



## Specifica Tecnica

### Informazioni sul documento

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



## Diario delle modifiche

Versione	Autore	Ruolo	Data	Descrizione
2.0.0	Francesco Agostini	Responsabile	2016-05-16	Approvazione documento
1.4.0	Alberto De Agostini	Verificatore	2016-05-03	Verifica sezione 4.25 e relative sotto-sezioni
1.3.1	Marco Boseggia	Progettista	2016-05-02	Incremento sezione 4.25 e relative sotto-sezioni
1.3.0	Davide Trevisan	Verificatore	2016-04-30	Verifica sezione 4.24 e relative sotto-sezioni
1.2.1	Andrea Giacomo Baldan	Progettista	2016-04-29	Incremento sezione 4.24 e relative sotto-sezioni
1.2.0	Michael Munaro	Verificatore	2016-04-28	Verifica sezione 4.23 e relative sotto-sezioni
1.1.0	Davide Trevisan	Verificatore	2016-04-28	Verifica sezione 4.22 e sottosezione 4.22.4
1.0.2	Giacomo Vanin	Progettista	2016-04-28	Incremento sezione 4.23 e relative sotto-sezioni
1.0.1	Marco Boseggia	Progettista	2016-04-27	Incremento sezione Driver 4.22 e sottosezione 4.22.4
1.0.0	Marco Boseggia	Responsabile	2016-04-08	Approvazione documento
0.8.0	Andrea Giacomo Baldan	Verificatore	2016-04-03	Verifica appendice
0.7.1	Giacomo Vanin	Progettista	2016-04-03	Stesura appendice documento
0.7.0	Alberto De Agostini	Verificatore	2016-04-02	Verifica tracciamenti (sezione 9)
0.6.1	Francesco Agostini	Progettista	2016-04-02	Stesura sezione 9 comprensivo di tutto il tracciamento
0.6.0	Andrea Giacomo Baldan	Verificatore	2016-04-01	Verifica sezioni 7 e 8
0.5.2	Alberto De Agostini	Progettista	2016-03-31	Stesura sezione 8
0.5.1	Alberto De Agostini	Progettista	2016-03-30	Stesura sezione 7 e sottosezioni
0.5.0	Andrea Giacomo Baldan	Verificatore	2016-03-28	Verifica sezione 6 e sottosezioni
0.4.1	Davide Trevisan	Progettista	2016-03-27	Stesura sezione 6 e sottosezioni
0.4.0	Michael Munaro	Verificatore	2016-03-25	Verifica sezione 5
0.3.1	Giacomo Vanin	Progettista	2016-03-24	Stesura sezione 5
0.3.0	Alberto De Agostini	Verificatore	2016-03-22	Verifica sezioni da 4.6 a 4.13
0.2.2	Marco Boseggia	Progettista	2016-03-22	Stesura sezioni 4.10 4.11 4.12 4.13
0.2.1	Marco Boseggia	Progettista	2016-03-21	Stesura sezioni 4.6 4.7 4.8 4.9
0.2.0	Michael Munaro	Verificatore	2016-03-21	Verifica sottosezioni 3.1 3.2 4.1 4.2 4.3 4.4 4.5



---

0.1.2	Andrea Giacomo Baldan	Progettista	2016-03-17	Stesura sottosezioni 4.1 4.2 4.3 4.4 4.5
0.1.1	Andrea Giacomo Baldan	Progettista	2016-03-17	Stesura sottosezioni 3.1 3.2
0.1.0	Michael Munaro	Verificatore	2016-03-16	Verifica sezioni 1 e 2
0.0.2	Marco Boseggia	Progettista	2016-02-26	Stesura sezioni 1 e 2
0.0.1	Andrea Giacomo Baldan	Amministratore	2016-02-26	Creazione scheletro del documento

# Indice

<b>1 Sommario</b>	<b>1</b>
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
<b>2 Tecnologie utilizzate</b>	<b>3</b>
2.1 Akka . . . . .	3
2.2 Scala . . . . .	3
2.3 sbt . . . . .	4
<b>3 Descrizione architettura</b>	<b>5</b>
3.1 Metodo e formalismo di specifica . . . . .	5
3.2 Architettura generale . . . . .	5
3.3 Protocollo di comunicazione Client-Server . . . . .	5
<b>4 Componenti</b>	<b>6</b>
4.1 actorbase . . . . .	6
4.1.1 Descrizione . . . . .	6
4.1.2 Package contenuti . . . . .	7
4.2 actorbase::actorsystem . . . . .	7
4.2.1 Descrizione . . . . .	7
4.2.2 Package contenuti . . . . .	8
4.2.3 Interazioni con altre componenti . . . . .	8
4.3 actorbase::actorsystem::utils . . . . .	9
4.3.1 Descrizione . . . . .	10
4.3.2 Interazioni con altre componenti . . . . .	10
4.3.3 Classi . . . . .	10
4.3.3.1 KeyRange . . . . .	10
4.3.3.1.1 Descrizione . . . . .	10
4.3.3.1.2 Utilizzo . . . . .	10
4.3.3.1.3 Realizza . . . . .	10
4.3.3.1.4 Interazioni con altre classi . . . . .	10
4.3.3.2 CollectionRange . . . . .	11
4.3.3.2.1 Descrizione . . . . .	11
4.3.3.2.2 Utilizzo . . . . .	11
4.3.3.2.3 Realizza . . . . .	11
4.3.3.2.4 Interazioni con altre classi . . . . .	11
4.3.3.3 CryptoUtils . . . . .	11
4.3.3.3.1 Descrizione . . . . .	11
4.3.3.3.2 Utilizzo . . . . .	11
4.3.3.3.3 Interazioni con altre classi . . . . .	11



4.3.3.4	ActorbaseCollection . . . . .	11
4.3.3.4.1	Descrizione . . . . .	11
4.3.3.4.2	Utilizzo . . . . .	12
4.3.3.4.3	Realizza . . . . .	12
4.3.3.4.4	Interazioni con altre classi . . . . .	12
4.4	actorbase::actorsystem::clientactor . . . . .	14
4.4.1	Descrizione . . . . .	15
4.4.2	Package contenuti . . . . .	15
4.4.3	Interazioni con altre componenti . . . . .	15
4.4.4	Classi . . . . .	15
4.4.4.1	actorbase::actorsystem::clientactor::Authenticator . . . . .	15
4.4.4.1.1	Descrizione . . . . .	15
4.4.4.1.2	Utilizzo . . . . .	15
4.4.4.2	actorbase::actorsystem::clientactor::CollectionApi . . . . .	15
4.4.4.2.1	Descrizione . . . . .	15
4.4.4.2.2	Utilizzo . . . . .	15
4.4.4.2.3	Interazioni con altre classi . . . . .	16
4.4.4.3	actorbase::actorsystem::clientactor::ItemApi . . . . .	16
4.4.4.3.1	Descrizione . . . . .	16
4.4.4.3.2	Utilizzo . . . . .	16
4.4.4.3.3	Interazioni con altre classi . . . . .	16
4.4.4.4	actorbase::actorsystem::clientactor::UserApi . . . . .	16
4.4.4.4.1	Descrizione . . . . .	16
4.4.4.4.2	Utilizzo . . . . .	16
4.4.4.4.3	Interazioni con altre classi . . . . .	16
4.4.4.5	actorbase::actorsystem::clientactor::ClientActor . . . . .	16
4.4.4.5.1	Descrizione . . . . .	16
4.4.4.5.2	Utilizzo . . . . .	17
4.4.4.5.3	Classi ereditate . . . . .	17
4.4.4.5.4	Interazioni con altre classi . . . . .	17
4.5	actorbase::actorsystem::clientactor::messages . . . . .	18
4.5.1	Descrizione . . . . .	18
4.5.2	Classi . . . . .	18
4.5.2.1	actorbase::actorsystem::clientactor::messages::Response . . . . .	18
4.5.2.1.1	Descrizione . . . . .	18
4.5.2.1.2	Utilizzo . . . . .	18
4.5.2.1.3	Interazioni con altre classi . . . . .	19
4.5.2.2	actorbase::actorsystem::clientactor::messages::MapResponse . . . . .	19
4.5.2.2.1	Descrizione . . . . .	19
4.5.2.2.2	Utilizzo . . . . .	19
4.5.2.2.3	Interazioni con altre classi . . . . .	19
4.6	actorbase::actorsystem::httpserver . . . . .	20
4.6.1	Descrizione . . . . .	20
4.6.2	Package contenuti . . . . .	20
4.6.3	Interazioni con altre componenti . . . . .	20



4.6.4 Classi . . . . .	21
4.6.4.1 actorbase::actorsystem::httpserver::HTTPServer . . . . .	21
4.6.4.1.1 Descrizione . . . . .	21
4.6.4.1.2 Utilizzo . . . . .	21
4.6.4.1.3 Classi ereditate . . . . .	21
4.6.5 Interazioni con altre componenti . . . . .	21
4.6.5.1 actorbase::actorsystem::httpserver::SslConfiguration . . . . .	21
4.6.5.1.1 Descrizione . . . . .	21
4.6.5.1.2 Utilizzo . . . . .	21
4.7 actorbase::actorsystem::warehouseman . . . . .	22
4.7.1 Descrizione . . . . .	22
4.7.2 Package contenuti . . . . .	22
4.7.3 Interazioni con altre componenti . . . . .	23
4.7.4 Classi . . . . .	23
4.7.4.1 actorbase::actorsystem::warehouseman::Warehouseman . . . . .	23
4.7.4.1.1 Descrizione . . . . .	23
4.7.4.1.2 Utilizzo . . . . .	23
4.7.4.1.3 Classi ereditate . . . . .	23
4.7.4.1.4 Interazioni con altre classi . . . . .	23
4.8 actorbase::actorsystem::warehouseman::messages . . . . .	24
4.8.1 Descrizione . . . . .	24
4.8.2 Interazioni con altre componenti . . . . .	24
4.8.2.1 actorbase::actorsystem::warehouseman::messages::Save . . . . .	24
4.8.2.1.1 Descrizione . . . . .	24
4.8.2.1.2 Utilizzo . . . . .	25
4.8.2.1.3 Interazioni con altre classi . . . . .	25
4.8.2.2 actorbase::actorsystem::warehouseman::messages::Clean . . . . .	25
4.8.2.2.1 Descrizione . . . . .	25
4.8.2.2.2 Utilizzo . . . . .	25
4.8.2.2.3 Interazioni con altre classi . . . . .	25
4.8.2.3 actorbase::actorsystem::warehouseman::messages::RemoveSfFolder . . . . .	25
4.8.2.3.1 Descrizione . . . . .	25
4.8.2.3.2 Utilizzo . . . . .	25
4.8.2.3.3 Interazioni con altre classi . . . . .	25
4.8.2.4 actorbase::actorsystem::warehouseman::messages::Read . . . . .	26
4.8.2.4.1 Descrizione . . . . .	26
4.8.2.4.2 Utilizzo . . . . .	26
4.9 actorbase::actorsystem::main . . . . .	27
4.9.1 Descrizione . . . . .	27
4.9.2 Package contenuti . . . . .	27
4.9.3 Interazioni con altre componenti . . . . .	28
4.9.4 Classi . . . . .	28
4.9.4.1 actorbase::actorsystem::main::Main . . . . .	28
4.9.4.1.1 Descrizione . . . . .	28
4.9.4.1.2 Utilizzo . . . . .	28



4.9.4.1.3	Classi ereditate . . . . .	28
4.9.4.1.4	Interazioni con altre classi . . . . .	28
4.10	actorbase::actorsystem::main::messages . . . . .	29
4.10.1	Descrizione . . . . .	29
4.10.2	Interazioni con altre componenti . . . . .	30
4.10.3	Classi . . . . .	30
4.10.3.1	actorbase::actorsystem::main::messages::AddUser . . . . .	30
4.10.3.1.1	Descrizione . . . . .	30
4.10.3.1.2	Utilizzo . . . . .	30
4.10.3.2	actorbase::actorsystem::main::messages::Login . . . . .	30
4.10.3.2.1	Descrizione . . . . .	30
4.10.3.2.2	Utilizzo . . . . .	30
4.10.3.3	actorbase::actorsystem::main::messages::Insert . . . . .	30
4.10.3.3.1	Descrizione . . . . .	30
4.10.3.3.2	Utilizzo . . . . .	30
4.10.3.4	actorbase::actorsystem::main::messages::RemoveItemFrom . . . . .	31
4.10.3.4.1	Descrizione . . . . .	31
4.10.3.4.2	Utilizzo . . . . .	31
4.10.3.5	actorbase::actorsystem::main::messages::GetItemFrom . . . . .	31
4.10.3.5.1	Descrizione . . . . .	31
4.10.3.5.2	Utilizzo . . . . .	31
4.10.3.6	actorbase::actorsystem::main::messages::GetItemFromResponse . . . . .	31
4.10.3.6.1	Descrizione . . . . .	31
4.10.3.6.2	Utilizzo . . . . .	31
4.10.3.7	actorbase::actorsystem::main::messages::CreateCollection . . . . .	31
4.10.3.7.1	Descrizione . . . . .	31
4.10.3.7.2	Utilizzo . . . . .	32
4.10.3.8	actorbase::actorsystem::main::messages::RemoveCollection . . . . .	32
4.10.3.8.1	Descrizione . . . . .	32
4.10.3.8.2	Utilizzo . . . . .	32
4.10.3.9	actorbase::actorsystem::main::messages::AddContributor . . . . .	32
4.10.3.9.1	Descrizione . . . . .	32
4.10.3.9.2	Utilizzo . . . . .	32
4.10.3.10	actorbase::actorsystem::main::messages::RemoveContributor . . . . .	32
4.10.3.10.1	Descrizione . . . . .	32
4.10.3.10.2	Utilizzo . . . . .	32
4.10.3.11	actorbase::actorsystem::main::messages::Ack . . . . .	32
4.10.3.11.1	Descrizione . . . . .	32
4.10.3.11.2	Utilizzo . . . . .	33
4.10.3.12	actorbase::actorsystem::main::messages::DuplicationRequestSF . . . . .	33
4.10.3.12.1	Descrizione . . . . .	33
4.10.3.12.2	Utilizzo . . . . .	33
4.10.3.12.3	Interazioni con altre classi . . . . .	33
4.10.3.13	actorbase::actorsystem::main::messages::UpdateCollectionSize . . . . .	33
4.10.3.13.1	Descrizione . . . . .	33



4.10.3.13.2 Utilizzo . . . . .	33
4.10.3.13.3 Interazioni con altre classi . . . . .	33
4.11 actorbase::actorsystem::userfinder . . . . .	34
4.11.1 Descrizione . . . . .	34
4.11.2 Package contenuti . . . . .	34
4.11.3 Interazioni con altre componenti . . . . .	35
4.11.4 Classi . . . . .	35
4.11.4.1 actorbase::actorsystem::userfinder::Userfinder . . . . .	35
4.11.4.1.1 Descrizione . . . . .	35
4.11.4.1.2 Utilizzo . . . . .	35
4.11.4.1.3 Classi ereditate . . . . .	35
4.11.4.1.4 Interazioni con altre classi . . . . .	35
4.12 actorbase::actorsystem::userfinder::messages . . . . .	36
4.12.1 Descrizione . . . . .	36
4.12.2 Interazioni con altre componenti . . . . .	36
4.12.3 Classi . . . . .	37
4.12.3.1 actorbase::actorsystem::userfinder::messages::GetPasswordOf . . . . .	37
4.12.3.1.1 Descrizione . . . . .	37
4.12.3.1.2 Utilizzo . . . . .	37
4.12.3.2 actorbase::actorsystem::userfinder::messages::InsertTo . . . . .	37
4.12.3.2.1 Descrizione . . . . .	37
4.12.3.2.2 Utilizzo . . . . .	37
4.12.3.3 actorbase::actorsystem::userfinder::messages::GetCollectionOf . . . . .	37
4.12.3.3.1 Descrizione . . . . .	37
4.12.3.3.2 Utilizzo . . . . .	37
4.12.3.4 actorbase::actorsystem::userfinder::messages::ChangePasswordOf . . . . .	37
4.12.3.4.1 Descrizione . . . . .	37
4.12.3.4.2 Utilizzo . . . . .	38
4.12.3.5 actorbase::actorsystem::userfinder::messages::RemoveCollectionFrom . . . . .	38
4.12.3.5.1 Descrizione . . . . .	38
4.12.3.5.2 Utilizzo . . . . .	38
4.12.3.5.3 Interazioni con altre classi . . . . .	38
4.12.3.6 actorbase::actorsystem::userfinder::messages::AddCollectionTo . . . . .	38
4.12.3.6.1 Descrizione . . . . .	38
4.12.3.6.2 Utilizzo . . . . .	38
4.12.3.6.3 Interazioni con altre classi . . . . .	38
4.12.3.7 actorbase::actorsystem::userfinder::messages::UpdateCollectionSizeTo . . . . .	38
4.12.3.7.1 Descrizione . . . . .	38
4.12.3.7.2 Utilizzo . . . . .	39
4.12.3.7.3 Interazioni con altre classi . . . . .	39
4.13 actorbase::actorsystem::userkeeper . . . . .	39
4.13.1 Descrizione . . . . .	40
4.13.2 Package contenuti . . . . .	40
4.13.3 Interazioni con altre componenti . . . . .	40
4.13.4 Classi . . . . .	40



4.13.4.1 actorbase::actorsystem::userkeeper::Userkeeper . . . . .	40
4.13.4.1.1 Descrizione . . . . .	40
4.13.4.1.2 Utilizzo . . . . .	40
4.13.4.1.3 Classi ereditate . . . . .	40
4.13.4.1.4 Interazioni con altre classi . . . . .	40
4.14 actorbase::actorsystem::userkeeper::messages . . . . .	41
4.14.1 Descrizione . . . . .	41
4.14.2 Interazioni con altre componenti . . . . .	41
4.14.3 Classi . . . . .	41
4.14.3.1 actorbase::actorsystem::userkeeper::messages::GetCollections . . . . .	41
4.14.3.1.1 Descrizione . . . . .	41
4.14.3.1.2 Utilizzo . . . . .	42
4.14.3.2 actorbase::actorsystem::userkeeper::messages::ChangePassword . . . . .	42
4.14.3.2.1 Descrizione . . . . .	42
4.14.3.2.2 Utilizzo . . . . .	42
4.14.3.3 actorbase::actorsystem::userkeeper::messages::GetPassword . . . . .	42
4.14.3.3.1 Descrizione . . . . .	42
4.14.3.3.2 Utilizzo . . . . .	42
4.14.3.4 actorbase::actorsystem::userkeeper::messages::RemoveCollection . . . . .	42
4.14.3.4.1 Descrizione . . . . .	42
4.14.3.4.2 Utilizzo . . . . .	42
4.14.3.4.3 Interazioni con altre classi . . . . .	43
4.14.3.5 actorbase::actorsystem::userkeeper::messages::AddCollection . . . . .	43
4.14.3.5.1 Descrizione . . . . .	43
4.14.3.5.2 Utilizzo . . . . .	43
4.14.3.5.3 Interazioni con altre classi . . . . .	43
4.14.3.6 actorbase::actorsystem::userkeeper::messages::BindClient . . . . .	43
4.14.3.6.1 Descrizione . . . . .	43
4.14.3.6.2 Utilizzo . . . . .	43
4.14.3.7 actorbase::actorsystem::userkeeper::messages::UpdateCollectionSize . . . . .	43
4.14.3.7.1 Descrizione . . . . .	43
4.14.3.7.2 Utilizzo . . . . .	43
4.14.3.7.3 Interazioni con altre classi . . . . .	44
4.15 actorbase::actorsystem::storefinder . . . . .	44
4.15.1 Descrizione . . . . .	45
4.15.2 Package contenuti . . . . .	45
4.15.3 Interazioni con altre componenti . . . . .	45
4.15.4 Classi . . . . .	45
4.15.4.1 actorbase::actorsystem::storefinder::Storefinder . . . . .	45
4.15.4.1.1 Descrizione . . . . .	45
4.15.4.1.2 Utilizzo . . . . .	45
4.15.4.1.3 Classi ereditate . . . . .	45
4.15.4.1.4 Interazioni con altre classi . . . . .	45
4.16 actorbase::actorsystem::storefinder::messages . . . . .	46
4.16.1 Descrizione . . . . .	46



4.16.2	Interazioni con altre componenti	46
4.16.3	Classi	47
4.16.3.1	actorbase::actorsystem::storefinder::messages::DuplicationRequestSK	47
4.16.3.1.1	Descrizione	47
4.16.3.1.2	Utilizzo	47
4.16.3.1.3	Interazioni con altre classi	47
4.16.3.2	actorbase::actorsystem::storefinder::messages::Ack	47
4.16.3.2.1	Descrizione	47
4.16.3.2.2	Utilizzo	47
4.16.3.3	actorbase::actorsystem::storefinder::messages::GetItem	47
4.16.3.3.1	Descrizione	47
4.16.3.3.2	Utilizzo	48
4.16.3.4	actorbase::actorsystem::storefinder::messages::RemoveItem	48
4.16.3.4.1	Descrizione	48
4.16.3.4.2	Utilizzo	48
4.16.3.5	actorbase::actorsystem::storefinder::messages::Insert	48
4.16.3.5.1	Descrizione	48
4.16.3.5.2	Utilizzo	48
4.16.3.6	actorbase::actorsystem::storefinder::messages::GetAllItem	48
4.16.3.6.1	Descrizione	48
4.16.3.6.2	Utilizzo	48
4.16.3.7	actorbase::actorsystem::storefinder::messages::GetAllItemResponse	49
4.16.3.7.1	Descrizione	49
4.16.3.7.2	Utilizzo	49
4.16.3.8	actorbase::actorsystem::storefinder::messages::UpdateCollectionSize	49
4.16.3.8.1	Descrizione	49
4.16.3.8.2	Utilizzo	49
4.16.3.8.3	Interazioni con altre classi	49
4.17	actorbase::actorsystem::storekeeper	50
4.17.1	Descrizione	50
4.17.2	Package contenuti	51
4.17.3	Interazioni con altre componenti	51
4.17.4	Classi	51
4.17.4.1	actorbase::actorsystem::storekeeper::Storekeeper	51
4.17.4.1.1	Descrizione	51
4.17.4.1.2	Utilizzo	51
4.17.4.1.3	Classi ereditate	51
4.17.4.1.4	Interazioni con altre classi	51
4.18	actorbase::actorsystem::storekeeper::messages	52
4.18.1	Descrizione	52
4.18.2	Interazioni con altre componenti	52
4.18.3	Classi	53
4.18.3.1	actorbase::actorsystem::storekeeper::messages::GetItem	53
4.18.3.1.1	Descrizione	53
4.18.3.1.2	Utilizzo	53



4.18.3.2 actorbase::actorsystem::storekeeper::messages::GetAllItem . . . . .	53
4.18.3.2.1 Descrizione . . . . .	53
4.18.3.2.2 Utilizzo . . . . .	53
4.18.3.3 actorbase::actorsystem::storekeeper::messages::Insert . . . . .	53
4.18.3.3.1 Descrizione . . . . .	53
4.18.3.3.2 Utilizzo . . . . .	53
4.18.3.4 actorbase::actorsystem::storekeeper::messages::RemoveItem . . . . .	53
4.18.3.4.1 Descrizione . . . . .	53
4.18.3.4.2 Utilizzo . . . . .	54
4.18.3.5 actorbase::actorsystem::storekeeper::messages::UpdateOwnerOfSk . . . . .	54
4.18.3.5.1 Descrizione . . . . .	54
4.18.3.5.2 Utilizzo . . . . .	54
4.18.3.5.3 Interazioni con altre classi . . . . .	54
4.18.3.6 actorbase::actorsystem::storekeeper::messages::Update . . . . .	54
4.18.3.6.1 Descrizione . . . . .	54
4.18.3.6.2 Utilizzo . . . . .	54
4.18.3.7 actorbase::actorsystem::storekeeper::messages::BecomeSK . . . . .	54
4.18.3.7.1 Descrizione . . . . .	54
4.18.3.7.2 Utilizzo . . . . .	54
4.18.3.8 actorbase::actorsystem::storekeeper::messages::BecomeNinja . . . . .	55
4.18.3.8.1 Descrizione . . . . .	55
4.18.3.8.2 Utilizzo . . . . .	55
4.18.3.9 actorbase::actorsystem::storekeeper::messages::Persist . . . . .	55
4.18.3.9.1 Descrizione . . . . .	55
4.18.3.9.2 Utilizzo . . . . .	55
4.19 actorbase::actorsystem::manager . . . . .	56
4.19.1 Descrizione . . . . .	57
4.19.2 Package contenuti . . . . .	57
4.19.3 Interazioni con altre componenti . . . . .	57
4.19.4 Classi . . . . .	57
4.19.4.1 actorbase::actorsystem::manager::Manager . . . . .	57
4.19.4.1.1 Descrizione . . . . .	57
4.19.4.1.2 Utilizzo . . . . .	57
4.19.4.1.3 Classi ereditate . . . . .	57
4.19.4.1.4 Interazioni con altre classi . . . . .	57
4.20 actorbase::actorsystem::manager::messages . . . . .	58
4.20.0.1 Descrizione . . . . .	58
4.20.0.1 actorbase::actorsystem::manager::messages::DuplicationRequestSK . . . . .	58
4.20.0.1.1 Descrizione . . . . .	58
4.20.0.1.2 Utilizzo . . . . .	58
4.20.0.1.3 Interazioni con altre classi . . . . .	59
4.20.0.2 actorbase::actorsystem::manager::messages::DuplicationRequestSF . . . . .	59
4.20.0.2.1 Descrizione . . . . .	59
4.20.0.2.2 Utilizzo . . . . .	59
4.20.0.2.3 Interazioni con altre classi . . . . .	59



4.21 actorbase::driver . . . . .	60
4.21.1 Descrizione . . . . .	60
4.21.2 Interazioni con altre componenti . . . . .	60
4.21.3 Package contenuti . . . . .	60
4.21.4 Classi . . . . .	61
4.21.4.1 ActorbaseDriver . . . . .	61
4.21.4.1.1 Descrizione . . . . .	61
4.21.4.1.2 Utilizzo . . . . .	61
4.21.4.1.3 Eredita . . . . .	61
4.21.4.1.4 Interazioni con le altre classi . . . . .	61
4.21.4.2 ActorbaseServices . . . . .	61
4.21.4.2.1 Descrizione . . . . .	61
4.21.4.2.2 Utilizzo . . . . .	61
4.21.4.2.3 Eredita . . . . .	61
4.21.4.2.4 Interazioni con le altre classi . . . . .	62
4.21.4.3 ActorbaseAdminServices . . . . .	62
4.21.4.3.1 Descrizione . . . . .	62
4.21.4.3.2 Utilizzo . . . . .	62
4.21.4.3.3 Eredita . . . . .	62
4.21.4.3.4 Interazioni con le altre classi . . . . .	62
4.21.4.4 Connection . . . . .	62
4.21.4.4.1 Descrizione . . . . .	62
4.21.4.4.2 Utilizzo . . . . .	62
4.21.4.4.3 Interazioni con le altre classi . . . . .	62
4.22 actorbase::driver::client . . . . .	63
4.22.1 Descrizione . . . . .	63
4.22.2 Interazioni con altre componenti . . . . .	63
4.22.3 Classi . . . . .	64
4.22.3.1 Client . . . . .	64
4.22.3.1.1 Descrizione . . . . .	64
4.22.3.1.2 Utilizzo . . . . .	64
4.22.3.2 SSLClient (abstract) . . . . .	64
4.22.3.2.1 Descrizione . . . . .	64
4.22.3.2.2 Utilizzo . . . . .	64
4.22.3.2.3 Realizza . . . . .	64
4.22.3.3 actorbase::driver::client::ActorbaseClient . . . . .	64
4.22.3.3.1 Descrizione . . . . .	64
4.22.3.3.2 Utilizzo . . . . .	64
4.22.3.3.3 Realizza . . . . .	65
4.22.3.3.4 Eredita . . . . .	65
4.22.3.4 Serializer (abstract) . . . . .	65
4.22.3.4.1 Descrizione . . . . .	65
4.22.3.4.2 Utilizzo . . . . .	65
4.22.3.4.3 Ereditata da . . . . .	65
4.22.3.5 actorbase::driver::client::Connector (abstract) . . . . .	65



4.22.3.5.1 Descrizione . . . . .	65
4.22.3.5.2 Utilizzo . . . . .	65
4.22.3.5.3 Eredita . . . . .	65
4.23 actorbase::driver::client::api . . . . .	66
4.23.1 Descrizione . . . . .	66
4.23.2 Interazioni con altre componenti . . . . .	66
4.23.3 Classi . . . . .	67
4.23.3.1 actorbase::driver::client::api::RequestMethod (abstract) . . . . .	67
4.23.3.1.1 Descrizione . . . . .	67
4.23.3.1.2 Utilizzo . . . . .	67
4.23.3.1.3 Interazioni con le altre classi . . . . .	67
4.23.3.2 actorbase::driver::client::api::GET (abstract) . . . . .	67
4.23.3.2.1 Descrizione . . . . .	67
4.23.3.2.2 Utilizzo . . . . .	67
4.23.3.2.3 Interazioni con le altre classi . . . . .	67
4.23.3.2.4 Eredita . . . . .	67
4.23.3.3 actorbase::driver::client::api::POST (abstract) . . . . .	68
4.23.3.3.1 Descrizione . . . . .	68
4.23.3.3.2 Utilizzo . . . . .	68
4.23.3.3.3 Interazioni con le altre classi . . . . .	68
4.23.3.3.4 Eredita . . . . .	68
4.23.3.4 actorbase::driver::client::api::PUT (abstract) . . . . .	68
4.23.3.4.1 Descrizione . . . . .	68
4.23.3.4.2 Utilizzo . . . . .	68
4.23.3.4.3 Interazioni con le altre classi . . . . .	68
4.23.3.4.4 Eredita . . . . .	68
4.23.3.5 actorbase::driver::client::api::DELETE (abstract) . . . . .	69
4.23.3.5.1 Descrizione . . . . .	69
4.23.3.5.2 Utilizzo . . . . .	69
4.23.3.5.3 Interazioni con le altre classi . . . . .	69
4.23.3.5.4 Eredita . . . . .	69
4.23.3.6 actorbase::driver::client::api::Request . . . . .	69
4.23.3.6.1 Descrizione . . . . .	69
4.23.3.6.2 Utilizzo . . . . .	69
4.23.3.6.3 Interazioni con le altre classi . . . . .	69
4.23.3.7 actorbase::driver::client::api::Response . . . . .	70
4.23.3.7.1 Descrizione . . . . .	70
4.23.3.7.2 Utilizzo . . . . .	70
4.23.3.7.3 Interazioni con le altre classi . . . . .	70
4.23.3.8 Status . . . . .	70
4.23.3.8.1 Descrizione . . . . .	70
4.23.3.8.2 Utilizzo . . . . .	70
4.23.3.8.3 Interazioni con le altre classi . . . . .	70
4.23.3.9 RequestBuilder . . . . .	70
4.23.3.9.1 Descrizione . . . . .	70



4.23.3.9.2 Utilizzo . . . . .	71
4.23.3.9.3 Interazioni con le altre classi . . . . .	71
4.24 actorbase::driver::client::api::data . . . . .	71
4.24.1 Descrizione . . . . .	71
4.24.2 Interazioni con altre componenti . . . . .	72
4.24.3 Classi . . . . .	72
4.24.3.1 ActorbaseObject . . . . .	72
4.24.3.1.1 Descrizione . . . . .	72
4.24.3.1.2 Utilizzo . . . . .	72
4.24.3.1.3 Interazioni con le altre classi . . . . .	72
4.24.3.2 actorbase::driver::data::ActorbaseCollection . . . . .	72
4.24.3.2.1 Descrizione . . . . .	72
4.24.3.2.2 Utilizzo . . . . .	72
4.24.3.2.3 Eredita . . . . .	72
4.24.3.2.4 Interazioni con le altre classi . . . . .	72
4.24.3.3 actorbase::driver::data::ActorbaseCollectionMap . . . . .	73
4.24.3.3.1 Descrizione . . . . .	73
4.24.3.3.2 Utilizzo . . . . .	73
4.24.3.3.3 Eredita . . . . .	73
4.24.3.3.4 Interazioni con le altre classi . . . . .	73
4.25 actorbase::driver::exceptions . . . . .	73
4.25.1 Descrizione . . . . .	73
4.25.2 Interazione con altre componenti . . . . .	74
4.25.3 Classi . . . . .	74
4.25.3.1 actorbase::driver::exceptions::WrongCredentialExc . . . . .	74
4.25.3.1.1 Descrizione . . . . .	74
4.25.3.1.2 Utilizzo . . . . .	74
4.25.3.1.3 Interazioni con altre classi . . . . .	74
4.25.3.2 actorbase::driver::exceptions::WrongPasswordExc . . . . .	74
4.25.3.2.1 Descrizione . . . . .	74
4.25.3.2.2 Utilizzo . . . . .	74
4.25.3.2.3 Interazioni con altre classi . . . . .	74
4.25.3.3 actorbase::driver::exceptions::WrongNewPasswordExc . . . . .	74
4.25.3.3.1 Descrizione . . . . .	74
4.25.3.3.2 Utilizzo . . . . .	75
4.25.3.3.3 Interazioni con altre classi . . . . .	75
4.25.3.4 actorbase::driver::exceptions::CollectionAlreadyExistsExc . . . . .	75
4.25.3.4.1 Descrizione . . . . .	75
4.25.3.4.2 Utilizzo . . . . .	75
4.25.3.4.3 Interazioni con altre classi . . . . .	75
4.25.3.5 actorbase::driver::exceptions::UndefinedCollectionExc . . . . .	75
4.25.3.5.1 Descrizione . . . . .	75
4.25.3.5.2 Utilizzo . . . . .	75
4.25.3.5.3 Interazioni con altre classi . . . . .	75
4.25.3.6 actorbase::driver::exceptions::UndefinedUsernameExc . . . . .	76



4.25.3.6.1 Descrizione . . . . .	76
4.25.3.6.2 Utilizzo . . . . .	76
4.25.3.6.3 Interazioni con altre classi . . . . .	76
4.25.3.7 actorbase::driver::exceptions::UsernameAlreadyExistsExc . . . . .	76
4.25.3.7.1 Descrizione . . . . .	76
4.25.3.7.2 Utilizzo . . . . .	76
4.25.3.7.3 Interazioni con altre classi . . . . .	76
4.25.3.8 actorbase::driver::exceptions::DuplicateKeyExc . . . . .	76
4.25.3.8.1 Descrizione . . . . .	76
4.25.3.8.2 Utilizzo . . . . .	76
4.25.3.8.3 Interazioni con altre classi . . . . .	77
4.25.3.9 actorbase::driver::exceptions::UndefinedFileExc . . . . .	77
4.25.3.9.1 Descrizione . . . . .	77
4.25.3.9.2 Utilizzo . . . . .	77
4.25.3.9.3 Interazioni con altre classi . . . . .	77
4.25.3.10 actorbase::driver::exceptions::MalformedFileExc . . . . .	77
4.25.3.10.1 Descrizione . . . . .	77
4.25.3.10.2 Utilizzo . . . . .	77
4.25.3.10.3 Interazioni con altre classi . . . . .	77
4.26 actorbase::cli . . . . .	78
4.26.1 Descrizione . . . . .	78
4.26.2 Interazioni con altre componenti . . . . .	79
4.26.3 Package contenuti . . . . .	79
4.27 actorbase::cli::views . . . . .	80
4.27.1 Descrizione . . . . .	81
4.27.2 Interazioni con altre componenti . . . . .	81
4.27.3 Classi . . . . .	81
4.27.3.1 actorbase::cli::views::CommandLoop . . . . .	81
4.27.3.1.1 Descrizione . . . . .	81
4.27.3.1.2 Utilizzo . . . . .	81
4.27.3.1.3 Interazioni con altre classi . . . . .	81
4.27.3.2 actorbase::cli::views::ActorbaseBanner . . . . .	81
4.27.3.2.1 Descrizione . . . . .	81
4.27.3.2.2 Utilizzo . . . . .	81
4.27.3.2.3 Interazioni con altre classi . . . . .	82
4.27.3.3 actorbase::cli::views::PromptProvider . . . . .	82
4.27.3.3.1 Descrizione . . . . .	82
4.27.3.3.2 Utilizzo . . . . .	82
4.27.3.3.3 Interazioni con altre classi . . . . .	82
4.27.3.4 actorbase::cli::views::ActorbasePrompt . . . . .	82
4.27.3.4.1 Descrizione . . . . .	82
4.27.3.4.2 Utilizzo . . . . .	82
4.27.3.4.3 Classi ereditate . . . . .	82
4.27.3.5 actorbase::cli::views::Observer . . . . .	82
4.27.3.5.1 Descrizione . . . . .	82



4.27.3.5.2 Utilizzo . . . . .	83
4.27.3.5.3 Interazioni con altre classi . . . . .	83
4.27.3.6 actorbase::cli::views::ResultView . . . . .	83
4.27.3.6.1 Descrizione . . . . .	83
4.27.3.6.2 Utilizzo . . . . .	83
4.27.3.6.3 Classi ereditate . . . . .	83
4.27.3.6.4 Interazioni con altre classi . . . . .	83
4.28 actorbase::cli::controllers . . . . .	84
4.28.1 Descrizione . . . . .	84
4.28.2 Interazioni con altre componenti . . . . .	85
4.28.3 Classi . . . . .	85
4.28.3.1 actorbase::cli::controllers::GrammarParser . . . . .	85
4.28.3.1.1 Descrizione . . . . .	85
4.28.3.1.2 Utilizzo . . . . .	85
4.28.3.1.3 Interazioni con altre classi . . . . .	85
4.29 actorbase::cli::models . . . . .	86
4.29.1 Descrizione . . . . .	87
4.29.2 Classi . . . . .	87
4.29.2.1 actorbase::cli::models::Observable . . . . .	87
4.29.2.1.1 Descrizione . . . . .	87
4.29.2.1.2 Utilizzo . . . . .	87
4.29.2.1.3 Interazioni con altre classi . . . . .	87
4.29.2.2 actorbase::cli::models::CommandInvoker . . . . .	87
4.29.2.2.1 Descrizione . . . . .	87
4.29.2.2.2 Utilizzo . . . . .	87
4.29.2.2.3 Classi ereditate . . . . .	87
4.29.2.2.4 Interazioni con altre classi . . . . .	88
4.29.2.3 actorbase::cli::models::Command . . . . .	88
4.29.2.3.1 Descrizione . . . . .	88
4.29.2.3.2 Utilizzo . . . . .	88
4.29.2.3.3 Interazioni con altre classi . . . . .	88
4.29.2.4 actorbase::cli::models::FindCommand . . . . .	88
4.29.2.4.1 Descrizione . . . . .	88
4.29.2.4.2 Utilizzo . . . . .	88
4.29.2.4.3 Classi ereditate . . . . .	88
4.29.2.5 actorbase::cli::models::LogoutCommand . . . . .	88
4.29.2.5.1 Descrizione . . . . .	88
4.29.2.5.2 Utilizzo . . . . .	89
4.29.2.5.3 Classi ereditate . . . . .	89
4.29.2.6 actorbase::cli::models::ResetPasswordCommand . . . . .	89
4.29.2.6.1 Descrizione . . . . .	89
4.29.2.6.2 Utilizzo . . . . .	89
4.29.2.6.3 Classi ereditate . . . . .	89
4.29.2.7 actorbase::cli::models::ChangePasswordCommand . . . . .	89
4.29.2.7.1 Descrizione . . . . .	89



4.29.2.7.2 Utilizzo . . . . .	89
4.29.2.7.3 Classi ereditate . . . . .	89
4.29.2.8 actorbase::cli::models::ListCommand . . . . .	90
4.29.2.8.1 Descrizione . . . . .	90
4.29.2.8.2 Utilizzo . . . . .	90
4.29.2.8.3 Classi ereditate . . . . .	90
4.29.2.9 actorbase::cli::models::AddContributorCommand . . . . .	90
4.29.2.9.1 Descrizione . . . . .	90
4.29.2.9.2 Utilizzo . . . . .	90
4.29.2.9.3 Classi ereditate . . . . .	90
4.29.2.10 actorbase::cli::models::LoginCommand . . . . .	90
4.29.2.10.1 Descrizione . . . . .	90
4.29.2.10.2 Utilizzo . . . . .	90
4.29.2.10.3 Classi ereditate . . . . .	91
4.29.2.11 actorbase::cli::models::RemoveCollectionCommand . . . . .	91
4.29.2.11.1 Descrizione . . . . .	91
4.29.2.11.2 Utilizzo . . . . .	91
4.29.2.11.3 Classi ereditate . . . . .	91
4.29.2.12 actorbase::cli::models::RemoveContributorCommand . . . . .	91
4.29.2.12.1 Descrizione . . . . .	91
4.29.2.12.2 Utilizzo . . . . .	91
4.29.2.12.3 Classi ereditate . . . . .	91
4.29.2.13 actorbase::cli::models::RenameCollectionCommand . . . . .	91
4.29.2.13.1 Descrizione . . . . .	91
4.29.2.13.2 Utilizzo . . . . .	92
4.29.2.13.3 Classi ereditate . . . . .	92
4.29.2.14 actorbase::cli::models::AddUserCommand . . . . .	92
4.29.2.14.1 Descrizione . . . . .	92
4.29.2.14.2 Utilizzo . . . . .	92
4.29.2.14.3 Classi ereditate . . . . .	92
4.29.2.15 actorbase::cli::models::HelpCommand . . . . .	92
4.29.2.15.1 Descrizione . . . . .	92
4.29.2.15.2 Utilizzo . . . . .	92
4.29.2.16 actorbase::cli::models::RemoveItemCommand . . . . .	92
4.29.2.16.1 Descrizione . . . . .	92
4.29.2.16.2 Utilizzo . . . . .	93
4.29.2.16.3 Classi ereditate . . . . .	93
4.29.2.17 actorbase::cli::models::RemoveUserCommand . . . . .	93
4.29.2.17.1 Descrizione . . . . .	93
4.29.2.17.2 Utilizzo . . . . .	93
4.29.2.17.3 Classi ereditate . . . . .	93
4.29.2.18 actorbase::cli::models::ImportCommand . . . . .	93
4.29.2.18.1 Descrizione . . . . .	93
4.29.2.18.2 Utilizzo . . . . .	93
4.29.2.18.3 Classi ereditate . . . . .	93



4.29.2.19 actorbase::cli::models::InsertItemCommand . . . . .	94
4.29.2.19.1 Descrizione . . . . .	94
4.29.2.19.2 Utilizzo . . . . .	94
4.29.2.19.3 Classi ereditate . . . . .	94
4.29.2.20 actorbase::cli::models::CreateCollectionCommand . . . . .	94
4.29.2.20.1 Descrizione . . . . .	94
4.29.2.20.2 Utilizzo . . . . .	94
4.29.2.20.3 Classi ereditate . . . . .	94
4.29.2.21 actorbase::cli::models::ExportCommand . . . . .	94
4.29.2.21.1 Descrizione . . . . .	94
4.29.2.21.2 Utilizzo . . . . .	94
4.29.2.21.3 Classi ereditate . . . . .	95
4.29.2.22 actorbase::cli::models::CommandReceiver . . . . .	95
4.29.2.22.1 Descrizione . . . . .	95
4.29.2.22.2 Utilizzo . . . . .	95
4.29.2.22.3 Interazioni con altre classi . . . . .	95
<b>5 Diagrammi di Attività</b>	<b>96</b>
5.1 Visione generale . . . . .	96
5.2 Operazioni su collezioni e/o item . . . . .	98
5.2.1 Creazione collezione . . . . .	98
5.2.2 Cancellazione collezione . . . . .	98
5.2.3 Visualizza collezioni . . . . .	99
5.2.4 Modifica nome collezione . . . . .	100
5.2.5 Inserimento item . . . . .	101
5.2.6 Rimozione item . . . . .	102
5.2.7 Aggiunta collaboratore . . . . .	103
5.2.8 Rimozione collaboratore . . . . .	104
5.2.9 Import . . . . .	105
5.3 Interrogazione del database . . . . .	106
5.4 Modifica password . . . . .	107
5.5 Gestione utenti . . . . .	108
<b>6 Design Pattern</b>	<b>110</b>
6.1 Design Pattern Architetturali . . . . .	110
6.1.1 MVC . . . . .	110
6.1.2 Dependance Injection . . . . .	110
6.2 Design Pattern Creazionali . . . . .	111
6.2.1 Singleton . . . . .	111
6.2.2 Builder . . . . .	111
6.3 Design Pattern Strutturali . . . . .	111
6.3.1 Decorator . . . . .	111
6.4 Design Pattern Comportamentali . . . . .	112
6.4.1 Command Pattern . . . . .	112
6.4.2 Observer Pattern . . . . .	112



<b>7 Stime di fattibilità e di bisogno di risorse</b>	<b>114</b>
<b>8 Tracciamento</b>	<b>115</b>
8.1 Tracciamento Componenti-Requisiti . . . . .	115
8.2 Tracciamento Requisiti-Componenti . . . . .	124
<b>A Descrizione Design Pattern</b>	<b>139</b>
A.1 Design Pattern Architetturali . . . . .	139
A.1.1 MVC . . . . .	139
A.2 Design Pattern Creazionali . . . . .	140
A.2.1 Singleton . . . . .	140
A.2.2 Builder . . . . .	140
A.3 Design Pattern Strutturali . . . . .	141
A.3.1 Decorator . . . . .	141
A.4 Design Pattern Comportamentali . . . . .	142
A.4.1 Command . . . . .	142
A.4.2 Observer . . . . .	143



## 1 Sommario

### 1.1 Scopo del documento

Il seguente documento ha lo scopo di descrivere la progettazione ad alto livello che il gruppo *ScalateKids* ha scelto per il progetto **Actorbase**.

Sarà descritta l'architettura<sub>G</sub> generale del progetto, in particolare la scelta dei design pattern<sub>G</sub> e delle componenti che andranno a comporre il software.

### 1.2 Scopo del Prodotto

Implementazione di un database NoSQL<sub>G</sub> di tipo key-value<sub>G</sub> orientato alla gestione di grandi moli di dati utilizzando il modello ad attori<sub>G</sub> su JVM<sub>G</sub>, comprensivo di un *Domain Specific Language* (DSL<sub>G</sub>) 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 v3.0.0](#); 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 v3.0.0](#).

#### 1.4.2 Informativi

- **Piano di Progetto:** [Piano di Progetto v3.0.0](#)
- **Dispense fornite dall'insegnamento Ingegneria del Software mod.A:**  
<http://www.math.unipd.it/~tullio/IS-1/2015/>
- **Documentazione di Akka<sub>G</sub> su Scala<sub>G</sub>:** <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<sub>G</sub> (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<sub>G</sub>(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<sub>G</sub> distribuiti, rappresenta un sistema di localizzazione e richiesta risorse stateless<sub>G</sub>, poggia su HTTP<sub>G</sub> come protocollo di comunicazione [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)



## 2 Tecnologie utilizzate

In questa sezione saranno descritte le tecnologie scelte per lo sviluppo del progetto **Actorbase** e le motivazioni che ci hanno spinto a sceglierle.

### 2.1 Akka

La scelta della libreria Akka<sub>G</sub> è stata dettata dal capitolato, tuttavia l'avremmo scelta comunque per i seguenti motivi:

- **Implementazione del modello ad attori<sub>G</sub>:** Akka<sub>G</sub> rende semplice la creazione e l'utilizzo degli attori<sub>G</sub>. Grazie a ciò è possibile creare facilmente architetture ad eventi estremamente concorrenti e viene naturale la distribuibilità del progetto e l'asincronia tra i processi;
- **Tolleranza agli errori:** Akka<sub>G</sub> implementa un sistema di gerarchia di supervisori. Questa gerarchia consente ai supervisori di un attore<sub>G</sub>, nel caso in cui quest'ultimo lanci un'eccezione, di mandarlo in crash<sub>G</sub> e farlo poi ripartire. In questo modo il sistema è resiliente agli errori;
- **Persistenza:** Ogni messaggio ricevuto da ciascun attore<sub>G</sub> può essere salvato in maniera tale che al riavvio esso possa riprendere dallo stato precedente al suo spegnimento.

### 2.2 Scala

Il capitolato richiede l'utilizzo di un linguaggio di programmazione a scelta tra Scala<sub>G</sub> e Java<sub>G</sub>. Abbiamo scelto Scala<sub>G</sub> per i seguenti motivi:

- **Programmazione funzionale<sub>G</sub>:** Questa tecnica di programmazione evita che ci siano degli effetti collaterali tra funzioni, rendendole quindi thread-safe;
- **Implementazione di Akka:** Akka<sub>G</sub> risulta più facilmente implementabile in Scala<sub>G</sub> che in Java<sub>G</sub>. Questo perché la programmazione funzionale<sub>G</sub> con i suoi paradigmi si integra meglio con il modello ad attori<sub>G</sub>;
- **Implementazione di DSL:** Scala si presta alla creazione di DSL<sub>G</sub> in maniera nativa, sia di tipo esterno che di tipo interno, senza necessariamente utilizzare librerie esterne o appoggiarsi a framework appositi;
- **Integrazione con librerie Java:** In Scala è possibile usufruire di molte librerie sviluppate in ambiente Java in quanto utilizza lo stesso modello di compilazione generando dei file<sub>G</sub> in bytecode<sub>G</sub>.

Tuttavia presenta alcuni svantaggi:

- **Curva di apprendimento:** Scala introduce paradigmi di programmazione funzionale applicati al paradigma ad oggetti, inoltre ha una sintassi molto espressiva e questo può rallentare significativamente il processo di apprendimento per i neofiti;
- **Librerie Java:** La possibilità di usufruire di molte librerie Java, pur essendo considerato positivo, ha anche il lato negativo di doversi appoggiare a documentazioni specifiche per linguaggio Java, e questo può portare a difficili comprensioni del comportamento della libreria in Scala, specie nel caso in cui la compatibilità non sia al 100%.



## 2.3 sbt

Sbt (Simple Build Tool) è un ambiente di build<sub>G</sub> simile alla controparte Java Maven<sub>G</sub> o Ant<sub>G</sub>. Fornito di console<sub>G</sub> e strumenti di compilazione ed esecuzione per progetti Scala, è stato scelto per i seguenti vantaggi:

- Semplice configurazione mediante file;
- REPL<sub>G</sub> Scala integrata;
- Facilmente integrabile all'interno di IDE<sub>G</sub> e editor;
- Gestione semplificata delle dipendenze;
- Gestione semplificata del processo di build<sub>G</sub>.



## 3 Descrizione architettura

### 3.1 Metodo e formalismo di specifica

L'esposizione dell'architettura segue la metodologia top-down<sub>G</sub>, descrivendo l'architettura iniziando dalle componenti generali e scendendo nel dettaglio. Seguirà quindi la descrizione dei package<sub>G</sub> seguiti dalla descrizione in dettaglio delle classi e interfacce che vi appartengono, specificando per ognuna la funzione e le relazioni in ingresso ed uscita. Il formalismo cromatico utilizzato sarà conforme a quanto specificato in *Norme Di Progetto*. Successivamente si illustreranno i Design Pattern<sub>G</sub> e la loro applicazione all'interno delle componenti, con un approfondimento del loro funzionamento in [appendice A](#).

Nel trattare le componenti, si chiarisce che sono da intendersi come package<sub>G</sub> e i due termini verranno quindi usati come sinonimi.

### 3.2 Architettura generale

Macroscopicamente il sistema si suddivide in due componenti principali, assimilabili ad un paradigma Client<sub>G</sub>-Server<sub>G</sub>, il sistema ad attori dove risiedono la struttura della base di dati ed un server<sub>G</sub> HTTP (Hyper Text Transport Protocol) pronto a ricevere comandi dall'esterno rappresentano la componente server<sub>G</sub>, la Command Line Interface (CLI<sub>G</sub>) rappresenta la componente client<sub>G</sub> mediante la quale è possibile inviare comandi al sistema e ricevere l'output prodotto da essi. Essa farà uso della componente driver<sub>G</sub> per comunicare con il sistema lato server<sub>G</sub> mediante il protocollo di comunicazione REST.

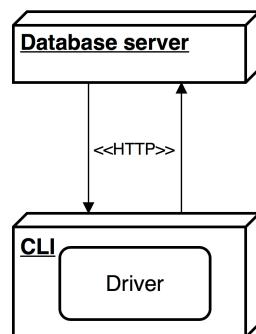


Figura 1: Diagramma architettura concettuale

### 3.3 Protocollo di comunicazione Client-Server

**Actorbase** è stato concepito come sistema RESTful<sub>G</sub> con possibilità di utilizzo come web-service<sub>G</sub>. La connessione è quindi di tipo Stateless, in quanto ogni richiesta non ha nulla a che fare con lo storico di richieste precedenti.



## 4 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**.

### 4.1 actorbase

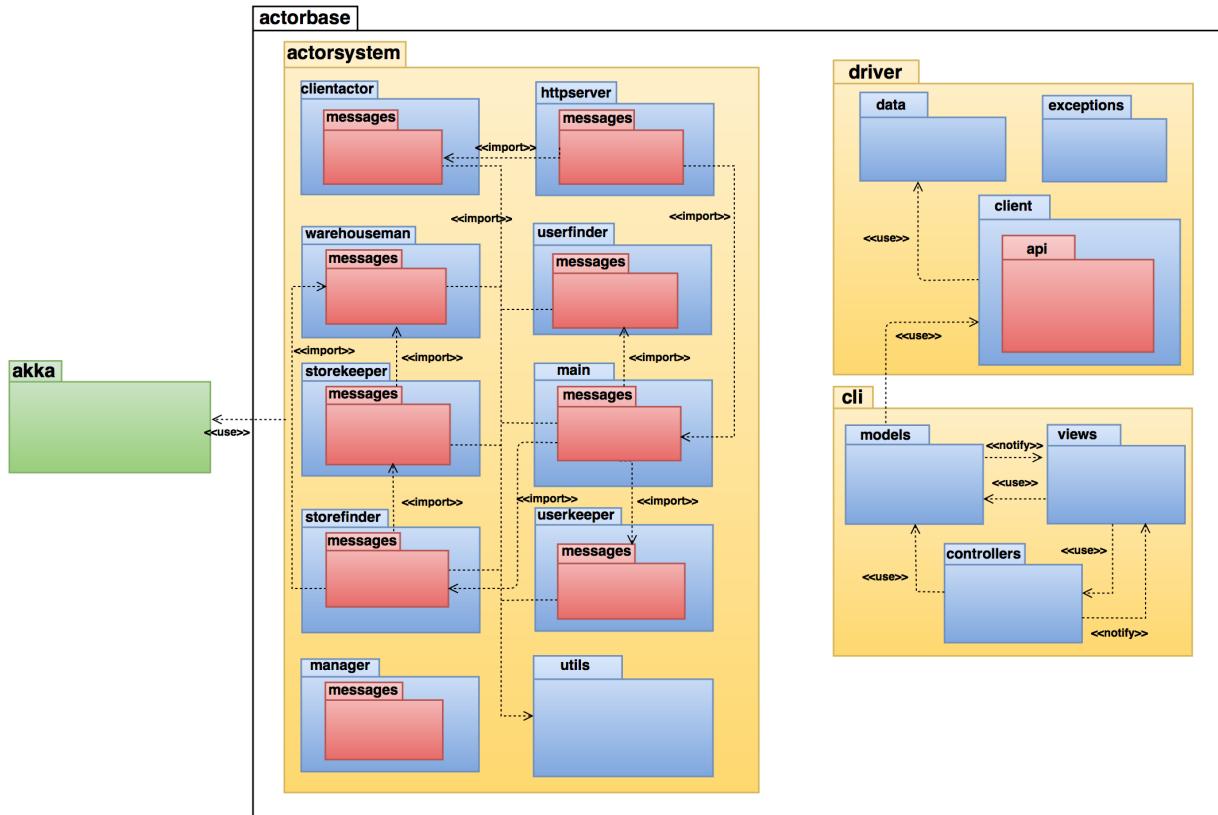


Figura 2: Diagramma componenti principali

#### 4.1.1 Descrizione

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

#### **4.1.2 Package contenuti**

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

Rappresentano le tre macro-componenti del sistema.

## 4.2 actorbase::actorsystem

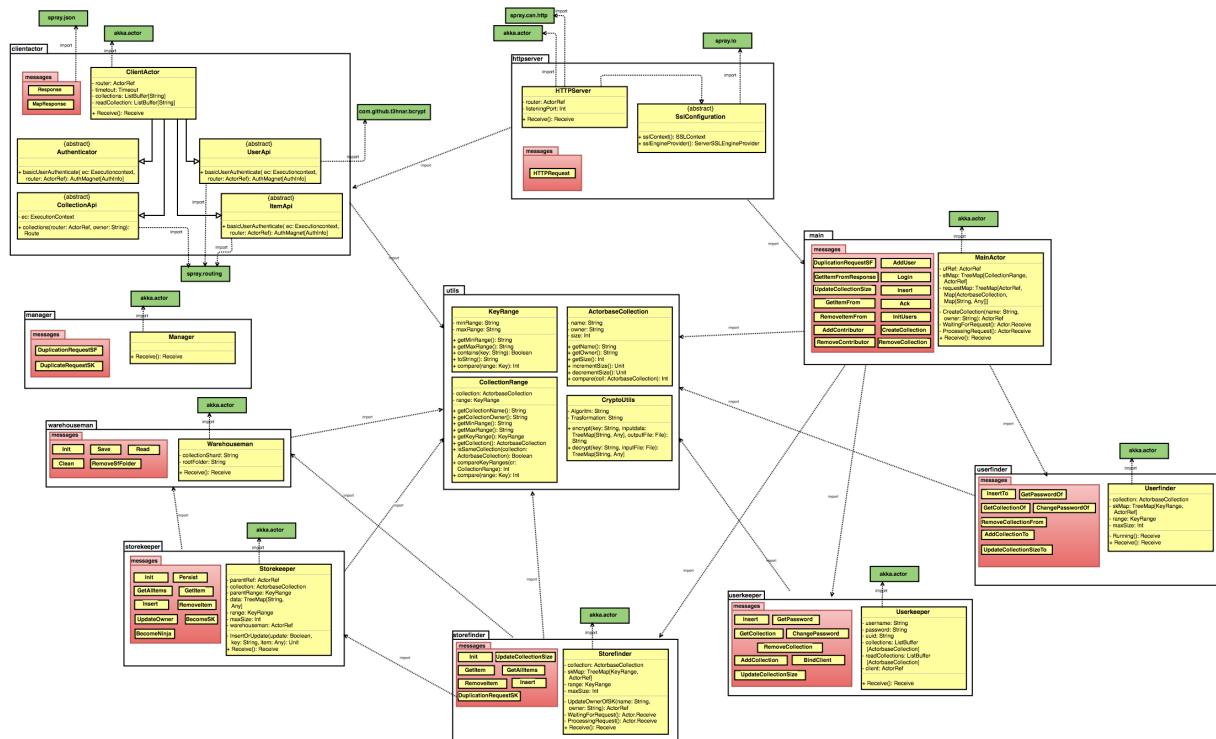


Figura 3: Actorsystem, visione generale

#### 4.2.1 Descrizione

Package<sub>G</sub> che raggruppa tutte le componenti del sistema che rappresentano il server<sub>G</sub>



#### 4.2.2 Package contenuti

- `actorbase::actorsystem::clientactor;`
- `actorbase::actorsystem::httpserver;`
- `actorbase::actorsystem::storefinder;`
- `actorbase::actorsystem::storekeeper;`
- `actorbase::actorsystem::userfinder;`
- `actorbase::actorsystem::userkeeper;`
- `actorbase::actorsystem::warehouseman;`
- `actorbase::actorsystem::manager;`
- `actorbase::actorsystem::main;`
- `actorbase::actorsystem::utils.`

#### 4.2.3 Interazioni con altre componenti

- `actorbase::driver;`
- Akka.



### 4.3 actorbase::actorsystem::utils

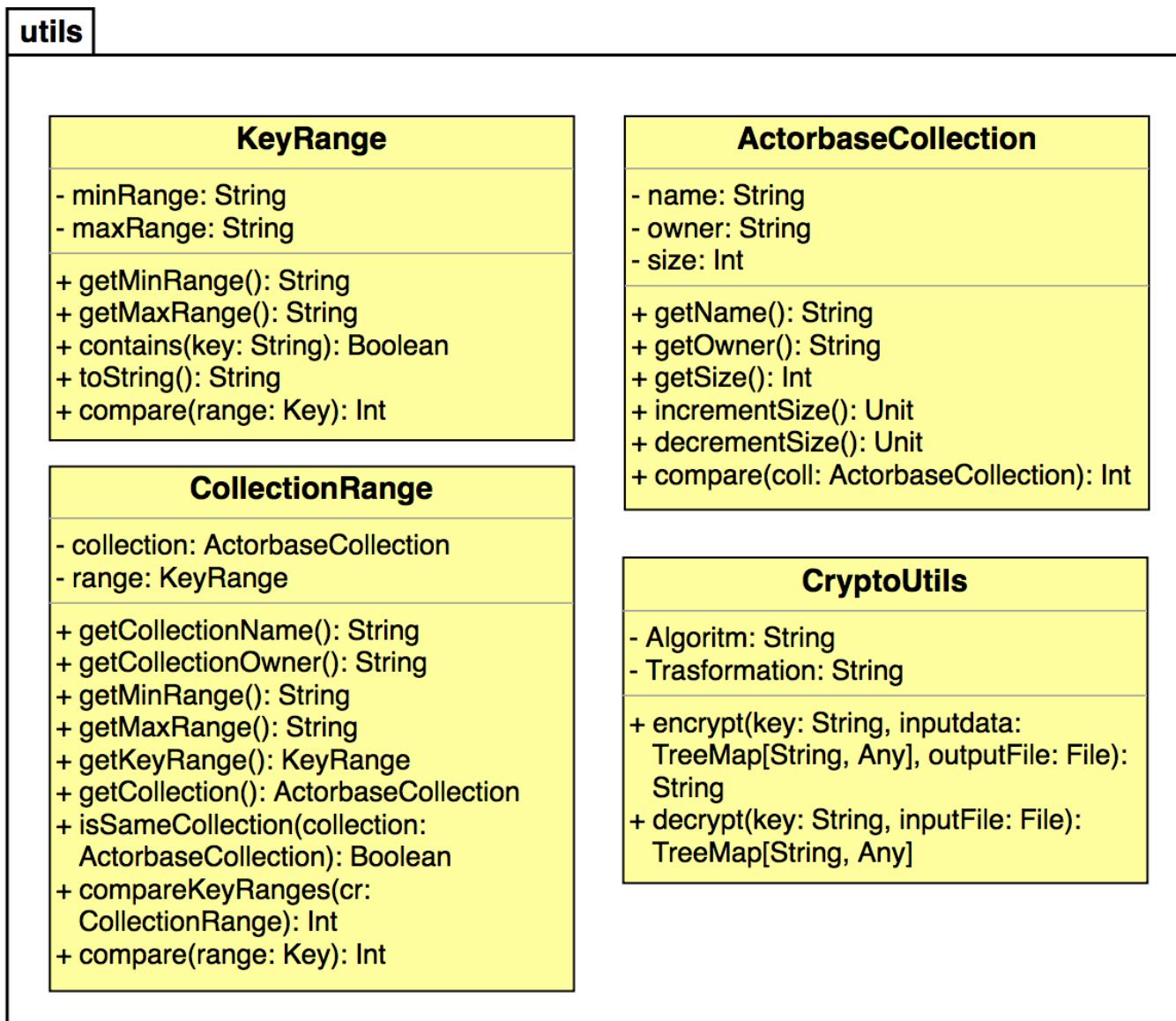


Figura 4: Package actorsystem::utils



### 4.3.1 Descrizione

Package<sub>G</sub> contenente diverse classi di utilità.

### 4.3.2 Interazioni con altre componenti

- actorbase::actorsystem::storekeeper;
- actorbase::actorsystem::storefinder;
- actorbase::actorsystem::userkeeper;
- actorbase::actorsystem::warehouseman;
- actorbase::actorsystem::main;
- actorbase::actorsystem::clientactor.

### 4.3.3 Classi

#### 4.3.3.1 KeyRange

##### 4.3.3.1.1 Descrizione

Classe atta a modellare un intervallo di chiavi.  
Viene utilizzata da diversi attori.

##### 4.3.3.1.2 Utilizzo

Viene utilizzata da diversi attori del sistema. Questa classe espone metodi getter, un metodo compare e un metodo per controllare se una determinata stringa fa parte di questo intervallo di chiavi.

##### 4.3.3.1.3 Realizza

- trait scala.math.Ordered.orderingToOrdered

#### 4.3.3.1.4 Interazioni con altre classi

- actorbase::actorsystem::storekeeper::Storekeeper;
- actorbase::actorsystem::storefinder::Storefinder;
- actorbase::actorsystem::main::Main.



#### 4.3.3.2 CollectionRange

##### 4.3.3.2.1 Descrizione

Classe atta a modellare un intervallo di chiavi di una specifica collezione.

##### 4.3.3.2.2 Utilizzo

Viene utilizzata da diversi attori del sistema. Questa classe espone metodi getter, un metodo per il confronto della collezione, un metodo per il confronto del range delle chiavi e un metodo compare.

##### 4.3.3.2.3 Realizza

- trait scala.math.Ordered.orderingToOrdered

##### 4.3.3.2.4 Interazioni con altre classi

- actorbase::actorsystem::storefinder::Storefinder;
- actorbase::actorsystem::main::Main.

#### 4.3.3.3 CryptoUtils

##### 4.3.3.3.1 Descrizione

Classe atta a criptare e decriptare i dati per il salvataggio e la lettura dei dati su disco.

##### 4.3.3.3.2 Utilizzo

Viene utilizzata dall'attore di tipo Warehouseman<sub>G</sub> per il salvataggio e la lettura del database su disco.

##### 4.3.3.3.3 Interazioni con altre classi

- actorbase::actorsystem::warehouseman::Warehouseman.

#### 4.3.3.4 ActorbaseCollection

##### 4.3.3.4.1 Descrizione

Classe atta a modellare una collezione del sistema actorbase.



#### 4.3.3.4.2 Utilizzo

Viene utilizzata da diversi attori del sistema. Questa classe espone metodi getter e un metodo per il confronto con un altro oggetto ActorbaseCollection.

#### 4.3.3.4.3 Realizza

- trait scala.math.Ordered

#### 4.3.3.4.4 Interazioni con altre classi

- actorbase::actorsystem::storekeeper::Storekeeper;
- actorbase::actorsystem::storefinder::Storefinder;
- actorbase::actorsystem::userkeeper::Userkeeper;
- actorbase::actorsystem::main::Main;
- actorbase::actorsystem::clientactor::ClientActor.



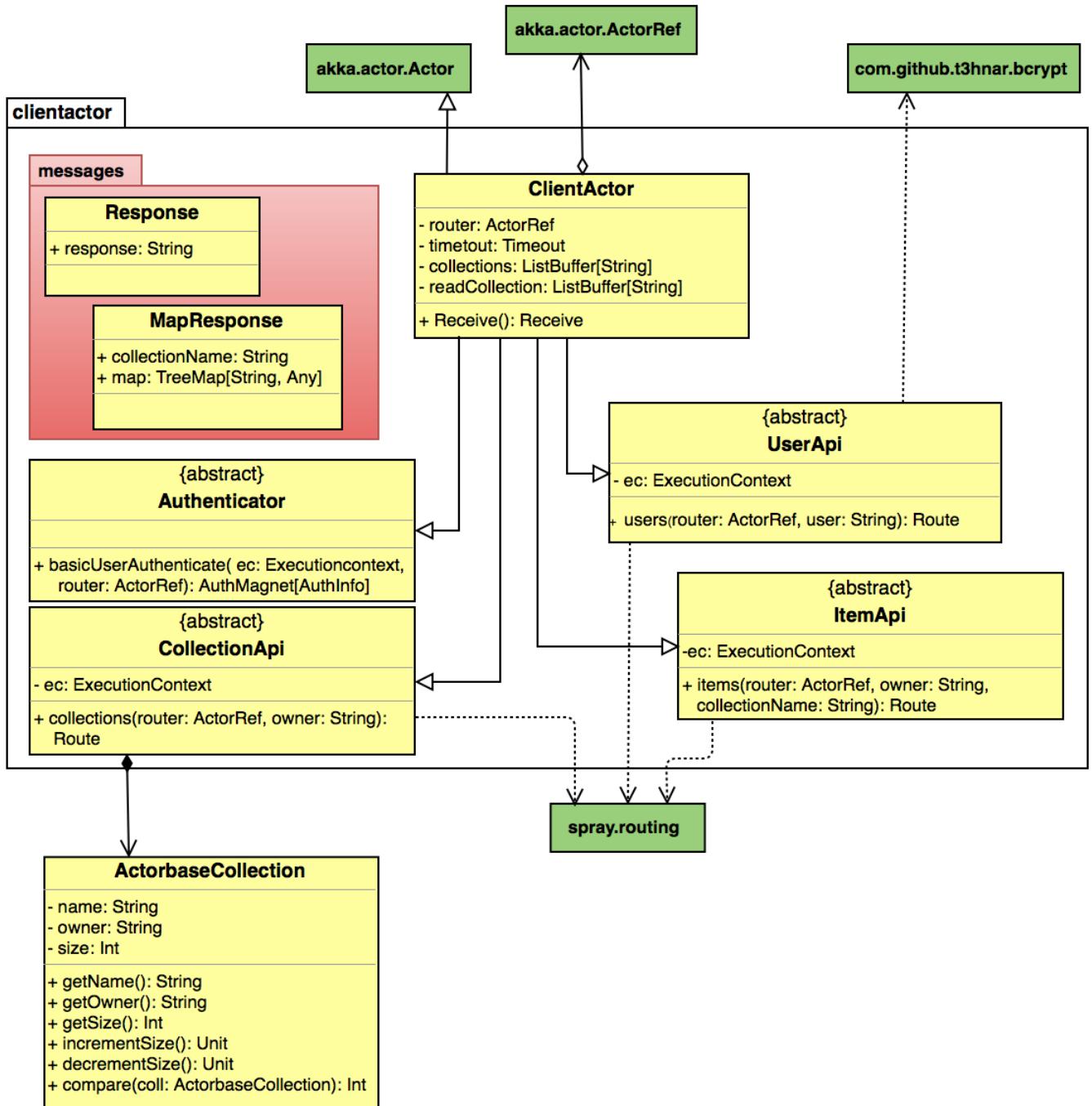
ScalateKids

4 COMPONENTI

---



#### 4.4 actorbase::actorsystem::clientactor





#### 4.4.1 Descrizione

Package<sub>G</sub> per l'attore<sub>G</sub> con cui si interfacerà il driver<sub>G</sub>.

#### 4.4.2 Package contenuti

- `actorbase::actorsystem::clientactor::messages.`

#### 4.4.3 Interazioni con altre componenti

- `actorbase::actorsystem::httpserver;`
- `actorbase::driver::client::ActorbaseClient;`
- `spray.json;`
- `spray.routing;`
- `akka.actor;`
- `com.github.t3hnar.bcrypt.`

#### 4.4.4 Classi

##### 4.4.4.1 `actorbase::actorsystem::clientactor::Authenticator`

###### 4.4.4.1.1 Descrizione

Classe astratta usata per effettuare la validazione di un login.

###### 4.4.4.1.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare il controllo delle credenziali di login.

##### 4.4.4.2 `actorbase::actorsystem::clientactor::CollectionApi`

###### 4.4.4.2.1 Descrizione

Classe astratta usata per analizzare le route che richiedono operazioni su collezioni.

###### 4.4.4.2.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare l'analisi delle route che richiedono operazioni su collezioni.



#### 4.4.4.2.3 Interazioni con altre classi

- 
- `actorbase::actorsystem::utils::ActorbaseCollection`;
- `spray.routing`.

#### 4.4.4.3 `actorbase::actorsystem::clientactor::ItemApi`

##### 4.4.4.3.1 Descrizione

Classe astratta usata per analizzare le route che richiedono operazioni su item.

##### 4.4.4.3.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare l'analisi delle route che richiedono operazioni su item.

#### 4.4.4.3 Interazioni con altre classi

- `spray.routing`.

#### 4.4.4.4 `actorbase::actorsystem::clientactor::UserApi`

##### 4.4.4.4.1 Descrizione

Classe astratta usata per analizzare le route che richiedono operazioni su utenti.

##### 4.4.4.4.2 Utilizzo

Questa classe viene usata dalla classe che la realizza per effettuare l'analisi delle route che richiedono operazioni su utenti.

#### 4.4.4.4.3 Interazioni con altre classi

- `spray.routing`.

#### 4.4.4.5 `actorbase::actorsystem::clientactor::ClientActor`

##### 4.4.4.5.1 Descrizione

Classe che rappresenta l'attore<sub>G</sub> con cui si interfaccia il driver<sub>G</sub> dopo la connessione.



#### 4.4.4.5.2 Utilizzo

Questo attore<sub>G</sub> riceve le richieste da actorbase::driver::client::Connection e si occupa di inviare messaggi al Main.

Questa classe inoltre si occupa di capire che tipo di richiesta è stata fatta dal driver<sub>G</sub> basandosi sulla route di quest'ultima.

#### 4.4.4.5.3 Classi ereditate

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

#### 4.4.4.5.4 Interazioni con altre classi

- actorbase::actorsystem::httpserver::HTTPServer;
- actorbase::driver::client::ActorbaseClient.



## 4.5 actorbase::actorsystem::clientactor::messages

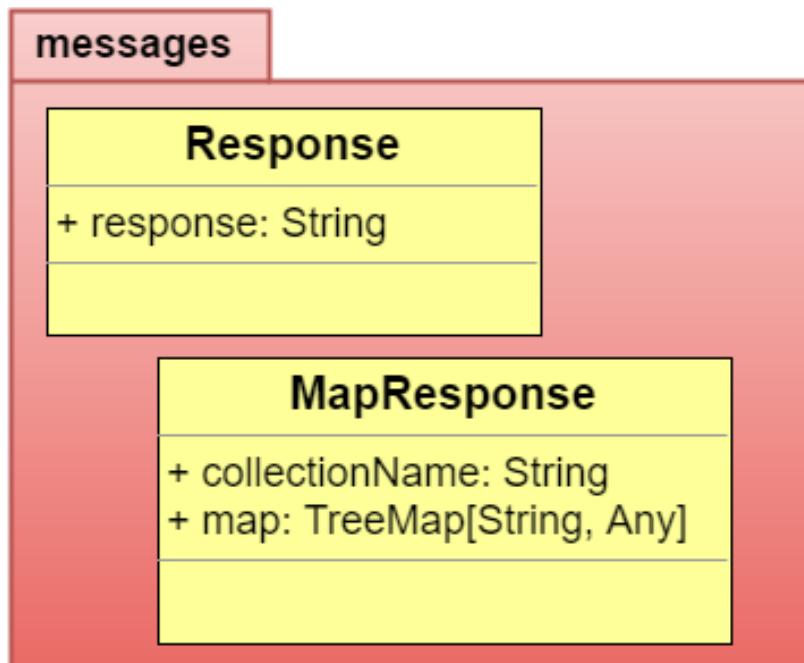


Figura 6: Actorsystem: package messages del clientactor

### 4.5.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti dal `ClientActor`.

### 4.5.2 Classi

#### 4.5.2.1 actorbase::actorsystem::clientactor::messages::Response

##### 4.5.2.1.1 Descrizione

Messaggio che rappresenta la risposta da parte del server<sub>G</sub> da inoltrare al driver<sub>G</sub>.

##### 4.5.2.1.2 Utilizzo

Quando `actorbase::actorsystem::clientactor::ClientActor` riceve questo messaggio il contenuto di quest'ultimo viene mandato a `actorbase::driver::client::ActorbaseClient`.



#### 4.5.2.1.3 Interazioni con altre classi

- spray.json.

#### 4.5.2.2 actorbase::actorsystem::clientactor::messages::MapResponse

##### 4.5.2.2.1 Descrizione

Messaggio che rappresenta la risposta da parte del server<sub>G</sub> ad una ricerca da inoltrare al driver<sub>G</sub>.

##### 4.5.2.2.2 Utilizzo

Quando ClientActor riceve questo messaggio il contenuto di quest'ultimo viene mandato a actorbase::driver::client::ActorbaseClient.

##### 4.5.2.2.3 Interazioni con altre classi

- spray.json.



## 4.6 actorbase::actorsystem::httpserver

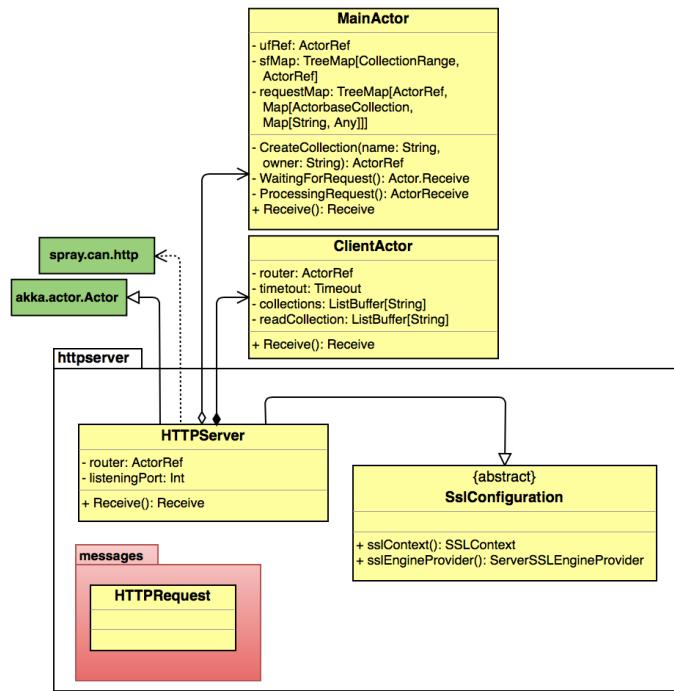


Figura 7: `httpserver`, visione generale del package

### 4.6.1 Descrizione

Package<sub>G</sub> per l'attore<sub>G</sub> con cui si interfacerà il driver<sub>G</sub> per la connessione iniziale.

### 4.6.2 Package contenuti

- `actorbase::actorsystem::httpserver::messages`.

### 4.6.3 Interazioni con altre componenti

- `actorbase::actorsystem::clientactor`;
- `actorbase::actorsystem::main`.



#### 4.6.4 Classi

##### 4.6.4.1 actorbase::actorsystem::httpserver::HTTPServer

###### 4.6.4.1.1 Descrizione

Classe che rappresenta l'attore<sub>G</sub> con cui si interfaccia il driver<sub>G</sub> per istanziare la connessione.

###### 4.6.4.1.2 Utilizzo

Questo attore<sub>G</sub> riceve la richiesta di connessione da [ActorbaseClient](#) e si occupa di associare al client<sub>G</sub> un attore<sub>G</sub> di tipo [ClientActor](#) per continuare le comunicazioni.

###### 4.6.4.1.3 Classi ereditate

- [actorbase::actorsystem::httpserver::SslConfiguration](#);
- [akka::actor::Actor](#).

#### 4.6.5 Interazioni con altre componenti

- [actorbase::actorsystem::clientactor::ClientActor](#);
- [actorbase::actorsystem::main::Main](#).

##### 4.6.5.1 actorbase::actorsystem::httpserver::SslConfiguration

###### 4.6.5.1.1 Descrizione

Classe che contiene metodi di utilità per applicare algoritmi di crittografia ssl.

###### 4.6.5.1.2 Utilizzo

Questa classe verrà utilizzata per applicare la crittografia SSL.



## 4.7 actorbase::actorsystem::warehouseman

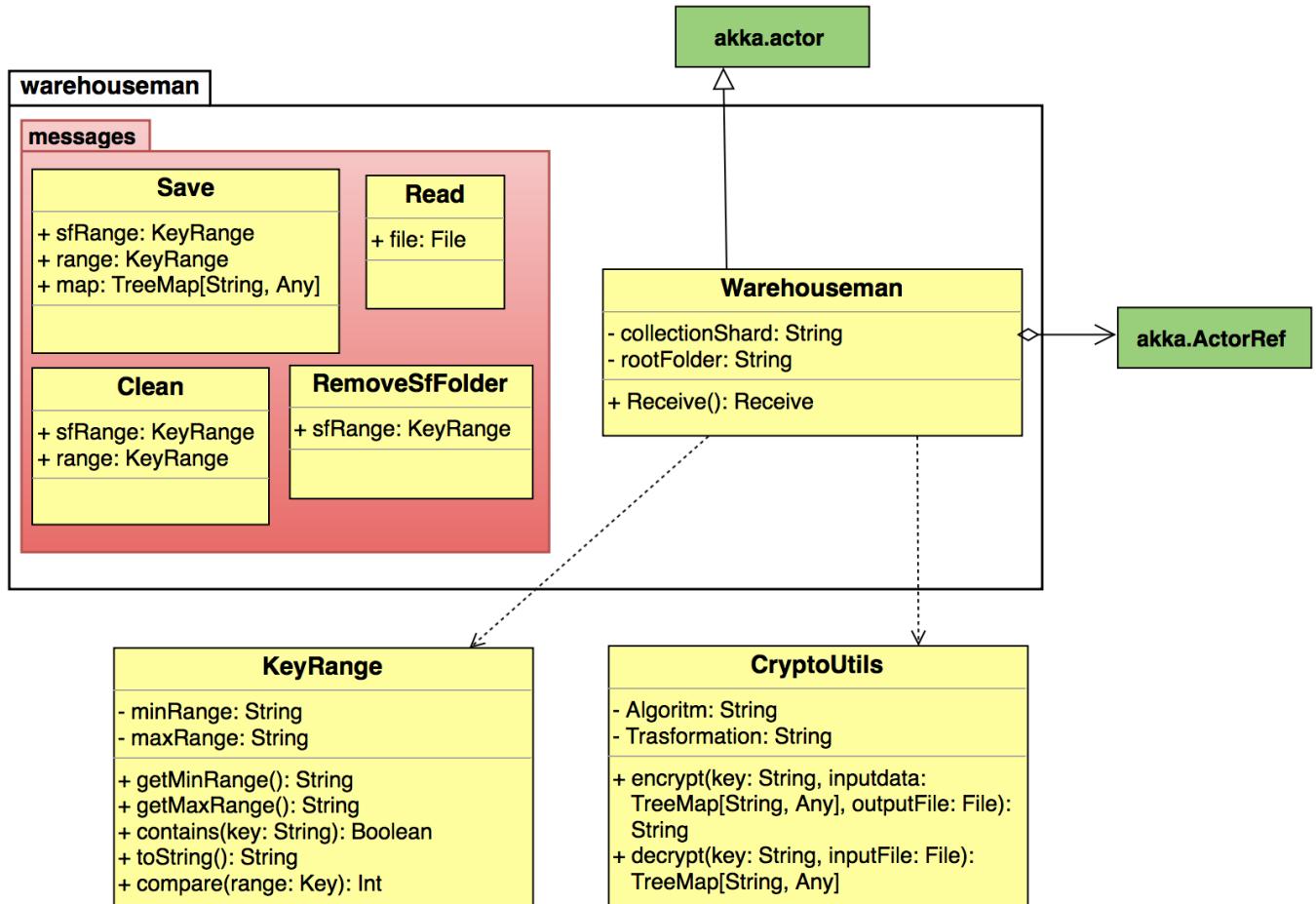


Figura 8: warehouseman, visione generale del package

### 4.7.1 Descrizione

Package<sub>G</sub> che rappresenta l'attore<sub>G</sub> che si occuperà della persistenza<sub>G</sub> su disco dei dati.

### 4.7.2 Package contenuti

- `actorbase::actorsystem::warehouseman::messages`.



#### 4.7.3 Interazioni con altre componenti

- `actorbase::actorsystem::utils;`
- `actorbase::actorsystem::storekeeper;`
- `actorbase::actorsystem::storefinder.`

#### 4.7.4 Classi

##### 4.7.4.1 `actorbase::actorsystem::warehouseman::Warehouseman`

###### 4.7.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Warehouseman<sub>G</sub>.

###### 4.7.4.1.2 Utilizzo

Questa classe viene utilizzata per effettuare il salvataggio su filesystem del database<sub>G</sub> e per caricare i dati da filesystem.

###### 4.7.4.1.3 Classi ereditate

- `akka::actor::Actor.`

###### 4.7.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::CryptoUtils;`
- `actorbase::actorsystem::utils::KeyRange;`
- `actorbase::actorsystem::storekeeper::Storekeeper;`
- `actorbase::actorsystem::storefinder::Storefinder.`



## 4.8 actorbase::actorsystem::warehouseman::messages

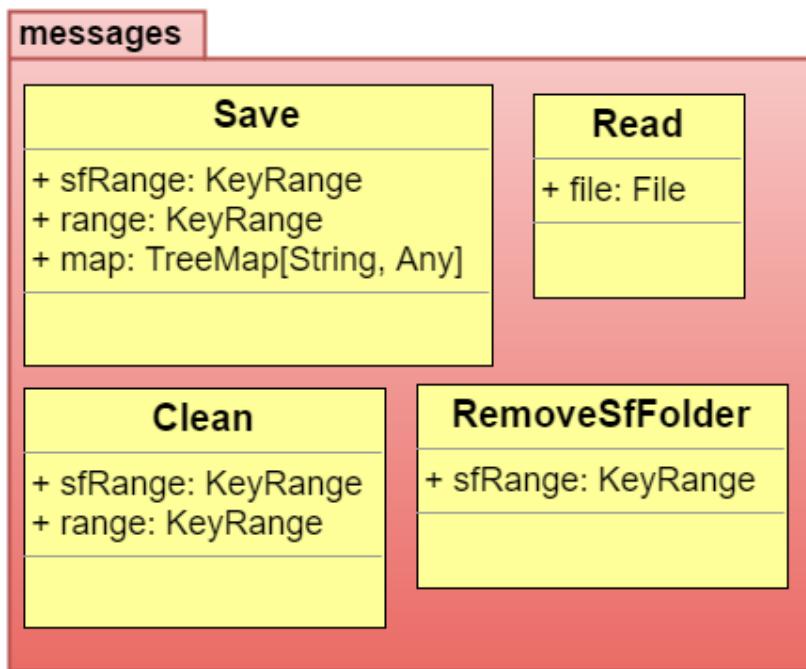


Figura 9: Actorsystem: package messages del warehouseman

### 4.8.1 Descrizione

Package<sub>G</sub> che racchiude tutti i messaggi che gli attori di tipo Warehouseman<sub>G</sub> possono ricevere.

### 4.8.2 Interazioni con altre componenti

- actorbase::actorsystem::utils.

#### 4.8.2.1 actorbase::actorsystem::warehouseman::messages::Save

##### 4.8.2.1.1 Descrizione

Messaggio che porta al salvataggio dei dati su disco.



#### 4.8.2.1.2 Utilizzo

Quando `actorbase::actorsystem::warehouseman::Warehouseman` riceve questo messaggio salverà i dati su disco sfruttando `actorbase::actorsystem::utils::CryptoUtils`.

#### 4.8.2.1.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`.

#### 4.8.2.2 `actorbase::actorsystem::warehouseman::messages::Clean`

##### 4.8.2.2.1 Descrizione

Messaggio che porta alla cancellazione dal filesystem di alcuni dati.

##### 4.8.2.2.2 Utilizzo

Quando `actorbase::actorsystem::warehouseman::Warehouseman` riceve questo messaggio rimuoverà i dati dal filesystem.

#### 4.8.2.2.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`.

#### 4.8.2.3 `actorbase::actorsystem::warehouseman::messages::RemoveSfFolder`

##### 4.8.2.3.1 Descrizione

Messaggio che porta alla rimozione di una cartella da disco.

##### 4.8.2.3.2 Utilizzo

Quando `actorbase::actorsystem::warehouseman::Warehouseman` riceve questo messaggio rimuoverà i dati da disco.

#### 4.8.2.3.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`.



#### 4.8.2.4 actorbase::actorsystem::warehouseman::messages::Read

##### 4.8.2.4.1 Descrizione

Messaggio che porta alla lettura di dati da disco.

##### 4.8.2.4.2 Utilizzo

Quando `actorbase::actorsystem::warehouseman::Warehouseman` riceve questo messaggio leggerà i dati da disco sfruttando `actorbase::actorsystem::utils::CryptoUtils`.



## 4.9 actorbase::actorsystem::main

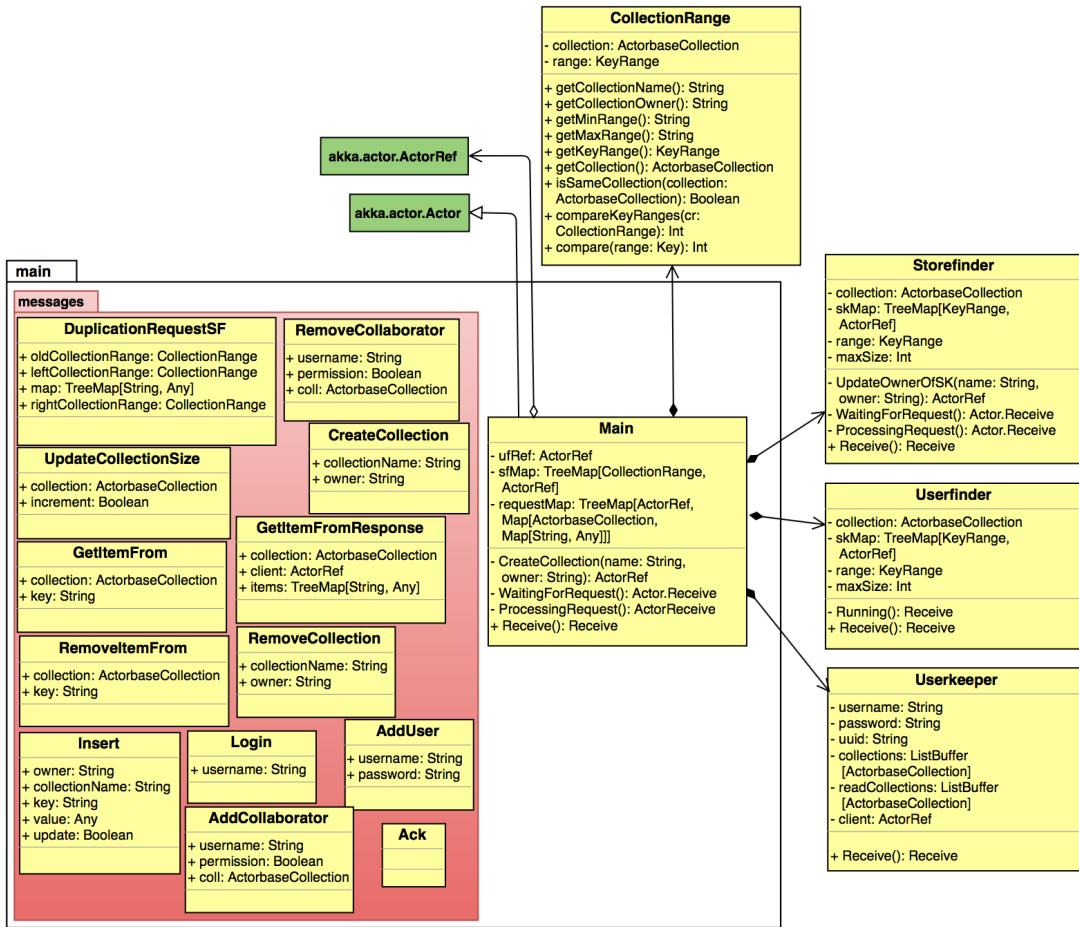


Figura 10: Package dell'attore di tipo Main

### 4.9.1 Descrizione

Package<sub>G</sub> che rappresenta l'attore<sub>G</sub> che si occuperà di gestire le richieste al server<sub>G</sub>.

### 4.9.2 Package contenuti

- `actorbase::actorsystem::main::messages`.



#### 4.9.3 Interazioni con altre componenti

- `actorbase::actorsystem::httpserver;`
- `actorbase::actorsystem::utils;`
- `actorbase::actorsystem::storefinder;`
- `actorbase::actorsystem::userkeeper;`
- `actorbase::actorsystem::userfinder.`

#### 4.9.4 Classi

##### 4.9.4.1 `actorbase::actorsystem::main::Main`

###### 4.9.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Main<sub>G</sub>.

###### 4.9.4.1.2 Utilizzo

Questa classe viene utilizzata per gestire le richieste ricevute al server<sub>G</sub>.

###### 4.9.4.1.3 Classi ereditate

- `akka::actor::Actor.`

##### 4.9.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::httpserver::HTTPServer;`
- `actorbase::actorsystem::utils::KeyRange;`
- `actorbase::actorsystem::utils::CollectionRange;`
- `actorbase::actorsystem::utils::ActorbaseCollection;`
- `actorbase::actorsystem::storefinder::Storefinder;`
- `actorbase::actorsystem::userkeeper::Userkeeper;`
- `actorbase::actorsystem::userfinder::Userfinder.`



## 4.10 actorbase::actorsystem::main::messages

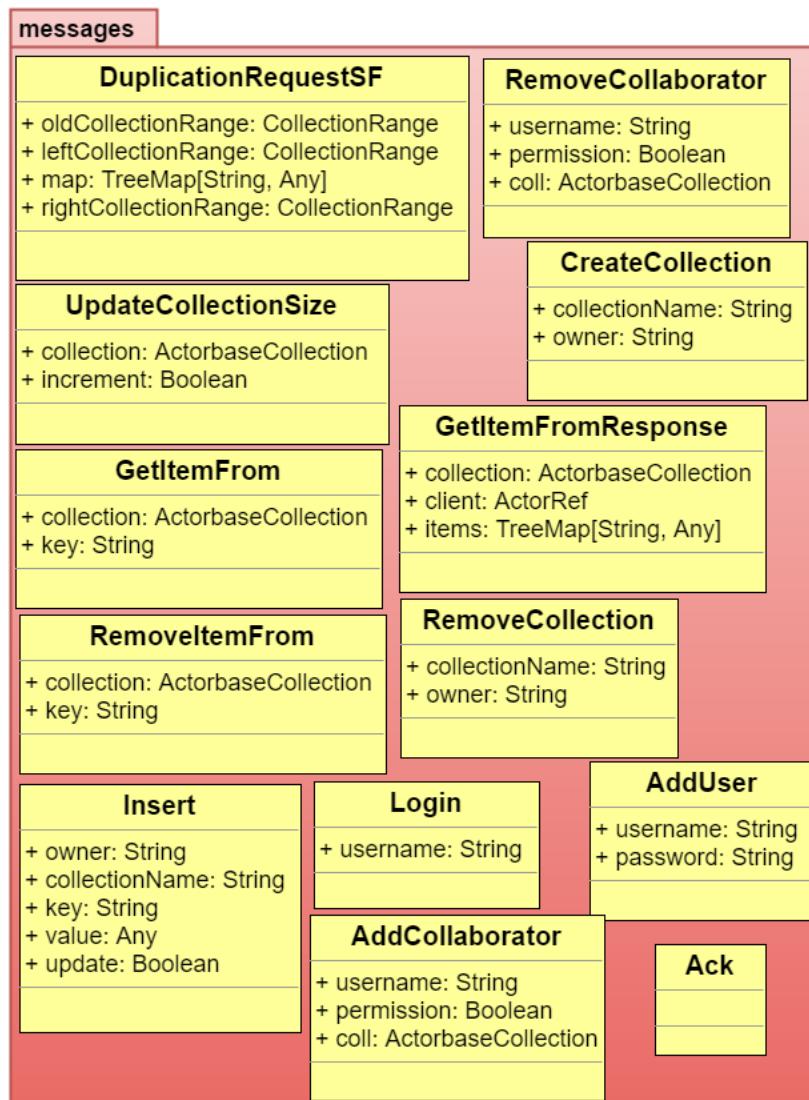


Figura 11: Actorsystem: package messages del main

### 4.10.1 Descrizione

Package<sub>G</sub> che racchiude tutti i messaggi che gli attori di tipo Main<sub>G</sub> possono ricevere.



#### 4.10.2 Interazioni con altre componenti

- `actorbase::actorsystem::utils.`

#### 4.10.3 Classi

##### 4.10.3.1 `actorbase::actorsystem::main::messages::AddUser`

###### 4.10.3.1.1 Descrizione

Messaggio che porta alla creazione di un utente<sub>G</sub>.

Viene ricevuto quando un amministratore di actorbase aggiunge un utente al sistema.

###### 4.10.3.1.2 Utilizzo

Quando `actorbase::actorsystem::main::Main` riceve questo messaggio inoltra la richiesta di aggiunta utente allo `Userfinder` di cui ha riferimento.

##### 4.10.3.2 `actorbase::actorsystem::main::messages::Login`

###### 4.10.3.2.1 Descrizione

Messaggio che indica una richiesta di autenticazione al sistema.

###### 4.10.3.2.2 Utilizzo

Quando `actorbase::actorsystem::main::Main` riceve questo messaggio manderà un messaggio di tipo `GetPasswordOf` all'attore di tipo `Userfinder` di cui ha riferimento.

##### 4.10.3.3 `actorbase::actorsystem::main::messages::Insert`

###### 4.10.3.3.1 Descrizione

Messaggio che indica una richiesta di inserimento di un item.

###### 4.10.3.3.2 Utilizzo

Quando `Main` riceve questo messaggio inoltrerà il messaggio allo `Storefinder` che si occupa del range di chiavi che contiene la chiave inserita.



#### 4.10.3.4 actorbase::actorsystem::main::messages::RemoveItemFrom

##### 4.10.3.4.1 Descrizione

Messaggio che indica una richiesta di rimozione di un item.

##### 4.10.3.4.2 Utilizzo

Quando **Main** riceve questo messaggio inoltrerà il messaggio allo **Storefinder** che si occupa del range di chiavi che contiene la chiave inserita.

#### 4.10.3.5 actorbase::actorsystem::main::messages::GetItemFrom

##### 4.10.3.5.1 Descrizione

Messaggio che indica una richiesta di ricerca di un item<sub>G</sub> da una collezione<sub>G</sub>.

##### 4.10.3.5.2 Utilizzo

Quando **Main** riceve questo messaggio cercherà gli attori di tipo **Storefinder** corrispondenti alla collezione<sub>G</sub> su cui effettuare la ricerca e inoltrerà lui la richiesta dell'item<sub>G</sub> con la chiave da cercare.

#### 4.10.3.6 actorbase::actorsystem::main::messages::GetItemFromResponse

##### 4.10.3.6.1 Descrizione

Messaggio che serve per comporre l'intera collezione da restituire all'utente

##### 4.10.3.6.2 Utilizzo

Quando **Main** riceve questo messaggio cercherà gli attori di tipo **Storefinder** corrispondenti alla collezione<sub>G</sub> su cui effettuare la ricerca e inoltrerà lui la richiesta dell'item<sub>G</sub> con la chiave da cercare.

#### 4.10.3.7 actorbase::actorsystem::main::messages::CreateCollection

##### 4.10.3.7.1 Descrizione

Messaggio che indica una richiesta di creazione di una collezione<sub>G</sub>.



#### 4.10.3.7.2 Utilizzo

Quando **Main** riceve questo messaggio provvederà a creare uno **Storefinder** adibito alla gestione della collezione e aggiornerà la propria struttura dati aggiungendo questo riferimento.

#### 4.10.3.8 actorbase::actorsystem::main::messages::RemoveCollection

##### 4.10.3.8.1 Descrizione

Messaggio che indica una richiesta di rimozione di una collezione<sub>G</sub>.

##### 4.10.3.8.2 Utilizzo

Quando **Main** riceve questo messaggio provvederà a terminare gli attori storefinder che rappresentano la collezione da rimuovere.

#### 4.10.3.9 actorbase::actorsystem::main::messages::AddContributor

##### 4.10.3.9.1 Descrizione

Messaggio che indica una richiesta di aggiunta di un collaboratore ad una collezione<sub>G</sub>.

##### 4.10.3.9.2 Utilizzo

Quando **Main** riceve questo messaggio inoltrerà la richiesta all'attore **Userfinder**.

#### 4.10.3.10 actorbase::actorsystem::main::messages::RemoveContributor

##### 4.10.3.10.1 Descrizione

Messaggio che indica una richiesta di rimozione di un collaboratore da una collezione<sub>G</sub>.

##### 4.10.3.10.2 Utilizzo

Quando **Main** riceve questo messaggio inoltrerà la richiesta all'attore **Userfinder**.

#### 4.10.3.11 actorbase::actorsystem::main::messages::Ack

##### 4.10.3.11.1 Descrizione

Messaggio che indica il completamento della richiesta precedente; **Main** torna nello stato `waitForRequests`.



#### 4.10.3.11.2 Utilizzo

Quando [Main](#) riceve questo messaggio cambia il suo contesto passando a `waitForRequests`.

#### 4.10.3.12 actorbase::actorsystem::main::messages::DuplicationRequestSF

##### 4.10.3.12.1 Descrizione

Messaggio che indica l'avvenuto duplicamento di uno [Storefinder](#).

##### 4.10.3.12.2 Utilizzo

Quando [Main](#) riceve questo messaggio provvederà a aggiornare la propria struttura dati contenente la mappatura degli [Storefinder](#).

#### 4.10.3.12.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`;
- `actorbase::actorsystem::utils::CollectionRange`.

#### 4.10.3.13 actorbase::actorsystem::main::messages::UpdateCollectionSize

##### 4.10.3.13.1 Descrizione

Messaggio che incrementa il numero delle coppie chiave-valore alla collezione

##### 4.10.3.13.2 Utilizzo

Quando [Main](#) riceve questo messaggio provvederà ad inoltrare allo [Userfinder](#) l'incremento della collezione.

#### 4.10.3.13.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.



## 4.11 actorbase::actorsystem::userfinder

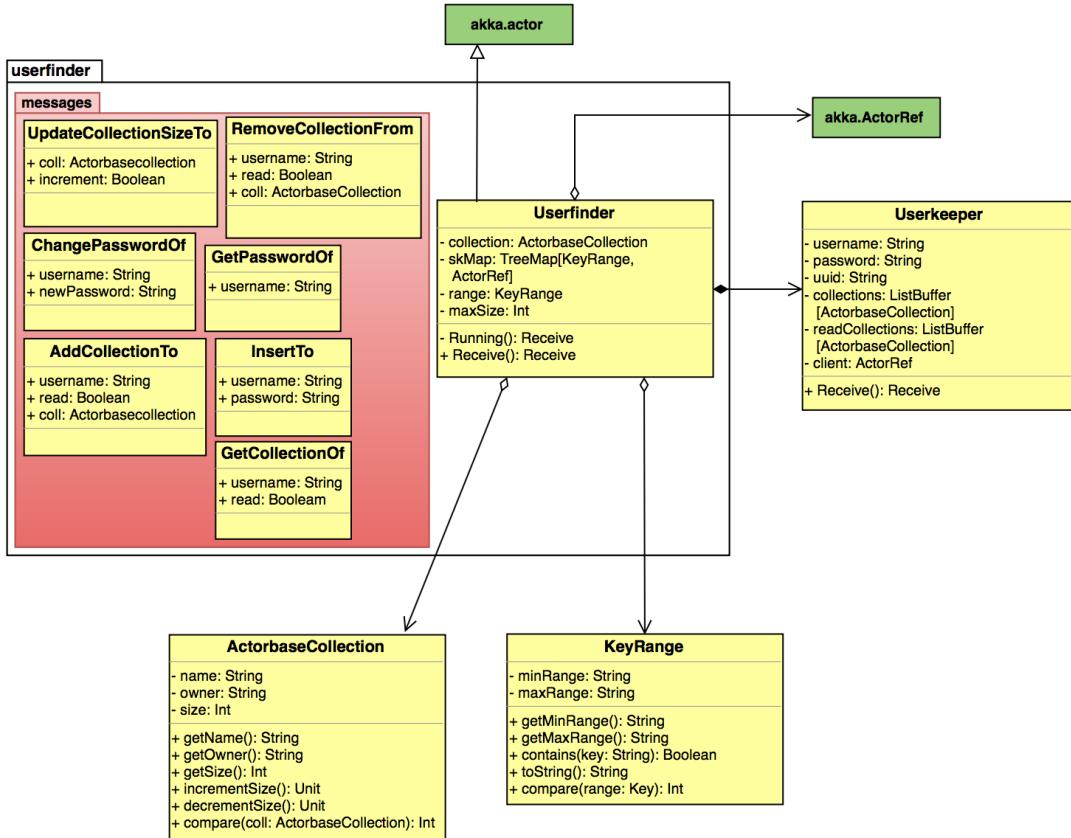


Figura 12: Userfinder, visione generale del package

### 4.11.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Userfinder<sub>G</sub>.

### 4.11.2 Package contenuti

- `actorbase::actorsystem::userfinder::messages`.



### 4.11.3 Interazioni con altre componenti

- `actorbase::actorsystem::main;`
- `actorbase::actorsystem::utils;`
- `actorbase::actorsystem::userkeeper.`

### 4.11.4 Classi

#### 4.11.4.1 `actorbase::actorsystem::userfinder::Userfinder`

##### 4.11.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Userfinder<sub>G</sub>.

##### 4.11.4.1.2 Utilizzo

Questa classe viene utilizzata per tener traccia degli attori di tipo `actorbase::actorsystem::userkeeper::Userkeeper`.

##### 4.11.4.1.3 Classi ereditate

- `akka::actor::Actor.`

##### 4.11.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::main::Main;`
- `actorbase::actorsystem::utils::KeyRange;`
- `actorbase::actorsystem::utils::ActorbaseCollection;`
- `actorbase::actorsystem::userkeeper::Userkeeper.`



## 4.12 actorbase::actorsystem::userfinder::messages

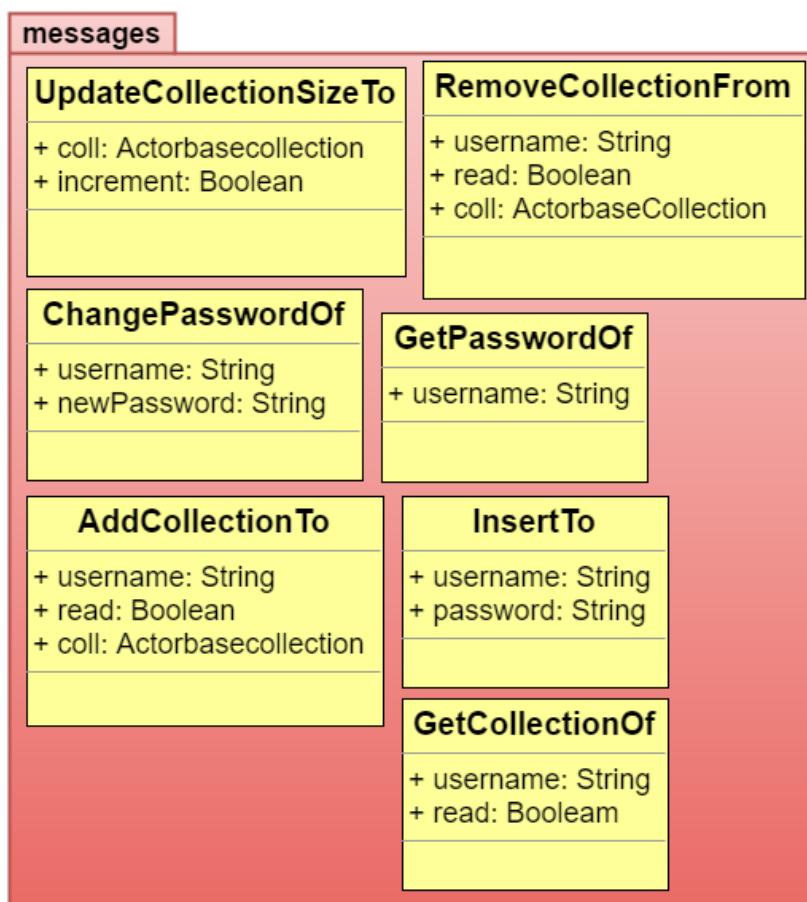


Figura 13: Actorsystem: package messages dello userfinder

### 4.12.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti da `actorbase::actorsystem::userfinder::Userfinder`.

### 4.12.2 Interazioni con altre componenti

- `actorbase::actorsystem::utils`.



### 4.12.3 Classi

#### 4.12.3.1 **actorbase::actorsystem::userfinder::messages::GetPasswordOf**

##### 4.12.3.1.1 Descrizione

Messaggio che indica la richiesta di autenticazione da parte di un utente.

##### 4.12.3.1.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a inoltrare la richiesta allo **Userkeeper** relativo allo username ricevuto.

#### 4.12.3.2 **actorbase::actorsystem::userfinder::messages::InsertTo**

##### 4.12.3.2.1 Descrizione

Messaggio che indica la richiesta di un aggiunta di un utente nel sistema, corrisponde con la creazione di un attore **actorbase::actorsystem::userkeeper::Userkeeper**.

##### 4.12.3.2.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a creare un attore di tipo **actorbase::actorsystem::userkeeper::Userkeeper** e l'inserimento del suo riferimento nella propria struttura dati.

#### 4.12.3.3 **actorbase::actorsystem::userfinder::messages::GetCollectionOf**

##### 4.12.3.3.1 Descrizione

Messaggio che indica la necessità di tornare le collezioni con il tipo di permessi immesso di un utente.

##### 4.12.3.3.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a inoltrare la richiesta all'attore di tipo **actorbase::actorsystem::userkeeper::Userkeeper** che rappresenta l'utente richiesto dal parametro di ingresso.

#### 4.12.3.4 **actorbase::actorsystem::userfinder::messages::ChangePasswordOf**

##### 4.12.3.4.1 Descrizione

Messaggio che indica la richiesta di una modifica password di un utente.



#### 4.12.3.4.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a inoltrare la richiesta di cambio password all'attore di tipo `actorbase::actorsystem::userkeeper::Userkeeper` corrispondente all'utente corretto.

#### 4.12.3.5 `actorbase::actorsystem::userfinder::messages::RemoveCollectionFrom`

##### 4.12.3.5.1 Descrizione

Messaggio che indica la richiesta di rimozione di una collezione dalla struttura dati dello `actorbase::actorsystem::userkeeper::Userkeeper` corrispondente.

##### 4.12.3.5.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a inoltrare la richiesta all'attore di tipo `actorbase::actorsystem::userkeeper::Userkeeper` corrispondente all'utente corretto.

#### 4.12.3.5.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

#### 4.12.3.6 `actorbase::actorsystem::userfinder::messages::AddCollectionTo`

##### 4.12.3.6.1 Descrizione

Messaggio che indica la richiesta di un aggiunta di una collezione con i permessi inseriti ad un utente nel sistema.

##### 4.12.3.6.2 Utilizzo

Quando l'attore riceve questo messaggio provvederà a inoltrare la richiesta all'attore di tipo `actorbase::actorsystem::userkeeper::Userkeeper` corrispondente all'utente su cui effettuare l'operazione.

#### 4.12.3.6.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

#### 4.12.3.7 `actorbase::actorsystem::userfinder::messages::UpdateCollectionSizeTo`

##### 4.12.3.7.1 Descrizione

Messaggio che indica la necessità di aggiornare il numero di item di una collezione.



#### 4.12.3.7.2 Utilizzo

Quando `actorbase::actorsystem::userfinder::Userfinder` riceve questo messaggio provvederà ad inoltrare all'attore di tipo `actorbase::actorsystem::userkeeper::Userkeeper`.

#### 4.12.3.7.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

### 4.13 actorbase::actorsystem::userkeeper

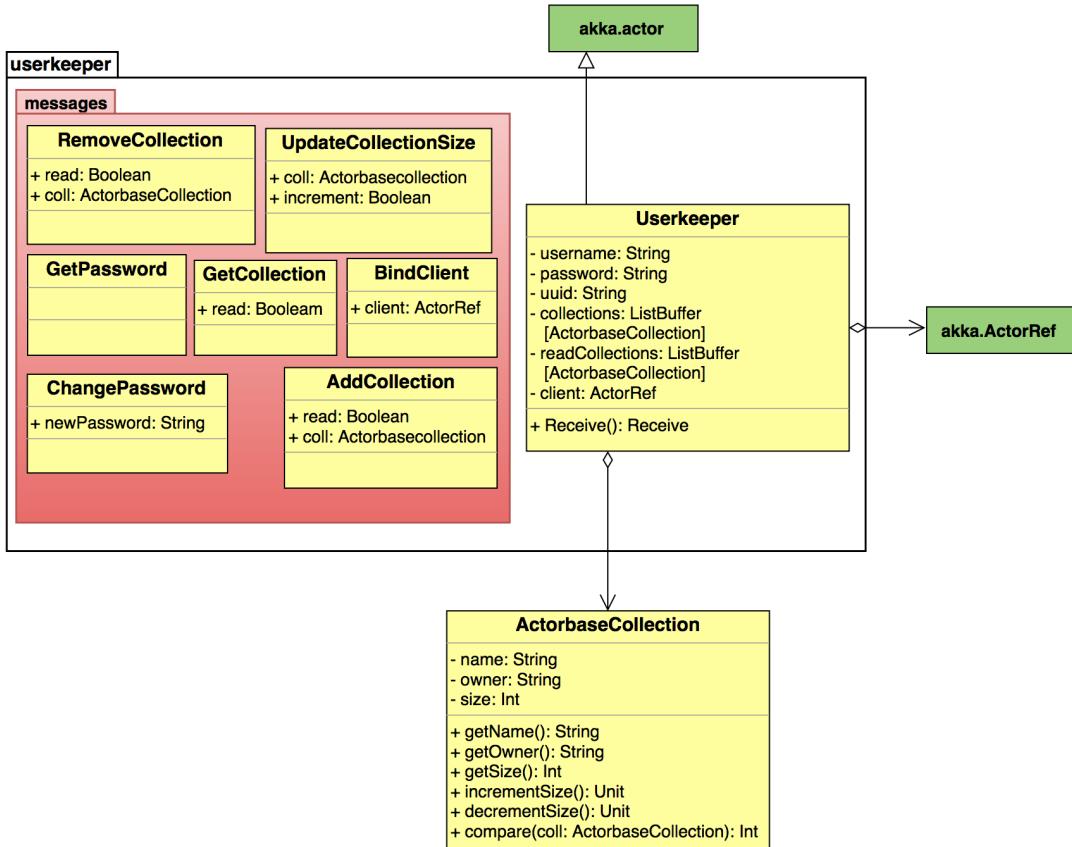


Figura 14: Userkeeper, visione generale del package



#### 4.13.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Userkeeper<sub>G</sub>.

#### 4.13.2 Package contenuti

- actorbase::actorsystem::userkeeper::messages.

#### 4.13.3 Interazioni con altre componenti

- actorbase::actorsystem::main;
- actorbase::actorsystem::utils;
- actorbase::actorsystem::userfinder.

#### 4.13.4 Classi

##### 4.13.4.1 actorbase::actorsystem::userkeeper::Userkeeper

###### 4.13.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Userkeeper<sub>G</sub>.

###### 4.13.4.1.2 Utilizzo

Questa classe viene utilizzata per salvare gli utenti e i loro dati nel database<sub>G</sub>.

###### 4.13.4.1.3 Classi ereditate

- akka::actor::Actor.

###### 4.13.4.1.4 Interazioni con altre classi

- actorbase::actorsystem::main::Main;
- actorbase::actorsystem::utils::ActorbaseCollection;
- actorbase::actorsystem::userfinder::Userfinder.



## 4.14 actorbase::actorsystem::userkeeper::messages

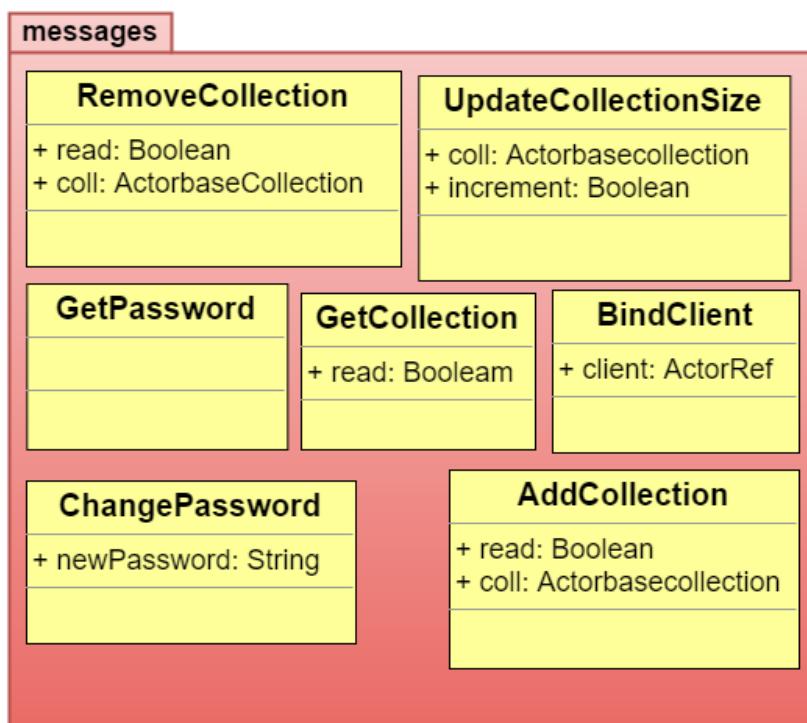


Figura 15: Actorsystem: package messages dello userkeeper

### 4.14.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti da `actorbase::actorsystem::userkeeper::Userkeeper`.

### 4.14.2 Interazioni con altre componenti

- `actorbase::actorsystem::utils`.

### 4.14.3 Classi

#### 4.14.3.1 actorbase::actorsystem::userkeeper::messages::GetCollections

##### 4.14.3.1.1 Descrizione

Messaggio che indica la richiesta della lista di collezioni<sub>G</sub> di cui l'utente ha permessi in lettura e scrittura.



#### 4.14.3.1.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà a restituire all'attore<sub>G</sub> di tipo `actorbase::actorsystem::clientactor::ClientActor` la lista di collezioni<sub>G</sub> di cui l'utente ha permessi in lettura e scrittura.

#### 4.14.3.2 `actorbase::actorsystem::userkeeper::messages::ChangePassword`

##### 4.14.3.2.1 Descrizione

Messaggio che indica la richiesta di cambiamento della password.

##### 4.14.3.2.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà a modificare la password dell'utente.

#### 4.14.3.3 `actorbase::actorsystem::userkeeper::messages::GetPassword`

##### 4.14.3.3.1 Descrizione

Messaggio che indica la richiesta della password.

##### 4.14.3.3.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà a inviare la password all'attore<sub>G</sub> di tipo `actorbase::actorsystem::clientactor::ClientActor`.

#### 4.14.3.4 `actorbase::actorsystem::userkeeper::messages::RemoveCollection`

##### 4.14.3.4.1 Descrizione

Messaggio che indica la richiesta di rimozione di una collezione<sub>G</sub> dalla lista contenente le collezioni<sub>G</sub> di cui l'utente ha permessi in lettura o in scrittura.

##### 4.14.3.4.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà a rimuovere la collezione<sub>G</sub> desiderata dalla lista contenente le collezioni<sub>G</sub> di cui l'utente ha permessi in lettura o in scrittura.



#### 4.14.3.4.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

#### 4.14.3.5 `actorbase::actorsystem::userkeeper::messages::AddCollection`

##### 4.14.3.5.1 Descrizione

Messaggio che indica la richiesta di aggiunta di una collezione<sub>G</sub> dalla lista contenente le collezioni<sub>G</sub> di cui l'utente ha permessi in lettura o in lettura e scrittura.

##### 4.14.3.5.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà ad aggiungere la collezione<sub>G</sub> desiderata dalla lista contenente le collezioni<sub>G</sub> di cui l'utente ha permessi in lettura o in lettura e scrittura.

#### 4.14.3.5.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

#### 4.14.3.6 `actorbase::actorsystem::userkeeper::messages::BindClient`

##### 4.14.3.6.1 Descrizione

Messaggio che indica la richiesta di collegamento con un ClientActor<sub>G</sub>.

##### 4.14.3.6.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà a salvare il riferimento dell'attore<sub>G</sub> di tipo `actorbase::actorsystem::clientactor::ClientActor` che ha inviato questo messaggio.

#### 4.14.3.7 `actorbase::actorsystem::userkeeper::messages::UpdateCollectionSize`

##### 4.14.3.7.1 Descrizione

Messaggio che aggiorna il numero di item di una collezione.

##### 4.14.3.7.2 Utilizzo

Quando `actorbase::actorsystem::userkeeper::Userkeeper` riceve questo messaggio provvederà ad aggiornare il contatore relativo al numero di oggetti nella collezione.



#### 4.14.3.7.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection`.

### 4.15 actorbase::actorsystem::storefinder

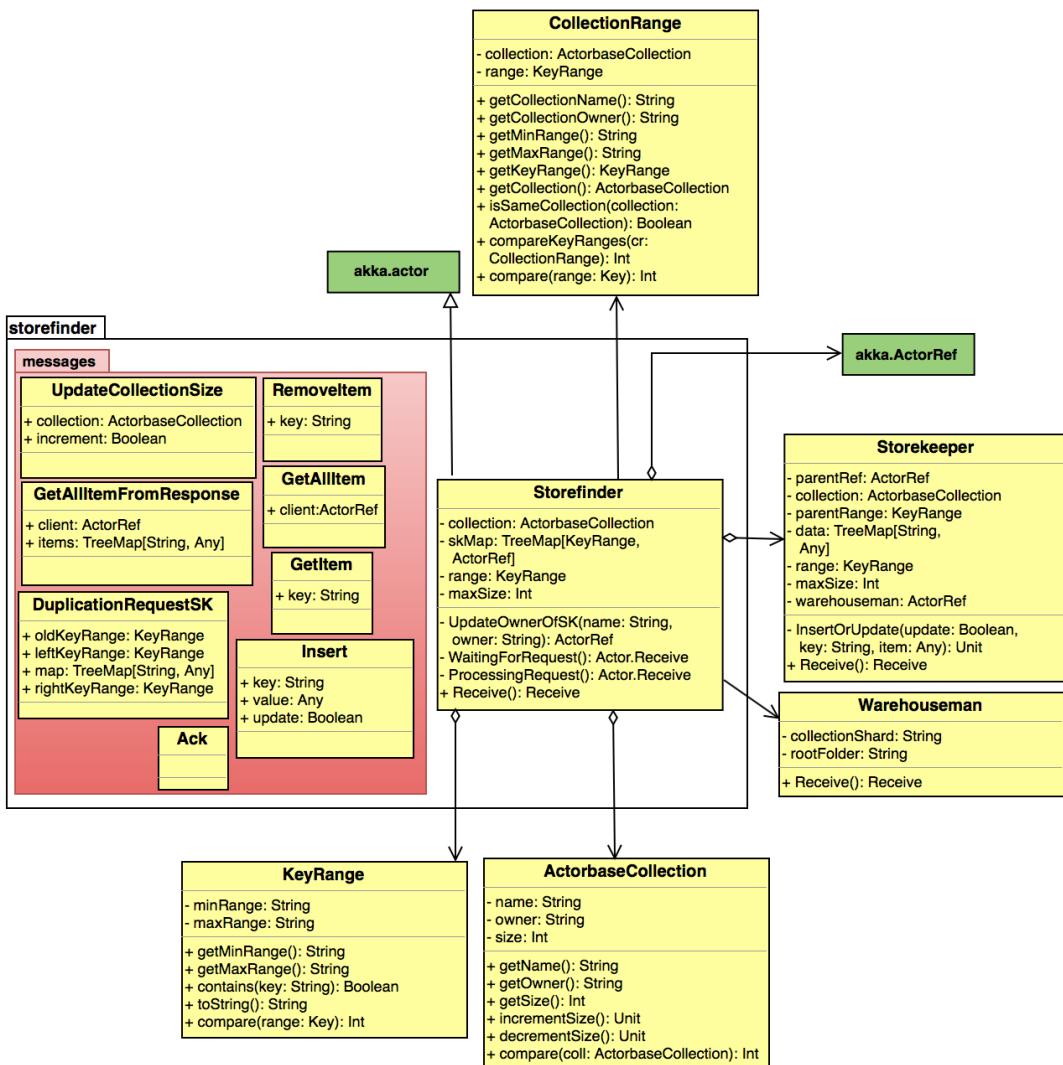


Figura 16: Storefinder, visione generale del package



#### 4.15.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Storefinder<sub>G</sub>.

#### 4.15.2 Package contenuti

- `actorbase::actorsystem::storefinder::messages.`

#### 4.15.3 Interazioni con altre componenti

- `actorbase::actorsystem::main;`
- `actorbase::actorsystem::utils;`
- `actorbase::actorsystem::storekeeper;`
- `actorbase::actorsystem::warehouseman.`

#### 4.15.4 Classi

##### 4.15.4.1 `actorbase::actorsystem::storefinder::Storefinder`

###### 4.15.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Storefinder<sub>G</sub>.

###### 4.15.4.1.2 Utilizzo

Questa classe viene utilizzata per tener traccia degli attori di tipo `actorbase::actorsystem::storekeeper::Storekeeper`.

###### 4.15.4.1.3 Classi ereditate

- `akka::actor::Actor.`

###### 4.15.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::main::Main;`
- `actorbase::actorsystem::utils::ActorbaseCollection;`
- `actorbase::actorsystem::utils::KeyRange;`
- `actorbase::actorsystem::utils::CollectionRange;`
- `actorbase::actorsystem::storekeeper::Storekeeper;`



- `actorbase::actorsystem::warehouseman::Warehouseman.`

## 4.16 `actorbase::actorsystem::storefinder::messages`

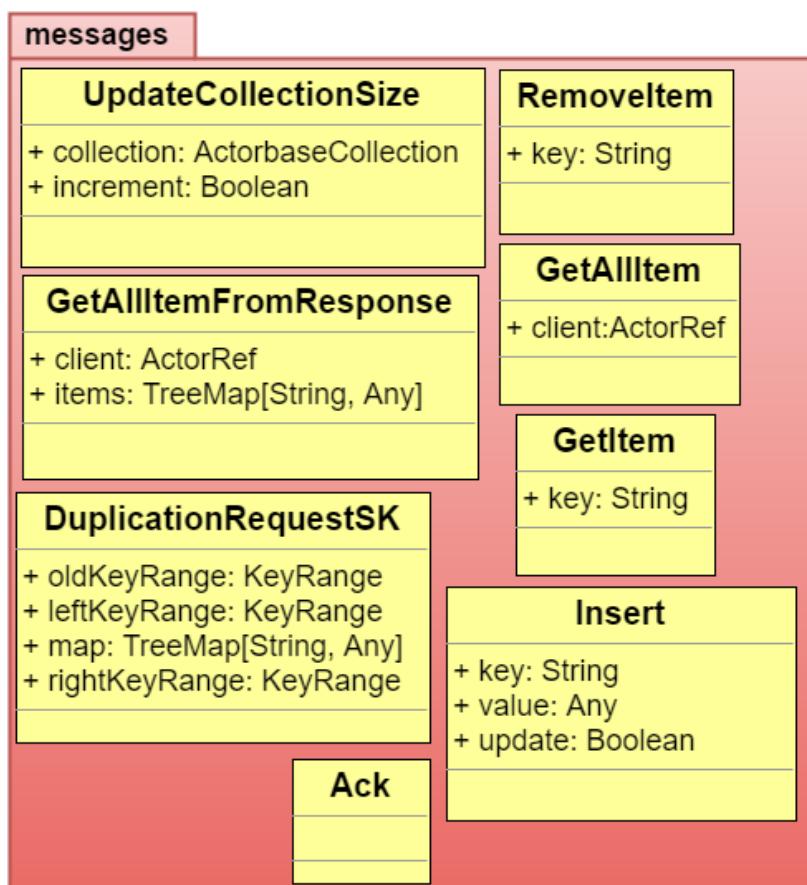


Figura 17: Actorsystem: package messages dello storefinder

### 4.16.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti da `actorbase::actorsystem::storefinder::Storefinder`.

### 4.16.2 Interazioni con altre componenti

- `actorbase::actorsystem::utils.`



### 4.16.3 Classi

#### 4.16.3.1 actorbase::actorsystem::storefinder::messages::DuplicationRequestSK

##### 4.16.3.1.1 Descrizione

Messaggio atto ad aggiornare uno Storefinder a seguito di una duplicazione di un attore di tipo Storekeeper.

##### 4.16.3.1.2 Utilizzo

Quando `actorbase::actorsystem::storefinder::Storefinder` riceve questo messaggio provvederà ad aggiornare la sua struttura dati.

In particolare aggiornerà i range di chiavi contenuti dagli Storekeeper presenti nella sua struttura dati.

##### 4.16.3.1.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`.

#### 4.16.3.2 actorbase::actorsystem::storefinder::messages::Ack

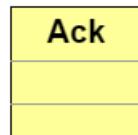


Figura 18: Classe `actorsystem::storefinder::messages::Ack`

##### 4.16.3.2.1 Descrizione

Messaggio che indica il completamento della richiesta precedente; `Storefinder` torna nello stato `waitForRequests`.

##### 4.16.3.2.2 Utilizzo

Quando `Storefinder` riceve questo messaggio cambia il suo contesto passando a `waitForRequests`.

#### 4.16.3.3 actorbase::actorsystem::storefinder::messages::GetItem

##### 4.16.3.3.1 Descrizione

Messaggio che indica la richiesta di un item<sub>g</sub>.



#### 4.16.3.3.2 Utilizzo

Quando `actorbase::actorsystem::storefinder::Storefinder` riceve questo messaggio provvederà a inoltrare la richiesta all'attore<sub>G</sub> di tipo `actorbase::actorsystem::storekeeper::Storekeeper` che contiene l'item<sub>G</sub> cercato.

#### 4.16.3.4 actorbase::actorsystem::storefinder::messages::RemoveItem

##### 4.16.3.4.1 Descrizione

Messaggio che indica la richiesta di rimozione di un item<sub>G</sub>.

##### 4.16.3.4.2 Utilizzo

Quando `actorbase::actorsystem::storefinder::Storefinder` riceve questo messaggio provvederà a inoltrare la richiesta all'attore<sub>G</sub> di tipo `actorbase::actorsystem::storekeeper::Storekeeper` che contiene l'item<sub>G</sub> da rimuovere.

#### 4.16.3.5 actorbase::actorsystem::storefinder::messages::Insert

##### 4.16.3.5.1 Descrizione

Messaggio che indica la richiesta di inserimento di un item<sub>G</sub>.

##### 4.16.3.5.2 Utilizzo

Quando `actorbase::actorsystem::storefinder::Storefinder` riceve questo messaggio provvederà a inoltrare la richiesta all'attore<sub>G</sub> di tipo `actorbase::actorsystem::storekeeper::Storekeeper` destinato a ricevere l'item<sub>G</sub> da inserire.

#### 4.16.3.6 actorbase::actorsystem::storefinder::messages::GetAllItem

##### 4.16.3.6.1 Descrizione

Messaggio che indica la richiesta di restituzione di tutti gli item<sub>G</sub> mappati dallo Storefinder.

##### 4.16.3.6.2 Utilizzo

Quando `actorbase::actorsystem::storefinder::Storefinder` riceve questo messaggio provvederà a inoltrare la richiesta a tutti gli attori di tipo `actorbase::actorsystem::storekeeper::Storekeeper` che dovranno restituire tutti gli items<sub>G</sub> contenuti in essi.



#### 4.16.3.7 **actorbase::actorsystem::storefinder::messages::GetAllItemResponse**

##### 4.16.3.7.1 Descrizione

Messaggio che serve per comporre l'intera collezione da restituire all'utente

##### 4.16.3.7.2 Utilizzo

Quando **Storefinder** riceve questo messaggio non farà altro che richiamare il metodo [GetItemFromResponse](#) del **Main**.

#### 4.16.3.8 **actorbase::actorsystem::storefinder::messages::UpdateCollectionSize**

##### 4.16.3.8.1 Descrizione

Messaggio che incrementa il numero delle coppie chiave-valore alla collezione

##### 4.16.3.8.2 Utilizzo

Quando **Storefinder** riceve questo messaggio non farà altro che richiamare il metodo [UpdateCollectionSize](#) del **Main**.

##### 4.16.3.8.3 Interazioni con altre classi

- [actorbase::actorsystem::utils::ActorbaseCollection](#).



## 4.17 actorbase::actorsystem::storekeeper

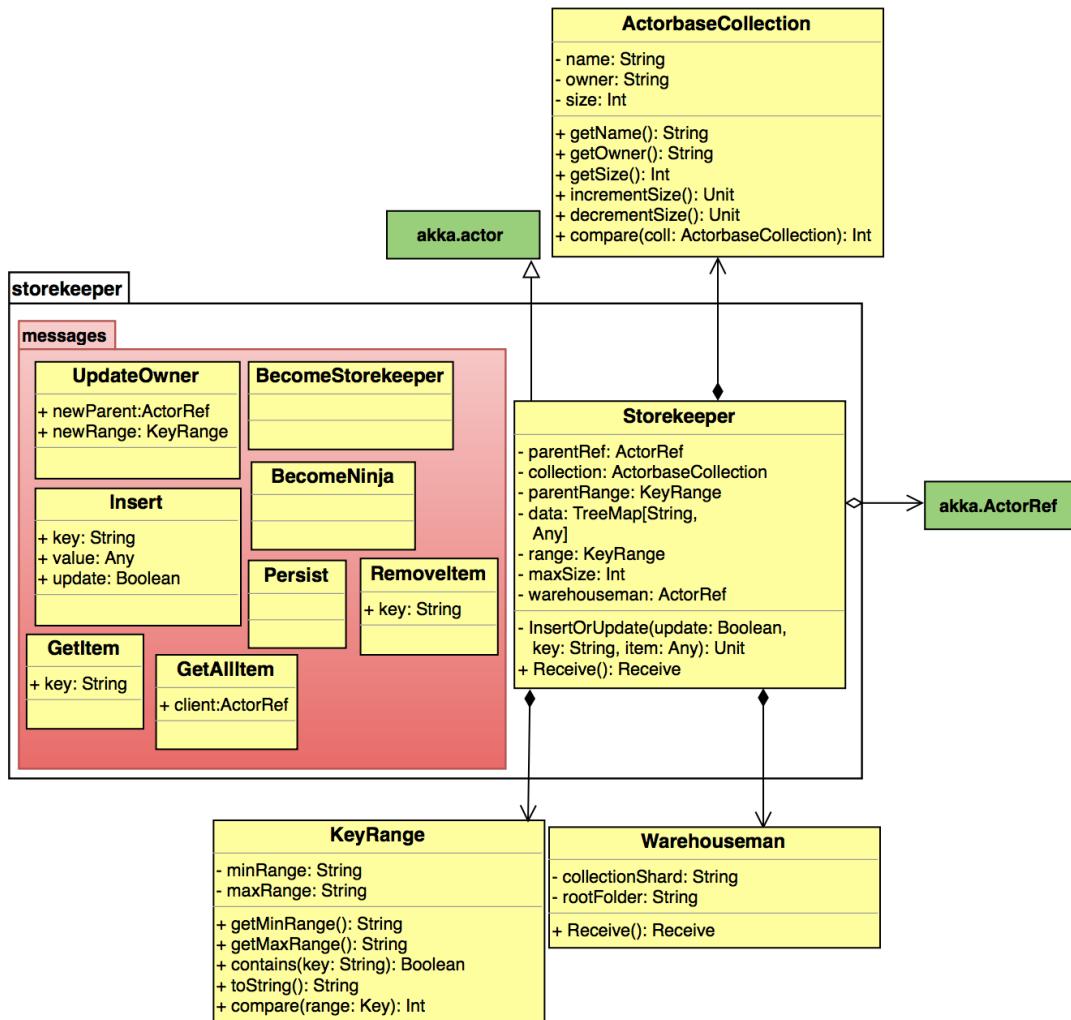


Figura 19: Storekeeper, visione generale del package

### 4.17.1 Descrizione

Package<sub>G</sub> che rappresenta l'attore<sub>G</sub> di tipo Storekeeper<sub>G</sub>.



#### 4.17.2 Package contenuti

- `actorbase::actorsystem::storekeeper::messages.`

#### 4.17.3 Interazioni con altre componenti

- `actorbase::actorsystem::utils;`
- `actorbase::actorsystem::storefinder;`
- `actorbase::actorsystem::warehouseman.`

#### 4.17.4 Classi

##### 4.17.4.1 `actorbase::actorsystem::storekeeper::Storekeeper`

###### 4.17.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Storekeeper<sub>G</sub>.

###### 4.17.4.1.2 Utilizzo

Questa classe viene utilizzata per salvare nel database<sub>G</sub> gli item.

###### 4.17.4.1.3 Classi ereditate

- `akka::actor::Actor.`

###### 4.17.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::utils::ActorbaseCollection;`
- `actorbase::actorsystem::utils::KeyRange;`
- `actorbase::actorsystem::storefinder::Storefinder;`
- `actorbase::actorsystem::warehouseman::Warehouseman.`



## 4.18 actorbase::actorsystem::storekeeper::messages

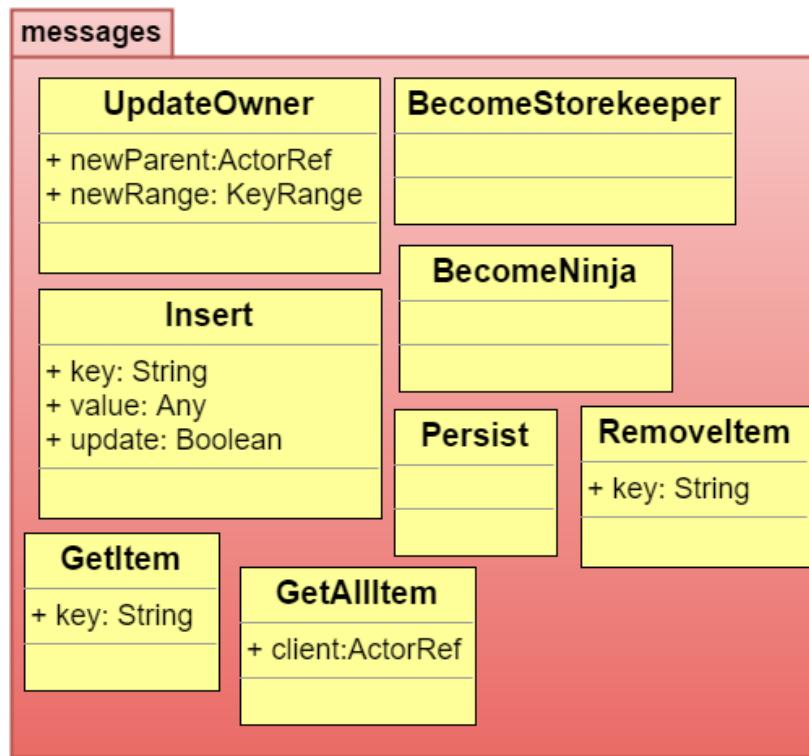


Figura 20: Actorsystem: package messages dello storekeeper

### 4.18.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti dall'attore<sub>G</sub> di tipo `actorbase::actorsystem::storekeeper::Storekeeper`.

### 4.18.2 Interazioni con altre componenti

- `actorbase::actorsystem::utils`.



### 4.18.3 Classi

#### 4.18.3.1 actorbase::actorsystem::storekeeper::messages::GetItem

##### 4.18.3.1.1 Descrizione

Messaggio che richiede un item<sub>G</sub> contenuto dall'attore<sub>G</sub> Storekeeper<sub>G</sub> che lo riceve.

##### 4.18.3.1.2 Utilizzo

Quando actorbase::actorsystem::storekeeper::Storekeeper riceve questo messaggio risponde mandando l'item<sub>G</sub> cercato all'attore<sub>G</sub> di tipo actorbase::actorsystem::clientactor::ClientActor che aveva effettuato la richiesta.

#### 4.18.3.2 actorbase::actorsystem::storekeeper::messages::GetAllItem

##### 4.18.3.2.1 Descrizione

Messaggio che richiede gli item<sub>G</sub> contenuti dall'attore<sub>G</sub> Storekeeper<sub>G</sub> che lo riceve.

##### 4.18.3.2.2 Utilizzo

Quando actorbase::actorsystem::storekeeper::Storekeeper riceve questo messaggio risponde mandando gli item<sub>G</sub> all'attore<sub>G</sub> di tipo actorbase::actorsystem::clientactor::ClientActor che aveva effettuato la richiesta.

#### 4.18.3.3 actorbase::actorsystem::storekeeper::messages::Insert

##### 4.18.3.3.1 Descrizione

Messaggio che richiede l'inserimento di un item<sub>G</sub>.

##### 4.18.3.3.2 Utilizzo

Quando actorbase::actorsystem::storekeeper::Storekeeper riceve questo messaggio inserisce l'item<sub>G</sub> nel suo attributo data.

#### 4.18.3.4 actorbase::actorsystem::storekeeper::messages::RemoveItem

##### 4.18.3.4.1 Descrizione

Messaggio che richiede la rimozione di un item<sub>G</sub>.



#### 4.18.3.4.2 Utilizzo

Quando `actorbase::actorsystem::storekeeper::Storekeeper` riceve questo messaggio rimuove l'item<sub>G</sub>.

#### 4.18.3.5 actorbase::actorsystem::storekeeper::messages::UpdateOwnerOfSk

##### 4.18.3.5.1 Descrizione

Messaggio che richiede l'update del campo dati che rappresenta lo `Storefinder` che contiene lo `Storekeeper` che ha ricevuto il messaggio.

##### 4.18.3.5.2 Utilizzo

Quando `Storekeeper` riceve questo messaggio aggiorna il campo dati relativo allo `Storefinder` che lo contiene.

#### 4.18.3.5.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::KeyRange`.

#### 4.18.3.6 actorbase::actorsystem::storekeeper::messages::Update

##### 4.18.3.6.1 Descrizione

Messaggio che richiede l'update dell'attore<sub>G</sub> che lo riceve.

##### 4.18.3.6.2 Utilizzo

Quando `actorbase::actorsystem::storekeeper::Storekeeper` riceve questo messaggio aggiorna gli item<sub>G</sub> che contiene.

#### 4.18.3.7 actorbase::actorsystem::storekeeper::messages::BecomeSK

##### 4.18.3.7.1 Descrizione

Messaggio che richiede che l'attore<sub>G</sub> di tipo `NinjaG` diventi di tipo `StorekeeperG`.

##### 4.18.3.7.2 Utilizzo

Quando `actorbase::actorsystem::storekeeper::Storekeeper` riceve questo messaggio ed è attualmente un attore di tipo `NinjaG`, esso cambia il proprio contesto e potrà ricevere tutti i messaggi di `actorbase::actorsystem::storekeeper::messages` eccetto `actorbase::actorsystem::storekeeper::messages::BecomeNinja` e `actorbase::actorsystem::storekeeper::messages::Update`.



#### 4.18.3.8 `actorbase::actorsystem::storekeeper::messages::BecomeNinja`

##### 4.18.3.8.1 Descrizione

Messaggio che richiede che l'attore<sub>G</sub> di tipo Storekeeper<sub>G</sub> diventi di tipo Ninja<sub>G</sub>.

##### 4.18.3.8.2 Utilizzo

Quando `actorbase::actorsystem::storekeeper::Storekeeper` riceve questo messaggio cambia il proprio contesto diventando un attore di tipo Ninja<sub>G</sub>. Questo comporta l'accesso ai messaggi `actorbase::actorsystem::storekeeper::messages::BecomeNinja`, `actorbase::actorsystem::storekeeper::messages::Update`.

#### 4.18.3.9 `actorbase::actorsystem::storekeeper::messages::Persist`

##### 4.18.3.9.1 Descrizione

Messaggio che richiede che il salvataggio su disco dei dati contenuti.

##### 4.18.3.9.2 Utilizzo

Quando lo `Storekeeper` riceve questo messaggio si occupa di mandare un messaggio al `Warehouseman` per il salvataggio dei propri dati su disco.



#### 4.19 actorbase::actorsystem::manager

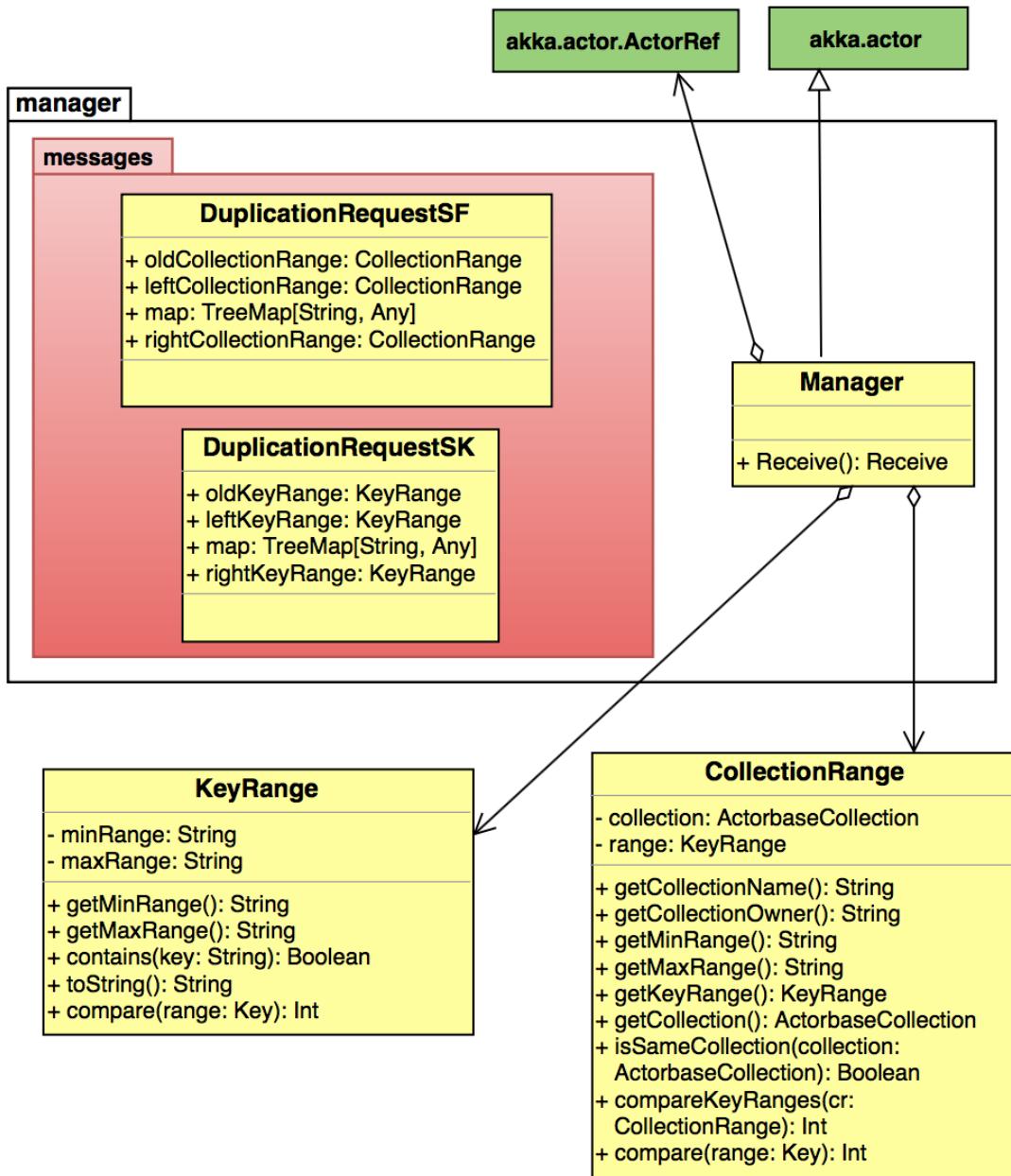


Figura 21: Manager, visione generale del package



#### 4.19.1 Descrizione

Package<sub>G</sub> che rappresenta l'attore<sub>G</sub> di tipo Manager<sub>G</sub>.

#### 4.19.2 Package contenuti

- `actorbase::actorsystem::manager::messages.`

#### 4.19.3 Interazioni con altre componenti

- `actorbase::actorsystem::utils.`

#### 4.19.4 Classi

##### 4.19.4.1 `actorbase::actorsystem::manager::Manager`

###### 4.19.4.1.1 Descrizione

Classe che rappresenta un attore<sub>G</sub> di tipo Manager<sub>G</sub>.

###### 4.19.4.1.2 Utilizzo

Questa classe viene utilizzata per gestire le richieste di divisione di attori di tipo Storekeeper e Storefinder.

###### 4.19.4.1.3 Classi ereditate

- `akka::actor::Actor.`

###### 4.19.4.1.4 Interazioni con altre classi

- `actorbase::actorsystem::utils::CollectionRange;`
- `actorbase::actorsystem::utils::KeyRange.`



## 4.20 actorbase::actorsystem::manager::messages

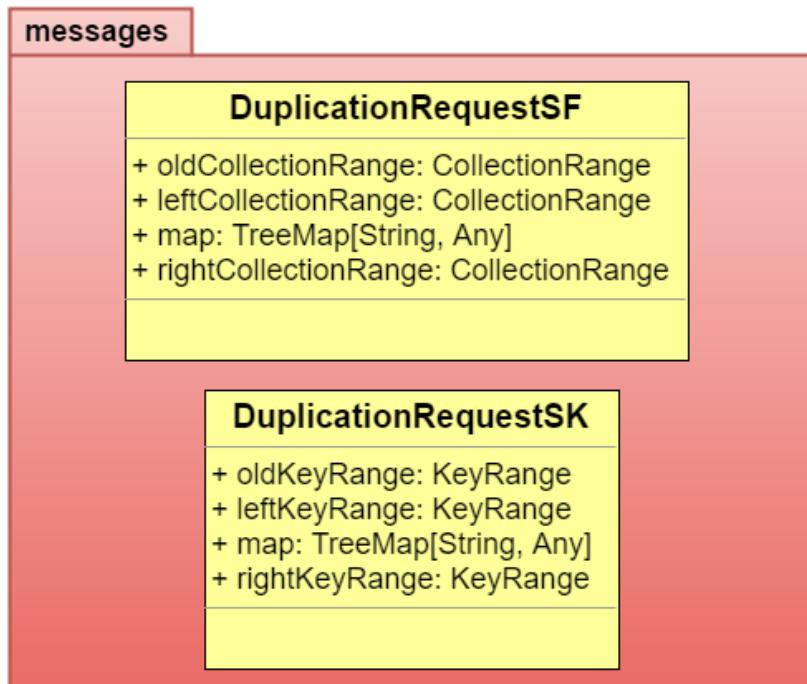


Figura 22: Actorsystem: package messages del manager

### 4.20.0.0.1 Descrizione

Package<sub>G</sub> che contiene tutti i messaggi che possono essere ricevuti dall'attore<sub>G</sub> di tipo [actorbase::actorsystem::manager::Manager](#)

#### 4.20.0.1 actorbase::actorsystem::manager::messages::DuplicationRequestSK

##### 4.20.0.1.1 Descrizione

Messaggio che richiede la divisione a metà dell'attore<sub>G</sub> di tipo [Storekeeper](#).

##### 4.20.0.1.2 Utilizzo

Quando [Manager](#) riceve questo messaggio provvederà a dividere a metà l'attore<sub>G</sub> di tipo [Storekeeper](#) che dovrà essere diviso.

Manderà inoltre un messaggio [DuplicateSKNotify](#) all'attore<sub>G</sub> di tipo [Storefinder](#) per aggiornare i suoi riferimenti.



#### 4.20.0.1.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::CollectionRange`;
- `actorbase::actorsystem::utils::KeyRange`.

#### 4.20.0.2 `actorbase::actorsystem::manager::messages::DuplicationRequestSF`

##### 4.20.0.2.1 Descrizione

Messaggio che richiede la divisione a metà dell'attore<sub>G</sub> di tipo Storefinder<sub>G</sub>.

##### 4.20.0.2.2 Utilizzo

Quando `actorbase::actorsystem::manager::Manager` riceve questo messaggio provvederà a dividere a metà l'attore<sub>G</sub> di tipo `actorbase::actorsystem::storefinder::Storefinder` che dovrà essere diviso. Manderà inoltre un messaggio all'attore<sub>G</sub> di tipo `actorbase::actorsystem::main::Main` per aggiornare i suoi riferimenti.

#### 4.20.0.2.3 Interazioni con altre classi

- `actorbase::actorsystem::utils::CollectionRange`;
- `actorbase::actorsystem::utils::KeyRange`.



## 4.21 actorbase::driver

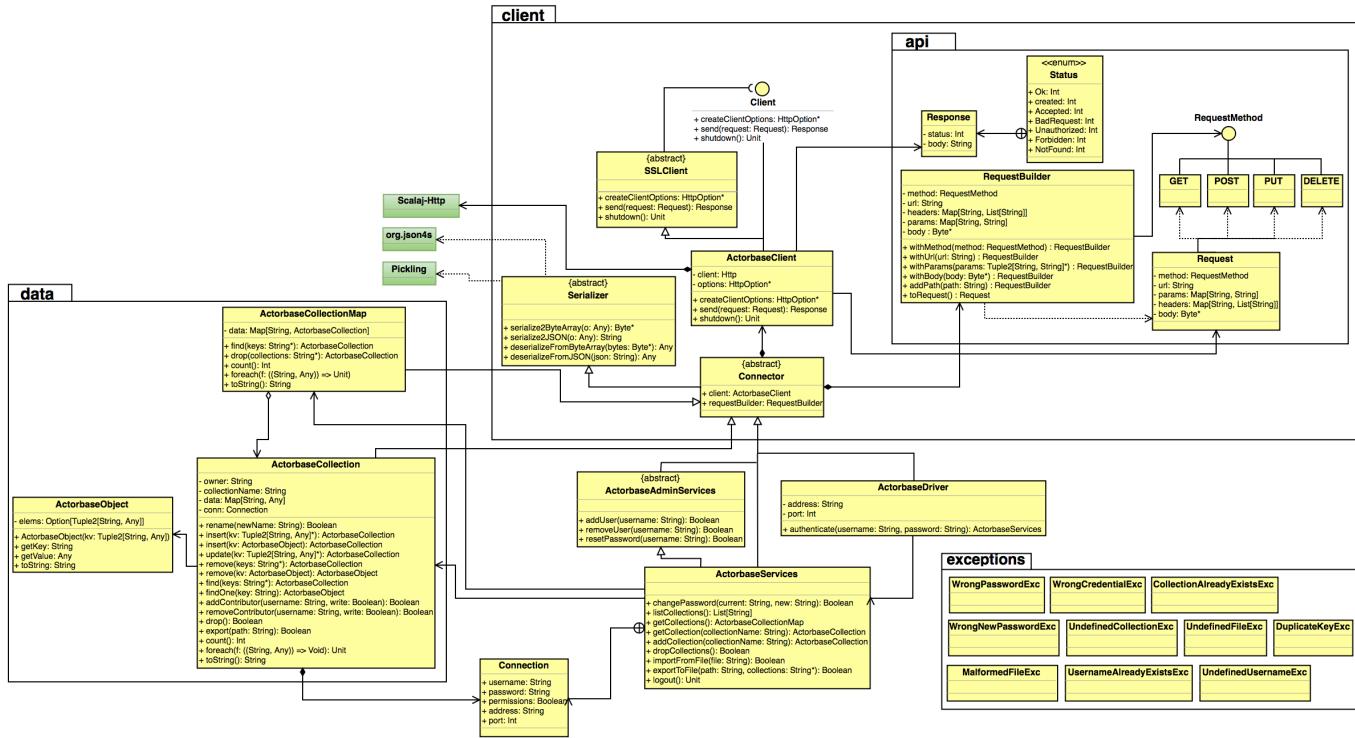


Figura 23: CLI:architettura MVC variante push model

### 4.21.1 Descrizione

Package<sub>G</sub> per la componente driver<sub>G</sub> da utilizzare in un programma Scala<sub>G</sub>, rappresenta l'interfaccia contenente i metodi designati all'utilizzo finale in un programma Scala<sub>G</sub>.

### 4.21.2 Interazioni con altre componenti

- actorbase::cli;
- actorbase::actorsystem.

### 4.21.3 Package contenuti

- actorbase::driver::client;



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

#### 4.21.4 Classi

##### 4.21.4.1 ActorbaseDriver

###### 4.21.4.1.1 Descrizione

Classe principale della componente  $\text{driver}_G$ , rappresenta l'interfaccia di connessione e autenticazione con il server $_G$ .

###### 4.21.4.1.2 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.

###### 4.21.4.1.3 Eredita

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

###### 4.21.4.1.4 Interazioni con le altre classi

- actorbase::driver::ActorbaseServices

##### 4.21.4.2 ActorbaseServices

###### 4.21.4.2.1 Descrizione

Classe principale della componente  $\text{driver}_G$ , rappresenta l'interfaccia contenente i metodi designati all'utilizzo di **Actorbase** in un programma Scala $_G$ .

###### 4.21.4.2.2 Utilizzo

Espone i metodi di interrogazione, ricerca e gestione dei dati remoti all'interno del database $_G$ .

###### 4.21.4.2.3 Eredita

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



#### 4.21.4.2.4 Interazioni con le altre classi

- actorbase::driver::client::ActorbaseCollection;
- actorbase::driver::client::ActorbaseCollectionMap.

#### 4.21.4.3 ActorbaseAdminServices

##### 4.21.4.3.1 Descrizione

Classe principale della componente  $\text{driver}_G$ , rappresenta l'interfaccia contenente i metodi designati all'utilizzo di **Actorbase** in un programma Scala $_G$ .

##### 4.21.4.3.2 Utilizzo

Aggiunge funzionalità amministrative all'oggetto [ActorbaseServices](#), ossia operazioni di gestione degli utenti.

##### 4.21.4.3.3 Eredità

- actorbase::driver::client::ActorbaseServices.

#### 4.21.4.3.4 Interazioni con le altre classi

- actorbase::driver::client::ActorbaseCollection;
- actorbase::driver::client::ActorbaseCollectionMap.

#### 4.21.4.4 Connection

##### 4.21.4.4.1 Descrizione

Classe interna all'oggetto [ActorbaseServices](#) contenente i parametri di connessione al server $_G$  e le credenziali del profilo **Actorbase**.

##### 4.21.4.4.2 Utilizzo

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

#### 4.21.4.4.3 Interazioni con le altre classi

- actorbase::driver::client::data::ActorbaseCollection.



## 4.22 actorbase::driver::client

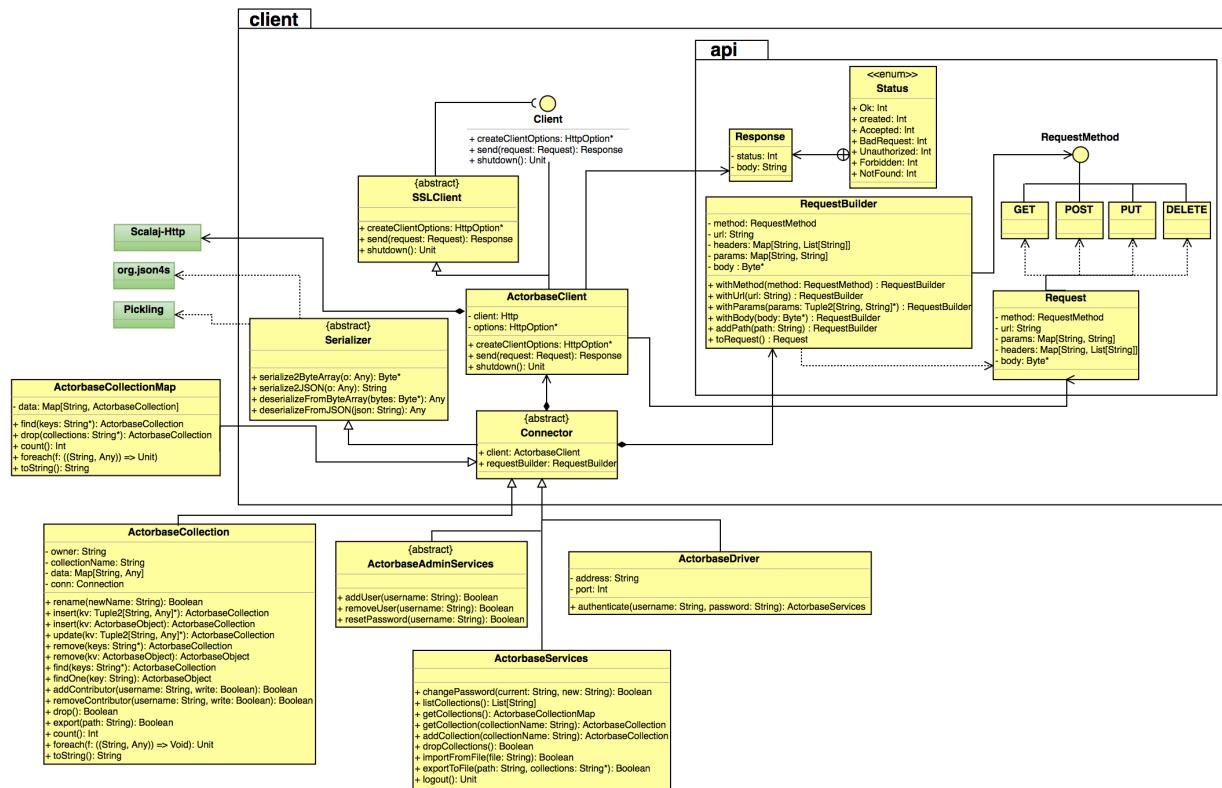


Figura 24: Driver: Packager client

### 4.22.1 Descrizione

Package<sub>G</sub> che si occupa di comunicare sia con la componente cli<sub>G</sub> che con la componente server<sub>G</sub> del sistema.

### 4.22.2 Interazioni con altre componenti

- actorbase::cli::models;
- actorbase::actorsystem::clientactor.



### 4.22.3 Classi

#### 4.22.3.1 Client

##### 4.22.3.1.1 Descrizione

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

##### 4.22.3.1.2 Utilizzo

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

#### 4.22.3.2 SSLClient (abstract)

##### 4.22.3.2.1 Descrizione

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

##### 4.22.3.2.2 Utilizzo

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

##### 4.22.3.2.3 Realizza

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

#### 4.22.3.3 actorbase::driver::client::ActorbaseClient

##### 4.22.3.3.1 Descrizione

Classe che rappresenta il client utilizzato per la connessione con la componente server<sub>G</sub> del sistema.

##### 4.22.3.3.2 Utilizzo

Viene utilizzata dalla componente Connector per fornire la comunicazione con la componente server<sub>G</sub> del sistema alle classi che lo necessitano.



#### 4.22.3.3.3 Realizza

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

#### 4.22.3.3.4 Eredita

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

#### 4.22.3.4 Serializer (abstract)

##### 4.22.3.4.1 Descrizione

Classe astratta atta a fornire metodi di serializzazione e deserializzazione nelle varianti  $\text{JSON}_G$  e  $\text{array}_G$  di  $\text{byte}_G$ .

##### 4.22.3.4.2 Utilizzo

Viene utilizzata mediante estensione dalle classi che necessitano di operazioni di serializzazione e deserializzazione in formato  $\text{JSON}_G$  o  $\text{array}_G$  di  $\text{byte}$  o entrambi.

##### 4.22.3.4.3 Ereditata da

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

#### 4.22.3.5 actorbase::driver::client::Connector (abstract)

##### 4.22.3.5.1 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.

##### 4.22.3.5.2 Utilizzo

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

##### 4.22.3.5.3 Eredita

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



## 4.23 actorbase::driver::client::api

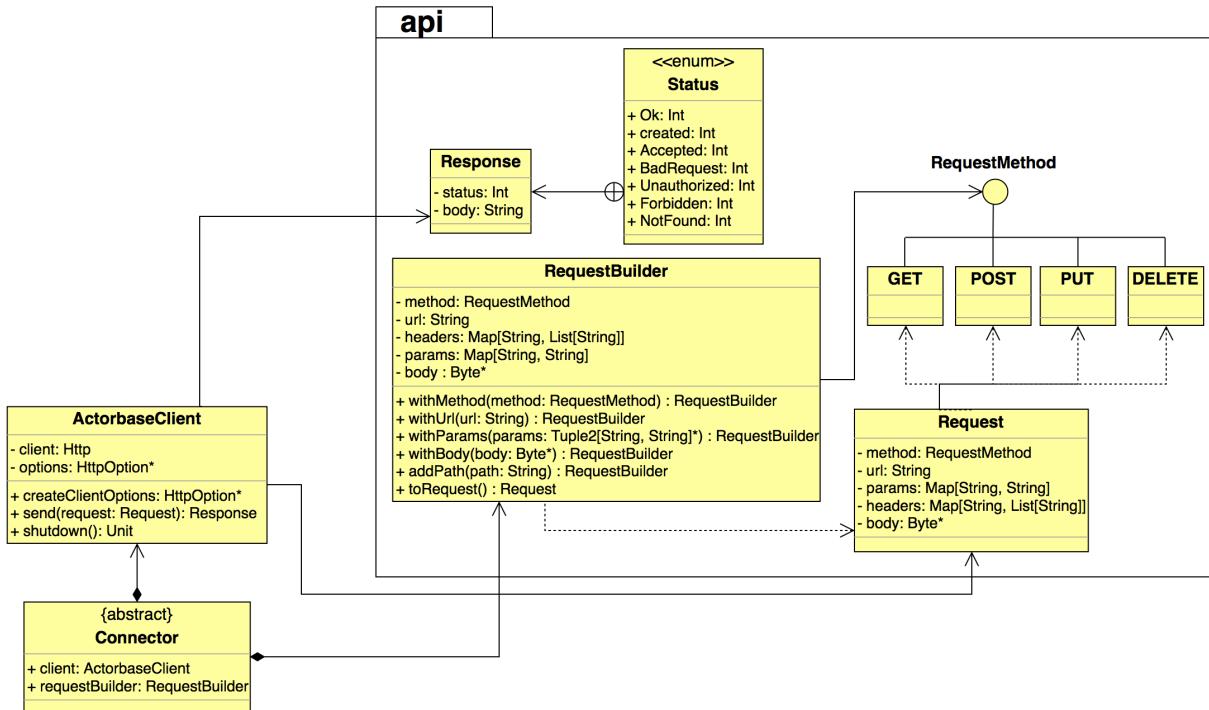


Figura 25: Driver: Package api e interazioni con package client

### 4.23.1 Descrizione

Package<sub>G</sub> che offre le strutture dati necessarie per la comunicazione con il server<sub>G</sub>, con la cli<sub>G</sub> o eventuali programmi esterni. Si occupa inoltre della gestione delle richieste REST<sub>G</sub> da inviare utilizzando il protocollo HTTP<sub>G</sub>.

### 4.23.2 Interazioni con altre componenti

- actorbase::driver::client;



### 4.23.3 Classi

#### 4.23.3.1 actorbase::driver::client::api::RequestMethod (abstract)

##### 4.23.3.1.1 Descrizione

Classe astratta che rappresenta una generica richiesta  $\text{HTTP}_G$ , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>.

##### 4.23.3.1.2 Utilizzo

Viene utilizzata come classe padre per l'implementazione delle richieste  $\text{HTTP}_G$  da utilizzare nella comunicazione con il server.

##### 4.23.3.1.3 Interazioni con le altre classi

- `actorbase::driver::client::api::RequestBuilder`.

#### 4.23.3.2 actorbase::driver::client::api::GET (abstract)

##### 4.23.3.2.1 Descrizione

Classe astratta che rappresenta una generica richiesta  $\text{HTTP}_G$  di tipo GET<sub>G</sub>, implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>.

##### 4.23.3.2.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste  $\text{HTTP}_G$  di tipo GET<sub>G</sub> da utilizzare nella comunicazione con il server.

##### 4.23.3.2.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Request`.

##### 4.23.3.2.4 Eredità

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



#### 4.23.3.3 **actorbase::driver::client::api::POST (abstract)**

##### 4.23.3.3.1 Descrizione

Classe astratta che rappresenta una generica richiesta  $\text{HTTP}_G$  di tipo  $\text{POST}_G$ , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>.

##### 4.23.3.3.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste  $\text{HTTP}_G$  di tipo  $\text{POST}_G$  da utilizzare nella comunicazione con il server.

##### 4.23.3.3.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Request`.

##### 4.23.3.4 Eredità

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

#### 4.23.3.4 **actorbase::driver::client::api::PUT (abstract)**

##### 4.23.3.4.1 Descrizione

Classe astratta che rappresenta una generica richiesta  $\text{HTTP}_G$  di tipo  $\text{PUT}_G$ , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>.

##### 4.23.3.4.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste  $\text{HTTP}_G$  di tipo  $\text{PUT}_G$  da utilizzare nella comunicazione con il server.

##### 4.23.3.4.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Request`.

##### 4.23.3.4.4 Eredità

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



#### 4.23.3.5 actorbase::driver::client::api::DELETE (abstract)

##### 4.23.3.5.1 Descrizione

Classe astratta che rappresenta una generica richiesta  $\text{HTTP}_G$  di tipo  $\text{DELETE}_G$ , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>.

##### 4.23.3.5.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste  $\text{HTTP}_G$  di tipo  $\text{DELETE}_G$  da utilizzare nella comunicazione con il server.

##### 4.23.3.5.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Request`.

##### 4.23.3.5.4 Eredità

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

#### 4.23.3.6 actorbase::driver::client::api::Request

##### 4.23.3.6.1 Descrizione

Classe che rappresenta una generica richiesta  $\text{HTTP}_G$ , implementata utilizzando la specifica funzionalità *case class* di Scala, in modo da poter usufruirne mediante costrutti e algoritmi di pattern matching<sub>G</sub>, racchiude al suo interno i parametri di richiesta, quali il metodo, di tipo `RequestMethod`, l' $\text{URL}_G$  a cui inviare la richiesta, gli header<sub>G</sub>  $\text{HTTP}_G$  associati, i parametri e il payload<sub>G</sub>.

##### 4.23.3.6.2 Utilizzo

Viene utilizzata come classe per l'implementazione delle richieste  $\text{HTTP}_G$  di tipo  $\text{DELETE}_G$  da utilizzare nella comunicazione con il server. Viene creata mediante il design pattern `Builder`<sub>G</sub> nella classe `RequestBuilder`.

##### 4.23.3.6.3 Interazioni con le altre classi

- `actorbase::driver::client::api::RequestBuilder`.
- `actorbase::driver::client::ActorbaseClient`.



#### 4.23.3.7 actorbase::driver::client::api::Response

##### 4.23.3.7.1 Descrizione

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

##### 4.23.3.7.2 Utilizzo

Viene utilizzata come classe per l'incapsulamento delle risposte HTTP<sub>G</sub> da utilizzare nella comunicazione con il server.

##### 4.23.3.7.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Status`;
- `actorbase::driver::client::ActorbaseClient`.

#### 4.23.3.8 Status

##### 4.23.3.8.1 Descrizione

Enum interna a `hyperref[sec:actorbase::driver::client::api::Response]Response`, rappresenta il codice di ritorno della richiesta HTTP<sub>G</sub> inviata al server.

##### 4.23.3.8.2 Utilizzo

Viene utilizzata come classe per l'incapsulamento dei codici di ritorno HTTP<sub>G</sub> da utilizzare nella comunicazione con il server.

##### 4.23.3.8.3 Interazioni con le altre classi

- `actorbase::driver::client::api::Response`.

#### 4.23.3.9 RequestBuilder

##### 4.23.3.9.1 Descrizione

Rappresenta una classe builder per la creazione di richieste HTTP<sub>G</sub>, racchiude al suo interno i parametri di richiesta, quali il metodo, di tipo `RequestMethod`, l'URL<sub>G</sub> a cui inviare la richiesta, gli header<sub>G</sub> HTTP<sub>G</sub> associati, i parametri e il payload<sub>G</sub> e i metodi `with` per aggiungere attributi alla richiesta di tipo `Request` da generare.



#### 4.23.3.9.2 Utilizzo

Viene utilizzata come classe buider per la costruzione delle richieste HTTP<sub>G</sub> di tipo **Request** da utilizzare nella comunicazione con il server.

#### 4.23.3.9.3 Interazioni con le altre classi

- actorbase::driver::client::api::Request;
- actorbase::driver::client::Connector.

### 4.24 actorbase::driver::client::api::data

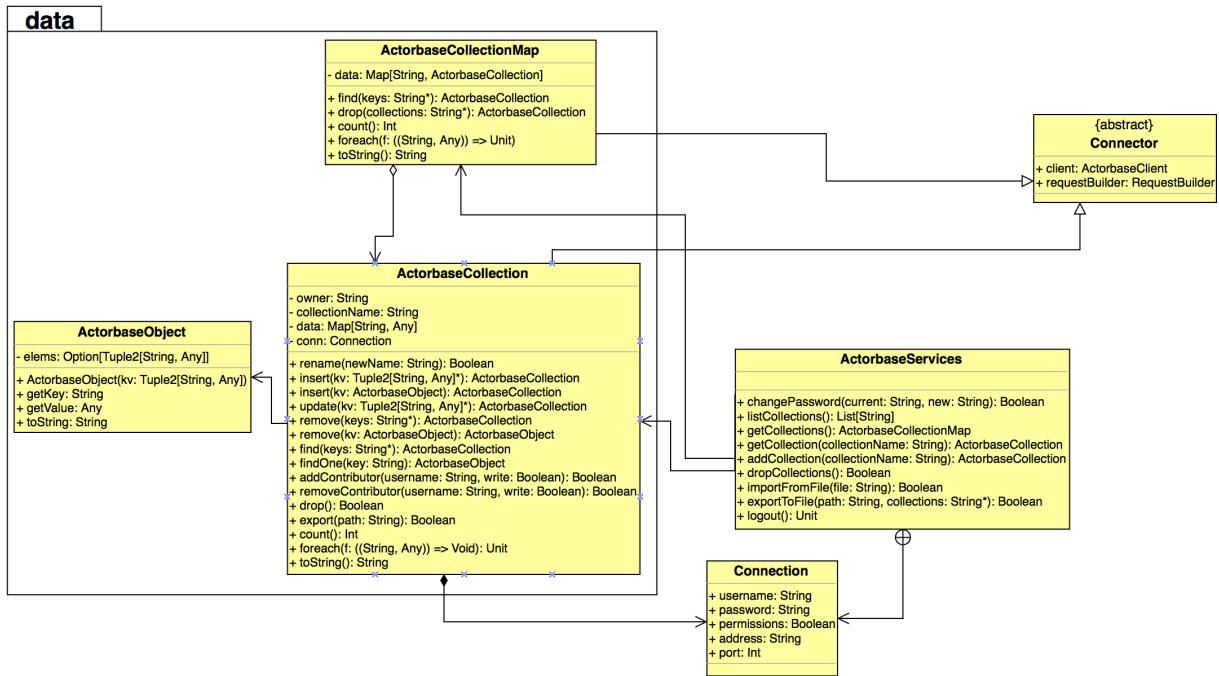


Figura 26: Driver: Package data e interazioni con package padre

#### 4.24.1 Descrizione

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



#### 4.24.2 Interazioni con altre componenti

- `actorbase::driver::client`.

#### 4.24.3 Classi

##### 4.24.3.1 ActorbaseObject

###### 4.24.3.1.1 Descrizione

Interfaccia che rappresenta un oggetto generico contenente una coppia chiave-valore ricevuto dal server<sub>G</sub>.

###### 4.24.3.1.2 Utilizzo

Classe utilizzata per agevolare le ricerche specifiche all'interno delle collezioni<sub>G</sub> del database<sub>G</sub>.

###### 4.24.3.1.3 Interazioni con le altre classi

- `actorbase::driver::client::data::ActorbaseCollection`.

##### 4.24.3.2 actorbase::driver::data::ActorbaseCollection

###### 4.24.3.2.1 Descrizione

Classe che rappresenta una singola collezione<sub>G</sub> del database<sub>G</sub>.

###### 4.24.3.2.2 Utilizzo

Questa classe viene utilizzata per la rappresentazione di una singola collezione<sub>G</sub> 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<sub>G</sub>. Offre metodi di utilità e navigabilità dei contenuti.

###### 4.24.3.2.3 Eredità

- `actorbase::driver::data::Serializer`.

###### 4.24.3.2.4 Interazioni con le altre classi

- `actorbase::driver::client::data::ActorbaseObject`.
- `actorbase::driver::client::data::ActorbaseCollectionMap`.



#### 4.24.3.3 actorbase::driver::data::ActorbaseCollectionMap

##### 4.24.3.3.1 Descrizione

Classe che rappresenta una collezione<sub>G</sub> di collezione<sub>G</sub> del sistema.

##### 4.24.3.3.2 Utilizzo

Questa classe viene utilizzata per la rappresentazione dell'aggregazione di collezioni<sub>G</sub>, è formata da una mappa di coppie nomecollezione - [ActorbaseCollection](#).

##### 4.24.3.3.3 Eredità

- actorbase::driver::data::Serializer.

##### 4.24.3.3.4 Interazioni con le altre classi

- actorbase::driver::client::ActorbaseDriver.
- actorbase::driver::client::data::ActorbaseCollection.

#### 4.25 actorbase::driver::exceptions

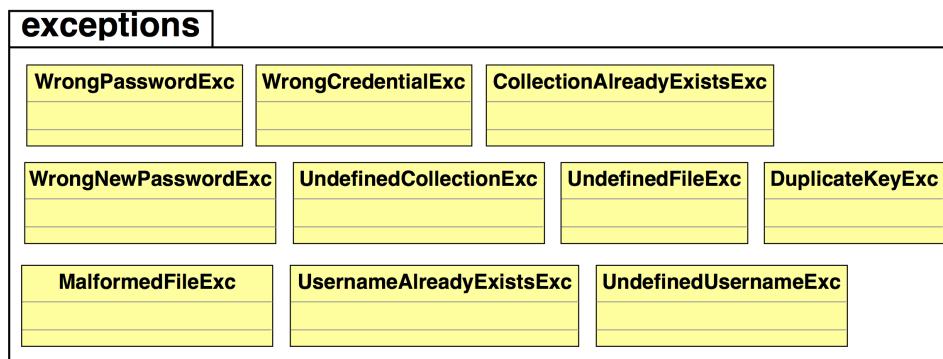


Figura 27: Driver: Package exceptions

##### 4.25.1 Descrizione

Package<sub>G</sub> che contiene le possibili eccezioni che si possono verificare nel driver<sub>G</sub>



## 4.25.2 Interazione con altre componenti

- `actorbase::driver::client;`

## 4.25.3 Classi

### 4.25.3.1 `actorbase::driver::exceptions::WrongCredentialExc`

#### 4.25.3.1.1 Descrizione

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

#### 4.25.3.1.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire errori nelle credenziali di accesso.

#### 4.25.3.1.3 Interazioni con altre classi

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

### 4.25.3.2 `actorbase::driver::exceptions::WrongPasswordExc`

#### 4.25.3.2.1 Descrizione

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

#### 4.25.3.2.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire il caso in cui alla modifica della propria password, quando dovrei inserire la vecchia password questa non corrisponda.

#### 4.25.3.2.3 Interazioni con altre classi

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

### 4.25.3.3 `actorbase::driver::exceptions::WrongNewPasswordExc`

#### 4.25.3.3.1 Descrizione

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



#### 4.25.3.3.2 Utilizzo

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

#### 4.25.3.3 Interazioni con altre classi

- actorbase::driver::client::ActorbaseClient.

#### 4.25.3.4 actorbase::driver::exceptions::CollectionAlreadyExistsExc

##### 4.25.3.4.1 Descrizione

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

##### 4.25.3.4.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire il caso in cui alla creazione di una nuova collezione, il nome della nuova collezione corrisponda al nome di una collezione già esistente.

##### 4.25.3.4.3 Interazioni con altre classi

- actorbase::driver::client::ActorbaseClient.

#### 4.25.3.5 actorbase::driver::exceptions::UndefinedCollectionExc

##### 4.25.3.5.1 Descrizione

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

##### 4.25.3.5.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire il caso in cui alla creazione di una nuova collezione, venga omesso il nome di quest'ultima.

##### 4.25.3.5.3 Interazioni con altre classi

- actorbase::driver::client::ActorbaseClient.



#### 4.25.3.6 **actorbase::driver::exceptions::UndefinedUsernameExc**

##### 4.25.3.6.1 Descrizione

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

##### 4.25.3.6.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire il caso in cui alla registrazione di un nuovo utente, venga omesso l'username.

##### 4.25.3.6.3 Interazioni con altre classi

- [actorbase::driver::client::ActorbaseClient](#).

#### 4.25.3.7 **actorbase::driver::exceptions::UsernameAlreadyExistsExc**

##### 4.25.3.7.1 Descrizione

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

##### 4.25.3.7.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> 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.

##### 4.25.3.7.3 Interazioni con altre classi

- [actorbase::driver::client::ActorbaseClient](#).

#### 4.25.3.8 **actorbase::driver::exceptions::DuplicateKeyExc**

##### 4.25.3.8.1 Descrizione

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

##### 4.25.3.8.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> per gestire il caso in cui all'inserimento di una nuova chiave, questa corrisponda ad una chiave già esistente nella stessa collezione.



#### 4.25.3.8.3 Interazioni con altre classi

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

#### 4.25.3.9 `actorbase::driver::exceptions::UndefinedFileExc`

##### 4.25.3.9.1 Descrizione

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

##### 4.25.3.9.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> 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.

#### 4.25.3.9.3 Interazioni con altre classi

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

#### 4.25.3.10 `actorbase::driver::exceptions::MalformedFileExc`

##### 4.25.3.10.1 Descrizione

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

##### 4.25.3.10.2 Utilizzo

Questa classe viene utilizzata dal driver<sub>G</sub> 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.

#### 4.25.3.10.3 Interazioni con altre classi

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

## 4.26 actorbase::cli

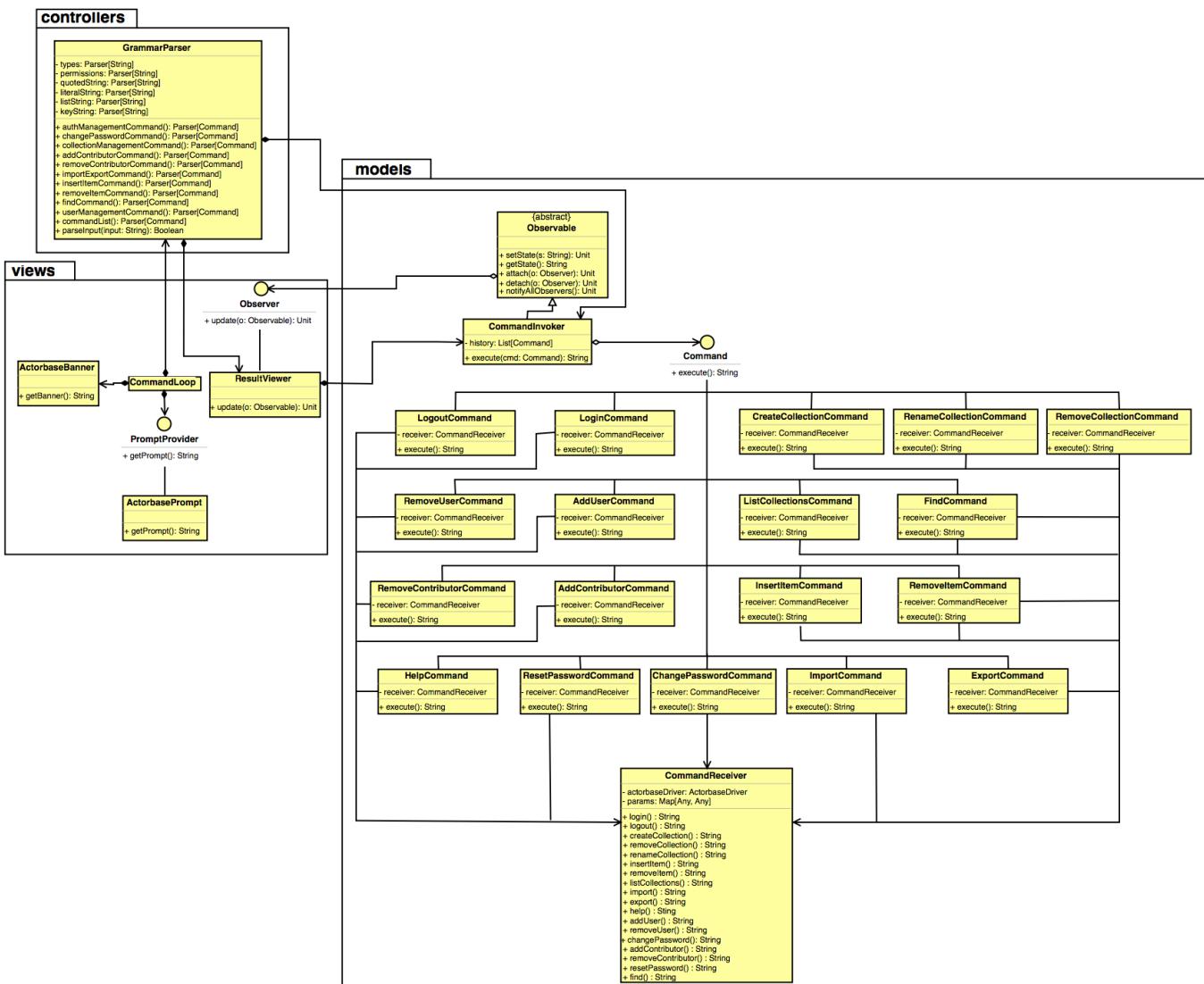


Figura 28: CLI, architettura MVC variante push model

#### **4.26.1 Descrizione**

Package<sub>G</sub> per la parte di client<sub>G</sub> rappresentata dalla cli<sub>G</sub>. Questa componente viene rappresentata usando il design pattern<sub>G</sub> MVC<sub>G</sub> in versione push model<sub>G</sub>.



#### 4.26.2 Interazioni con altre componenti

- `actorbase::driver`.

#### 4.26.3 Package contenuti

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



## 4.27 actorbase::cli::views

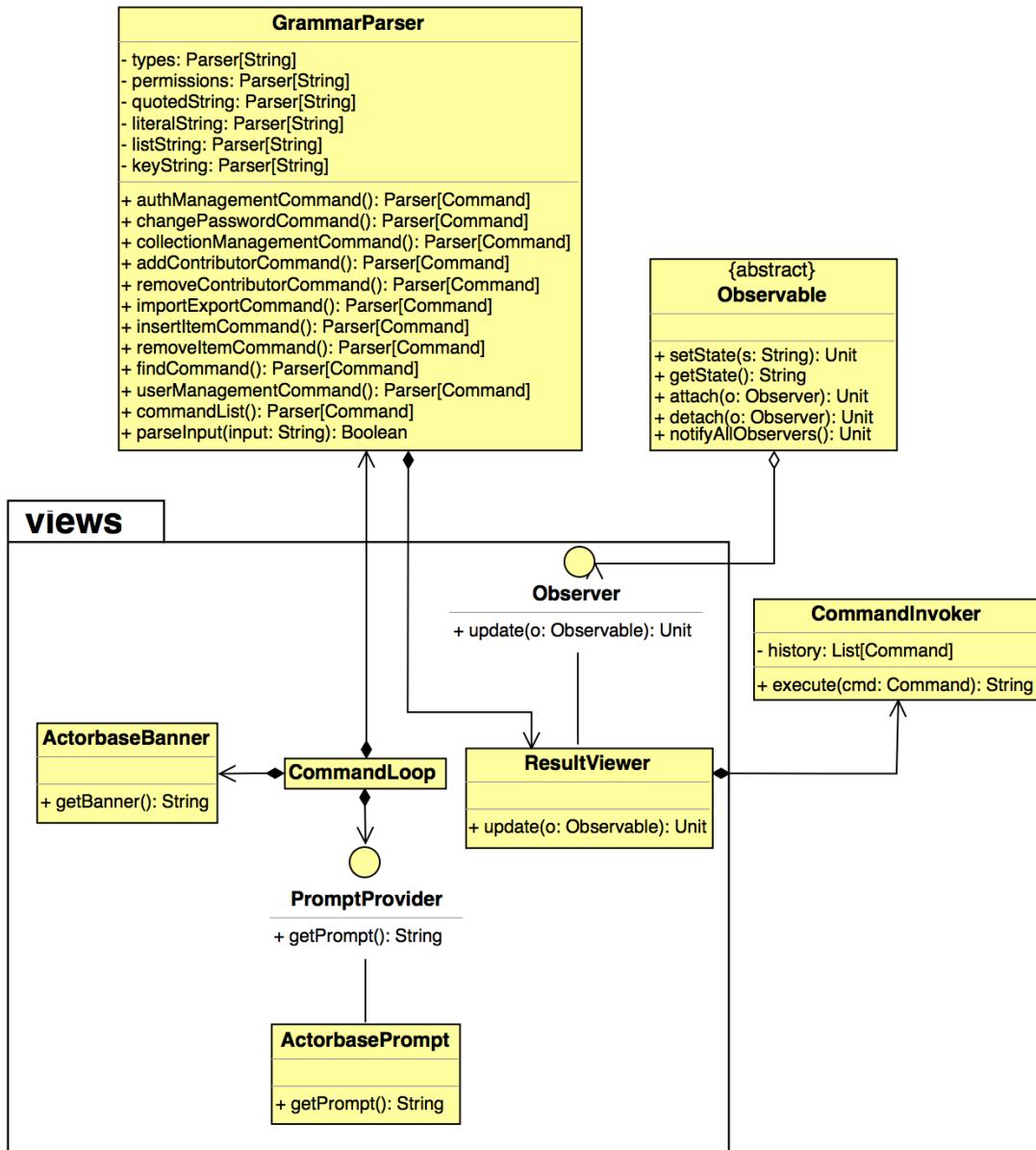


Figura 29: CLI, views package, interazioni con controllers e models



#### 4.27.1 Descrizione

Package<sub>G</sub> per la parte view<sub>G</sub> della cli<sub>G</sub>. Questa componente gestisce l'output e offre l'interfaccia tramite cui l'utente può inserire i comandi.

#### 4.27.2 Interazioni con altre componenti

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

#### 4.27.3 Classi

##### 4.27.3.1 actorbase::cli::views::CommandLoop

###### 4.27.3.1.1 Descrizione

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

###### 4.27.3.1.2 Utilizzo

Viene utilizzata per la lettura dei comandi inseriti dall'utente.

###### 4.27.3.1.3 Interazioni con altre classi

- actorbase::cli::views::PromptProvider;
- actorbase::cli::views::ActorbaseBanner;
- actorbase::cli::views::ResultView;
- actorbase::cli::controllers::GrammarParser.

##### 4.27.3.2 actorbase::cli::views::ActorbaseBanner

###### 4.27.3.2.1 Descrizione

Classe per la creazione del banner<sub>G</sub> da mostrare all'avvio della CLI<sub>G</sub>.

###### 4.27.3.2.2 Utilizzo

Viene utilizzata per la creazione di un banner<sub>G</sub> di presentazione all'avvio della CLI<sub>G</sub>.



#### 4.27.3.2.3 Interazioni con altre classi

- `actorbase::cli::views::CommandLoop`.

#### 4.27.3.3 `actorbase::cli::views::PromptProvider`

##### 4.27.3.3.1 Descrizione

Interfaccia per la creazione del prompt<sub>G</sub> dei comandi.

##### 4.27.3.3.2 Utilizzo

Offre un'interfaccia per la creazione di un prompt<sub>G</sub> generico.

#### 4.27.3.3.3 Interazioni con altre classi

- `actorbase::cli::views::CommandLoop`.

#### 4.27.3.4 `actorbase::cli::views::ActorbasePrompt`

##### 4.27.3.4.1 Descrizione

Classe che implementa l'interfaccia `actorbase::cli::views::PromptProvider` per la creazione di un prompt<sub>G</sub> dei comandi.

##### 4.27.3.4.2 Utilizzo

Viene utilizzata per la creazione di un prompt<sub>G</sub> per l'inserimento dei comandi e la visualizzazione delle informazioni relative alla connessione effettuata.

##### 4.27.3.4.3 Classi ereditate

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

#### 4.27.3.5 `actorbase::cli::views::Observer`

##### 4.27.3.5.1 Descrizione

Interfaccia per l'implementazione del design pattern<sub>G</sub> Observer<sub>G</sub> che permette di aggiornare la view<sub>G</sub> ad ogni risposta della componente driver<sub>G</sub>.



#### 4.27.3.5.2 Utilizzo

Offre un'interfaccia per permettere di aggiornare la  $\text{view}_G$  ad ogni risposta ricevuta dal  $\text{driver}_G$ .

#### 4.27.3.5.3 Interazioni con altre classi

- `actorbase::cli::models::Observable`.

#### 4.27.3.6 `actorbase::cli::views::ResultView`

##### 4.27.3.6.1 Descrizione

Classe per la gestione e la formattazione in output dei risultati ottenuti dall'esecuzione del comando ricevuto in input tramite il design pattern<sub>G</sub> Observer<sub>G</sub>.

##### 4.27.3.6.2 Utilizzo

Viene utilizzata per la gestione e la formattazione in output dei risultati ottenuti dal server mediante l'inserimento dei comandi tramite il design pattern<sub>G</sub> Observer<sub>G</sub>.

##### 4.27.3.6.3 Classi ereditate

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

##### 4.27.3.6.4 Interazioni con altre classi

- `actorbase::cli::views::CommandLoop`.



## 4.28 actorbase::cli::controllers

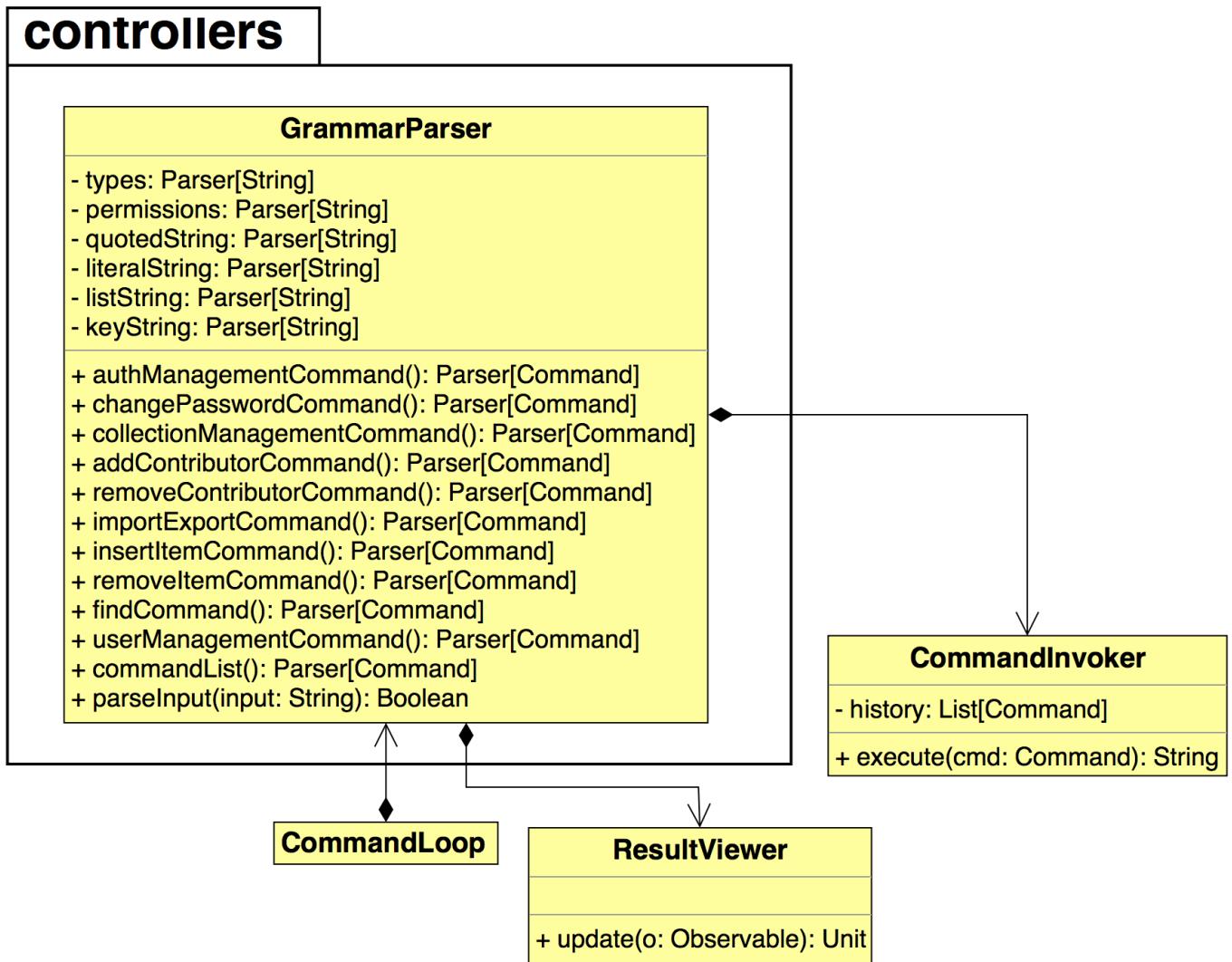


Figura 30: CLI: package controllers e interazioni con package models e views

### 4.28.1 Descrizione

Package<sub>G</sub> per la parte controller<sub>G</sub> della cli<sub>G</sub>. Questa componente si occupa di passare il comando ricevuto dalla view<sub>G</sub> al model<sub>G</sub>.



#### 4.28.2 Interazioni con altre componenti

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

#### 4.28.3 Classi

##### 4.28.3.1 `actorbase::cli::controllers::GrammarParser`

###### 4.28.3.1.1 Descrizione

Classe che si occupa di effettuare il parsing<sub>G</sub> del comando ricevuto dalla componente view<sub>G</sub> e di chiamare la componente model<sub>G</sub> con i parametri opportuni.

###### 4.28.3.1.2 Utilizzo

Viene utilizzata per effettuare il parsing<sub>G</sub> del comando ricevuto da `actorbase::cli::views::CommandLoop` e, in seguito, chiamare `actorbase::cli::models::CommandInvoker` con i parametri corretti per l'esecuzione del comando.

###### 4.28.3.1.3 Interazioni con altre classi

- `actorbase::cli::views::CommandLoop;`
- `actorbase::cli::models::CommandInvoker.`



## 4.29 actorbase::cli::models

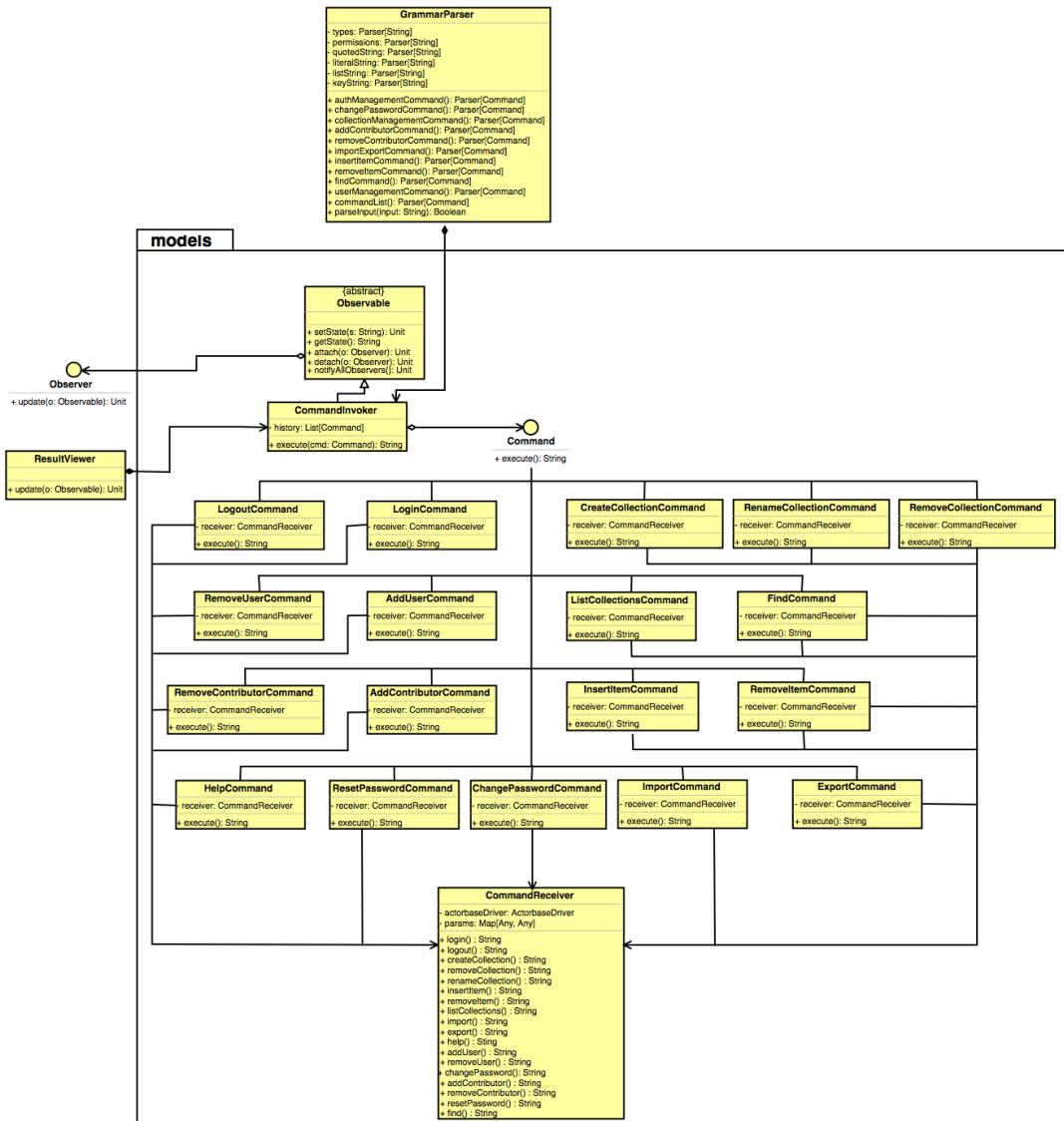


Figura 31: CLI: package models e interazioni con controllers e views



#### 4.29.1 Descrizione

Package<sub>G</sub> per la parte model<sub>G</sub> della cli<sub>G</sub>. Questa componente si occupa della comunicazione con la componente driver<sub>G</sub> e con la componente view<sub>G</sub> tramite il push model<sub>G</sub> del design pattern<sub>G</sub> observer<sub>G</sub>. Per la gestione dei comandi si è scelto di utilizzare il command pattern<sub>G</sub>.

#### 4.29.2 Classi

##### 4.29.2.1 actorbase::cli::models::Observable

###### 4.29.2.1.1 Descrizione

Questa classe rappresenta l'oggetto osservato nell'ambito del design pattern<sub>G</sub> observer<sub>G</sub>.

###### 4.29.2.1.2 Utilizzo

Viene utilizzata per notificare la classe actorbase::cli::views::Observer quando avviene una modifica al proprio stato, ossia quando il driver<sub>G</sub> ha ritornato un risultato.

###### 4.29.2.1.3 Interazioni con altre classi

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

##### 4.29.2.2 actorbase::cli::models::CommandInvoker

###### 4.29.2.2.1 Descrizione

Questa classe rappresenta l'Invoker<sub>G</sub> del command pattern<sub>G</sub>. Essa si occupa di eseguire il comando ricevuto dalla componente controller<sub>G</sub>.

###### 4.29.2.2.2 Utilizzo

Viene utilizzata per eseguire il comando ricevuto da actorbase::cli::controllers::GrammarParser chiamando il metodo opportuno che implementa l'interfaccia actorbase::cli::models::Command.

###### 4.29.2.2.3 Classi ereditate

- actorbase::cli::models::Observable.



#### 4.29.2.2.4 Interazioni con altre classi

- [actorbase::cli::controllers::GrammarParser](#);
- [actorbase::cli::models::Command](#).

#### 4.29.2.3 actorbase::cli::models::Command

##### 4.29.2.3.1 Descrizione

Interfaccia generica che rappresenta un comando.

##### 4.29.2.3.2 Utilizzo

Viene utilizzata per offrire un'interfaccia comune a tutti i comandi.

Ha un riferimento a [actorbase::cli::models::CommandReceiver](#) per l'esecuzione del comando.

#### 4.29.2.3.3 Interazioni con altre classi

- [actorbase::cli::models::CommandInvoker](#);
- [actorbase::view::cli::models::CommandReceiver](#).

#### 4.29.2.4 actorbase::cli::models::FindCommand

##### 4.29.2.4.1 Descrizione

Classe per il comando di ricerca.

##### 4.29.2.4.2 Utilizzo

Viene utilizzata per chiamare il metodo di [actorbase::cli::models::CommandReceiver](#) per la ricerca con i parametri immessi dall'utente.

##### 4.29.2.4.3 Classi ereditate

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

#### 4.29.2.5 actorbase::cli::models::LogoutCommand

##### 4.29.2.5.1 Descrizione

Classe per il comando di logout.



#### 4.29.2.5.2 Utilizzo

Viene utilizzata per chiamare il metodo di [actorbase::cli::models::CommandReceiver](#) per il logout.

#### 4.29.2.5.3 Classi ereditate

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

### 4.29.2.6 actorbase::cli::models::ResetPasswordCommand

#### 4.29.2.6.1 Descrizione

Classe per il comando di reset della password.

#### 4.29.2.6.2 Utilizzo

Viene utilizzata per chiamare il metodo di [actorbase::cli::models::CommandReceiver](#) per il reset della password dell'utente specificato in input.

#### 4.29.2.6.3 Classi ereditate

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

### 4.29.2.7 actorbase::cli::models::ChangePasswordCommand

#### 4.29.2.7.1 Descrizione

Classe per il comando di modifica della password.

#### 4.29.2.7.2 Utilizzo

Viene utilizzata per chiamare il metodo di [actorbase::cli::models::CommandReceiver](#) per la modifica della password..

#### 4.29.2.7.3 Classi ereditate

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



#### 4.29.2.8 **actorbase::cli::models::ListCommand**

##### 4.29.2.8.1 Descrizione

Classe per il comando di visualizzazione dell'elenco dei nomi di tutte le collezioni presenti nel database<sub>G</sub>.

##### 4.29.2.8.2 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<sub>G</sub>.

##### 4.29.2.8.3 Classi ereditate

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

#### 4.29.2.9 **actorbase::cli::models::AddContributorCommand**

##### 4.29.2.9.1 Descrizione

Classe per il comando di aggiunta collaboratore a una collezione.

##### 4.29.2.9.2 Utilizzo

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

##### 4.29.2.9.3 Classi ereditate

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

#### 4.29.2.10 **actorbase::cli::models::LoginCommand**

##### 4.29.2.10.1 Descrizione

Classe per il comando di login.

##### 4.29.2.10.2 Utilizzo

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



#### 4.29.2.10.3 Classi ereditate

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

#### 4.29.2.11 `actorbase::cli::models::RemoveCollectionCommand`

##### 4.29.2.11.1 Descrizione

Classe per il comando di rimozione una collezione<sub>G</sub>.

##### 4.29.2.11.2 Utilizzo

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

#### 4.29.2.11.3 Classi ereditate

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

#### 4.29.2.12 `actorbase::cli::models::RemoveContributorCommand`

##### 4.29.2.12.1 Descrizione

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

##### 4.29.2.12.2 Utilizzo

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

#### 4.29.2.12.3 Classi ereditate

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

#### 4.29.2.13 `actorbase::cli::models::RenameCollectionCommand`

##### 4.29.2.13.1 Descrizione

Classe per il comando rinominazione di una collezione<sub>G</sub>.



#### 4.29.2.13.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per la rinominazione di una collezione<sub>G</sub> con i parametri specificati in input.

#### 4.29.2.13.3 Classi ereditate

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

### 4.29.2.14 actorbase::cli::models::AddUserCommand

#### 4.29.2.14.1 Descrizione

Classe per il comando di aggiunta di un utente.

#### 4.29.2.14.2 Utilizzo

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

#### 4.29.2.14.3 Classi ereditate

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

### 4.29.2.15 actorbase::cli::models::HelpCommand

#### 4.29.2.15.1 Descrizione

Classe per la richiesta di aiuto.

#### 4.29.2.15.2 Utilizzo

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

### 4.29.2.16 actorbase::cli::models::RemoveItemCommand

#### 4.29.2.16.1 Descrizione

Classe per il comando di rimozione di un item<sub>G</sub>.



#### 4.29.2.16.2 Utilizzo

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

#### 4.29.2.16.3 Classi ereditate

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

### 4.29.2.17 actorbase::cli::models::RemoveUserCommand

#### 4.29.2.17.1 Descrizione

Classe per il comando di rimozione di un utente.

#### 4.29.2.17.2 Utilizzo

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

#### 4.29.2.17.3 Classi ereditate

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

### 4.29.2.18 actorbase::cli::models::ImportCommand

#### 4.29.2.18.1 Descrizione

Classe per il comando di importazione da un file JSON<sub>G</sub>.

#### 4.29.2.18.2 Utilizzo

Viene utilizzata per chiamare il metodo di `actorbase::cli::models::CommandReceiver` per l'importazione da un file JSON<sub>G</sub> con i parametri immessi dall'utente.

#### 4.29.2.18.3 Classi ereditate

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



#### 4.29.2.19 actorbase::cli::models::InsertItemCommand

##### 4.29.2.19.1 Descrizione

Classe per il comando di inserimento di un item<sub>G</sub> in una collezione<sub>G</sub>.

##### 4.29.2.19.2 Utilizzo

Viene utilizzata per chiamare il metodo di actorbase::cli::models::CommandReceiver per l' inserimento di un item<sub>G</sub> in una collezione<sub>G</sub> con i parametri immessi dall'utente.

##### 4.29.2.19.3 Classi ereditate

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

#### 4.29.2.20 actorbase::cli::models::CreateCollectionCommand

##### 4.29.2.20.1 Descrizione

Classe per il comando di creazione di una nuova collezione<sub>G</sub>.

##### 4.29.2.20.2 Utilizzo

Viene utilizzata per chiamare il metodo di actorbase::cli::models::CommandReceiver per la creazione di una nuova collezione<sub>G</sub> con i parametri immessi dall'utente.

##### 4.29.2.20.3 Classi ereditate

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

#### 4.29.2.21 actorbase::cli::models::ExportCommand

##### 4.29.2.21.1 Descrizione

Classe per il comando di esportazione in file JSON<sub>G</sub>.

##### 4.29.2.21.2 Utilizzo

Viene utilizzata per chiamare il metodo di actorbase::cli::models::CommandReceiver per l'esportazione in file JSON<sub>G</sub> con i parametri immessi dall'utente.



#### 4.29.2.21.3 Classi ereditate

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

#### 4.29.2.22 actorbase::cli::models::CommandReceiver

##### 4.29.2.22.1 Descrizione

Classe contenente i metodi per richiamare la componente driver<sub>G</sub> per l'esecuzione dei comandi o per restituire l'aiuto se richiesto.

##### 4.29.2.22.2 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.

##### 4.29.2.22.3 Interazioni con altre classi

- [actorbase::cli::models::Command](#);
- [actorbase::driver::client::ActorbaseClient](#).



## 5 Diagrammi di Attività

In questa sezione, vengono illustrati i diagrammi di attività che descrivono l’interazione dell’utente con **Actor-base** tramite l’utilizzo della  $CLI_G$  ad esso associata. È stato disegnato un diagramma ad alto livello in cui sono rappresentate le operazioni possibili sul  $database_G$ . Esse vengono poi illustrate tramite dei sotto-diagrammi per poterne comprendere meglio il funzionamento e facilitarne la lettura.

### 5.1 Visione generale

L’utente, dopo aver aperto la  $CLI_G$ , ha la possibilità di autenticarsi al  $database_G$  oppure chiedere dell’aiuto. Una volta autenticato, esso può operare su collezioni e/o item, interrogare il  $database_G$ , modificare la propria password, effettuare il logout oppure, se si tratta di un amministratore, gestire gli utenti del  $database_G$ .

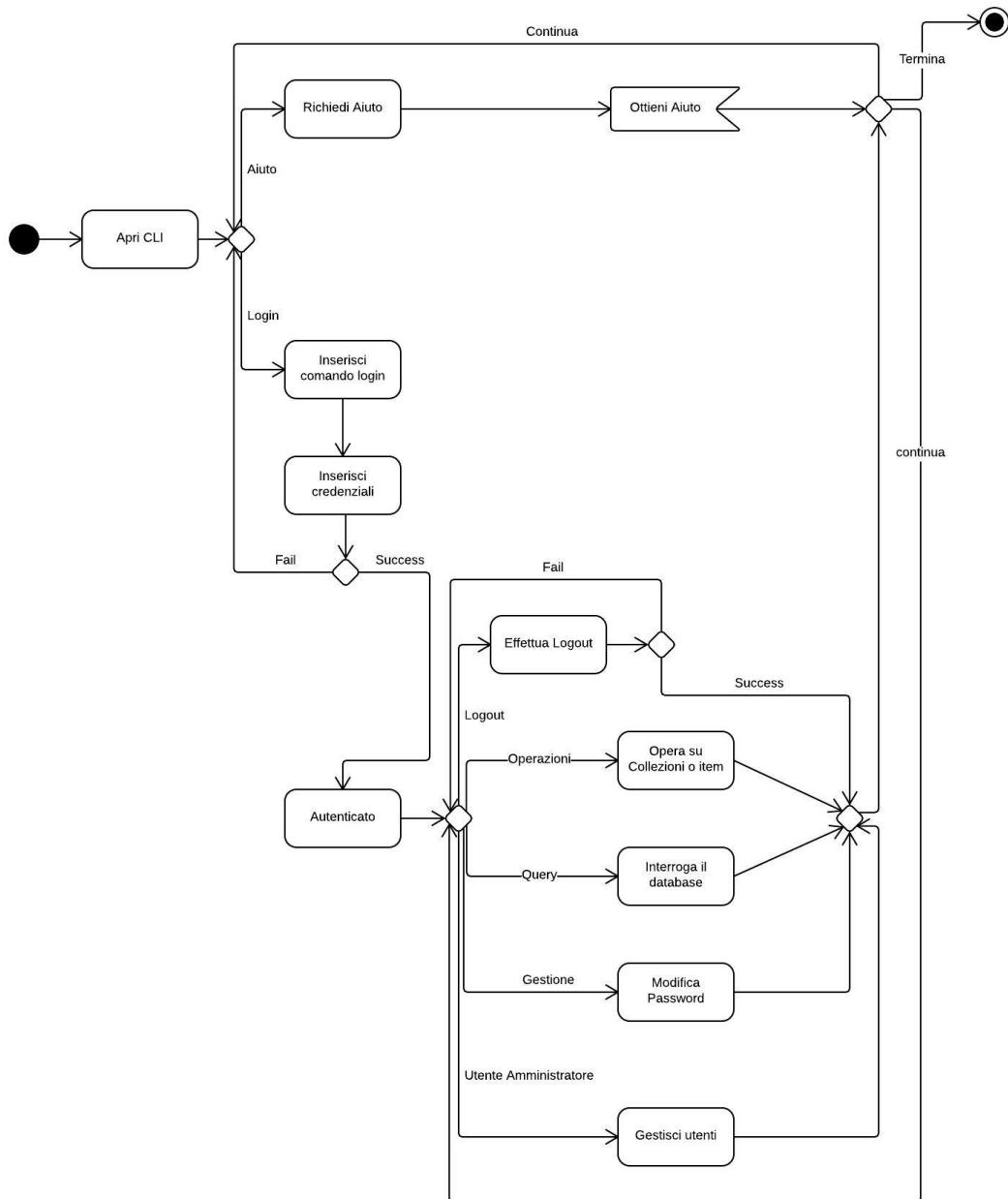


Figura 32: Diagrammi attività - Visione generale



## 5.2 Operazioni su collezioni e/o item

### 5.2.1 Creazione collezione

Questo tipo di operazione permette di inserire una collezione all'interno del database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la creazione della collezione, il nome della collezione stessa e i suoi parametri. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

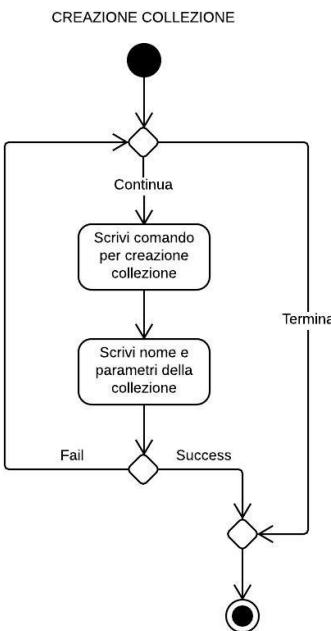


Figura 33: Diagrammi attività - Creazione collezione

### 5.2.2 Cancellazione collezione

Questo tipo di operazione permette di cancellare una collezione dal database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la cancellazione di una collezione e il nome della collezione stessa. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

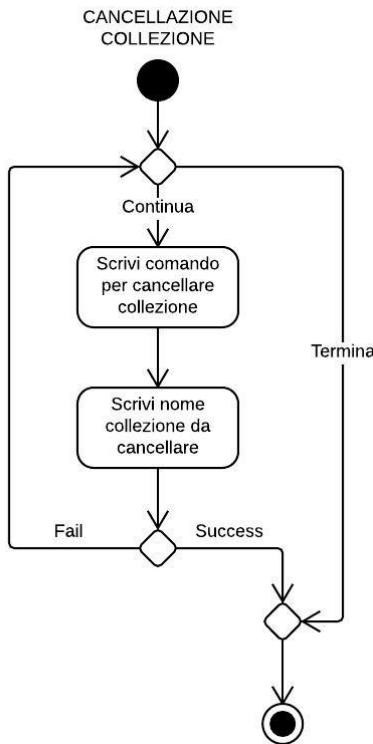


Figura 34: Diagrammi attività - Cancellazione collezione

### 5.2.3 Visualizza collezioni

Questo tipo di operazione permette di visualizzare le collezioni presenti all'interno del database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la visualizzazione delle collezioni. Una volta premuto il tasto invio, l'operazione andrà a buon termine (ricevendo la lista delle collezioni) se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

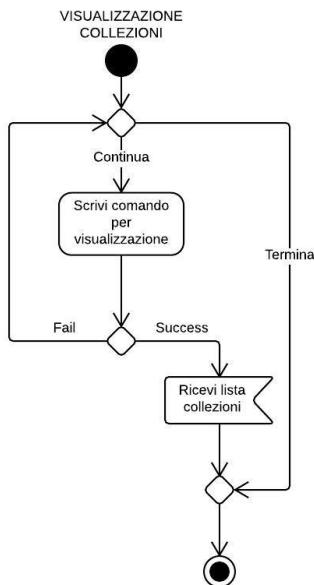


Figura 35: Diagrammi attività - Visualizzazione collezioni

#### 5.2.4 Modifica nome collezione

Questo tipo di operazione permette di modificare il nome di una collezione presente nel database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la rinominazione della collezione, il nome della collezione stessa e il nuovo nome per essa. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

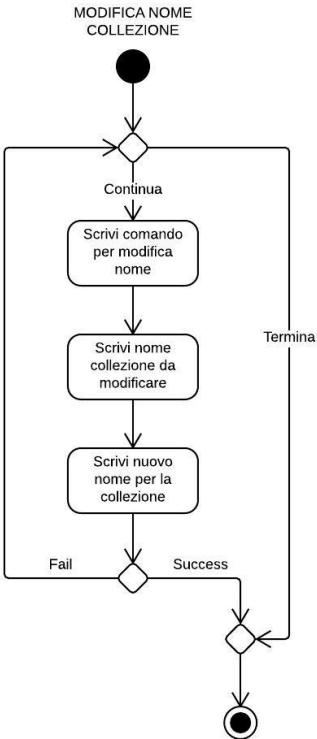


Figura 36: Diagrammi attività - Modifica nome collezione

### 5.2.5 Inserimento item

Questo tipo di operazione permette di inserire un item all'interno di una collezione del database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per l'inserimento item, il valore e parametri dell'item stesso e il nome della collezione dove inserire l'item. Una volta premuto il tasto invio, l'operazione andrà a buon termine (ricevendo la lista delle collezioni) se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

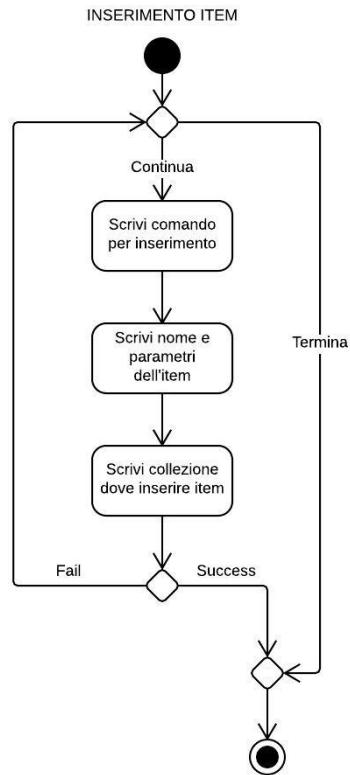


Figura 37: Diagrammi attività - Inserimento item

### 5.2.6 Rimozione item

Questo tipo di operazione permette di rimuovere un item dall'interno di una collezione<sub>G</sub> del database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per l'eliminazione di un item, il nome della collezione da dove rimuovere l'item e il nome dell'item stesso. Una volta premuto il tasto invio, l'operazione andrà a buon termine (ricevendo la lista delle collezioni) se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

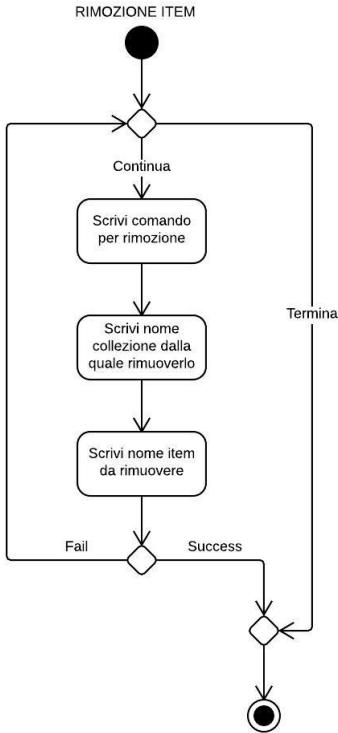


Figura 38: Diagrammi attività - Rimozione item

### 5.2.7 Aggiunta collaboratore

Questo tipo di operazione permette di aggiungere un collaboratore<sub>G</sub> ad una collezione presente nel database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per l'aggiunta di un collaboratore ad una collezione, lo username del collaboratore e il nome della collezione alla quale aggiungerlo. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

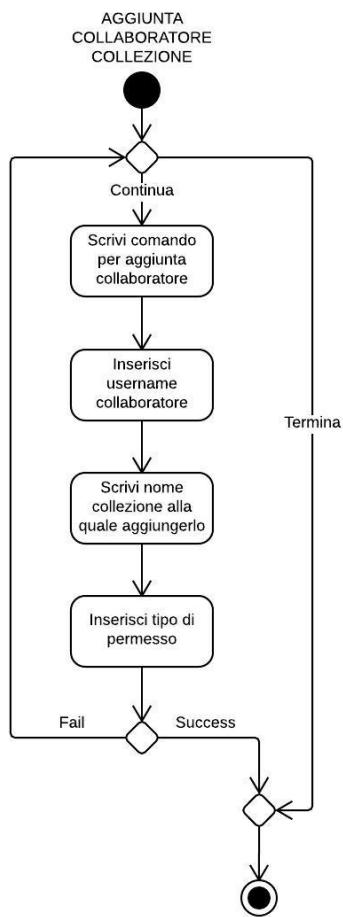


Figura 39: Diagrammi attività - Aggiunta collaboratore

### 5.2.8 Rimozione collaboratore

Questo tipo di operazione permette di rimuovere un collaboratore da una collezione presente nel database<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la rimozione di un collaboratore da una collezione, lo username del collaboratore e il nome della collezione dalla quale rimuoverlo. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

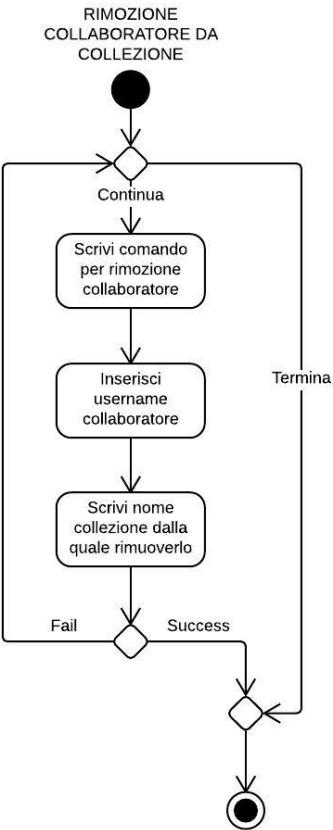


Figura 40: Diagrammi attività - Rimozione collaboratore

### 5.2.9 Import

Questo tipo di operazione permette di importare nel database<sub>G</sub> collezioni o item tramite file JSON<sub>G</sub>. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per l'importazione e il path che porta al file desiderato. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

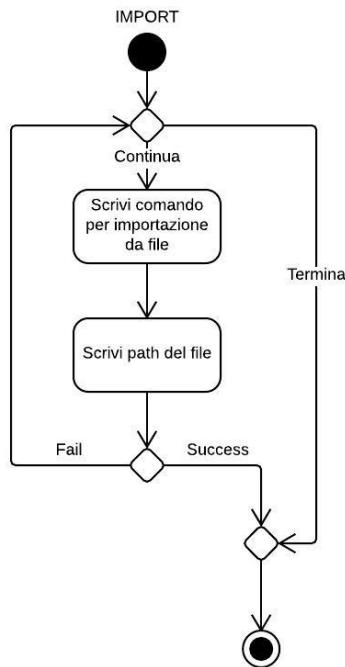


Figura 41: Diagrammi attività - Import

### 5.3 Interrogazione del database

Questo tipo di operazione permette di interrogare il database<sub>G</sub> tramite delle query. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per l'interrogazione del database<sub>G</sub> e i parametri per la ricerca. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando, ricevendo il risultato della query, altrimenti verrà visualizzato un messaggio di errore.

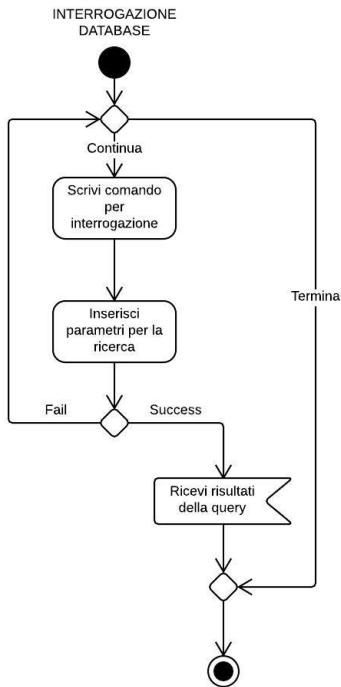


Figura 42: Diagrammi attività - Interrogazione del database<sub>G</sub>

#### 5.4 Modifica password

Questo tipo di operazione permette di modificare la propria password. Come si vede dal grafico che segue, l'utente dovrà inserire il comando per la modifica della password, la vecchia password, la nuova password e confermare quest'ultima. Una volta premuto il tasto invio, l'operazione andrà a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

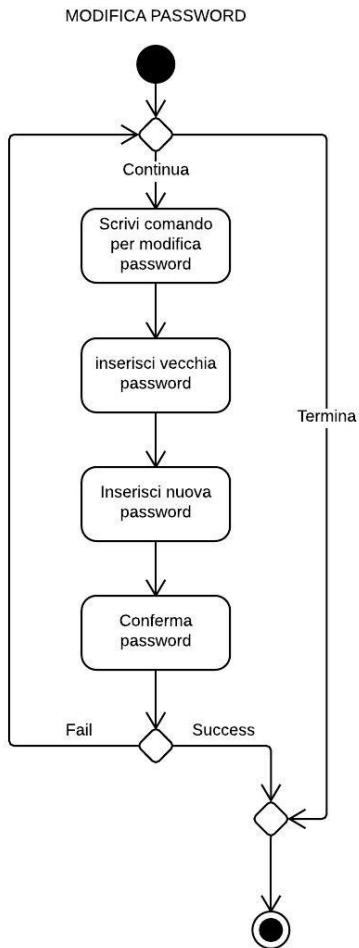


Figura 43: Diagrammi attività - Modifica password

## 5.5 Gestione utenti

La gestione utenti, possibile solo ad un utente amministratore, prevede la possibilità di aggiungere o rimuovere un utente dal database<sub>G</sub> e la possibilità di resettare la password di un utente. Per aggiungere o rimuovere un utente, l'amministratore dovrà inserire il rispettivo comando, lo username dell'utente da aggiungere/rimuovere dal database<sub>G</sub> e una conferma. Per resettare la password di un utente, l'amministratore dovrà scrivere il comando per il reset della password, lo username dell'utente interessato e una conferma. Una volta premuto il tasto invio, le operazioni andranno a buon termine se l'utente ha scritto correttamente il comando altrimenti verrà visualizzato un messaggio di errore.

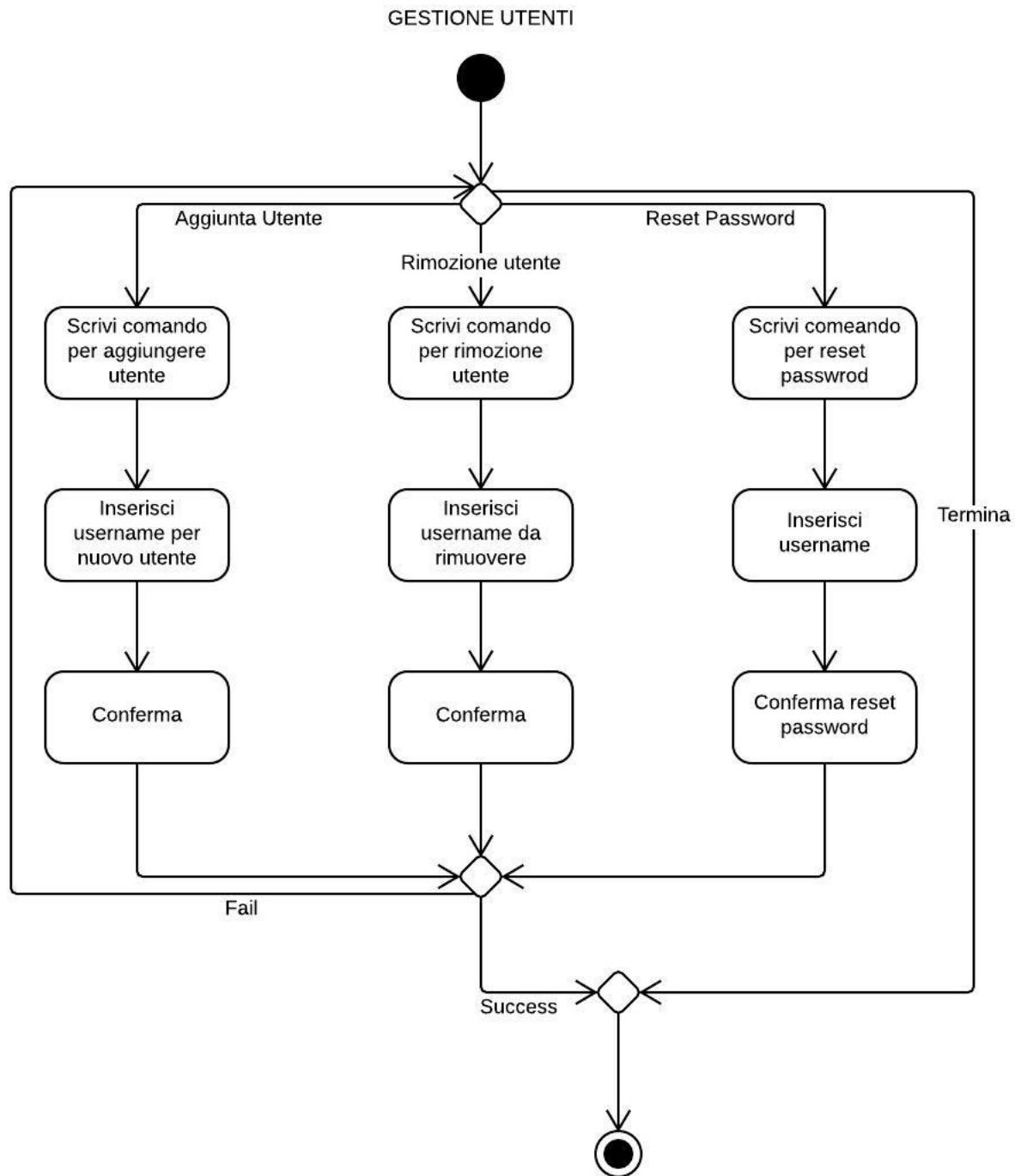


Figura 44: Diagrammi attività - Gestione utenti



## 6 Design Pattern

I Design Pattern<sub>G</sub> rappresentano soluzioni progettuali generali ad un problema ricorrente. Si tratta di modelli logici da applicare per la risoluzione di un problema ancor prima della definizione dell'algoritmo risolutivo vero e proprio, favoriscono il riutilizzo del codice e rendono l'architettura più manutenibile.

I Design Pattern<sub>G</sub> possono essere suddivisi in:

- **architetturali:** operano al livello più alto, esprimono schemi di base per impostare l'organizzazione strutturale di un sistema software;
- **creazionali:** nascondono i costruttori delle classi e mettono dei metodi al loro posto creando un'interfaccia, in questo modo forniscono un'astrazione del processo di instanziazione degli oggetti;
- **strutturali:** consentono di riutilizzare degli oggetti esistenti fornendo agli utilizzatori un'interfaccia più adatta alle loro esigenze;
- **comportamentali:** definiscono soluzioni per le interazioni tra oggetti.

Si rimanda all'Appendice A per un approfondimento dei Design Pattern<sub>G</sub> utilizzati nel progetto **Actorbase**.

### 6.1 Design Pattern Architetturali

#### 6.1.1 MVC

- **Scopo:** È stato scelto l'utilizzo del pattern Model<sub>G</sub> View<sub>G</sub> Controller<sub>G</sub> (MVC<sub>G</sub>) per l'alto grado di separazione tra la parte logica dell'applicazione e parte grafica che offre;
- **Utilizzo:** Viene utilizzato per delineare l'architettura generale della componente CLI<sub>G</sub> (Command Line Interface) del progetto. La parte View<sub>G</sub> funge da interfaccia con l'utente, si occupa di leggere l'input e inviarlo al Controller, il quale si occupa di validare l'input e inviare il comando extrapolato al Model<sub>G</sub>. Il Model<sub>G</sub> mediante un Command Pattern<sub>G</sub> (vedi [Command Pattern](#)) esegue il comando richiesto e notifica la View<sub>G</sub> utilizzando un Observer Pattern<sub>G</sub> (vedi [Observer Pattern](#)) secondo la variante Push model<sub>G</sub>.

#### 6.1.2 Dependancy Injection

- **Scopo:** Separa il comportamento di una componente dalla risoluzione delle sue dipendenze, offrendo un maggiore grado di astrazione;
- **Utilizzo:** Viene utilizzato nella componente driver, più precisamente nella classe [ActorbaseServices](#), sfruttando la variante cake-pattern<sub>G</sub> offerta dal linguaggio Scala<sub>G</sub> per aggiungere dipendenze amministrative fornite da ActorbaseAdminServices alla classe dedicata alle operazioni comuni. Tale pattern<sub>G</sub> viene anche utilizzato nella classe [ActorbaseClient](#) per fornire crittografia TLS/SSL<sub>G</sub> (Transport Layer Security / Secure Socket Layer) al client<sub>G</sub> utilizzato per la connessione.



## 6.2 Design Pattern Creazionali

### 6.2.1 Singleton

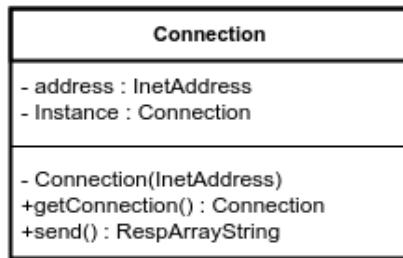


Figura 45: Connection Singleton

- **Scopo:** Viene utilizzato per le classi di cui è preferibile avere un'unica istanza durante l'esecuzione dell'applicazione;
- **Utilizzo:** È stato utilizzato una sola volta per la classe di connessione all'interno della componente driver<sub>G</sub>, in modo da offrire un unico punto di accesso al server<sub>G</sub>.

### 6.2.2 Builder

- **Scopo:** Il pattern Builder<sub>G</sub> viene usato per separare la costruzione di un oggetto complesso dalla sua rappresentazione. L'algoritmo di creazione dell'oggetto complesso è indipendente dalle varie parti che costituiscono l'oggetto e da come vengono assemblate. In questo modo è possibile costruire diverse rappresentazioni dell'oggetto utilizzando lo stesso oggetto Builder<sub>G</sub>.
- **Utilizzo:** Viene implementato all'interno della componente driver<sub>G</sub>, precisamente dalla classe RequestBuilder per la costruzione delle diverse richieste HTTP<sub>G</sub> da inviare alla componente server<sub>G</sub>.

## 6.3 Design Pattern Strutturali

### 6.3.1 Decorator

- **Scopo:** Il pattern Decorator<sub>G</sub> viene usato per aggiungere dinamicamente nuove funzionalità ad oggetti già esistenti, e permette di estenderli più facilmente;
- **Utilizzo:** Viene utilizzato all'interno della componente driver<sub>G</sub>, precisamente dalla classe ActorbaseClient per aggiungere funzionalità di protezione SSL/TLS<sub>G</sub> alla comunicazione HTTP<sub>G</sub> con la componente server<sub>G</sub>.



## 6.4 Design Pattern Comportamentali

### 6.4.1 Command Pattern

- **Scopo:** Viene utilizzato per separare l'implementazione di un'azione ( $\text{Receiver}_G$ ) dal richiamante dell'azione stessa ( $\text{Invoker}_G$ ).
- **Utilizzo:** Viene utilizzato per la gestione dei comandi nella componente  $\text{CLI}_G$ , e funge da  $\text{Model}_G$  dell'architettura  $\text{MVC}_G$  utilizzata.

Le classi che implementano i comandi realizzando l'interfaccia `Command` sono:

- `FindCommand`
- `ListCommand`
- `LoginCommand`
- `LogoutCommand`
- `HelpCommand`
- `InsertItemCommand`
- `RemoveItemCommand`
- `RenameCollectionCommand`
- `CreateCollectionCommand`
- `ChangePasswordCommand`
- `RemoveCollectionCommand`
- `ImportCommand`
- `ExportCommand`
- `AddContributorCommand`
- `RemoveContributorCommand`
- `ResetPasswordCommand`
- `AddUserCommand`
- `RemoveUserCommand`

La classe incaricata di eseguire i comandi è `CommandReceiver`, invocata mediante `CommandInvoker`, utilizzando i parametri ricevuti dalla parte  $\text{Controller}_G$  dell'architettura  $\text{MVC}_G$  scelta per la componente  $\text{CLI}_G$ .

### 6.4.2 Observer Pattern

- **Scopo:** Viene utilizzato per tenere sotto controllo lo stato di diversi oggetti, ed è parte dell'implementazione del Design Pattern<sub>G</sub>  $\text{MVC}_G$  con logica push model<sub>G</sub>, il funzionamento si basa su meccanismi di callback.



- **Utilizzo:** È stato utilizzato nella componente  $CLI_G$  del progetto in modo da ottenere una  $view_G$  aggiornante sull'output prodotto dai comandi inviati alla componente  $driver_G$ .  
Più in dettaglio, la classe `CommandInvoker` realizza l'interfaccia `Observable` aggiungendo l'oggetto di tipo `ResultView` alla propria lista di  $observers_G$ , all'esecuzione di ogni comando, notifica `resultview` con l'output prodotto.



## 7 Stime di fattibilità e di bisogno di risorse

Durante la progettazione dell'architettura, oltre alle tecnologie e librerie consigliate dal proponente, ne sono state ricercate e testate altre in modo da poter usufruire di funzionalità già esistenti.

Dopo un iniziale orientamento verso il framework<sub>G</sub> Spring<sub>G</sub>, più precisamente nella sua derivazione SpringShell<sub>G</sub>, in modo da poter generare una CLI<sub>G</sub> personalizzata, abbiamo deciso di rimanere su un'implementazione più leggera in linguaggio Scala<sub>G</sub> in quanto la scarsa personalizzazione offerta e il grosso carico di dipendenze che necessitava SpringShell<sub>G</sub> per il suo utilizzo è stato ritenuto eccessivamente gravoso sulla realizzazione della componente in questione.

È stato tuttavia deciso di utilizzare il framework<sub>G</sub> pickling<sub>G</sub> per effettuare serializzazione<sub>G</sub> della persistenza su disco; offre una buona documentazione e numerosi esempi di utilizzo, rende inoltre possibile la definizione di protocolli di serializzazione<sub>G</sub> personalizzati.

Per la gestione della comunicazione HTTP<sub>G</sub> sono stati utilizzate due librerie:

- **Lato server:** Spray permette di gestire efficacemente le richieste in ingresso di tipo HTTP<sub>G</sub> asincronamente, appositamente sviluppata per funzionare in congiunzione con il modello ad attori offerto da akka<sub>G</sub>;
- **Lato client:** Scalaj-http è una semplice libreria Scala<sub>G</sub> che funge da client HTTP<sub>G</sub> per la comunicazione con la componente server<sub>G</sub>.



## 8 Tracciamento

### 8.1 Tracciamento Componenti-Requisiti

Componente	Requisiti
actorbase	DEF3 DEV8 DEV9 OBF1 OBF2 OBQ12 OBV4 OBV5 OBV6 OBV7
actorbase::actorsystem	OBF1
actorbase::actorsystem::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::messages	OBF1.1.2.1 OBF1.1.2.2
actorbase::actorsystem::httpserver	OBF1.1 OBF1.1.1 OBF1.1.1.1 OBF1.1.10
actorbase::actorsystem::httpserver::messages	OBF1.1.1.1.1



actorbase::actorsystem::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	OBF1.1.3.1 OBF1.1.3.2 OBF1.1.3.3 OBF1.1.3.4 OBF1.1.3.5 OBF1.1.3.6 OBF1.1.3.7 OBF1.1.3.8 OBF1.1.3.9 OBF1.1.3.11 OBF1.1.3.12 OBF1.1.3.12 OBF1.1.3.13
actorbase::actorsystem::manager	DEF1.1.8 OBF1.1.10.3 OBF1.1.10.3.1
actorbase::actorsystem::manager::messages	DEF1.1.8.1 DEF1.1.8.2
actorbase::actorsystem::ninja	DEF1.1.7 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
actorbase::actorsystem::ninja::messages	DEF1.1.7.1 DEF1.1.7.2
actorbase::actorsystem::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	OBF1.1.4.1 OBF1.1.4.2 OBF1.1.4.3 OBF1.1.4.4 OBF1.1.4.5 OBF1.1.4.6 OBF1.1.4.7
actorbase::actorsystem::storekeeper	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	OBF1.1.5.1 OBF1.1.5.2 OBF1.1.5.3 OBF1.1.5.4 OBF1.1.5.5 OBF1.1.5.6 OBF1.1.5.7
actorbase::actorsystem::userfinder	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 OBF1.1.11
actorbase::actorsystem::userfinder::messages	OBF1.1.11.1 OBF1.1.11.2 OBF1.1.11.3 OBF1.1.11.4 OBF1.1.11.5 OBF1.1.11.6 OBF1.1.11.7
actorbase::actorsystem::userkeeper	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 OBF1.1.9
actorbase::actorsystem::userkeeper::messages	OBF1.1.9.1 OBF1.1.9.2 OBF1.1.9.3 OBF1.1.9.4 OBF1.1.9.5 OBF1.1.9.6 OBF1.1.9.9



actorbase::actorsystem::utils	DEF1.1.7 OBF1.1.11 OBF1.1.2 OBF1.1.3 OBF1.1.4 OBF1.1.5 OBF1.1.6 OBF1.1.9
actorbase::actorsystem::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	OBF1.1.6.1 OBF1.1.6.2 OBF1.1.6.3 OBF1.1.6.4
actorbase::cli	OBF2
actorbase::cli::controllers	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	DEF2.1.2.7 DEF2.1.2.7.1 DEF2.1.3.1.2 DEF2.1.3.1.2.1 DEF2.1.4 DEF2.1.5 DEF2.1.5.1 OBF2.1.1 OBF2.1.1.1 OBF2.1.1.2 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 OBF2.1.2.8 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 OBF2.1.6 OBF2.1.6.1 OBF2.1.6.2 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	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 DEF2.1.3.2 DEF2.1.3.3 DEF2.1.4 DEF2.1.5 DEF2.1.5.1 OBF2 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 OBF2.1.2.10 OBF2.1.2.11 OBF2.1.2.12 OBF2.1.2.13 OBF2.1.2.14 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 OBF2.1.2.8 OBF2.1.2.9 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 OBF2.1.3.4 OBF2.1.3.6 OBF2.1.3.6.1 OBF2.1.3.6.2 OBF2.1.3.5 OBF2.1.6 OBF2.1.6.1 OBF2.1.6.2 OBF2.1.7 OBF2.1.7.1 OBF2.1.7.2 OBF2.1.7.3 OBF2.1.7.4 OBF2.1.7.5 OBF2.1.7.6 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 OBF2.1.9.4 OBF2.1.9.5 OBF2.1.9.6
-----------------------	--



actorbase::cli::views::Observer	DEF3.1.2 DEF3.1.3 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	DEF3



actorbase::driver::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::api	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 DEF3.2.7.3 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 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 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 DEF3.7 DEF3.7.1 DEF3.7.2
--------------------------------	--



actorbase::driver::data	DEF3.2 DEF3.2.1 DEF3.2.1.1 DEF3.2.1.2 DEF3.2.3 DEF3.2.3.1 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 DEF3.3 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 DEF3.3.1.4 DEF3.3.2 DEF3.3.2.1 DEF3.3.2.2 DEF3.3.2.3 DEF3.4 DEF3.4.1 DEF3.4.2 DEF3.8 DEF3.8.1 DEF3.8.2 DEF3.8.3
actorbase::driver::exceptions	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

## 8.2 Tracciamento Requisiti-Componenti

Requisito	Componenti
DEF1.1.7	actorbase::actorsystem::ninja actorbase::actorsystem::utils
DEF1.1.7.1	actorbase::actorsystem::ninja::messages
DEF1.1.7.2	actorbase::actorsystem::ninja::messages
DEF1.1.8	actorbase::actorsystem::manager
DEF1.1.8.1	actorbase::actorsystem::manager::messages
DEF1.1.8.2	actorbase::actorsystem::manager::messages
DEF2.1.2.7	actorbase::cli::models actorbase::cli::views



DEF2.1.2.7.1	actorbase::cli::models actorbase::cli::views
DEF2.1.2.7.2	actorbase::cli::views
DEF2.1.3.1.2	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
DEF2.1.3.1.2.1	actorbase::cli::models actorbase::cli::views
DEF2.1.3.2	actorbase::cli::views
DEF2.1.3.3	actorbase::cli::views
DEF2.1.4	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
DEF2.1.5	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
DEF2.1.5.1	actorbase::cli::models actorbase::cli::views
DEF3	actorbase actorbase::driver
DEF3.1	actorbase::driver::client actorbase::driver::client::api
DEF3.1.1	actorbase::driver::client actorbase::driver::client::api
DEF3.1.2	actorbase::cli::views::Observer actorbase::driver::client actorbase::driver::client::api
DEF3.1.3	actorbase::cli::views::Observer actorbase::driver::client actorbase::driver::exceptions
DEF3.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.1.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.1.2	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions



DEF3.2.2	actorbase::driver::client actorbase::driver::client::api
DEF3.2.3	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.3.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.3.2	actorbase::driver::client actorbase::driver::exceptions
DEF3.2.4	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.4.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.4.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.4.3	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.2.4.4	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.2.5	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.5.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.5.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.5.3	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.2.5.4	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.2.5.5	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data



DEF3.2.6	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.6.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.6.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.2.6.3	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.2.7	actorbase::driver::client actorbase::driver::client::api
DEF3.2.7.1	actorbase::driver::client actorbase::driver::client::api
DEF3.2.7.2	actorbase::driver::client actorbase::driver::client::api
DEF3.2.7.3	actorbase::driver::client actorbase::driver::client::api
DEF3.3	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.1.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.1.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.1.3	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.1.4	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.1.2	actorbase::driver::client actorbase::driver::client::api



DEF3.3.1.2.1	actorbase::driver::client actorbase::driver::client::api
DEF3.3.1.2.2	actorbase::driver::client actorbase::driver::exceptions
DEF3.3.1.2.3	actorbase::driver::client actorbase::driver::exceptions
DEF3.3.1.4	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.3.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.2.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.2.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.3.2.3	actorbase::driver::client actorbase::driver::data actorbase::driver::exceptions
DEF3.4	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.4.1	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.4.2	actorbase::driver::client actorbase::driver::client::api actorbase::driver::data
DEF3.5	actorbase::driver::client actorbase::driver::client::api
DEF3.6	actorbase::driver::client actorbase::driver::client::api
DEF3.6.1	actorbase::driver::client actorbase::driver::client::api
DEF3.6.1.1	actorbase::driver::client actorbase::driver::client::api
DEF3.6.2	actorbase::driver::client actorbase::driver::client::api
DEF3.6.2.1	actorbase::driver::client actorbase::driver::client::api



DEF3.6.3	actorbase::driver::client actorbase::driver::client::api
DEF3.6.3.1	actorbase::driver::client actorbase::driver::client::api
DEF3.6.4	actorbase::driver::client actorbase::driver::exceptions
DEF3.6.5	actorbase::driver::client actorbase::driver::exceptions
DEF3.6.6	actorbase::driver::client actorbase::driver::exceptions
DEF3.7	actorbase::driver::client actorbase::driver::client::api
DEF3.7.1	actorbase::driver::client actorbase::driver::client::api
DEF3.7.2	actorbase::driver::client actorbase::driver::client::api
DEF3.7.3	actorbase::driver::client actorbase::driver::exceptions
DEF3.7.4	actorbase::driver::client actorbase::driver::exceptions
DEF3.8	actorbase::driver::client actorbase::driver::data
DEF3.8.1	actorbase::driver::client actorbase::driver::data
DEF3.8.2	actorbase::driver::client actorbase::driver::data
DEF3.8.3	actorbase::driver::client actorbase::driver::data
DEV8	actorbase
DEV9	actorbase
OBF1	actorbase actorbase::actorsystem
OBF1.1	actorbase::actorsystem::httpserver
OBF1.1.1	actorbase::actorsystem::httpserver
OBF1.1.1.1	actorbase::actorsystem::httpserver
OBF1.1.1.1.1	actorbase::actorsystem::httpserver::messages
OBF1.1.10	actorbase::actorsystem::clientactor actorbase::actorsystem::httpserver
OBF1.1.10.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper



OBF1.1.10.1.1	actorbase::actorsystem::clientactor actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.2	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.2.1.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main
OBF1.1.10.2.2	actorbase::actorsystem::clientactor
OBF1.1.10.2.3	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.2.3.1	actorbase::actorsystem::clientactor actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.2.3.2	actorbase::actorsystem::clientactor
OBF1.1.10.2.4	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.4.1	actorbase::actorsystem::clientactor



OBF1.1.10.2.5	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.5.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.5.2	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.5.3	actorbase::actorsystem::clientactor
OBF1.1.10.2.5.4	actorbase::actorsystem::clientactor actorbase::actorsystem::storefinder
OBF1.1.10.2.6	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.2.6.1	actorbase::actorsystem::clientactor
OBF1.1.10.2.7	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman



OBF1.1.10.3	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::manager actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper actorbase::actorsystem::warehouseman
OBF1.1.10.3.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::manager actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::warehouseman
OBF1.1.10.3.2	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.3.3	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper
OBF1.1.10.3.4	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::ninja actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper actorbase::actorsystem::warehouseman
OBF1.1.10.3.5	actorbase::actorsystem::clientactor
OBF1.1.10.4	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::storekeeper
OBF1.1.10.5	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.5.1	actorbase::actorsystem::clientactor actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.6	actorbase::actorsystem::clientactor



OBF1.1.10.7	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.7.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.7.1.1	actorbase::actorsystem::clientactor
OBF1.1.10.7.2	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder
OBF1.1.10.7.2.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder
OBF1.1.10.7.3	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder actorbase::actorsystem::userfinder actorbase::actorsystem::userkeeper
OBF1.1.10.7.3.1	actorbase::actorsystem::clientactor actorbase::actorsystem::main actorbase::actorsystem::storefinder
OBF1.1.11	actorbase::actorsystem::userfinder actorbase::actorsystem::utils
OBF1.1.11.1	actorbase::actorsystem::userfinder::messages
OBF1.1.11.2	actorbase::actorsystem::userfinder::messages
OBF1.1.11.3	actorbase::actorsystem::userfinder::messages
OBF1.1.11.4	actorbase::actorsystem::userfinder::messages
OBF1.1.11.5	actorbase::actorsystem::userfinder::messages
OBF1.1.11.6	actorbase::actorsystem::userfinder::messages
OBF1.1.11.7	actorbase::actorsystem::userfinder::messages
OBF1.1.2	actorbase::actorsystem::clientactor actorbase::actorsystem::utils
OBF1.1.2.1	actorbase::actorsystem::clientactor::messages
OBF1.1.2.2	actorbase::actorsystem::clientactor::messages
OBF1.1.3	actorbase::actorsystem::main actorbase::actorsystem::utils
OBF1.1.3.1	actorbase::actorsystem::main::messages
OBF1.1.3.10	actorbase::actorsystem::main::messages
OBF1.1.3.11	actorbase::actorsystem::main::messages



OBF1.1.3.12	actorbase::actorsystem::main::messages
OBF1.1.3.13	actorbase::actorsystem::main::messages
OBF1.1.3.2	actorbase::actorsystem::main::messages
OBF1.1.3.3	actorbase::actorsystem::main::messages
OBF1.1.3.4	actorbase::actorsystem::main::messages
OBF1.1.3.5	actorbase::actorsystem::main::messages
OBF1.1.3.6	actorbase::actorsystem::main::messages
OBF1.1.3.7	actorbase::actorsystem::main::messages
OBF1.1.3.8	actorbase::actorsystem::main::messages
OBF1.1.3.9	actorbase::actorsystem::main::messages
OBF1.1.4	actorbase::actorsystem::storefinder actorbase::actorsystem::utils
OBF1.1.4.1	actorbase::actorsystem::storefinder::messages
OBF1.1.4.2	actorbase::actorsystem::storefinder::messages
OBF1.1.4.3	actorbase::actorsystem::storefinder::messages
OBF1.1.4.4	actorbase::actorsystem::storefinder::messages
OBF1.1.4.5	actorbase::actorsystem::storefinder::messages
OBF1.1.4.6	actorbase::actorsystem::storefinder::messages
OBF1.1.4.7	actorbase::actorsystem::storefinder::messages
OBF1.1.5	actorbase::actorsystem::storekeeper actorbase::actorsystem::utils
OBF1.1.5.1	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.2	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.3	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.4	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.5	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.6	actorbase::actorsystem::storekeeper::messages
OBF1.1.5.7	actorbase::actorsystem::storekeeper::messages
OBF1.1.6	actorbase::actorsystem::utils actorbase::actorsystem::warehouseman
OBF1.1.6.1	actorbase::actorsystem::warehouseman::messages
OBF1.1.6.2	actorbase::actorsystem::warehouseman::messages
OBF1.1.6.3	actorbase::actorsystem::warehouseman::messages
OBF1.1.6.4	actorbase::actorystem::warehouseman::messages
OBF1.1.9	actorbase::actorsystem::userkeeper actorbase::actorsystem::utils
OBF1.1.9.1	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.2	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.3	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.4	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.5	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.6	actorbase::actorsystem::userkeeper::messages
OBF1.1.9.9	actorbase::actorsystem::userkeeper::messages



OBF2	actorbase actorbase::cli actorbase::cli::views
OBF2.1	actorbase::cli::controllers
OBF2.1.1	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.1.1	actorbase::cli::models actorbase::cli::views
OBF2.1.1.2	actorbase::cli::models actorbase::cli::views
OBF2.1.1.3	actorbase::cli::views
OBF2.1.2	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.1	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.1.1	actorbase::cli::models actorbase::cli::views
OBF2.1.2.10	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.11	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.12	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.13	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.14	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.2	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.3	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.3.1	actorbase::cli::models actorbase::cli::views
OBF2.1.2.4	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.4.1	actorbase::cli::models actorbase::cli::views



OBF2.1.2.4.2	actorbase::cli::models actorbase::cli::views
OBF2.1.2.5	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.5.1	actorbase::cli::models actorbase::cli::views
OBF2.1.2.5.2	actorbase::cli::models actorbase::cli::views
OBF2.1.2.5.3	actorbase::cli::models actorbase::cli::views
OBF2.1.2.6	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.2.6.1	actorbase::cli::models actorbase::cli::views
OBF2.1.2.6.2	actorbase::cli::models actorbase::cli::views
OBF2.1.2.8	actorbase::cli::models actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.2.9	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.3	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.3.1	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1.1	actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1.2	actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1.3	actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1.4	actorbase::cli::models actorbase::cli::views
OBF2.1.3.1.1.5	actorbase::cli::models actorbase::cli::views



OBF2.1.3.4	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.3.5	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.3.6	actorbase::cli::models actorbase::cli::views
OBF2.1.3.6.1	actorbase::cli::views
OBF2.1.3.6.2	actorbase::cli::views
OBF2.1.6	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.6.1	actorbase::cli::models actorbase::cli::views
OBF2.1.6.2	actorbase::cli::models actorbase::cli::views
OBF2.1.7	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.7.1	actorbase::cli::models actorbase::cli::views
OBF2.1.7.2	actorbase::cli::models actorbase::cli::views
OBF2.1.7.3	actorbase::cli::models actorbase::cli::views
OBF2.1.7.4	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.7.5	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.7.6	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.8	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.9	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.9.1	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.9.1.1	actorbase::cli::models actorbase::cli::views
OBF2.1.9.1.2	actorbase::cli::models actorbase::cli::views



OBF2.1.9.2	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.9.2.1	actorbase::cli::models actorbase::cli::views
OBF2.1.9.2.2	actorbase::cli::models actorbase::cli::views
OBF2.1.9.3	actorbase::cli::controllers actorbase::cli::models actorbase::cli::views
OBF2.1.9.3.1	actorbase::cli::models actorbase::cli::views
OBF2.1.9.3.2	actorbase::cli::models actorbase::cli::views
OBF2.1.9.4	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.9.5	actorbase::cli::views actorbase::cli::views::Observer
OBF2.1.9.6	actorbase::cli::views actorbase::cli::views::Observer
OBQ12	actorbase
OBV4	actorbase
OBV5	actorbase
OBV6	actorbase
OBV7	actorbase



## A Descrizione Design Pattern

### A.1 Design Pattern Architetturali

#### A.1.1 MVC

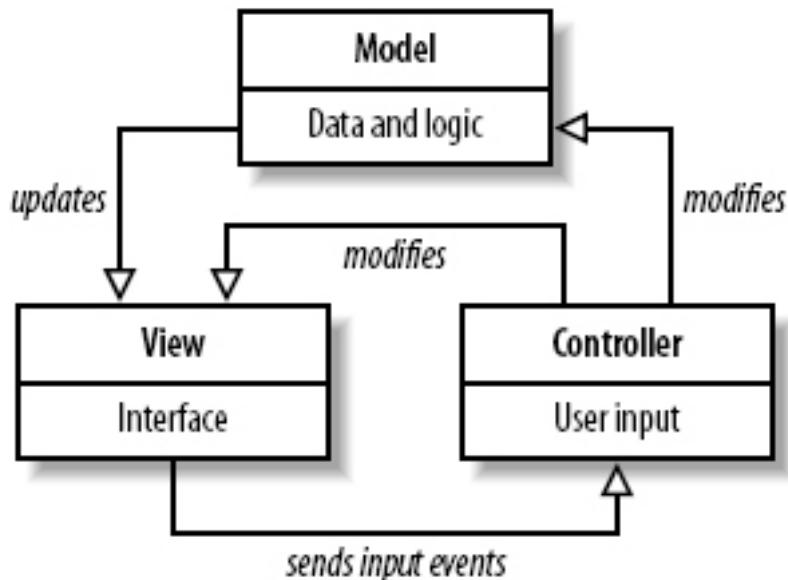


Figura 46: Esempio di applicazione del design pattern MVC

- **Scopo:** Disaccoppiare le tre seguenti componenti:
  - Model: parte che si occupa direttamente di dati, logica e regole dell'applicazione;
  - View: rappresentazione grafica sotto forma delle varie tipologie di output utilizzate per rappresentare i dati dell'applicazione;
  - Controller: parte che accetta gli input e li converte in comandi per il model o la view;
- **Motivazione:** Molte applicazioni hanno la necessità di recuperare dati e di mostrarli in maniera opportuna agli utenti. Poiché si tratta di una comunicazione tra i dati dei dati e la interfaccia utente, bisogna trovare il metodo per far comunicare le due parti senza accorparle assieme, cosa che comporterebbe codice pesante e scarsa manutenibilità, in quanto in genere la parte grafica si evolve più in fretta della parte di model e, viceversa, bisogna rendere l'interfaccia che si offre all'utente quanto più separata possibile dall'implementazione effettiva della gestione dei dati. La soluzione che è stata trovata è costituita dal design pattern<sub>6</sub> Model-View-Controller (MVC) che separa le tre componenti come già illustrato;

- **Applicabilità:** Il pattern MVC è adatto ad essere utilizzato nei seguenti casi:
    - se c'è la necessità che un insieme di oggetti debba essere considerato come un oggetto singolo;
    - se c'è la necessità di disaccoppiare il model e la view, creando un sistema di notifiche tra le due componenti;
    - Se c'è la necessità di avere più view per il medesimo model.

## A.2 Design Pattern Creazionali

### A.2.1 Singleton

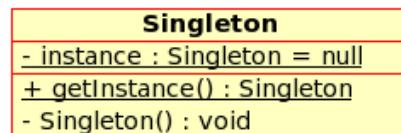


Figura 47: Esempio di applicazione del design pattern singleton

- **Scopo:** Assicurare che una classe abbia un'unica istanza ed avere un punto di accesso globale ad essa;
  - **Motivazione:** Per alcune classi è fondamentale assicurare che non abbiano più di un'istanza. Per questo motivo la classe tipicamente ha un costruttore privato e un metodo per poter accedere all'istanza della classe pubblico;
  - **Applicabilità:** Questo design pattern<sub>G</sub> può essere applicato nei seguenti casi:
    - Quando è essenziale che ci sia solo una istanza di una classe e tale istanza deve essere resa accessibile attraverso un unico punto di accesso;
    - Quando l'unica istanza della classe deve essere estesa in maniera tale da non dover avere client<sub>G</sub> con codici diversi.

### A.2.2 Builder

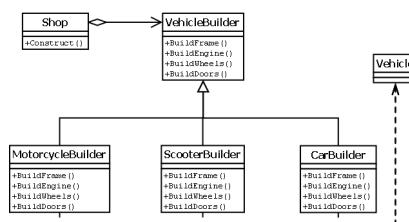


Figura 48: Esempio di applicazione del design pattern builder



- **Scopo:** Permettere la crezione di oggetti complessi indipendentemente dalla sua rappresentazione;
- **Motivazione:** Alcuni oggetti richiedono molti parametri di inizializzazione e possono avere molte rappresentazioni, questo porta spesso all'effetto telescoping<sub>G</sub> dei costruttori di tale oggetto, e il pattern<sub>G</sub> abstract factory<sub>G</sub> risulta non essere abbastanza adeguato allo scopo;
- **Applicabilità:** Questo design pattern<sub>G</sub> può essere applicato nei seguenti casi:
  - quando è necessaria la creazione di oggetti complessi con molte varianti di rappresentazione.

## A.3 Design Pattern Strutturali

### A.3.1 Decorator

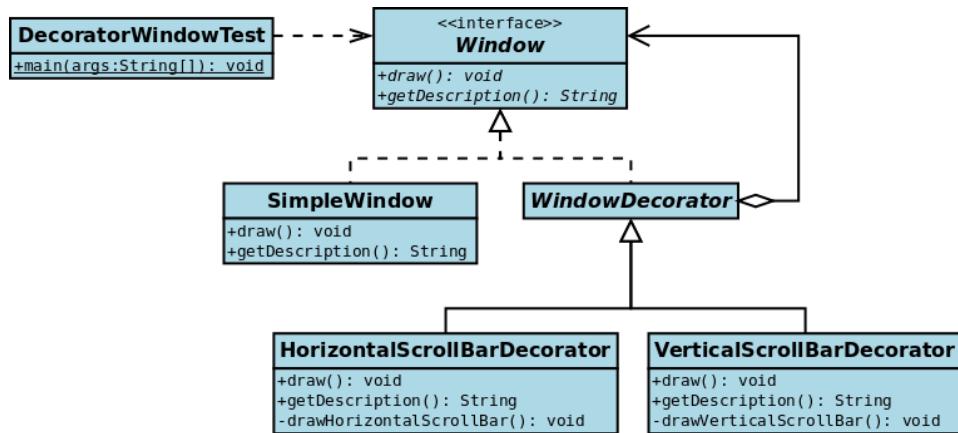


Figura 49: Esempio di applicazione del design pattern decorator

- **Scopo:** Aggiungere responsabilità e funzionalità dinamicamente ad oggetti già esistenti;
- **Motivazione:** Permettere l'estendibilità di oggetti e affinare le funzionalità offerte modificandole o aggiungendone di nuove in modo trasparente.
- **Applicabilità:** Questo design pattern<sub>G</sub> può essere applicato nei seguenti casi:
  - Quando si presenta la necessità di riutilizzare un oggetto per le sue funzionalità, necessitando tuttavia di altre aggiuntive.



## A.4 Design Pattern Comportamentali

### A.4.1 Command

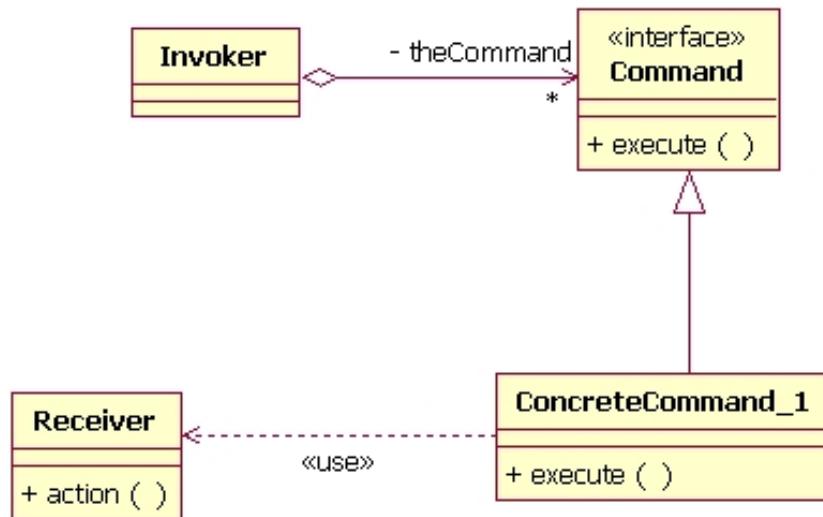


Figura 50: Esempio di applicazione del design pattern command

- **Scopo:** Incapsulare una richiesta in un oggetto, cosicché il client<sub>G</sub> sia indipendente dalle richieste, poiché le richieste hanno al loro interno tutti i dati necessari per essere risolte.
- **Motivazione:** C'è la necessità di gestire richieste di cui non si conoscono i particolari poiché i Toolkit associano ai propri elementi, richieste da eseguire. Una classe astratta, **Command**, definisce l'interfaccia per eseguire la richiesta che è un semplice oggetto, per cui è possibile rendere variabile la reazione del client<sub>G</sub> senza conoscere i dettagli dell'operazione stessa.
- **Applicabilità:** Il Command pattern si presta bene alla parametrizzazione di oggetti sull'azione da eseguire (Callback function) soprattutto nello specificare, accodare ed eseguire richieste molteplici volte. Vi è inoltre il Supporto alle operazioni di "Undo" e "Redo". Vi è inoltre il supporto a transazione in quanto è possibile encapsulare un'azione in una operazione atomica.



#### A.4.2 Observer

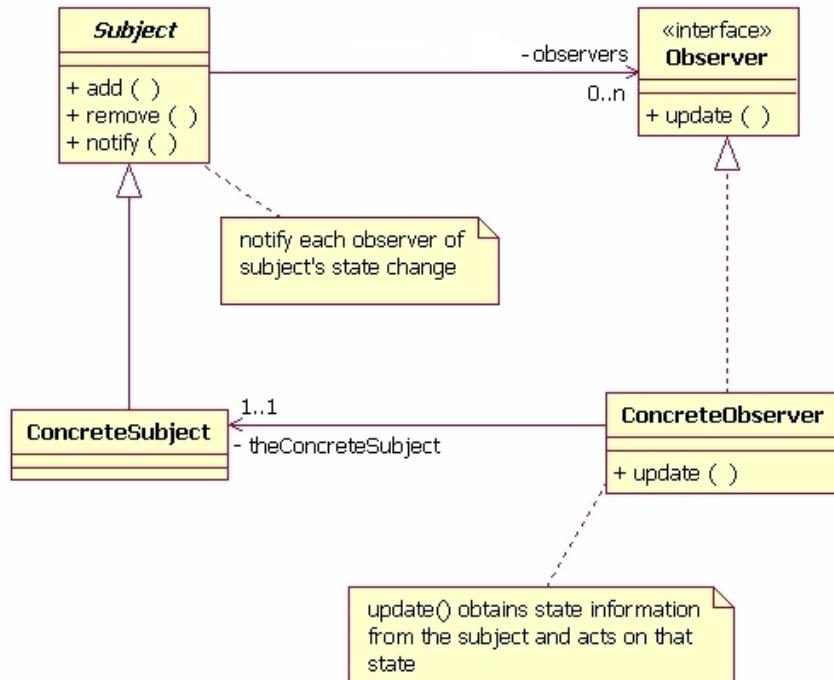


Figura 51: Esempio di applicazione del design pattern observer

- **Scopo:** Definisce una dipendenza “1..n” fra oggetti, riflettendo la modifica di un oggetto su quelli che ad esso dipendono.
- **Motivazione:** Mantenere la consistenza fra oggetti, oltre che al modello e alle viste ad esso collegate. Observer pattern definisce come implementare la relazione di dipendenza: infatti il modello prevede due attori principali che sono:
  - Subject, che effettua le notifiche e contiene le interfacce per registrare e rimuovere gli observer;
  - Observer, che si aggiorna in base alle notifiche che riceve dal subject.
- **Applicabilità:** È utile nei casi in cui si debba associare più “viste” differenti ad una astrazione. Il pattern comporta inoltre un aumento del grado di riuso dei singoli tipi; È indicato inoltre quando il cambiamento di un oggetto richiede il cambiamento di altri oggetti, oppure non si conosce quanti oggetti devono cambiare. È utile anche per notificare oggetti senza fare assunzioni su quali siano questi oggetti ed evita l'accoppiamento “forte”.

## **Elenco delle tabelle**

## Elenco delle figure

1	Diagramma architettura concettuale . . . . .	5
2	Diagramma componenti principali . . . . .	6
3	Actorsystem, visione generale . . . . .	7
4	Package actorsystem::utils . . . . .	9
5	ACtorsystem: package Clientactor . . . . .	14
6	Actorsystem: package messages del clientactor . . . . .	18
7	httpserver, visione generale del package . . . . .	20
8	warehouseman, visione generale del package . . . . .	22
9	Actorsystem: package messages del warehouseman . . . . .	24
10	Package dell'attore di tipo Main . . . . .	27
11	Actorsystem: package messages del main . . . . .	29
12	Userfinder, visione generale del package . . . . .	34
13	Actorsystem: package messages dello userfinder . . . . .	36
14	Userkeeper, visione generale del package . . . . .	39
15	Actorsystem: package messages dello userkeepern . . . . .	41
16	Storefinder, visione generale del package . . . . .	44
17	Actorsystem: package messages dello storefinder . . . . .	46
18	Classe actorsystem::storefinder::messages::Ack . . . . .	47
19	Storekeeper, visione generale del package . . . . .	50
20	Actorsystem: package messages dello storekeeper . . . . .	52
21	Manager, visione generale del package . . . . .	56
22	Actorsystem: package messages del manager . . . . .	58
23	CLI:architettura MVC variante push model . . . . .	60
24	Driver: Packager client . . . . .	63
25	Driver: Package api e interazioni con package client . . . . .	66
26	Driver: Package data e interazioni con package padre . . . . .	71
27	Driver: Package exceptions . . . . .	73
28	CLI, architettura MVC variante push model . . . . .	78
29	CLI, views package, interazioni con controllers e models . . . . .	80
30	CLI: package controllers e interazioni con package models e views . . . . .	84
31	CLI: package models e interazioni con controllers e views . . . . .	86
32	Diagrammi attività - Visione generale . . . . .	97
33	Diagrammi attività - Creazione collezione . . . . .	98
34	Diagrammi attività - Cancellazione collezione . . . . .	99
35	Diagrammi attività - Visualizzazione collezioni . . . . .	100
36	Diagrammi attività - Modifica nome collezione . . . . .	101
37	Diagrammi attività - Inserimento item . . . . .	102
38	Diagrammi attività - Rimozione item . . . . .	103
39	Diagrammi attività - Aggiunta collaboratore . . . . .	104
40	Diagrammi attività - Rimozione collaboratore . . . . .	105
41	Diagrammi attività - Import . . . . .	106
42	Diagrammi attività - Interrogazione del database <sub>c</sub> . . . . .	107
43	Diagrammi attività - Modifica password . . . . .	108



44	Diagrammi attività - Gestione utenti . . . . .	109
45	Connection Singleton . . . . .	111
46	Esempio di applicazione del design pattern MVC . . . . .	139
47	Esempio di applicazione del design pattern singleton . . . . .	140
48	Esempio di applicazione del design pattern builder . . . . .	140
49	Esempio di applicazione del design pattern decorator . . . . .	141
50	Esempio di applicazione del design pattern command . . . . .	142
51	Esempio di applicazione del design pattern observer . . . . .	143