pkg

<<record>>
ErrorResponseDTO
- message : String
- statusCode : int
- timestamp : LocalDateTime

0..*

1

<<controller advice>>
RestExceptionHandler

<<record>>
AgendaListResponseDTO
- id : long
- nome : String
- contatos : List<Contato>

<<controller>>
AgendaController
- agendaService : AgendaService
+ createAgenda(requestDto : CreateAgendaDTO) : ResponseEntity<AgendaTypedResponse>
+ getAgendas(nome : String) : Collection<AgendaResponseDTO>>
+ getAgendaById(id : long) : ResponseEntity<AgendaResponseDTO>
+ deleteAgenda(id : long) : ResponseEntity<Void>
+ addContatoToAgenda(agendaId : long, contatoRequest : ContatoRequestDTO) : ResponseEntity<ContatoResponseDTO>
+ removeContatoFromAgenda(agendaId : long, contatoId : long) : ResponseEntity<Void>
+ getContatos(agendaId : long, telefone : String, nome : String) : ResponseEntity<Collection<ContatoResponseDTO>>
+ convertAgendaType(id : long) : ResponseEntity<AgendaTypedResponse>
+ removeContatosByNome(agendaId : long, nome : String) : ResponseEntity<Void>

<<record>>
ContatoResponseDTO
- id : long
- nome : String
- telefone : String

<<interface>>
AgendaTypedResponse
- id : long
- nome : String
- contatos : Object

<<record>>
AgendaMapResponseDTO
- id : long
- nome : String
- contatos : Map<String,ContatoResponseDTO>

<<record>>
AgendaResponseDTO
- id : long
- nome : String

<<record>>
CreateAgendaDTO
- nome : String
- tipo : String

<<record>>
ContatoRequestDTO
- nome : String
- telefone : String

<<service>>
AgendaService
- agendaRepository : AgendaRepository
- entityManager : EntityManager
+ createAgenda(requestDto : CreateAgendaDTO) : AgendaTypedResponse
+ getAgendas(nome : String) : Collection<AgendaResponseDTO>
+ getAgendaById(id : long) : AgendaResponseDTO
+ deleteAgenda(id : long) : void
+ addContatoToAgenda(agendaId : long, contatoRequest : ContatoRequestDTO) : ContatoResponseDTO
+ removeContatoFromAgenda(agendaId : long, contatoId : long) : void
+ getContatos(agendaId : long, telefone : String, nome : String) : Collection<ContatoResponseDTO>
+ convertAgendaType(id : long) : AgendaTypedResponse
+ removeContatosByNome(id : long, nome : String) : void
- mapearAgendaResponseDTO(agenda : Agenda) : AgendaResponseDTO
- mapearContatoResponseDTO(contato : Contato) : ContatoResponseDTO
- mapearAgendaResponse(agenda : Agenda) : AgendaTypedResponse
- mapearParaAgendaListDTO(agendaList : AgendaList) : AgendaListResponseDTO
- mapearParaAgendaMapDTO(agendaMap : AgendaMap) : AgendaMapResponseDTO
+ mapearTipo(tipo : String) : String

<<repository>>
<<interface>>
ContatoRepository

<<entity>>
ContatoImpl
- nome : String
- telefone : String

<<interface>>
Contato
+ getNome() : String
+ getTelefone() : String
+ getId() : long
+ getAgenda() : Agenda
+ setNome(nome : String) : void
+ setTelefone(telefone : String) : void
+ setId(id : long) : void
+ setAgenda(agenda : Agenda) : void

<<repository>>
<<interface>>
AgendaRepository
+ findByNome(nome : String) : Optional<Agenda>
+ updateAgendaType(id : long, novoTipo : String) : void

<>
<<entity>>
Agenda
- id : long
- nome : String
+ getContatos() : Collection<Contato>
+ addContato(contato : Contato) : boolean
+ removeContato(contato : Contato) : boolean
+ getContato(telefone : String) : Contato
+ getId() : long
+ setId(id : long) : void
+ getNome() : void
+ setNome(nome : String) : void

<<entity>>
AgendaMap
- listaContato : HashMap

<<create>>

AgendaFactory
- instance : AgendaFactory = new AgendaFactory()
+ AGENDALIST : String = LIST
+ AGENDAMAP : String = MAP
+ getInstance() : AgendaFactory
+ createAgenda(tipo : String) : Agenda

<<entity>>
AgendaList
- listaContato : ArrayList
+ localizaContato(telefone : String) : int

<<create>>