

UserController
+ user:User
+ addRole(role:Role):void
+ deleteRole(role:Role):void
+ applyUsername(username:String):void
+ changeEmail(email:String):void
+ changeName(name:String):void
+ changeSurname(name:String):void
+ changePassword(password:String):void

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 + resetByMail():void

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):void
+ 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
<pre> + getAllUser():List<User> + addUser(user:User):void + editUser(user:User):void + deleteUser(user:User):void + addCarrierType(carriertype:CarrierType):void + editCarrierType(carriertype:CarrierType):void + 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? </pre>

ProzessschrittController
+ ps:ProzessSchritt
+ setAuftrag(a:Auftrag):void
+ getAuftrag():Auftrag
+ uploaded():boolean
+ setUploaded():void
+ getZustand():String
+ setZustand():void
+ getES():ExperimentierStation
+ getPSV():ProzesskettenVorlage
+ setPSLog(ps: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

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

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

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

```

AuftragController
+ auftrag:Auftrag

+ getId():int
+ getPKV():ProzessKettenVorlage
+ getLog():AuftragsLog
+ setLog(al:AuftragsLog):void
+ getPKZ(al:AuftragsLog):enum("INSTANZIIERT",
"FREIGEgeben", "GESTARTET",
"ABGEBROCHEN", "DURCHGEFÜHRT")
+ setPKZ():void
+ getPrior():enum("KEINE", "ETWAS",
"VIEL", "HOCH", "SEHR HOCH")
+ setPrior():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

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

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(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(Set<String>):void + getTraegerArten():Set<String>

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