UserController user:User + setID(id:int):void getID():int - addRole(role:Enum<Role>):void + getRole():Set<Enum<Role>> removeRole(role:Enum<Role>):void applyUsername(username:String):void setEmail(email:String):void ⊦ getEmail():String + setName(name:String):void + getName():String setSurname(name:String):void + getSurname():String + setPassword(password:String):void + getPassword():Byte[] + setPhone(pn:String):void + getPhone():String + setVerified(v:boolean):void getVerified():boolean setCreationDate(d:java.time.LocalDateTime):void + getCreationDate():java.time.LocalDateTime

IndexBean

RegisterBean

ResetBean

resetByUsername():void

+ password:String

⊦ username:Strina

- surname:String

- password:String

username:String

resetByMail():void

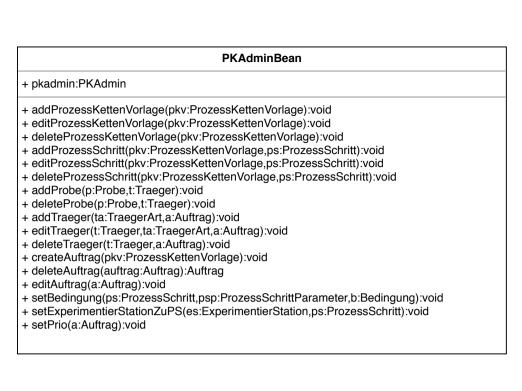
email:String

name:String

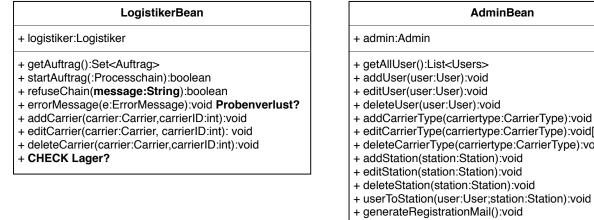
email:String

+ register():void

- login():void



TechnologeBean technologe:Technologe - getAuftrag(List<Stationen>):List<Auftrag> getStationen(List<Auftrag>):List<Stationen> - assignToAuftrag(List<Auftrag): boolean F prioSort(List<Auftrag>):void errorMessage(e:ErrorMessage):void - addComment(ps:Prozessschritt,c:String):void editComment(ps:Prozessschritt,c:String):void deleteComment(ps:Prozessschritt,c:String):void - addProbenComment(p:Probe,c:String):void editProbenComment(p:Probe,c:String):void - deleteProbenComment(p:Probe,c:String):void nextStep(psza:ProzessSchrittZustandsAutomat):void upload():void



AdminBean - admin:Admin - getAllUser():List<Users> - addUser(user:User):void - editUser(user:User):void deleteUser(user:User):void addCarrierType(carriertype:CarrierType):void editCarrierType(carriertype:CarrierType):void[Fehleranfäl - deleteCarrierType(carriertype:CarrierType):void - addStation(station:Station):void - editStation(station:Station):void + deleteStation(station:Station):void

TransporterBean + transporter:Transporter

«extends QualitativeEigenschaftController» QuantitativeEigenschaftController QualitativeEigenschaftController qualitativeEigenschaft:QualitativeEigenschaft - setWert(wert:java.lang.Number):void setEigenschaft(s:String):void getWert():java.lang.Number + getEigenschaft():String - setEinheit(einheit:uom.si.SI):void getEinheit():uom.si.SI

- auftrag:Auftrag

- setPKZ():void

setPrio():void

StandortController

+ setStandort(s:String):void

+ getTraeger():Set<Traeger>

+ setAuftrag(a:Auftrag):void + getAuftrag():Auftrag

ArchivController

- setDatum(d:java.time.LocalDateTime):void

+ getDatum():java.time.LocalDateTime

standort:Standort

+ getStandort():String

+ archiv:Archiv

getLog():AuftragsLog

⊦ getID():int

AuftragController

getPKV():ProzessKettenVorlage

getPKZ():enum("INSTANZIIERT",

"FREIGEGEBEN", "GESTARTET". "ABGEBROCHEN", "DURCHGEFÜHRT"

+ getPrio():enum("KEINE","ETWAS",

"VIEL","HOCH","SEHR HOCH")

- getPS():Set<Prozesschritt>

setLog(al:AuftragsLog):void

ProzessSchrittParameterController + psp:ProzessSchrittParameter - setName(n:String):void

+ getName():String + getEigenschaften():Set<QualitativeEigenschaft> - setEigenschaften(eigenschaft:Set<QualitativeEigenschaft>):void

BedingungController setPSZ(psz:Set<ProzessSchrittParameter>):void + getPSZ():Set<ProzessSchrittParamter> + setEigenschaften(eigenschaften:Set<QualitativeEigenschaft>):void + getEigenschaften():Set<QualitativeEigenschaft> + setBedingung(bedingung:Pair<Set<ProzessSchrittParameter>,Set<QualitativeEigenschaft>>):void getBedingung():Pair<Set<ProzessSchrittParameter>,Set<QualitativeEigenschaft>>

ProzessschrittController + ps:ProzessSchritt + setAuftrag(a:Auftrag):void + getAuftrag():Auftrag + uploaded():boolean + setUploaded():void + getZustand():String setZustand():void + getES():ExperimentierStation + getPSV():ProzessKettenVorlage + setPSLog(psl:ProzessSchrittLog):void + getPSLog():ProzessSchrittLog + setTransportAuftrag(ta:TransportAuftrag):void

ProzessSchrittLogController - pslog:ProzessSchrittLog

+ getTransportAuftrag():TransportAuftrag

setStart(java.time.LocalDateTime):void + getStart():java.time.LocalDateTime

setBeendet(t:java.time.LocalDateTime):

+ getBeendet():java.time.LocalDateTime setArchiviert(t:java.time.LocalDateTime)

+ getArchiviert():java.time.LocalDateTime

ProzessSchrittZustandsAutomatController

+ pszav:ProzessSchrittZustandsAutomatVorlage

ProzessSchrittZustandsAutomatVorlageController

- setProzessSchrittZustandsAutomatVorlageZustaende(pszavz:Set<String>):void

+ getProzessSchrittZustandsAutomatVorlageZustaende():Set<String>

+ psza:ProzessSchrittZustandsAutomat

+ setZustand():void

+ getZustand():String

ProzessschrittVorlageController

psv:ProzessschrittVorlage

- getES():Set<ExperimentierStation>

setES(ExperimentierStation):void

- setID(i:int):void

getID():int

+ controller:Controller

- getZustaende():Set<ProzessSchrittZustand>

setZustaende(psz:Set<ProzessSchrittZustand>):void

- getAusgabeTraeger():Set<Traegerart>

- setAusgabeTraeger(Set<Traegerart>):void

- getEingabeTraeger():Set<Traegerart>

- setEingabeTraeger(Set<Traegerart>):void

- setDauer(d:java.time.Duration):void

getDauer():java.time.Duration

ProzessSchrittArtController

+ setProzessSchrittArt(psa:Set<String>):void

+ getProzessSchrittArt():Set<String>

- psa: ProzessSchrittArt

-set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat): void a set Prozess Schritt Zust and s Automat (psza: Prozess Schritt Zust and s Automat (psz: Prozess Schritt Zust and s Automat (psz: Prozess Schritt Zust and s Automat (psz: Prozess Schrit

get Prozess Schritt Zust and s Automat (): Prozess Schritt Zust and s Automat

setProzessSchrittArt(psa:ProzessSchrittArt):void

getProzessSchrittArt():ProzessSchrittArt

- setProzessSchrittParameter(psp:Set<ProzessSchrittParameter>):void - getProzessSchrittParameter():Set<ProzessSchrittParameter>

- setStart(java.time.LocalDateTime):void

⊦ log:AuftragsLog

getStart():java.time.LocalDateTime

- setBeendet(t:java.time.LocalDateTime):void

AuftragsLogController

- getBeendet():java.time.LocalDateTime

- setArchiviert(t:java.time.LocalDateTime):void

- getArchiviert():java.time.LocalDateTime

TraegerController

traeger:Traeger

setID(id:int):void

getID():int

setArt():void

getArt():String - setTraegerArten(ta:TraegerArt):void

- getTraegerArten():Set<String>

- setStandort(s:Standort):void

getStandort():Standort

- setProben(probe:Set<Probe>):void

- getProben():Set<Probe>

TraegerArtController

traegerart:TraegerArt

setTraegerArten(a:Set<String>):void - getTraegerArten():Set<String>

ProzessKettenVorlageController

BACKUP?

+ pkv:ProzessKettenVorlage

- setID(id:int):void

+ getID():int

+ setPSV(psv:Set<ProzessSchrittVorlage>):void

+ getPSV():Set<ProzessSchrittVorlage>

TransportAuftragController

+ ta:TransportAuftrag

- setZustand(e:enum("ABGEHOLT","ABGELIEFERT")):void

+ getZustand():enum("ABGEHOLT","ABGELIEFERT")

ProbeController

+ probe:Probe

setID(id:String):void

+ addComment(p:Pair<java.time.LocalDateTime,s:String>):void

- getComment():Pair<java.time.LocalDateTime,String>

+ setZustand(e:enum("KAPUTT","VERLOREN","VORHANDEN")):void

+ getZustand():enum("KAPUTT","VERLOREN","VORHANDEN")

+ setStandort(s:Standort):void

+ getStandort():Standort

+ setArchiv(a:Archiv):void

+ getArchiv():Archiv

+ getEigenschaften():Set<QualitativeEigenschaft> + setEigenschaften(eigenschaft:Set<QualitativeEigenschaft>):void

ExperimentierStationController

- experimentierstation: Experimentier Station

+ getESID():int

+ getESStandort():String

+ setStatus(e:enum("VERFÜGBAR","BESETZT","KAPUTT")):void

+ getStatus():enum("VERFÜGBAR","BESETZT","KAPUTT")

+ addNextPS(ps:ProzessSchritt):void

+ peekNextPS():ProzessSchritt

+ popNextPS():void