

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

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.LocalDate):void
+ getStart():java.time.LocalDate
+ setBeendet(t:java.time.LocalDate):void
+ getBeendet():java.time.LocalDate
+ setArchiviert(t:java.time.LocalDate):void
+ getArchiviert():java.time.LocalDate

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

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

PKAdminBean
+ pkadmin:PKAdmin
+ addProzessKettenVorlage

ProzessschrittVorlageController	
	+ psv:ProzessschrittVorlage
	+ getES():Set<Experimentierstation>
	+ setES(ExperimentierStation):void
	+ setID(i:int):void
	+ getID():int
	+ getZustaende():Set<ProzessSchrittZustand>
	+ setZustaende(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(psa:ProzessSchrittZustandsAutomat):void
	+ getProzessSchrittZustandsAutomat():ProzessSchrittZustandsAutomat
	+ setProzessSchrittArt(psa:ProzessSchrittArt):void
	+ getProzessSchrittArt():ProzessSchrittArt

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

```

+ 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():void
+ upload():void
+ weiterleiten???

```

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

AuftragsLogController
+ log:AuftragsLog
+ setStart(java.time.LocalDate):void
+ getStart():java.time.LocalDate
+ setBeendet(t:java.time.LocalDate):void
+ getBeendet():java.time.LocalDate
+ setArchiviert(t:java.time.LocalDate):void
+ getArchiviert():java.time.LocalDate

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>

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

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

TransporterBean
+ transporter:Transporter
+

```

AuftragController
+ Auftrag:Auftrag

+ getID():int
+ getPKV():ProzesskettenVorlage
+ getLog():AuftragsLog
+ setLog(al:AuftragsLog):void
+ getPKZ():enum("INSTANZIERT",
"FREIGEgeben", "GESTARTET",
"ABGEBROCHEN", "DURCHGEFÜHRT")
+ setPKZ():void
+ getPrior():enum("KEINE", "ETWAS",
"VIEL", "HOCH", "SEHR HOCH")
+ setPrior():void
+ getPS():Set<Prozessschritt>

```

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

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

	<b>AdminBean</b>
	+ admin:Admin
ist?	+ getAllUser():List<User> + addUser(user:User):void + editUser(user:User):void + deleteUser(user:User):void + addCarrierType(carriertype:CarrierType):void + editCarrierType(carriertype:CarrierType):void <b>Fehleranfä</b> + 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?

ProzesskettenVorlageController
+ pkv:ProzesskettenVorlage
+ setID(id:int):void + getID():int + getPSV(Set<ProzessschrittVorlage>):void + getPSV():Set<ProzessschrittVorlage>

TransportAuftragController
+ ta:TransportAuftrag
+ setZustand(enum("ABGEHOLT","ABGELIEFERT")):void + getZustand():enum("ABGEHOLT","ABGELIEFERT")

ProbeController
+ probe:Probe
+ setID(id:String):void + getID():String + addComment(p:Pair<java.time.LocalDate,String>):void + getComment():Pair<java.time.LocalDate,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"})
+ getStatus():enum{"VERFÜGBAR","BESETZT"}
+ addNextPS(ProzessSchritt):void
+ peekNextPS():ProzessSchritt
+ popNextPS(ProzessSchritt):void

```