

DOCUMENTACIÓ TÈCNICA

DOGINDER

Funcions server node:

Login → Aquesta funció ens permet iniciar sessió, rebent com a paràmetres la contrasenya introduïda i el correu electrònic. Comprova si les dades rebudes són correctes fent una consulta a la base de dades i retorna l'usuari complet en cas de ser-ho.

sendMailWithToken → Per tal de recuperar la contrasenya, rebem el mail de l'usuari, es genera un token aleatori i fent servir nodemailer enviem un correu electrònic, així com retornar el mateix token a l'aplicació per comprovar si és correcte.

checkToken → Complementant la funció anterior, aquesta funció rep per parametres el token que ha introduït l'usuari i comprova si és el mateix que s'havia generat.

changePassword → Per últim en quant a canviar la contrasenya, aquesta funció rep com a paràmetre la nova contrasenya de l'usuari, la encripta i la insereix a la base de dades.

findNearbyUsers → Aquesta funció rep com a paràmetre l'identificador de l'usuari, la seva ubicació i la distància que vol fer servir com a filtre. Retorna la informació dels usuaris que troba.

findAllNearbyUsers → Aquesta funció fa el mateix que l'anterior, però sense tenir en compte si ja s'ha interactuat anteriorment amb ell.

createUser → Funció per registrar nous usuaris. Rep com a paràmetre totes les dades tant de l'usuari com de la mascota i fa els 'inserts' necessaris.

updateSocket → per poder mantenir la comunicació a temps reals amb l'aplicació es va registrant i actualitzant l'identificador del socket de l'usuari.

getMatches → Aquesta funció, rebent com a paràmetre l'identificador de l'usuari, retorna els usuaris que han interactuant positivament entre ells.

bloquearUsuario → Amb aquesta funció un usuari pot bloquejar a un altre en cas d'haver qualsevol problema. Rep com a paràmetres tots dos identificadors.

validateMail → En diferents casos volem comprovar si el correu electrònic que l'usuari introdueix i es rep com a paràmetre sigui correcte o estigui ja a la base de dades.

interaccion → Amb aquesta funció es queden registrades les interaccions que fa cada usuari amb un altre a l'apartat de "swipe", és a dir, si un usuari dona like o dislike a un altre.

Rep com a paràmetres els dos identificadors. També és la funció que comprova si es farà match entre els dos usuaris.

getDislikedUsers → Aquesta funció fa una consulta a la taula de les interaccions i retorna tots els usuaris als quals l'usuari que la crida ha donat dislike.

Estructura aplicació android:

Carpeta Activities:

Chat Activity:

Pantalla que ens mostra el chat amb un usuari. Menu a la cantonada superior dreta per netejar el xat, bloquejar i reportar. Fletxa per marxar del xat, imatge i nom de la mascota de l'usuari amb el qual estàs parlant. Un scrollView amb els missatges, un edit text per escriure el missatge i un botó per enviar.

Fa crides al servidor per la part del socket per enviar els missatges i rebre'ls.

Login Activity:

Pantalla amb dos Edit text per correu electrònic i contrasenya i un botó per enviar. Tenim dos texts clickables, el primer ens permet anar a la RegisterActivity i l'altre a RecoveryPassActivity.

Fa crides al servidor a la funció de Login.

Main Activity:

Pantalla que conté els tres fragments amb les funcions principals que veurem més endavant (Swipe, Perfil i Chat).

Demana els permisos de notificació i de localització.

RecoveryPassActivity:

Pantalla amb un EditText per introduir el correu electrònic al qual s'enviarà el token i un botó per enviar. Prèviament hi ha un Text View amb un breu explicació. Si el correu electrònic s'ha enviat correctament es mostrarà un altre EditText per introduir el token i un botó per comprobar.

Es fan crides al servidor per validar el mail i per comprovar el token.

RegisterActivity:

Pantalla amb un conjunt de Edit Texts per introduir les dades del nou usuari. Primer es demana nom, cognom, correu electrònic, contrasenya, edat (DatePicker amb restringió per

a menors d'edats) i imatge de perfil. Quan presionem el botó de continuar es comprovarà tota la informació i veurem les dades de la mascota.

Introduïrem el nom, la raça (Spinner), edat (DatePicker sense restringió), imatge, sexe, descripció i relació amb humans i altres mascotes.

Fa crides al servidor per comprovar el mail i per registrar l'usuari.

ResetPasswordActivity:

Pantalla final de la part de recuperar la contrasenya, veurem dos Edit Text per introduir la nova contrasenya i un botó per confirmar.

Fa una crida al servidor per canviar la contrasenya.

Carpeta Fragments:

FragmentChat:

Fragment en el qual podem veure els usuaris amb els quals ja has fet match, i amb un RecyclerView podem veure el nom de la mascota i el seu nom. Si fem click ens porta a la ChatActivity d'aquest usuari.

Fa una crida al servidor per recuperar els usuaris amb els quals has fet match.

FragmentPerfil:

Fragment que ens mostra les dades del nostre usuari, tant de la mascota com de la persona. Primer, a un apartat "Sobre tu mascota" veurem amb un ImageView la fotografia, el símbol del sexe i el nom de la mascota. A continuació la resta de dades, i per últim l'apartat "Sobre tu", amb la teva imatge i dades.

FragmentSwiper:

Fragment que té un petit menú amb tres opcions: Ver dislikes, Ver todo i Filtro (per la distancia). A continuació, fent servir la llibreria de Koloda, veurem un seguit de cartes amb la informació de cadascun dels usuaris que tenim a prop segons una distancia predeterminada. Podem lliscar la carta a dreta (like) i esquerra (dislike) així com fer-ho amb uns botons a sota de la carta. També podem lliscar a dalt i a baix per veure tota la informació de la mascota o prémer el botó que hi ha a la part superior per veure la informació de l'usuari de la mascota.

RecoveryFragment:

Carpeta Helpers:

ChatAdapter:

Complementa al recycler View del FragmentChat per mostrar els usuaris amb els quals pots xatejar.

ChatDatabaseHelper:

Base de dades local SQLite per guardar els missatges que s'envien al xat.

DataBaseHelper:

Base de dades local SQLite per guardar la informació de l'usuari que inicia sessió.

LocationHelper:

Classe que ens permet demanar la localització actual de l'usuari.

MatchListener:

Classe que ens serveix per a rebre la notificació del socket per quan es fa un match.

Settings:

Classe auxiliar per guardar constants.

SocketListener:

Classe auxiliar al Socket Manager.

SocketManager:

Classe que conté les funcions principals del socket, com connectar, desconectar i rebre el match.

SwipeAdapter:

Complementa a la classe Koloda per mostrar les cartes d'usuari.

UserNoAdapter:

Complementa al RecyclerView per mostrar els usuaris als quals has donat que no.

doginderAPI:

Interfície amb les rutes per fer les crides al servidor amb retrofit.

Carpeta Objects:

Aquí simplement tenim objectes que ens serveixen per agrupar variables, no és necessari aprofundir en les variables que conté cadascun.

- BloquearUsuario
- ChangePass
- ChatItem
- Mascota
- Mensaje
- Token
- User
- UserNearbyRequest
- UserRegistrationRequest
- UserRequest
- UserResponse
- Usuario
- Usuario2