





ReviewFacade	GraphQLFacade	MaterialFacade	MemberFacade	ProblemFacade	ReservationFacade	PermissionFacade
+ find(id: UUID): Try[Review] + create(input: CreateReviewInput): Review - updateRequestStatus(requestId: UUID): Boolean - updateReservation(requestId: UUID): Boolean - findRequest(requestId: UUID): RequestEntity - createReview(reviewEntity: ReviewEntity): ReviewEntity - createReservations(requestEntity: RequestEntity): Boolean - requiredApproval: Int	+ executeQuery(form: GraphQLQuery): Future[Json]+ findAll: List[Department] + executeQuery(document: Document, operationName: Option[String], variables: Option[Json]): Future[Json]+ roles(id: UUID): List[Role]	+ find(id: UUID): Material + create(input: CreateMaterialInput): Material + delete(input: DeleteMaterialInput): Department	+ find(id: UUID): Member + findByUsername(username: String): Member + requests(id: UUID): List[Request] + reviews(id: UUID): List[Review] + reservations(id: UUID): List[Reservation] + roles(id: UUID): List[Role] + departments(id: UUID): List[Department]	+ find(id: UUID): Try[Problem] + create(input: CreateProblemInput): Try[Problem]	+ find(id: UUID): Reservation	+ find(name: String): Permission + permissions: List[Permission]

BaseFacade
+ findById(id: UUID): Member + getIdentity(member: Member): Identity + login(username: String, password: String): MemberWithToken - findOrElseCreateMember(existingMember: ExistingMemberEntity): MemberEntity - createToken(memberId: UUID): String

AuthenticationFacade	AuthorizationFacade	RoleFacade	DepartmentFacade	RequestFacade	SpaceFacade	ImageUploadFacade
+ findById(id: UUID): Member + getIdentity(member: Member): Identity + login(username: String, password: String): MemberWithToken - findOrElseCreateMember(existingMember: ExistingMemberEntity): MemberEntity - createToken(memberId: UUID): String	+ hasAccess(access: Access, accesses: List[Access]): Boolean + permissions(memberId: UUID, departmentId: UUID): List[Permission] + accesses(memberId: UUID, departmentId: UUID): List[Access] + accesses(memberId: UUID, departmentName: String): List[Access] - getAccessValidator(access: Access)(memberAccesses: List[Access]): Boolean	+ find(id: UUID): Try[Role] + findByName(department: String, name: String): Role + members(id: UUID): List[Member] + create(input: CreateRoleInput): Role + update(input: UpdateRoleInput): Role + delete(input: DeleteRoleInput): Department + assign(input: AssignRoleInput): Role + revoke(input: RevokeRoleInput): Role	+ find(id: UUID): Department + findAll: List[Department] + findByName(name: String): Department + roles(id: UUID): List[Role] + materials(id: UUID): List[Material] + spaces(id: UUID): List[Space] + create(input: CreateDepartmentInput): Department - isDepartmentNameValid(name: String): Boolean	+ find(id: UUID): Request + reviews(id: UUID): List[Review] + create(input: CreateRequestInput): Request + cancel(input: CancelRequestInput): Member - isMemberOwnRequest(member: Member, request: RequestEntity) + findByReviewerId(reviewerId: UUID): List[ReviewEntity]	+ find(id: UUID): Try[Space] + findByName(department: String, name: String): Space + searchByName(name: String): List[Space] + findAll: List[Space] + requests(space: Space): List[Request] + reservations(space: Space): List[Reservation] + problems(space: Space): List[Problem] + create(input: CreateSpaceInput): Space + update(input: UpdateSpaceInput): Space - isSpaceNameValid(name: String): Boolean	+ createSpaceUploadSignUrl(spaceId: String): String - createSignatureString(spaceId: String, expiredAt: Long): String + signString(s: String, privateKey: PrivateKey): String - getPrivateKey: PrivateKey - removePrivateKeyTag(privateKey: String): String

ProblemPersist	SpacePersist	MemberRolePersist	MaterialPersist	ReviewPersist	ReservationPersist
+ find(id: UUID): Option[ProblemEntity] + findBySpaceId(spaceId: UUID): List[ProblemEntity] + create(problem: ProblemEntity): Boolean	+ space(query: String, department: Option[String], tags: List[String], capacity: Option[Int]): List[SpaceEntity]	+ find(memberId: UUID, roleId: UUID): Option[MemberRoleEntity] + insert(memberRoleEntity: MemberRoleEntity): Boolean + delete(memberId: UUID, roleId: UUID): Boolean	+ find(id: UUID): Option[MaterialEntity] + findByDepartmentId(departmentId: UUID): List[MaterialEntity] + create(materialEntity: MaterialEntity): Boolean + delete(id: UUID): Boolean	+ find(id: UUID): Option[ReviewEntity] + findByRequestId(requestId: UUID): List[ReviewEntity] + findByReviewerId(reviewerId: UUID): List[ReviewEntity] + insert(reviewEntity: ReviewEntity): Boolean	+ find(id: UUID): Option[ReservationEntity] + findBySpaceId(spaceId: UUID): List[ReservationEntity] + findByClientId(clientId: UUID): List[ReservationEntity] + insert(reservationEntity: ReservationEntity): Boolean

AuthenticationPersist	RolePersist	DepartmentPersist	MemberPersist	RequestPersist	SearchPersist
+ findByMemberId(memberId: UUID): List[MemberDepartmentRelationEntity]	+ find(id: UUID): Option[RoleEntity] + findByName(departmentName: String, name: String): Option[RoleEntity] + findByDepartmentId(departmentId: UUID): List[RoleEntity] + findByMemberId(memberId: UUID): List[RoleEntity] + insert(roleEntity: RoleEntity): Boolean + update(roleEntity: RoleEntity): Boolean + delete(id: UUID): Boolean	+ find(id: UUID): Option[DepartmentEntity] + findAll: List[DepartmentEntity] + findByName(name: String): Option[DepartmentEntity] + findByRoleId(roleId: UUID): List[DepartmentEntity] + insert(departmentEntity: DepartmentEntity): Boolean	+ find(id: UUID): Option[MemberEntity] + findByUsername(username: String): Option[MemberEntity] + findByRoleId(roleId: UUID): List[MemberEntity] + insert(member: MemberEntity): Option[MemberEntity]	+ find(id: UUID): Option[MemberEntity] + findByUsername(username: String): Option[MemberEntity] + findByRoleId(roleId: UUID): List[MemberEntity] + insert(member: MemberEntity): Option[MemberEntity]	+ find(id: UUID): Option[SpaceEntity] + searchByName(name: String): List[SpaceEntity] + findByName(departmentName: String, spaceName: String): Option[SpaceEntity] + findByDepartmentId(departmentId: UUID): List[SpaceEntity] + findAll: List[SpaceEntity] + insert(space: SpaceEntity): Boolean + update(space: SpaceEntity): Boolean + delete(id: UUID): Boolean



