Co.edu.unbosque.exception

DontExistException

PasswordNotEncryptedException

PasswordNotEncryptedException

PasswordNotEncryptedException(String mensaje)

co.edu.unbosque.model co.edu.unbosque.model.entity co.edu.unbosque.model.dtos UserDto ResidentDto ServiceRFQDto -resident:boolean -idResident: Int Resource User rfqDateTime: LocalDateTime -blocked:boolean -idProperty:PropertyDto ServiceRFQ -idResource: Int -nameUser:String -userName:UserDto - userName: User -resident:boolean -owner:boolean propertyId: PropertyDto -attempt:int -blocked:boolean -descripcion:String rfqDateTime: LocalDateTime -password:String svcProviderId: ServiceProviderDto -nameUser:String -type:String +ResidentDto() - userName: User -lastLogin:LocalDateTime - requestDescription: String -attempt:int +ResidentDto(int:idResidentidResident, Property:idProperty, +Resource() -emailUser:String propertyId: Property -password:String User:userName, boolean:owner) + ServiceRFQDTO() -propertyAdmin:boolean +Resource(int:idResource, String:descripcion, String:type) svcProviderId: ServiceProvider -lastLogin:LocalDateTime + ServiceRFQDTO(int rfqId, LocalDateTime rqstDateTime, User +getters() - requestDescription: String userName, PropertyDto:propertyId, ServiceProviderDto:svcProviderId, -emailUser:String +setters() +UserDto(boolean:resident, boolean:blocked, String:nameUser, String requestDescription, LocalDateTime svcDateTime) -propertyAdmin:boolean + ServiceRFQ() int:attempt, String:password, LocalDateTime:lastLogin, ResourceDto + ServiceRFQ(int rfqId, LocalDateTime rqstDateTime, User String:emailUser, boolean:property) + setters userName, Property propertyld, ServiceProvider svcProviderId, , -idResource: Int +User(boolean:resident, boolean:blocked, String:nameUser, ResourceBooking String requestDescription, LocalDateTime svcDateTime) -descripcion:String int:attempt, String:password, LocalDateTime:lastLogin, +setters() -type:String ServiceRequestDto String:emailUser, boolean:property) - bookingId: int + setters - userName: User +getters() +ResourceDto() PropertyDto +setters() propertyResourceId: PropertyResource ResourceDto(int:idResource, String:descripcion, String:type) rqstDateTime: LocalDateTime bookingDateTime: LocalDateTime - userName: UserDto -idProperty: int +setters() propertyId: PropertyDto bookingStartDate: LocalDateTime -nameProperty: String svcProviderId: ServiceProviderDto bookingEndDate: LocalDateTime -city: String Visitor requestDescription: String bookingCost: double -address: String svcDateTime: LocalDateTime -idVisitor: Int paymentComplete: boolean -area:int -nameProperty:String PropertyResourceDto -nameVisitor: String -priceProperty:double + ServiceRequestDto() -city:String + ResourceBooking() -advisor: String -room:int -proResId: int -address:String + ServiceRequestDto(int rqstId, LocalDateTime rqstDateTime, UserDto + ResourceBooking(int bookingId, String userName, int -bathroom:int -resld:ResourceDto -dateTimeVisitor: LocalDateTime userName, PropertyDto propertyId, ServiceProviderDto svcProviderId, propertyResourceId, LocalDateTime bookingDateTime, -prold:PropertyDto -descripcion:String -priceProperty:double -idProperty:Property String requestDescription, LocalDateTime svcDateTime) LocalDateTime bookingStartDate, LocalDateTime -Admi:User -minPrice:double l -room:int + getters -bathroom:String -minTimeH:int -rent:boolean bookingEndDate, double bookingCost, boolean + setters -description:String +Visitor(int:idVisitor, String:nameVisitor, String:advisot, -sale:boolean capacity:int paymentComplete) -propertyAdmi:User LocalDateTime:dateTimeVisitor, Property:idProperty) -bookEmail:String + getters -sale:boolean +PropertyDto() +getters() + setters +PropertyDto(int:idProperty, String:nameProperty, String:city, +PropertyResourceDto() +PropertyResourceDto(int:proResId, Resource:resId, Property:proId, String:address, int:area, double:priceProperty, int:room, int:bathdhrom, ServiceProviderDto +Property(int:idProperty, String:nameProperty, String:city, String:descripcion, User:Admi, boolean:Rent, boolean:sale) double:minPrice, int:minTimeH, int:capacity, String:bookEmail) String:address, int:area, double:priceProperty, int:room, ServiceProvider +getters() providerId: int int:bathdhrom, String:descripcion, User:Admi, boolean:Rent, +setters() +setters() - providerEmail: String providerId: int | boolean:sale) - serviceDescription: String ServiceRequest providerEmail: String +getters() - serviceType: String serviceDescription: String - servicePrice: double +setters() serviceType: String rqstDateTime: LocalDateTime ResourceBookingDto servicePrice: double - userName: User + ServiceProviderDto(int providerId, String providerEmail, String -idVisitor: Int bookingId: int propertyId: Property ServiceProvider() -nameVisitor: String serviceDescription, String serviceType, double servicePrice) - userName: UserDto - svcProviderId: ServiceProvider PropertyResource ServiceProvider(int providerId, String providerEmail, String -advisor: String propertyResourceId: PropertyResourceDto - requestDescription: String serviceDescription, String serviceType, double servicePrice) -dateTimeVisitor: LocalDateTime bookingDateTime: LocalDateTime + setters -prold: int + getters -idProperty:PropertyDto bookingStartDate: LocalDateTime -resld:Resource bookingEndDate: LocalDateTime + setters -prold:Property + ServiceRequest(int rqstld, LocalDateTime rqstDateTime, User bookingCost: double -minPrice:double +VisitorDto(int:idVisitor, String:nameVisitor, String:advisot, userName, Property propertyId, ServiceProvider svcProviderId, paymentComplete: boolean LocalDateTime:dateTimeVisitor, PropertyDto:dProperty) -minTimeH:int String requestDescription) Resident -availabily:boolean +getters() + ResourceBookingDto() + getters -idResident: Int ResourceBookingDto(int bookingId, String userName, int +setters() -bookEmail:String + setters -idProperty:Property PropertyDto:propertyResourceId, LocalDateTime bookingDateTime, +PropertyResource() -userName:User LocalDateTime bookingStartDate, LocalDateTime bookingEndDate, +PropertyResource(int:prold, Resource:resld, Property:prold, double bookingCost, boolean paymentComplete) -owner:boolean double:minPrice, int:minTimeH, int:availabily, String:bookEmail) + setters +getters() +Resident(int:idResidentidResident, Property:idProperty, +setters() User:userName, boolean:owner) +setters() co.edu.unbosque.model.persistence ServiceRequiestDao ServiceRequiestDao -em:EntityManager co.edu.unbosque.model.view -em:EntityManager +create(ServiceRequiestDao serviceRequiestDao):void +create(ServiceRequiestDao serviceRequiestDao):void +update(ServiceRequiestDao serviceRequiestDao):void ShowPropertyBean +update(ServiceRequiestDao serviceRequiestDao):void AdmiUserBean AdmiPropertyBean +delete(ServiceRequiestDao serviceRequiestDao):void +delete(ServiceRequiestDao serviceRequiestDao):void PropertyDao -logger:Logger +getAll():List<ServiceRequiestDao> -logger:Logger +getAll():List<ServiceRequiestDao> -logger:Logger -showPropertyService:ShowPropertyService -admiUserService:AdmiUserService +find(Integer id) :ServiceRequiestDao -em:EntityManager +find(Integer id) :ServiceRequiestDao -admiPropertiesService:AdmiPropertiesService -nameUsers:List<String> -userDto:UserDto -propertyDto:PropertyDto +create(PropertyDao propertyDao):void -propertiesDto:List<PropertyDto> -residentDto:ResidentDto -administradores:List<String> +update(PropertyDao propertyDao):void -visitorDto:VisitorDto -listNameProperty:List<String> -minDateTime:Date +delete(PropertyDao propertyDao):void VisitorDao | +init:void +init:void +getAll():List<PropertyDao> +createProperty():void +init:void -em:EntityManager +createUser():String +find(Integer id) :PropertyDao +update():void +createAppointmentVisitor():void +createResident():String +create(VisitorDao visitorDao):void +getters() +update():void +update():void +update(VisitorDao visitorDao):void +setters() +getters() +getters() PropertyResourceDao +delete(VisitorDao visitorDao):void +setters() +setters() +getAll():List<VisitorDao> -em:EntityManager <<InterfaceDao<T,K>>> +find(Integer id) :VisitorDao AdmiPropertyResourceBean ProveedorServiceBean +create(PropertyResourceDao propertyResourceDao):void -logger:Logger AdmiResourceBean +update(PropertyResourceDao propertyResourceDao):void -logger:Logger -admiPropertyResourceService:AdmiPropertyResourceService +create(T t):void +delete(PropertyResourceDao propertyResourceDao):void -proveedorAPIServiceProveedorAPIService -listNameProperty:List<String> -logger:Logger +update(T t):void +getAll():List<PropertyDao> userDto:UserDto -admiResourceService:AdmiResourceService -listTypeResource:List<String> ServiceRFQDao +delete(T t):void +find(Integer id) :PropertyResourceDao - propertyResourceDto:PropertyResourceDto -resourceDto:ResourceDto -listEmailUser:List<String> +getAll():List<T> -providerDto:ServiceProviderDto -em:EntityManager -propertyResourceDto:PropertyResourceDto +find(K id) :T +init:void -serviceRFQDto:ServiceRFQDto +create(ServiceRFQDao serviceRFQDao):void +createResource():void -serviceProviderDtoList:List<String> ResourceBookingDao +update(ServiceRFQDao serviceRFQDao):void +update():void +createPropertyResource():void -serviceRequestDto:ServiceRequestDto -em:EntityManager +delete(ServiceRFQDao serviceRFQDao):void +update():void +getters() +getAll():List<ServiceRFQDao> +setters() +init:void +getters() +create(ResourceBookingDao resourceBookingDao):void +createServiceRfq():void _ _ _ _ +find(Integer id) :ServiceRFQDao +setters() +update(ResourceBookingDao resourceBookingDao):void +createServiceRequest():void +delete(ResourceBookingDao resourceBookingDao):void +update():void +getAll():List<ResourceBookingDao> UserBean +getters() LogBean +find(Integer id) :ResourceBookingDao +setters() -logger:Logger -logger:Logger -userService:UserService -serviceLog:LogService -propertyResources:List<PropertyResourceDto> UserDao AdmiReportBean -userDto:UserDto -resourceBookingDto:ResourceBookingDto ResourceDao -logout():String -em:EntityManager ResidentDao -propertyResource:PropertyResourceDto -admiReportService:AdmiReportService -userDto:UserDto +init:void -em:EntityManager -em:EntityManager +create(UserDao userDao):void -resourceBookings:List<ResourceBookingDto> +log():String +update(UserDao userDao):void +init:void +create(ResourceDao resourceDao):void -residentDtos:List<ResidentDto> +create(ResidentDao residentDao):void +update():void +delete(UserDao userDao):void +ValorFinal():String +update(ResourceDao resourceDao):void +update(ResidentDao residentDao):void +getters() +getAll():List<UserDao> +init:void +prepareResourceBooking(PropertyResourceDto propertyResourceDto):String +delete(ResourceDao resourceDao):void +delete(ResidentDao residentDao):void +setters() +find(String id) :UserDao +update():void +createResourceBooking():String +getAll():List<ResidentDao> +getAll():List<ResidentDao> +getters() +find(Integer id) :ResidentDao +update():void +find(Integer id) :ResidentDao +setters() +getters() +setters() co.edu.unbosque.model.service LogService UserService AdmiPropertiesService() -logger:Logger | -logger:Logger -logger:Logger -userDao:Interface<User,String> -InterfaceDao<PropertyResource,Integer> propertyResourceDao; -propertyDao:Interface<Property,Integer> -modelMapper:ModelMapper -residentDao:InterfaceDao<Resident,Integer> -userDao:Interface<User,String> -visitorDao:InterfaceDao<Visitor, Integer> +LogService() -resourceBookingDao:InterfaceDao<ResourceBooking,Integer> +AdmiPropertiesService() +log(UserDto userDto):UserDto -userDao:InterfaceDao<User, String> +createProperties(PropertyDto:propertiesDto):boolean +redireccionar(UserDto userDto):String -modelMapper:ModelMapper +listarAdmi():List<String> encryptedPassword(String password):String -NotificationService notificationService +UserService() AdmiPropertyResourceService NotificationService +crearReserva(ResourceBookingDto resourceBookingDto,UserDto -logger:Logger -logger:Logger userDto):boolean -propertyDao:Interface<Property,Integer> +obtenerReservas(UserDto userDto):List<PropertyResourceDto> +NotificationService() -userDao:Interface<User,String> +calcularPrecio (ResourceBookingDto resourceBookingDto):double +send(String to, String subject, String content):void -propertyDao:Interface<Property,Integer> +calcularTiempoMin(ResourceBookingDto resourceBookingDto):boolean +notificationBooking(ResourceBookingDto resourceBookingDto, UserDto -resourceDao:InterfaceDao<Resource,Integer> +calcular(ResourceBookingDto resourceBookingDto):double userDto, double tiempoReserva):void -propertyResourceDao:InterfaceDao<PropertyResource,Integer> +notificationResource(List<String> correos, PropertyResourceDto -residentDao:InterfaceDao<Resident,Integer> resourceDto):void ShowPropertyService -notificationService:NotificationService +notificationRequest(ServiceRequestDto serviceRequestDto):void -modelMapper:ModelMapper -logger:Logger +notificationService(ServiceRFQDto serviceRFQDto):void -propertyDao:InterfaceDao<Property, Integer> +AdmiPropertyResourceService() -visitorDao:InterfaceDao<Visitor, Integer> +listEmilUser():List<String> ProveedorAPIService -userDao:InterfaceDao<User, String> +findPropertyByName(PropertyDto propertyDto):PropertyDto -modelMapper:ModelMapper +listNameProperty():List<String> -logger:Logger +findResourceByName(ResourceDto resourceDto):ResourceDto -URL:String +ShowPropertyService() +listTypeResource():List<String> - client:Client +listPropertyObject():List<PropertyDto> +emailResident(PropertyResourceDto propertyResourceDto):List<String> -gson:Gson +listNameUser():List<String> modelMapper:ModelMapper +createAppointmentVisitor(VisitorDto visitorDto):boolean serviceRFQDao:InterfaceDao<ServiceRFQ,Integer> AdmiReportService -serviceRequestDao:InterfaceDao<ServiceRequest,Integer> -notificationService:NotificationService AdmiUserService -logger:Logger -residentDao:InterfaceDao<Resident,Integer> -logger:Logger -resourceBookingDao:InterfaceDao<ResourceBooking,Integer> +ProveedorAPIService() -userDao:Interface<User,String> -modelMapper:ModelMapper -modelMapper:ModelMapper +createServiceRFQ(ServiceRFQDto serviceRFQDto):boolean +AdmiReportService() +createServiceRequest(ServiceRequestDto serviceRequestDto):boolean +AdmiUserService() +listResourceBookings():List<ResourceBookingDto> +getServiceMaintenance():List<String> +createUser(UserDto userDto):boolean +listResident():List<ResidentDto> +getServiceTransport():ServiceProviderDto +createPropertyResident(ResidentDto residentDto):boolean +getProviders():List<ServiceProviderDto> +encryptedPassword(String password):String AdmiResourceService -logger:Logger -userDao:InterfaceDao<User,String> -resourceBookingDao:InterfaceDao<ResourceBooking,Integer> -modelMapper:ModelMapper +AdmiReportService() +listResourceBookings():List<ResourceBookingDto> +listResident():List<ResidentDto>