

Controller

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 + getCurrentLanguage():String + setDefaultLanguage(l:String):void

IndexBean
+ username:String + password:String
+ login():void + setLanguage(l:String):void + getCurrentLanguage():String

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

ResetBean
+ username:String + email:String
+ resetByUsername():void + resetByEmail():void + setLanguage(l:String):void + getCurrentLanguage():String

LoggedInBean
+ user:User
+ setView():void + logout():void

PKAdminBean
+ pkadmin:User
+ getAuftragsList():Set<Auftrag> + getProzessSchrittList():Set<ProzessSchritt> + addProzessKettenVorlage(pkv:ProzessKettenVorlage):void + editProzessKettenVorlage(pkv:ProzessKettenVorlage):void + deleteProzessKettenVorlage(pkv:ProzessKettenVorlage):void + addPSToAuftrag(ps:ProzessSchritt):void + addProzessSchrittVorlage(pkv:ProzessKettenVorlage,psv:ProzessSchrittVorlage):void + editProzessSchrittVorlage(pkv:ProzessKettenVorlage,psv:ProzessSchrittVorlage):void + deleteProzessSchrittVorlage(pkv:ProzessKettenVorlage,psv:ProzessSchrittVorlage):void + addProbe(p:Probe):void + deleteProbe(p:Probe):void + addTraeger(t:Traeger):void + editTraeger(t:Traeger):void + deleteTraeger(t:Traeger):void + createAuftrag(pkv:ProzessKettenVorlage):void + deleteAuftrag(auftrag:Auftrag):void + editAuftrag(a:Auftrag):void + setBedingung(ps:ProzessSchritt,b:Bedingung):void + editBedingung(ps:ProzessSchritt,b:Bedingung):void + removeBedingung(ps:ProzessSchritt,b:Bedingung):void + setExperimentierStationZuPS(es:ExperimentierStation,ps:ProzessSchritt):void + editExperementierStationZuPS(es:ExperementierStation,ps:ProzessSchritt):void + setPrio(a:Auftrag):void + editPrio(a:Auftrag):void + deleteProzessSchritt(ps:ProzessSchritt):void + createZustandsAutomat(pszav:ProzessSchrittZustandsAutomatVorlage):void + removeZustandsAutomat(pszav:ProzessSchrittZustandsAutomatVorlage):void + editZustandsAutomat(pszav:ProzessSchrittZustandsAutomatVorlage):void + setFreigegeben(a:Auftrag):void + stopAuftrag(a:Auftrag):void + viewESUsage():void + viewESUsage(es:ExperimentierStation):void + exportJSON():void

AdminBean
+ admin:User
+ getAllUser():Set<User> + addUser(user:User):void + editUser(user:User):void + deleteUser(user:User):void + addTraegerArt(ta:TraegerArt):void + editTraegerArt(ta:TraegerArt):void + deleteTraegerArt(ta:TraegerArt):void + addStation(es:ExperimentierStation):void + editStation(es:ExperimentierStation):void + deleteStation(es:ExperimentierStation):void + userToStation(user:User;es:ExperimentierStation):void + generateRegistrationMail():void + getES():Set<ExperimentierStation> + editAuftragTime(a:Auftrag):void + backup():void

TransporterBean
+ transporter:User
+ getAuftragList():Set<TransportAuftrag> + setAuftragStatus():void + reportLostProbe(p:Probe):void + reportLostProbe(id:int):void

BedingungController
+ controller: 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>>

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

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

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

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>

LogistikerBean
+ logistiker:User
+ getTraegerList():Set<Traeger> + createTraeger():Traeger + deleteTraeger(t:Traeger):void + getAuftrag():Set<Auftrag> + zuordnenProbe(a:Auftrag,p:Probe):void + zuordnenTraeger(a:Auftrag,t:Traeger):void + getProben():Set<Probe> + getProben(archiviert:boolean):Set<Probe> + getProbenStandort(p:Probe):Standort + startAuftrag(a:Auftrag):void + refuseAuftrag(a:Auftrag,message:String):void + errorMessage():String + zuordnen(a:Auftrag):void + getAngeforderteProben(a:Auftrag):Set<Pair<QualitativeEigenschaft,int>>

TechnologieBean
+ technologie:User
+ getStationen():Set<ExperimentierStation> + getAuftrag():Set<Auftrag> + setAuftragsZustand(a:Auftrag,zustand:Enum("INSTANZIERT", "FREIGEgeben","GESTARTET","ABGEBROCHEN", "DURCHGEFÜHRT")):void + reportBroken(es:ExperimentierStation):void + assignToAuftrag(a:Auftrag):void + prioSort(prio:Set<Auftrag>):void + createUrformend(id: String):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 + viewToBeUploaded():Set<Probe> + upload(p:Probe):void + reportLostProbe(p:Probe):void + reportLostProbe(id:int):void + errorMessage(e:String):void

ProzessSchrittVorlageController
+ psv:ProzessSchrittVorlage
+ getES(b: Set<Bedingung>):Set<ExperimentierStation> + setES(): void + setES(es: ExperimentierStation):void + setID(i:int):void + getID():int + getZustaende():Set<ProzessSchrittZustandsAutomat> + setZustaende(psz:Set<ProzessSchrittZustand>):void + getAusgabeTraeger():Set<TraegerArt> + setAusgabeTraeger(at:Set<TraegerArt>):void + getEingabeTraeger():Set<TraegerArt> + setEingabeTraeger(et: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(bsp:Set<ProzessSchrittParameter>):void + getProzessSchrittParameter():Set<ProzessSchrittParameter> + getEigenschaften():Set<QualitativeEigenschaft> + setEigenschaften(qe:Set<QualitativeEigenschaft>):void + setAngeforderteProben(Set<Pair<QualitativeEigenschaft,int>>):void

ProzessSchrittController
+ ps:ProzessSchritt
+ setAuftrag(a:Auftrag):void + getAuftrag():Auftrag + uploaded():boolean + setUploaded():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

ProzessSchrittZustandsAutomatVorlageController
+ pszav:ProzessSchrittZustandsAutomatVorlage
+ setProzessSchrittZustandsAutomatVorlageZustaende(pszavz:Set<String>):void + getProzessSchrittZustandsAutomatVorlageZustaende():Set<String> + getDefaults():Set<String>{"Angenommen","In Bearbeitung","Bearbeitet","Weitergeleitet"}

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

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

TransportAuftragController
+ ta:TransportAuftrag
+ setZustand(e:enum("ABGEHOLT","ABGELIEFERT")):void + getZustand():enum("ABGEHOLT","ABGELIEFERT") + getDestination():ExperimentierStation + getDefaults():Set<String>{"Abgeholt","Abgegeben"}

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

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

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

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

ProzessKettenVorlageController
+ pkv:ProzessKettenVorlage
+ setID(id:int):void + getID():int + setPSV(psv:Set<ProzessSchrittVorlage>):void + getPSV():Set<ProzessSchrittVorlage>

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(eigenschaften:Set<QualitativeEigenschaft>):void

ExperimentierStationController
+ experimentierstation:ExperimentierStation
+ getESID():int + getESStandort():Standort + setStatus(e:enum("VERFÜGBAR","BESETZT","KAPUTT")):void + getStatus():enum("VERFÜGBAR","BESETZT","KAPUTT") + addNextPS(ps:ProzessSchritt):void + peekNextPS():ProzessSchritt + popNextPS():ProzessSchritt + getUsage():int + setEingabe(ta: Set<TraegerArt>): void + setAusgabe(ta: Set<TraegerArt>): void + setBedingung(b: Set<Bedingung>): void