main` manderà un messaggio allo `Storefinder` relativo il quale manderà un messaggio allo `StoreKeeper` che dovrà salvarsi l'informazione sulla propria mappa.

A questo punto lo `Storekeeper` manderà un messaggio al `ClientActor` con la risposta (positiva o negativa che sia) il quale risponderà tramite risposta HTTP al `DriverG` con il risultato.

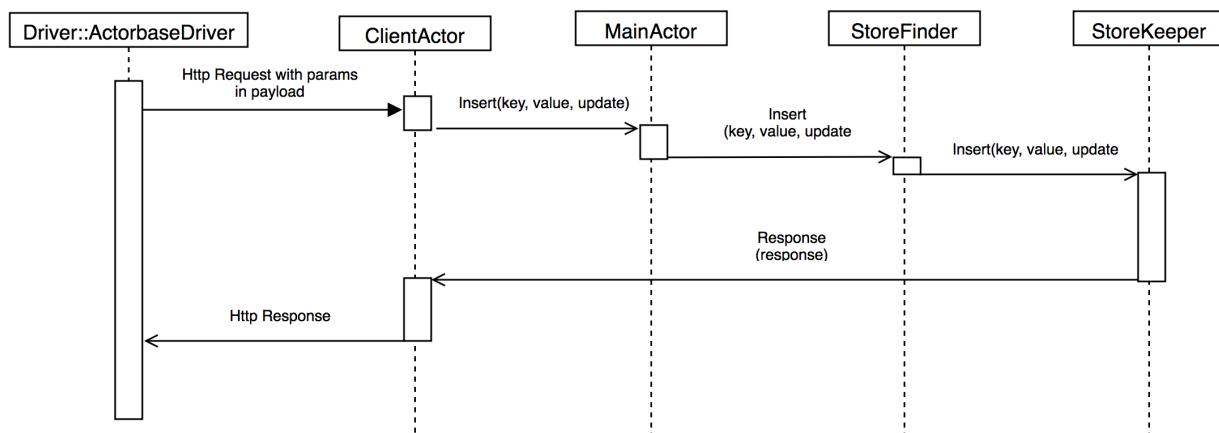


Figura 145: Diagramma di sequenza - Inserimento di un item

4.3 Inserimento di un Item con Storekeeper pieno

Questo diagramma rappresenta l'interazione tra le componenti nel caso di una richiesta di inserimento di un item_G.

La componente `ActorbaseDriver` manda tramite richiesta HTTP al `ClientActor` un payload contenente i dati per eseguire l'operazione scelta dalla route_G.

Da questo componente parte il messaggio di `Insert(key, value, update)` ad un `Main` scelto in base ad una politica di routing_G. Il `Main` manderà un messaggio allo `Storefinder` relativo il quale manderà un messaggio allo `StoreKeeper` che essendo pieno manda un messaggio `OneMore` ad un `Manager`.

Il `Manager` provvede a creare un nuovo attore `StoreKeeper` e provvederà ad inizializzarlo tramite messaggio `InitMn`.

A questo punto il nuovo `Storekeeper` manderà un messaggio al `ClientActor` con la risposta (positiva o negativa che sia) il quale risponderà tramite risposta HTTP al `DriverG` con il risultato.

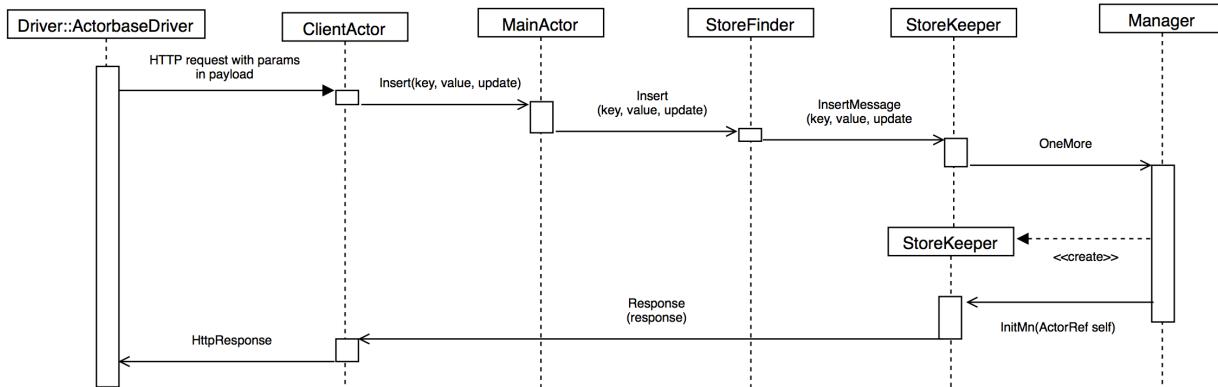


Figura 146: Diagramma di sequenza - Inserimento di un item con Storekeeper pieno

4.4 Autenticazione

Questo diagramma rappresenta l'interazione tra le componenti nel caso di una richiesta di autenticazione di un utente.

La componente **ActorbaseDriver** manda tramite richiesta di autenticazione HTTP al **ClientActor**.

Da questo componente parte il messaggio **Authenticate** contenente le credenziali inserite rivolto verso l'attore **AuthActor**.

Questo attore controllerà le credenziali inserite per vedere se corrispondono ad un utente registrato al sistema e risponderà al **ClientActor** tramite messaggio **Response** avvisando se l'autenticazione ha avuto successo o meno. Quest'ultimo attore risponderà al software richiedente attraverso una risposta HTTP indicando l'esito dell'operazione.

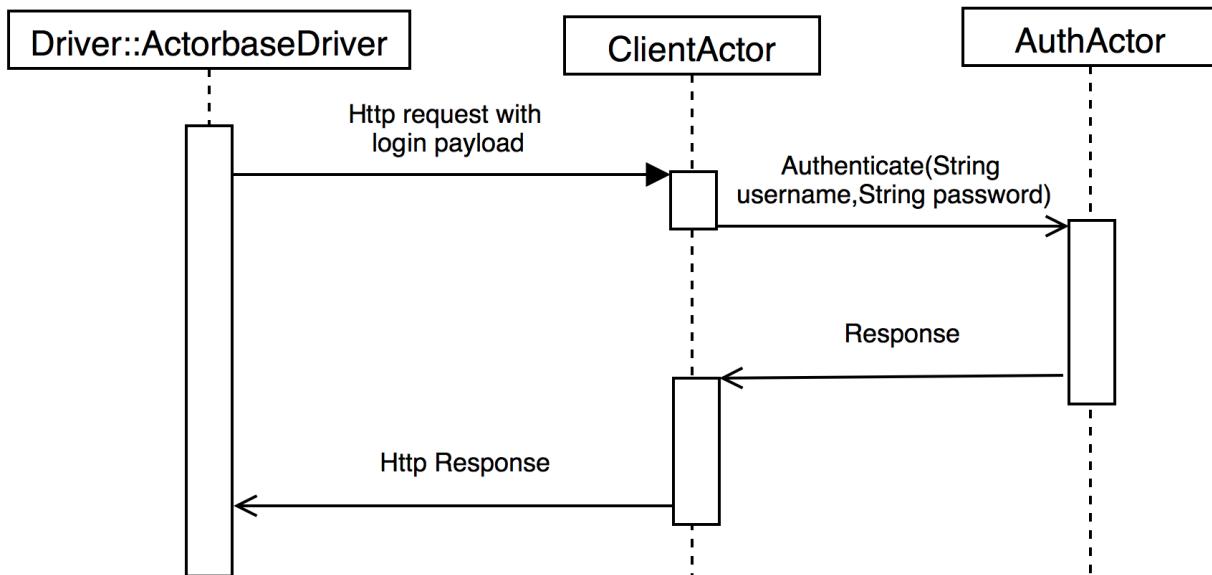


Figura 147: Diagramma di sequenza - Autenticazione

4.5 Salvataggio su filesystem

Questo diagramma rappresenta l'interazione tra le componenti nel caso venga effettuato un salvataggio dei dati su filesystem.

La componente **ActorbaseDriver** manda tramite richiesta HTTP al **ClientActor**.

Da questo componente parte il messaggio di **Insert(key, value, update)** ad un **Main** scelto in base ad una politica di routing_G. Il **Main** manderà un messaggio allo **Storefinder** relativo il quale manderà un messaggio allo **StoreKeeper** che dovrà salvarsi l'informazione sulla propria mappa.

A questo punto lo **StoreKeeper** manderà un messaggio al **Warehouseman** di tipo **Save** e successivamente manderà un messaggio al **ClientActor** con la risposta (positiva o negativa che sia) il quale risponderà sempre tramite risposta HTTP al **Driver_G** con il risultato.

Nel frattempo l'attore_G di tipo **Warehouseman** eseguirà un salvataggio su disco dei propri dati previa criptazione_G. Si noti che la frequenza con cui gli attori di tipo **Warehouseman** eseguiranno l'aggiornamento delle proprie strutture dati e il salvataggio sarà specificato tramite un parametro di configurazione del sistema.

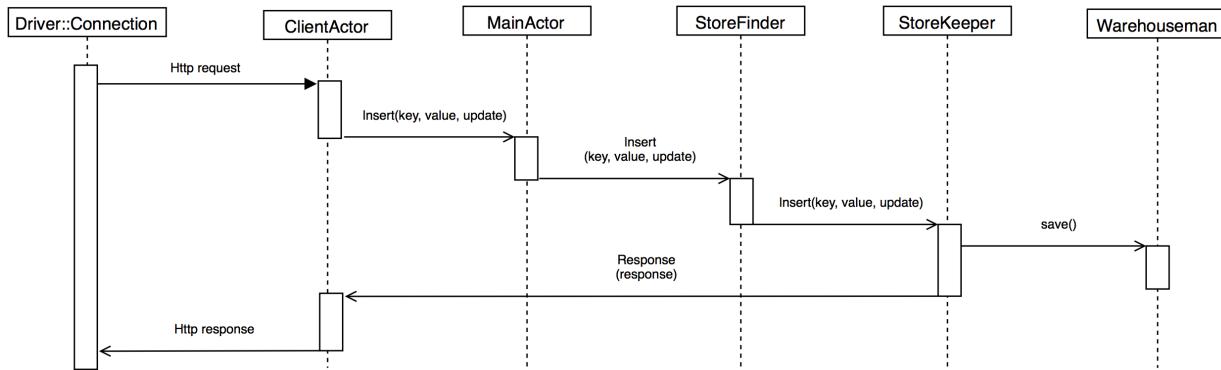


Figura 148: Diagramma di sequenza - Inserimento di un item con salvataggio su filesystem

4.6 Aggiunta collaboratore (in sola lettura)

Questo diagramma rappresenta l'interazione tra le componenti nel caso di una richiesta di autenticazione di un utente.

La componente [ActorbaseDriver](#) manda tramite richiesta HTTP al [ClientActor](#).

Da questo componente parte il messaggio di [addContributor\(collection, permission, username\)](#) ad un [Main](#) scelto in base ad una politica di routing_G.

Il [Main](#) provvederà ad aggiornare le proprie strutture dati per riflettere l'aggiornamento richiesto. Quando avrà finito risponderà all'attore richiedente con un messaggio [Response](#) avvisando se l'operazione ha avuto successo o meno. Quest'ultimo attore risponderà al software richiedente attraverso una risposta HTTP indicando l'esito dell'operazione.

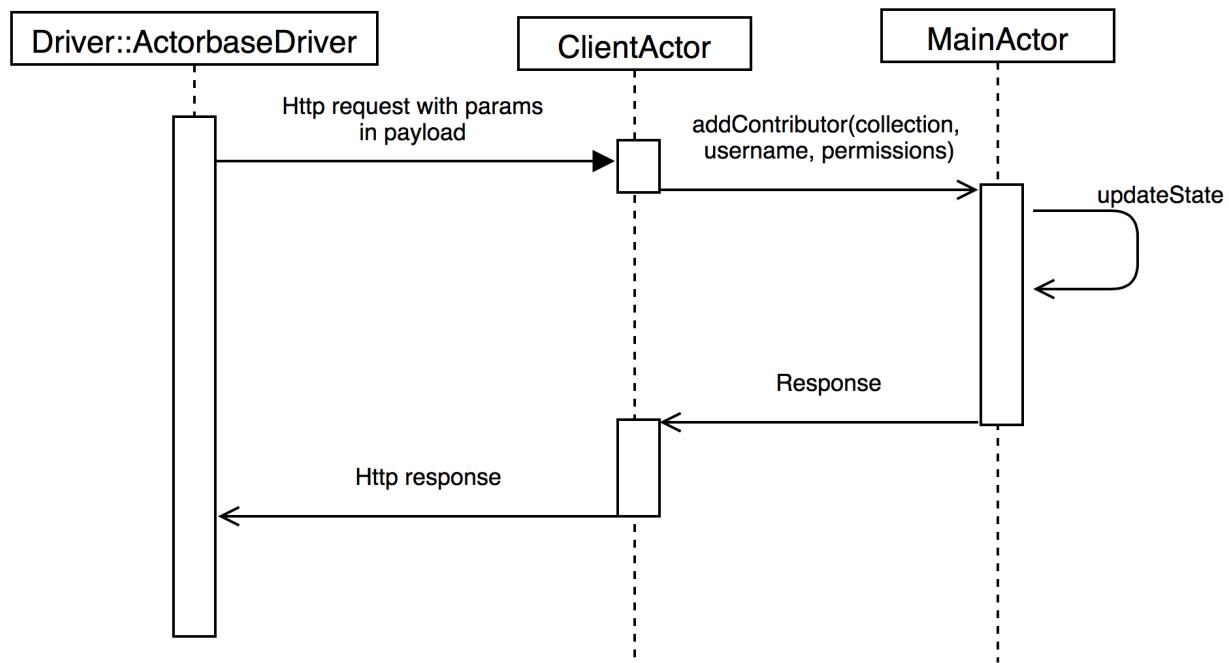


Figura 149: Diagramma di sequenza - aggiunta collaboratore



5 Tracciamento

6 Tracciamento Classi-Requisiti

Classi	Requisiti
actorbase::actorsystem::authactor::AuthActor	OBF1.1.10.1 OBF1.1.10.1.1 OBF1.1.10.2 OBF1.1.10.2.1 OBF1.1.10.2.3 OBF1.1.10.2.3.1 OBF1.1.10.2.4 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.6 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.2 OBF1.1.10.5 OBF1.1.10.5.1 OBF1.1.10.7 OBF1.1.10.7.1 OBF1.1.10.7.3
actorbase::actorsystem::authactor::messages::AddCollectionTo	OBF1.1.9.5
actorbase::actorsystem::authactor::messages::AddCredentials	OBF1.1.9.1
actorbase::actorsystem::authactor::messages::Authenticate	OBF1.1.9.4
actorbase::actorsystem::authactor::messages::ListCollectionsOf	OBF1.1.9.7
actorbase::actorsystem::authactor::messages::RemoveCollectionFrom	OBF1.1.9.6
actorbase::actorsystem::authactor::messages::RemoveCredentials	OBF1.1.9.3
actorbase::actorsystem::authactor::messages::UpdateCredentials	OBF1.1.9.2
actorbase::actorsystem::clientactor::Authenticator	OBF1.1.10.1 OBF1.1.10.1.1



actorbase::actorsystem::clientactor::Clientactor	OBF1.1.10 OBF1.1.10.1 OBF1.1.10.1.1 OBF1.1.10.2 OBF1.1.10.2.1 OBF1.1.10.2.1.1 OBF1.1.10.2.2 OBF1.1.10.2.3 OBF1.1.10.2.3.1 OBF1.1.10.2.3.2 OBF1.1.10.2.4 OBF1.1.10.2.4.1 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.5.3 OBF1.1.10.2.5.4 OBF1.1.10.2.6 OBF1.1.10.2.6.1 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.1 OBF1.1.10.3.2 OBF1.1.10.3.3 OBF1.1.10.3.4 OBF1.1.10.3.5 OBF1.1.10.4 OBF1.1.10.5 OBF1.1.10.5.1 OBF1.1.10.6 OBF1.1.10.7 OBF1.1.10.7.1 OBF1.1.10.7.1.1 OBF1.1.10.7.2 OBF1.1.10.7.2.1 OBF1.1.10.7.3 OBF1.1.10.7.3.1 OBF1.1.2
actorbase::actorsystem::clientactor::CollectionApi	OBF1.1.10.2
actorbase::actorsystem::clientactor::ItemApi	OBF1.1.10.3 OBF1.1.10.4
actorbase::actorsystem::clientactor::UserApi	OBF1.1.10.1 OBF1.1.10.5 OBF1.1.10.6 OBF1.1.10.7
actorbase::actorsystem::clientactor::messages::MapResponse	OBF1.1.2.2
actorbase::actorsystem::clientactor::messages::Response	OBF1.1.2.1
actorbase::actorsystem::httpserver::Httpserver	OBF1.1 OBF1.1.1 OBF1.1.1.1 OBF1.1.10
actorbase::actorsystem::httpserver::SslConfiguration	OBF1.1 OBF1.1.1
actorbase::actorsystem::main::Main	OBF1.1.10.1 OBF1.1.10.2 OBF1.1.10.2.1 OBF1.1.10.2.1.1 OBF1.1.10.2.3 OBF1.1.10.2.4 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.6 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.1 OBF1.1.10.3.2 OBF1.1.10.3.3 OBF1.1.10.3.4 OBF1.1.10.4 OBF1.1.10.5 OBF1.1.10.7 OBF1.1.10.7.1 OBF1.1.10.7.2 OBF1.1.10.7.2.1 OBF1.1.10.7.3 OBF1.1.10.7.3.1 OBF1.1.3
actorbase::actorsystem::main::messages::AddContributor	OBF1.1.3.5
actorbase::actorsystem::main::messages::CompleteTransaction	OBF1.1.3.8



actorbase::actorsystem::main::messages::CreateCollection	OBF1.1.3.1
actorbase::actorsystem::main::messages::GetItemFrom	OBF1.1.3.4
actorbase::actorsystem::main::messages::RemoveContributor	OBF1.1.3.7
actorbase::actorsystem::main::messages::RemoveItemFrom	OBF1.1.3.9
actorbase::actorsystem::manager::Manager	DEF1.1.8 OBF1.1.10.3 OBF1.1.10.3.1
actorbase::actorsystem::manager::messages::DuplicationRequestSK	DEF1.1.8.1
actorbase::actorsystem::storefinder::Storefinder	OBF1.1.10.1 OBF1.1.10.2 OBF1.1.10.2.1 OBF1.1.10.2.3 OBF1.1.10.2.4 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.5.4 OBF1.1.10.2.6 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.1 OBF1.1.10.3.3 OBF1.1.10.3.4 OBF1.1.10.4 OBF1.1.10.5 OBF1.1.10.7 OBF1.1.10.7.1 OBF1.1.10.7.2 OBF1.1.10.7.2.1 OBF1.1.10.7.3 OBF1.1.10.7.3.1 OBF1.1.4
actorbase::actorsystem::storefinder::messages::GetAllItems	OBF1.1.4.6
actorbase::actorsystem::storefinder::messages::GetItem	OBF1.1.4.3
actorbase::actorsystem::storefinder::messages::Insert	OBF1.1.4.5
actorbase::actorsystem::storefinder::messages::PartialMapTransaction	OBF1.1.4.7
actorbase::actorsystem::storefinder::messages::RemoveItem	OBF1.1.4.4
actorbase::actorsystem::storefinder::messages::UpdateCollectionSize	OBF1.1.4.1
actorbase::actorsystem::storekeeper::Storekeeper	DEF1.1.7 OBF1.1.10.2 OBF1.1.10.2.1 OBF1.1.10.2.4 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.6 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.1 OBF1.1.10.3.3 OBF1.1.10.3.4 OBF1.1.10.4 OBF1.1.5
actorbase::actorsystem::storekeeper::messages::GetAllItems	OBF1.1.5.3
actorbase::actorsystem::storekeeper::messages::GetItem	OBF1.1.5.2
actorbase::actorsystem::storekeeper::messages::Insert	OBF1.1.5.4
actorbase::actorsystem::storekeeper::messages::Persist	OBF1.1.5.1
actorbase::actorsystem::storekeeper::messages::RemoveItem	OBF1.1.5.5
actorbase::actorsystem::utils::ActorbaseCollection	DEF1.1.7 OBF1.1.2 OBF1.1.3 OBF1.1.4 OBF1.1.5
actorbase::actorsystem::utils::CryptoUtils	OBF1.1.6



actorbase::actorsystem::warehouseman::Warehouseman	OBF1.1.10.2 OBF1.1.10.2.4 OBF1.1.10.2.5 OBF1.1.10.2.5.1 OBF1.1.10.2.5.2 OBF1.1.10.2.6 OBF1.1.10.2.7 OBF1.1.10.3 OBF1.1.10.3.1 OBF1.1.10.3.4 OBF1.1.6
actorbase::actorsystem::warehouseman::messages::Clean	OBF1.1.6.3
actorbase::actorsystem::warehouseman::messages::Read	OBF1.1.6.1
actorbase::actorsystem::warehouseman::messages::Save	OBF1.1.6.2
actorbase::cli::controllers::GrammarParser	DEF2.1.2.7 DEF2.1.3.1.2 DEF2.1.4 DEF2.1.5 OBF2.1 OBF2.1.1 OBF2.1.2 OBF2.1.2.1 OBF2.1.2.2 OBF2.1.2.3 OBF2.1.2.4 OBF2.1.2.5 OBF2.1.2.6 OBF2.1.3 OBF2.1.3.1 OBF2.1.3.1.1 OBF2.1.6 OBF2.1.7 OBF2.1.8 OBF2.1.9 OBF2.1.9.1 OBF2.1.9.2 OBF2.1.9.3
actorbase::cli::models::AddContributorCommand	OBF2.1.2.5
actorbase::cli::models::AddUserCommand	OBF2.1.9.1
actorbase::cli::models::ChangePasswordCommand	OBF2.1.7
actorbase::cli::models::Command	DEF2.1.2.7 DEF2.1.3.1.2 OBF2.1.1 OBF2.1.2 OBF2.1.2.1 OBF2.1.2.2 OBF2.1.2.3 OBF2.1.2.4 OBF2.1.2.5 OBF2.1.2.6 OBF2.1.3 OBF2.1.3.1 OBF2.1.6 OBF2.1.7 OBF2.1.8 OBF2.1.9 OBF2.1.9.1 OBF2.1.9.2 OBF2.1.9.3
actorbase::cli::models::CommandInvoker	OBF2.1.2.10 OBF2.1.2.11 OBF2.1.2.12 OBF2.1.2.13 OBF2.1.2.14 OBF2.1.2.8 OBF2.1.2.9 OBF2.1.3.4 OBF2.1.3.5 OBF2.1.7.4 OBF2.1.7.5 OBF2.1.7.6 OBF2.1.9.4 OBF2.1.9.5 OBF2.1.9.6



actorbase::cli::models::CommandReceiver	DEF2.1.2.7 DEF2.1.3.2 DEF2.1.3.3 DEF2.1.4 DEF2.1.5 OBF2.1.1 OBF2.1.2 OBF2.1.2.1 OBF2.1.2.2 OBF2.1.2.3 OBF2.1.2.4 OBF2.1.2.5 OBF2.1.2.6 OBF2.1.3 OBF2.1.3.1 OBF2.1.6 OBF2.1.7 OBF2.1.8 OBF2.1.9 OBF2.1.9.1 OBF2.1.9.2 OBF2.1.9.3
actorbase::cli::models::CreateCollectionCommand	OBF2.1.2.1
actorbase::cli::models::ExportCommand	DEF2.1.2.7
actorbase::cli::models::FindCommand	OBF2.1.6
actorbase::cli::models::HelpCommand	DEF2.1.4 DEF2.1.5
actorbase::cli::models::ImportCommand	DEF2.1.3.1.2
actorbase::cli::models::InsertItemCommand	OBF2.1.3.1
actorbase::cli::models::ListCollectionCommand	OBF2.1.2.2
actorbase::cli::models::LoginCommand	OBF2.1.1
actorbase::cli::models::LogoutCommand	OBF2.1.1
actorbase::cli::models::Observable	OBF2.1.2.10 OBF2.1.2.11 OBF2.1.2.12 OBF2.1.2.13 OBF2.1.2.14 OBF2.1.2.8 OBF2.1.2.9 OBF2.1.3.4 OBF2.1.3.5 OBF2.1.7.4 OBF2.1.7.5 OBF2.1.7.6 OBF2.1.9.4 OBF2.1.9.5 OBF2.1.9.6
actorbase::cli::models::RemoveCollectionCommand	OBF2.1.2.3
actorbase::cli::models::RemoveContributorCommand	OBF2.1.2.6
actorbase::cli::models::RemoveItemCommand	OBF2.1.3.6
actorbase::cli::models::RemoveUserCommand	OBF2.1.9.2
actorbase::cli::models::RenameCollectionCommand	OBF2.1.2.4
actorbase::cli::models::ResetPasswordCommand	OBF2.1.9.3



actorbase::cli::views::CommandLoop	DEF2.1.2.7 DEF2.1.2.7.1 DEF2.1.2.7.2 DEF2.1.3.1.2 DEF2.1.3.1.2.1 DEF2.1.4 DEF2.1.5 DEF2.1.5.1 OBF2 OBF2.1 OBF2.1.1 OBF2.1.1.1 OBF2.1.1.2 OBF2.1.1.3 OBF2.1.2 OBF2.1.2.1 OBF2.1.2.1.1 OBF2.1.2.2 OBF2.1.2.3 OBF2.1.2.3.1 OBF2.1.2.4 OBF2.1.2.4.1 OBF2.1.2.4.2 OBF2.1.2.5 OBF2.1.2.5.1 OBF2.1.2.5.2 OBF2.1.2.5.3 OBF2.1.2.6 OBF2.1.2.6.1 OBF2.1.2.6.2 OBF2.1.3 OBF2.1.3.1 OBF2.1.3.1.1 OBF2.1.3.1.1.1 OBF2.1.3.1.1.2 OBF2.1.3.1.1.3 OBF2.1.3.1.1.4 OBF2.1.3.1.1.5 OBF2.1.3.6 OBF2.1.3.6.1 OBF2.1.3.6.2 OBF2.1.6 OBF2.1.6.1 O BF2.1.6.2 OBF2.1.7 OBF2.1.7.1 OBF2.1.7.2 OBF2.1.7.3 OBF2.1.8 OBF2.1.9 OBF2.1.9.1 OBF2.1.9.1.1 OBF2.1.9.1.2 OBF2.1.9.2 OBF2.1.9.2.1 OBF2.1.9.2.2 OBF2.1.9.3 OBF2.1.9.3.1 OBF2.1.9.3.2
actorbase::cli::views::ResultViewer	OBF2.1.2.10 OBF2.1.2.11 OBF2.1.2.12 OBF2.1.2.13 OBF2.1.2.14 OBF2.1.2.8 OBF2.1.2.9 OBF2.1.3.4 OBF2.1.3.5 OBF2.1.7.4 OBF2.1.7.5 OBF2.1.7.6 OBF2.1.9.4 OBF2.1.9.5 OBF2.1.9.6



actorbase::driver::ActorbaseAdminServices	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
actorbase::driver::ActorbaseDriver	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3
actorbase::driver::ActorbaseServices	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.5 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3



actorbase::driver::Connector	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4
------------------------------	---



actorbase::driver::client::ActorbaseClient	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
--	---



actorbase::driver::client::Client	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
-----------------------------------	---



actorbase::driver::client::Connector	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
--------------------------------------	---



actorbase::driver::client::SSLClient	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
--------------------------------------	---



actorbase::driver::client::Serializer	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.1.3 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7 DEF3.7.1 DEF3.7.2 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
actorbase::driver::client::api::Credentials	DEF3.1 DEF3.2 DEF3.2.1 DEF3.2.2 DEF3.2.3 DEF3.2.4 DEF3.2.5 DEF3.2.6 DEF3.2.7 DEF3.3 DEF3.3.1 DEF3.3.2 DEF3.4 DEF3.5 DEF3.6 DEF3.7



actorbase::driver::client::api::DELETE	DEF3.2 DEF3.2.3 DEF3.2.3.1 DEF3.3 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2
actorbase::driver::client::api::GET	DEF3.2 DEF3.2.2 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.4 DEF3.5
actorbase::driver::client::api::POST	DEF3.1 DEF3.1.1 DEF3.1.2 DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.6 DEF3.6.1 DEF3.6.1.1 DEF3.6.2 DEF3.6.2.1 DEF3.6.3 DEF3.6.3.1 DEF3.7 DEF3.7.1 DEF3.7.2
actorbase::driver::client::api::PUT	DEF3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.7 DEF3.7.1 DEF3.7.2



actorbase::driver::client::api::Request	DEF3.1 DEF3.2 DEF3.2.1 DEF3.2.2 DEF3.2.3 DEF3.2.4 DEF3.2.5 DEF3.2.6 DEF3.2.7 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.2 DEF3.4 DEF3.5 DEF3.6 DEF3.7
actorbase::driver::client::api::RequestMethod	DEF3.1 DEF3.2 DEF3.2.1 DEF3.2.2 DEF3.2.3 DEF3.2.4 DEF3.2.5 DEF3.2.6 DEF3.2.7 DEF3.3 DEF3.3.1 DEF3.3.2 DEF3.4 DEF3.5 DEF3.6 DEF3.7
actorbase::driver::client::api::Response	DEF3.1.3 DEF3.2.1.2 DEF3.2.3.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.6.3 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2.3 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7.3 DEF3.7.4 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
actorbase::driver::client::api::Status	DEF3.1.3 DEF3.2.1.2 DEF3.2.3.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.6.3 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2.3 DEF3.6.4 DEF3.6.5 DEF3.6.6 DEF3.7.3 DEF3.7.4



actorbase::driver::data::ActorbaseCollection	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
actorbase::driver::data::ActorbaseObject	DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.8 DEF3.8.3



actorbase::driver::data::CollectionMap	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.2 DEF3.2.3 DEF3.2.3.1 DEF3.2.3.2 DEF3.2.4 DEF3.2.4.1 DEF3.2.4.2 DEF3.2.4.3 DEF3.2.4.4 DEF3.2.5 DEF3.2.5.1 DEF3.2.5.2 DEF3.2.5.3 DEF3.2.5.4 DEF3.2.5.5 DEF3.2.6 DEF3.2.6.1 DEF3.2.6.2 DEF3.2.6.3 DEF3.2.7 DEF3.2.7.1 DEF3.2.7.2 DEF3.2.7.3 DEF3.3 DEF3.3.1 DEF3.3.1.1 DEF3.3.1.1.1 DEF3.3.1.1.2 DEF3.3.1.1.3 DEF3.3.1.1.4 DEF3.3.1.2 DEF3.3.1.2.1 DEF3.3.1.2.2 DEF3.3.1.2.3 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.5 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
--	---

7 Tracciamento Requisiti-Classi

Requisito	Classi
DEF1.1.7	actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::utils::ActorbaseCollection
DEF1.1.8	actorbase::actorsystem::manager::Manager
DEF1.1.8.1	actorbase::actorsystem::manager::messages::DuplicationRequestSK
DEF2.1.2.7	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::ExportCommand actorbase::cli::views::CommandLoop actorbase::cli::controllers
DEF2.1.2.7.1	actorbase::cli::views::CommandLoop
DEF2.1.2.7.2	actorbase::cli::views::CommandLoop



DEF2.1.3.1.2	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::ImportCommand actorbase::cli::views::CommandLoop actorbase::cli::controllers
DEF2.1.3.1.2.1	actorbase::cli::views::CommandLoop
DEF2.1.3.2	actorbase::cli::models::CommandReceiver
DEF2.1.3.3	actorbase::cli::models::CommandReceiver
DEF2.1.4	actorbase::cli::controllers::GrammarParser actorbase::cli::models::CommandReceiver actorbase::cli::models::HelpCommand actorbase::cli::views::CommandLoop actorbase::cli::controllers
DEF2.1.5	actorbase::cli::controllers::GrammarParser actorbase::cli::models::CommandReceiver actorbase::cli::models::HelpCommand actorbase::cli::views::CommandLoop actorbase::cli::controllers
DEF2.1.5.1	actorbase::cli::views::CommandLoop
DEF3.1	actorbase::driver::ActorbaseDriver actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod
DEF3.1.1	actorbase::driver::ActorbaseDriver actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.1.2	actorbase::driver::ActorbaseDriver actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST



DEF3.1.3	actorbase::driver::ActorbaseDriver actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status
DEF3.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::DELETE actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::client::api::PUT actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.1.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.1.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.3	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.3.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.3.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::PUT actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.4.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::client::api::PUT actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.4.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::client::api::PUT actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.4.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.4.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.5	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.5.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.5.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.5.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.5.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.5.5	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.6	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.6.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.6.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.6.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.7	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.2.7.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.7.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.2.7.3	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::GET actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap
DEF3.3.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap
DEF3.3.1.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap



DEF3.3.1.1.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.1.1.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap
DEF3.3.1.1.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap
DEF3.3.1.1.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.3.1.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.1.2.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.1.2.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.1.2.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.3.1.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.2.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.3.2.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::DELETE actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.3.2.3	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.4	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::GET actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.4.1	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.4.2	actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap



DEF3.5	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::GET actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.6	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod
DEF3.6.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.6.1.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST



DEF3.6.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.6.2.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.6.3	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.6.3.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST
DEF3.6.4	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status



DEF3.6.5	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status
DEF3.6.6	actorbase::driver::ActorbaseAdminServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status
DEF3.7	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Credentials actorbase::driver::client::api::POST actorbase::driver::client::api::PUT actorbase::driver::client::api::Request actorbase::driver::client::api::RequestMethod
DEF3.7.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::client::api::PUT



DEF3.7.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::POST actorbase::driver::client::api::PUT
DEF3.7.3	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status
DEF3.7.4	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::Connector actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::client::api::Status
DEF3.8	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap



DEF3.8.1	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.8.2	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::data::ActorbaseCollection actorbase::driver::data::CollectionMap
DEF3.8.3	actorbase::driver::ActorbaseAdminServices actorbase::driver::ActorbaseServices actorbase::driver::client::ActorbaseClient actorbase::driver::client::Client actorbase::driver::client::Connector actorbase::driver::client::SSLClient actorbase::driver::client::Serializer actorbase::driver::client::api::Response actorbase::driver::data::ActorbaseCollection actorbase::driver::data::ActorbaseObject actorbase::driver::data::CollectionMap
OBF1.1	actorbase::actorsystem::httpserver::Httpserver actorbase::actorsystem::httpserver::SslConfiguration
OBF1.1.1	actorbase::actorsystem::httpserver::Httpserver actorbase::actorsystem::httpserver::SslConfiguration
OBF1.1.1.1	actorbase::actorsystem::httpserver::Httpserver
OBF1.1.10	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::httpserver::Httpserver
OBF1.1.10.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Authenticator actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::UserApi actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder



OBF1.1.10.1.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Authenticator actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::CollectionApi actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.2.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper
OBF1.1.10.2.1.1	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::clientactor
OBF1.1.10.2.2	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.3	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.2.3.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.3.2	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.4	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.2.4.1	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.5	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman



OBF1.1.10.2.5.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.2.5.2	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.2.5.3	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.5.4	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.2.6	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.2.6.1	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.2.7	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.3	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::ItemApi actorbase::actorsystem::main::Main actorbase::actorsystem::manager::Manager actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.3.1	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::manager::Manager actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman



OBF1.1.10.3.2	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::authactor
OBF1.1.10.3.3	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper
OBF1.1.10.3.4	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.10.3.5	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.4	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::ItemApi actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::storekeeper::Storekeeper
OBF1.1.10.5	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::UserApi actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.5.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.6	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::UserApi
OBF1.1.10.7	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::clientactor::UserApi actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.7.1	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.7.1.1	actorbase::actorsystem::clientactor::Clientactor
OBF1.1.10.7.2	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder



OBF1.1.10.7.2.1	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.7.3	actorbase::actorsystem::authactor::AuthActor actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.10.7.3.1	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::main::Main actorbase::actorsystem::storefinder::Storefinder
OBF1.1.2	actorbase::actorsystem::clientactor::Clientactor actorbase::actorsystem::utils::ActorbaseCollection
OBF1.1.2.1	actorbase::actorsystem::clientactor::messages::Response
OBF1.1.2.2	actorbase::actorsystem::clientactor::messages::MapResponse
OBF1.1.3	actorbase::actorsystem::main::Main actorbase::actorsystem::utils::ActorbaseCollection
OBF1.1.3.1	actorbase::actorsystem::main::messages::CreateCollection
OBF1.1.3.4	actorbase::actorsystem::main::messages::GetItemFrom
OBF1.1.3.5	actorbase::actorsystem::main::messages::AddContributor
OBF1.1.3.7	actorbase::actorsystem::main::messages::RemoveContributor
OBF1.1.3.8	actorbase::actorsystem::main::messages::CompleteTransaction
OBF1.1.3.9	actorbase::actorsystem::main::messages::RemoveItemFrom
OBF1.1.4	actorbase::actorsystem::storefinder::Storefinder actorbase::actorsystem::utils::ActorbaseCollection
OBF1.1.4.1	actorbase::actorsystem::storefinder::messages::UpdateCollectionSize
OBF1.1.4.3	actorbase::actorsystem::storefinder::messages::GetItem
OBF1.1.4.4	actorbase::actorsystem::storefinder::messages::RemoveItem
OBF1.1.4.5	actorbase::actorsystem::storefinder::messages::Insert
OBF1.1.4.6	actorbase::actorsystem::storefinder::messages::GetAllItems
OBF1.1.4.7	actorbase::actorsystem::storefinder::messages::PartialMapTransaction
OBF1.1.5	actorbase::actorsystem::storekeeper::Storekeeper actorbase::actorsystem::utils::ActorbaseCollection
OBF1.1.5.1	actorbase::actorsystem::storekeeper::messages::Persist
OBF1.1.5.2	actorbase::actorsystem::storekeeper::messages::GetItem
OBF1.1.5.3	actorbase::actorsystem::storekeeper::messages::GetAllItems
OBF1.1.5.4	actorbase::actorsystem::storekeeper::messages::Insert
OBF1.1.5.5	actorbase::actorsystem::storekeeper::messages::RemoveItem
OBF1.1.6	actorbase::actorsystem::utils::CryptoUtils actorbase::actorsystem::warehouseman::Warehouseman
OBF1.1.6.1	actorbase::actorsystem::warehouseman::messages::Read
OBF1.1.6.2	actorbase::actorsystem::warehouseman::messages::Save
OBF1.1.6.3	actorbase::actorsystem::warehouseman::messages::Clean
OBF1.1.9.1	actorbase::actorsystem::authactor::messages::AddCredentials



OBF1.1.9.2	actorbase::actorsystem::authactor::messages::UpdateCredentials
OBF1.1.9.3	actorbase::actorsystem::authactor::messages::RemoveCredentials
OBF1.1.9.4	actorbase::actorsystem::authactor::messages::Authenticate
OBF1.1.9.5	actorbase::actorsystem::authactor::messages::AddCollectionTo
OBF1.1.9.6	actorbase::actorsystem::authactor::messages::RemoveCollectionFrom
OBF1.1.9.7	actorbase::actorsystem::authactor::messages::ListCollectionsOf
OBF2	actorbase::cli::views::CommandLoop
OBF2.1	actorbase::cli::controllers::GrammarParser actorbase::cli::views::CommandLoop
OBF2.1.1	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::LoginCommand actorbase::cli::models::LogoutCommand actorbase::cli::views::CommandLoop
OBF2.1.1.1	actorbase::cli::views::CommandLoop
OBF2.1.1.2	actorbase::cli::views::CommandLoop
OBF2.1.1.3	actorbase::cli::views::CommandLoop
OBF2.1.2	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.2.1	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::CreateCollectionCommand actorbase::cli::views::CommandLoop
OBF2.1.2.1.1	actorbase::cli::views::CommandLoop
OBF2.1.2.10	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.2.11	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.2.12	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.2.13	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.2.14	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer



OBF2.1.2.2	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::ListCollectionCommand actorbase::cli::views::CommandLoop
OBF2.1.2.3	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::RemoveCollectionCommand actorbase::cli::views::CommandLoop
OBF2.1.2.3.1	actorbase::cli::views::CommandLoop
OBF2.1.2.4	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::RenameCollectionCommand actorbase::cli::views::CommandLoop
OBF2.1.2.4.1	actorbase::cli::views::CommandLoop
OBF2.1.2.4.2	actorbase::cli::views::CommandLoop
OBF2.1.2.5	actorbase::cli::controllers::GrammarParser actorbase::cli::models::AddContributorCommand actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.2.5.1	actorbase::cli::views::CommandLoop
OBF2.1.2.5.2	actorbase::cli::views::CommandLoop
OBF2.1.2.5.3	actorbase::cli::views::CommandLoop
OBF2.1.2.6	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::RemoveContributorCommand actorbase::cli::views::CommandLoop
OBF2.1.2.6.1	actorbase::cli::views::CommandLoop
OBF2.1.2.6.2	actorbase::cli::views::CommandLoop
OBF2.1.2.8	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.2.9	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.3	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop



OBF2.1.3.1	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::InsertItemCommand actorbase::cli::views::CommandLoop
OBF2.1.3.1.1	actorbase::cli::controllers::GrammarParser actorbase::cli::views::CommandLoop
OBF2.1.3.1.1.1	actorbase::cli::views::CommandLoop
OBF2.1.3.1.1.2	actorbase::cli::views::CommandLoop
OBF2.1.3.1.1.3	actorbase::cli::views::CommandLoop
OBF2.1.3.1.1.4	actorbase::cli::views::CommandLoop
OBF2.1.3.1.1.5	actorbase::cli::views::CommandLoop
OBF2.1.3.4	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.3.5	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.3.6	actorbase::cli::models::RemoveItemCommand actorbase::cli::views::CommandLoop
OBF2.1.3.6.1	actorbase::cli::views::CommandLoop
OBF2.1.3.6.2	actorbase::cli::views::CommandLoop
OBF2.1.6	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::FindCommand actorbase::cli::views::CommandLoop
OBF2.1.6.1	actorbase::cli::views::CommandLoop
OBF2.1.6.2	actorbase::cli::views::CommandLoop
OBF2.1.7	actorbase::cli::controllers::GrammarParser actorbase::cli::models::ChangePasswordCommand actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.7.1	actorbase::cli::views::CommandLoop
OBF2.1.7.2	actorbase::cli::views::CommandLoop
OBF2.1.7.3	actorbase::cli::views::CommandLoop
OBF2.1.7.4	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.7.5	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer



OBF2.1.7.6	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.8	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.9	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.9.1	actorbase::cli::controllers::GrammarParser actorbase::cli::models::AddUserCommand actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::views::CommandLoop
OBF2.1.9.1.1	actorbase::cli::views::CommandLoop
OBF2.1.9.1.2	actorbase::cli::views::CommandLoop
OBF2.1.9.2	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::RemoveUserCommand actorbase::cli::views::CommandLoop
OBF2.1.9.2.1	actorbase::cli::views::CommandLoop
OBF2.1.9.2.2	actorbase::cli::views::CommandLoop
OBF2.1.9.3	actorbase::cli::controllers::GrammarParser actorbase::cli::models::Command actorbase::cli::models::CommandReceiver actorbase::cli::models::ResetPasswordCommand actorbase::cli::views::CommandLoop
OBF2.1.9.3.1	actorbase::cli::views::CommandLoop
OBF2.1.9.3.2	actorbase::cli::views::CommandLoop
OBF2.1.9.4	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.9.5	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer
OBF2.1.9.6	actorbase::cli::models::CommandInvoker actorbase::cli::models::Observable actorbase::cli::views::ResultViewer

Elenco delle figure

1	Diagramma componenti principali: Server, Driver e CLI	5
2	Package _G actorsystem, visione generale	7
3	Package actorsystem::utils	8
4	Classe actorsystem::utils::Permissions	9
5	Classe actorsystem::utils::Read	9
6	Classe actorsystem::utils::ReadWrite	10
7	Classe actorsystem::utils::CryptoUtils	11
8	Classe actorsystem::utils::ActorbaseCollection	13
9	Package _G messages	16
10	Package actorsystem::messages::authactormessages	17
11	Classe AddCredentials	18
12	Classe UpdateCredentials	19
13	Classe RemoveCredentials	20
14	Classe Authenticate	21
15	Classe AddCollectionTo	22
16	Classe RemoveCollectionFrom	23
17	Classe ListCollectionsOf	24
18	Package actorsystem::messages::clientactormessages	25
19	Classe ListResponse	26
20	Classe MapResponse	27
21	Classe Response	28
22	Package actorsystem::messages::mainmessages	29
23	Classe InsertTo	30
24	Classe GetFrom	31
25	Classe CompleteTransaction	32
26	Classe RemoveFrom	33
27	Classe AddContributor	34
28	Classe RemoveContributor	35
29	Classe CreateCollection	36
30	Package actorsystem::messages::storefindermessages	38
31	Classe Storefinder	39
32	Classe UpdateCollectionSize	39
33	Classe Get	40
34	Classe PartialMapTransaction	41
35	Classe Remove	42
36	Classe Insert	43
37	Package actorsystem::messages::storekeepermessages	44
38	Classe Persist	44
39	Classe InitMn	45
40	Classe GetAll	46
41	Classe GetItem	47
42	Classe InsertItem	48
43	Classe RemoveItem	49



44	Package actorsystem::messages::warehousemanmessages	50
45	Classe Clean	50
46	Classe Init	51
47	Classe Save	52
48	Classe Read	53
49	Package actorsystem::messages::managermessages	54
50	Classe OneMore	54
51	Package _G actors, visione generale	55
52	Package _G Clientactor e interazioni con Main	56
53	Classe actorsystem::actors::clientactor::Authenticator	57
54	Classe actorsystem::actors::clientactor::User	58
55	Classe actorsystem::actors::clientactor::AuthInfo	59
56	Classe actorsystem::actors::clientactor::CollectionApi	60
57	Classe actorsystem::actors::clientactor::ClientActor	61
58	httpserver, visione generale del package	63
59	Classe actorsystem::actors::httpserver::SslConfiguration	64
60	Classe actorsystem::actors::httpserver::HTTPServer	65
61	Package actorsystem::actors::warehouseman	67
62	Classe actorsystem::actors::warehouseman::Warehouseman	68
63	Package dell'attore di tipo Main	69
64	Classe actorsystem::actors::main::Main	70
65	AuthActor, visione generale del package	72
66	Classe actorsystem::actors::authactor::AuthActor	73
67	Classe actorsystem::actors::authactor::AuthActor	74
68	Storefinder, visione generale del package	77
69	Classe actorsystem::actors::storefinder::Storefinder	78
70	Storekeeper, visione generale del package	80
71	Classe actorsystem::actors::storekeeper::Storekeeper	81
72	Manager, visione generale del package	83
73	Classe actorsystem::actors::manager::Manager	84
74	Driver: Architettura generale	85
75	Driver: Classe ActorbaseDriver	86
76	Driver: Classe ActorbaseServices	87
77	Driver: Classe ActorbaseAdminServices	91
78	Driver: Classe Connection	93
79	Driver: Client	94
80	Driver: Client interface	95
81	Driver: Classe serializer	96
82	Driver: Classe SSLClient	98
83	Driver: Classe ActorbaseClient	99
84	Driver: Classe astratta Connector	100
85	Driver: Package api	102
86	Driver: RequestMethod interface	103
87	Driver: Oggetto rappresentante richiesta get	103
88	Driver: Oggetto rappresentante richiesta post	104



89	Driver: Oggetto rappresentante richiesta Put	104
90	Driver: Oggetto rappresentante richiesta Delete	105
91	Driver: Classe request	106
92	Driver: Classe request	107
93	Driver: Classe response	108
94	Driver: Classe response	109
95	Driver: Classe RequestBuilder	110
96	Driver: Package data	112
97	Driver: Classe ActorbaseObject	113
98	Driver: Classe ActorbaseCollection	116
99	Driver: Classe ActorbaseCollectionMap	121
100	Driver: Package exceptions	123
101	Driver: class WrongCredentialExc	123
102	Driver: class WrongPasswordExc	124
103	Driver: class WrongNewPasswordExc	124
104	Driver: class CollectionAlreadyExistsExc	125
105	Driver: class UndefinedCollectionExc	125
106	Driver: class UndefinedUsernameExc	126
107	Driver: class UsernameAlreadyExistsExc	126
108	Driver: class DuplicateKeyExc	127
109	Driver: class UndefinedFileExc	127
110	Driver: class MalformedFileExc	128
111	CLI: architettura MVC variante push model	129
112	CLI: views package, interazioni con controllers e models	132
113	CLI: interfaccia PromptProvider	133
114	CLI: interfaccia Observer	134
115	CLI: classe CommandLoop	135
116	CLI: classe ActorbaseBanner	136
117	CLI: classe ActorbasePrompt	136
118	CLI: classe ResultViewer	137
119	CLI: package Controllers	139
120	CLI: package Controllers	140
121	CLI: package Models	143
122	CLI: interfaccia Command	144
123	CLI: classe astratta Observable	145
124	CLI: classe CommandInvoker	147
125	CLI: classe ChangePasswordCommand	148
126	CLI: classe ChangePasswordCommand	149
127	CLI: classe HelpCommand	150
128	CLI: classe LogoutCommand	151
129	CLI: classe ResetPasswordCommand	152
130	CLI: classe ListCollectionsCommand	153
131	CLI: classe AddContributorCommand	154
132	CLI: classe LoginCommand	155
133	CLI: classe RemoveCollectionCommand	156



134 CLI: classe RemoveContributorCommand	157
135 CLI: classe RenameCollectionCommand	158
136 CLI: classe AddUserCommand	159
137 CLI: classe RemoveItemCommand	160
138 CLI: classe RemoveUserCommand	161
139 CLI: classe ImportCommand	162
140 CLI: classe InsertItemCommand	163
141 CLI: classe CreateCollectionCommand	164
142 CLI: classe ExportCommand	165
143 CLI: classe CommandReceiver	166
144 Diagramma di sequenza - Interazioni generali tra componenti	169
145 Diagramma di sequenza - Inserimento di un item	170
146 Diagramma di sequenza - Inserimento di un item con Storekeeper pieno	171
147 Diagramma di sequenza - Autenticazione	172
148 Diagramma di sequenza - Inserimento di un item con salvataggio su filesystem	173
149 Diagramma di sequenza - aggiunta collaboratore	174