

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

PKAdminBean
+ pkadmin:PKAdmin
+ addProzessKettenVorlage(pkv:ProzessKettenVorlage):void + editProzessKettenVorlage(pkv:ProzessKettenVorlage):void + deleteProzessKettenVorlage(pkv:ProzessKettenVorlage):void + addProzessSchritt(pkv:ProzessKettenVorlage,ps:ProzessSchritt):void + editProzessSchritt(pkv:ProzessKettenVorlage,ps:ProzessSchritt):void + deleteProzessSchritt(pkv:ProzessKettenVorlage,ps:ProzessSchritt):void + addProbe(p:Probe,t:Traeger):void + deleteProbe(p:Probe,t:Traeger):void + addTraeger(ta:TraegerArt,a:Auftrag):void + editTraeger(t:Traeger,ta:TraegerArt,a:Auftrag):void + deleteTraeger(t:Traeger,a:Auftrag):void + createAuftrag(pkv:ProzessKettenVorlage):void + deleteAuftrag(auftrag:Auftrag):Auftrag + editAuftrag(a:Auftrag):void + setBedingung(ps:ProzessSchritt,psp:ProzessSchrittParameter,b:Bedingung):void + setExperimentierStationZuPS(es:ExperimentierStation,ps:ProzessSchritt):void + setPrio(a:Auftrag):void

TechnologieBean
+ technologie:Technologie
+ getAuftrag(List<Stationen>):List<Auftrag> + getStationen(List<Auftrag>):List<Stationen> + assignToAuftrag(List<Auftrag>):boolean + 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

LogistikerBean
+ logistiker:Logistiker
+ getAuftrag():Set<Auftrag> + startAuftrag(:Processchain):boolean + refuseChain(message:String):boolean + errorMessage(e:ErrorMessage):void Probenverlust? + addCarrier(carrier:Carrier,carrierID:int):void + editCarrier(carrier:Carrier,carrierID:int):void + deleteCarrier(carrier:Carrier,carrierID:int):void + CHECK Lager?

TransporterBean
+ transporter:Transporter
+

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 Fehleranfäll + deleteCarrierType(carriertype:CarrierType):void + addStation(station:Station):void + editStation(station:Station):void + deleteStation(station:Station):void + userToStation(user:User,station:Station):void + generateRegistrationMail():void + BACKUP?

«extends QualitativeEigenschaftController» QuantitativeEigenschaftController
+ setWert(wert:java.lang.Number):void + getWert():java.lang.Number + setEinheit(einheit:uom.si.SI):void + getEinheit():uom.si.SI

QualitativeEigenschaftController
+ qualitativeEigenschaft:QualitativeEigenschaft
+ setEigenschaft(s:String):void + getEigenschaft():String

ProzessSchrittParameterController
+ psp:ProzessSchrittParameter
+ setName(n:String):void + getName():String + getEigenschaften():Set<QualitativeEigenschaft> + setEigenschaften(eigenschaft:Set<QualitativeEigenschaft>):void

BedingungController
+ controller:Controller
+ 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>>

ProzessschrittVorlageController
+ psv:ProzessschrittVorlage
+ getES():Set<ExperimentierStation> + setES(ExperimentierStation):void + setID(i:int):void + getID():int + 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 + setProzessSchrittZustandsAutomat(psza:ProzessSchrittZustandsAutomat):void + getProzessSchrittZustandsAutomat():ProzessSchrittZustandsAutomat + setProzessSchrittArt(psa:ProzessSchrittArt):void + getProzessSchrittArt():ProzessSchrittArt + setProzessSchrittParameter(psp:Set<ProzessSchrittParameter>):void + getProzessSchrittParameter():Set<ProzessSchrittParameter>

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 + getTransportAuftrag():TransportAuftrag

ProzessSchrittLogController
+ pslog:ProzessSchrittLog
+ setStart(java.time.LocalDateTime):void + getStart():java.time.LocalDateTime + setBeendet(t:java.time.LocalDateTime):void + getBeendet():java.time.LocalDateTime + setArchiviert(t:java.time.LocalDateTime):void + getArchiviert():java.time.LocalDateTime

ProzessSchrittZustandsAutomatController
+ psza:ProzessSchrittZustandsAutomat
+ setZustand():void + getZustand():String

ProzessSchrittArtController
+ psa: ProzessSchrittArt
+ setProzessSchrittArt(psa:Set<String>):void + getProzessSchrittArt():Set<String>

ProzessSchrittZustandsAutomatVorlageController
+ pszav:ProzessSchrittZustandsAutomatVorlage
+ setProzessSchrittZustandsAutomatVorlageZustaende(pszavz:Set<String>):void + getProzessSchrittZustandsAutomatVorlageZustaende():Set<String>

AuftragController
+ auftrag:Auftrag
+ getID():int + getPKV():ProzessKettenVorlage + getLog():AuftragsLog + setLog(al:AuftragsLog):void + getPKZ():enum("INSTANZIERT", "FREIGE GEBEN", "GESTARTET", "ABGEBROCHEN", "DURCHGEFÜHRT") + setPKZ():void + getPrio():enum("KEINE", "ETWAS", "VIEL", "HOCH", "SEHR HOCH") + setPrio():void + getPS():Set<Prozessschritt>

ProzessKettenVorlageController
+ 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 + getID():String + 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:ExperimentierStation
+ getESID():int + getESStandort():String + setStatus(e:enum("VERFÜGBAR", "BESETZT", "KAPUTT")):void + getStatus():enum("VERFÜGBAR", "BESETZT", "KAPUTT") + addNextPS(ps:ProzessSchritt):void + peekNextPS():ProzessSchritt + popNextPS():void

AuftragsLogController
+ log:AuftragsLog
+ setStart(java.time.LocalDateTime):void + getStart():java.time.LocalDateTime + setBeendet(t:java.time.LocalDateTime):void + 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>

StandortController
+ standort:Standort
+ setStandort(s:String):void + getStandort():String + getTraeger():Set<Traeger>

ArchivController
+ archiv:Archiv
+ setAuftrag(a:Auftrag):void + getAuftrag():Auftrag + setDatum(d:java.time.LocalDateTime):void + getDatum():java.time.LocalDateTime

TraegerArtController
+ traegerart:TraegerArt
+ setTraegerArten(a:Set<String>):void + getTraegerArten():Set<String>

IndexBean
+ username:String + password:String
+ login():void

RegisterBean
+ username:String + name:String + surname:String + email:String + password:String
+ register():void

ResetBean
+ username:String + email:String
+ resetByUsername():void + resetByEmail():void