Tracks.Model

Entity #guid : Guid #deleted : boolean = false #version : int = 0 #appliedEvents: array<Tracks.Event.Base> #handlers : array +loadHistory(history: array <Tracks.Event.Base>) +getGuid(): Guid +getVersion(): int +incVersion(increment : int = 1) +getAppliedEvents(): array <Tracks.Event.Base> +getAllAppliedEvents(): array <Tracks.Event.Base> +clearAppliedEvents() +clearAllAppliedEvents() +isCreated() : boolean +isDeleted(): boolean +getAllChildEntities(): array <Entity> +getAllEntities(): array <Entity> #handleDomainEvent(event : Base) #applyEvent(event : Base) #registerEvent(eventName : string, handlerMethod) -getHandlerName(event : Base) : string

AggregateRoot

Guid

+guid: string

+Constructor(guid : string = null)

+toString() : string +create() : Guid +uuid() : string

EntityList

-items : array

-guids : array<Tracks.Model.Guid>

-cursor : int = 0

+add(entity : Entity)

+find(guid : Guid) : Entity

+remove(guid : Guid)

+getAllAppliedEvents(): array <Tracks.Event.Base>

+clearAllAppliedEvents()

+getAllChildEntities(): array <Tracks.Model.Entity>

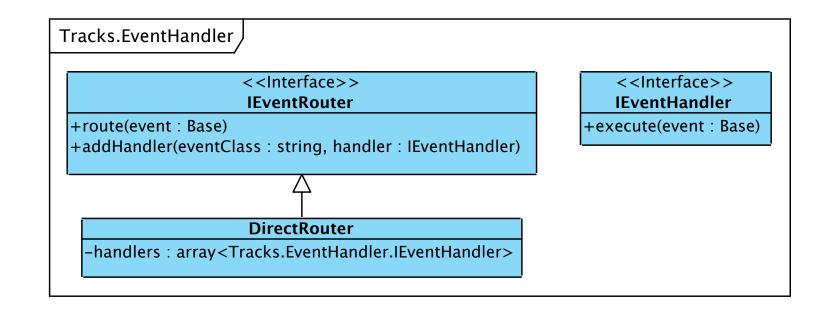
+getAllEntities(): array <Tracks.Model.Entity>

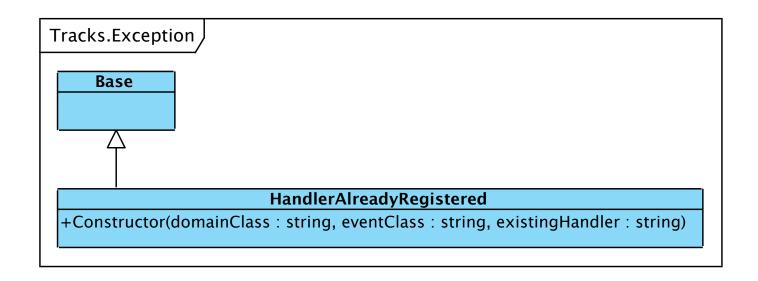
Tracks.Event

Base

+guid : Guid

+Constructor(guid : Guid) +getGuid() : Guid





Tracks.EventStore <<Interface>> <<Interface>> **IEventStore ISnapshotStore** +getAllEvents(guid : Guid) : array <Tracks.Event.Base> +load(guid : Guid) : Entity +getEventsFromVersion(guid : Guid, version : int) : array <Tracks.Event.Base> +save(entity : Entity) +getType(guid : Guid) : string +save(entity : Entity) EventStorage Memory -events : array<Tracks.Event.Base> -entities : array<Tracks.Model.Entity> -createEntity(entity : Entity) SnapshotStorage -incVersion(guid : Guid) File -directory = string ZendDb +Constructor(directory : string) -dbh : Zend Db Adapter Abstract -getFilename(guid : Guid) : string -createEvent(event : Base) -createEntity(entity : Entity) -incVersion(guid : Guid) Memory -snapshots = array -getEntityByGuid(guid : Guid)

Repository

- -snapshotFrequency: int
- -identityMap : array<Tracks.Model.Entity>

 $SNAPSHOT_FREQUENCY : int = 100$

- +Constructor(eventStore : IEventStore, router : IEventRouter, snapshotStore : ISnapshotStore)
- +load(guid : Guid) : AggregateRoot
- +setSnapshotFrequency(numEvents : int = Tracks.EventStore.Repository.SNAPSHOT_FREQUENCY)
- +save(aggregateRoot : AggregateRoot)
- -storeInIdentityMap(entity: Entity)
- -loadFromIdentityMap(guid : Guid) : Entity
- -loadFromHistory(guid : Guid, entity : Entity = null) : Entity
- -loadEntity(guid : Guid, entity : Entity = null) : Entity
- -routeEvents(aggregateRoot : AggregateRoot)
- -loadFromSnapshot(guid : Guid) : Entity
- -updateVerrsionsAndClearEvents(aggregateRoot : AggregateRoot)
- -saveSnapshots(aggregateRoot : AggregateRoot)
- -saveSnapshot(entity : Entity)