

Users

Client

+Client(firstName: String, lastName: String, email: String, password: SecretKey, salt: byte[], birthYear: String)
+Client(firstName: String, lastName: String, email: String, birthYear: String)
+isValid(): boolean
+getBirthYear(): Object

Courier

~context: Context
~firestore: FirebaseFirestore
+Courier(context: Context, firestore: FirebaseFirestore)
+sendRequest(request: Request): void
+acceptRequest(request: Request): void
+doesExist(request: Request): boolean

Landlord

+Landlord(firstName: String, lastName: String, email: String, password: SecretKey, salt: byte[], address: String)
+Landlord(firstName: String, lastName: String, email: String, address: String)
+isValid(): boolean
+getAddress(): Object

User

#userData: Map<String, Object>
+User(firstName: String, lastName: String, email: String, password: SecretKey, salt: byte[])
+User(firstName: String, lastName: String, email: String)
+isValid(): boolean
+getFirstName(): Object
+getLastName(): Object
+getEmail(): Object
+getRole(): Object
+getProperties(): Object
#getPassword(): Object
+setPassword(SecretKey pass, byte[] randomSalt): void

Tickets

Request

-requestData: HashMap<String, Object>
+Request(idClient: String, idLandlord: String, property: String)
+isValid(): boolean
+getClient(): String
+getLandlord(): String
+getProperty(): String
+getRequestData(): HashMap<String, Object>

Invitation

-invitationData: HashMap<String, Object>
+Invitation(idClient: String, idLandlord: String, property: String, commission: double, accepted: boolean)
+getCommission(): String
+getAccepted(): String
+setCommission(): void
+setAccepted(): void

Ticket

-ticketData: HashMap<String, Object>
+Ticket(idClient: String, idLandlord: String, property: String, text: String)
+isValid(): boolean
+getText(): String

PropertyMgr

-invitations: ArrayList<Invitation>
+PropertyMgr(firstName: String, lastName: String, email: String, password: SecretKey, salt: byte[])
+PropertyMgr(firstName: String, lastName: String, email: String)
+isValid(): boolean
+getInvitations(): List<String>
+addInvitation(invite: Invite): void
+removeInvitation(invite: Invite): void
+clearInvitations(): void

UserCreator

~context: Context
~firestore: FirebaseFirestore
+UserCreator(activeContext: Context, firestore: FirebaseFirestore)
#add(user: User, pass: String): boolean
+doesExist(email: String): boolean

User Factory and Helper

Hashing

~context: Context
+Hashing(context: Context): Hashing
+hashPassword(password: char[], salt: byte[]): SecretKey
+generateSalt(): byte[]
+verifyPassword(salt: byte[], passwordPlainText: String, password: SecretKey): boolean

<<Activities>>

<<instantiate>>

Properties

Property Factory

PropertyCreator

~context: Context
~firestore: FirebaseFirestore
+PropertyCreator(activeContext: Context, firestore: FirebaseFirestore)
+add(property: Property): boolean
+doesExist(address: String): boolean

Property

#propertyData: HashMap<String, Object>
+Property(address: String, type: String, floor: String, numRoom: String, numBathroom: String, numFloor: String, area: String, laundry: String, numParkingSpot: String, rent: String, utilities: String, landlord: String, manager: String)
+Property(address: String, type: String, floor: String, numRoom: String, numBathroom: String, numFloor: String, area: String, laundry: String, numParkingSpot: String, rent: String, utilities: String, landlord: String, manager: String, client: String)
+Property(address: String, type: String, floor: String, numRoom: String, numBathroom: String, numFloor: String, area: String, laundry: String, numParkingSpot: String, rent: String, heating: boolean, hydro: boolean, water: boolean, landlord: String, manager: String, client: String)
+getPropertyData(): HashMap<String, Object>
+isValid(): boolean
+getAddress(): String
+getType(): String
+getFloor(): String
+getNumRoom(): String
+getNumBathroom(): String
+getNumFloor(): String
+getArea(): String
+getLaundry(): String
+getNumParkingSpot(): String
+getRent(): String
+getHeating(): boolean
+getHydro(): boolean
+getWater(): boolean
+getLandlord(): String
+getManager(): String
+getClient(): String