Controllers

ControllerAddPath -TAG : String = "ControllerAddPath" +createPathWithGPX(name: String, difficulties: String, duration: int, disableAccessibility: boolean, description: String, uriGPX: Uri, userEmail: String, activity: Activity): void -GPXParser(path : Uri) : ArrayList<GeoPoint> getPathFromGPX(parsedGPX : Gpx) : ArrayList<GeoPoint> -GPXTrackAdapter(GPXTracks : List<Track>) : ArrayList<GeoPoint> -GPXRouteAdapter(GPXRoute : List<Route>) : ArrayList<GeoPoint> -GPXWayPointAdapter(GPXTracks : List<WayPoint>) : ArrayList<GeoPoint> -openInsertPointOfInterest(response: JSONObject, activity: Activity, path: NaPath, emailUser: String): void +addPathtoDatabase(naPath : NaPath, activity : Activity) : void +createNaPath(name: String, difficulties: String, duration: int, disableAccessibility: boolean, description: String, path: ArrayList<GeoPoint>, userEmail: String, country: String, region: String, activity: Activity): void -path|SONAdapter(path : ArrayList<GeoPoint>) : |SONArray -difficultyEnumAdapter(difficulties : String) : DifficultyEnum +addPointOfInterestFromExternalUser(sights: ArrayList<Sights>, naPath: NaPath, activity: Activity, emailUser: String): void +addPointOfInterestToDatabase(sights: ArrayList<Sights>, naPath: NaPath, activity: Activity, emailUser: String): void -openAddPhotoToPhoto(activity: Activity, naPath: NaPath, emailUser: String): void +checkInputDuration(durationEditText : EditText) : String +checkInputName(namePath : String) : String

ControllerRegistration -TAG : String = "ControllerRegistration" +checkName(name : String, regex : String) : String +checkPassword(password : String, regex : String, confirmedPassword : String) : String

+nameAlreadyExist(naPaths : ArrayList<NaPath>, name : String) : boolean

ControllerSearch +searchUserWith(data : String, users_result : ArrayList<User>) : ArrayList<User> +searchPathWith(name : String, paths : ArrayList<NaPath>, naPathFilters : NaPathFilters) : ArrayList<NaPath> -filterDisableAccess(paths : ArrayList<NaPath>, accessible : Boolean) : ArrayList<NaPath> -filterDifficult(paths : ArrayList<NaPath>, diffcult : DifficultyEnum) : ArrayList<NaPath> -filterCountry(paths : ArrayList<NaPath>, country : String) : ArrayList<NaPath> -filterRegion(paths : ArrayList<NaPath>, region : String) : ArrayList<NaPath> -filterDuration(paths : ArrayList<NaPath>, minDuration : Integer) : ArrayList<NaPath>

ControllerAccount -TAG: String = "ControllerAccount" <<Property>> -user: User +setUser(intent: Intent): void

ControllerPath

-TAG : String = "ControllerPath"

+jsonToNaPath(naPathJSON : JSONObject) : NaPath

auth0 auth0AuthenticationEmbedded <u>-TAG : String = "auth0Authentica</u>tion" -auth0ClientId : String -auth0Domain: String -auth0Account : Auth0 -authenticationAPIclientAuth0: AuthenticationAPIClient -databaseConnections : String -userToken : String +auth0AuthenticationEmbedded(auth0ClientId: String, auth0Domain: String, databaseConnections: String) +loginFacebook(context: Context, onSuccess: Consumer<User>, onError: FailListener): void +loginGoogle(context: Context, onSuccess: Consumer<User>, onError: FailListener): void +login2(email: String, password: String, context: Context, onSuccess: Consumer<String>, onError: FailListener): void +signup2(email: String, password: String, username: String, name: String, surname: String, onSuccess: Consumer<User>, onError: FailListener): void -getEmailUserSigned() : String -getUsernameUserSigned() : String -getNameUserSigned() : String -getGivenName() : String