

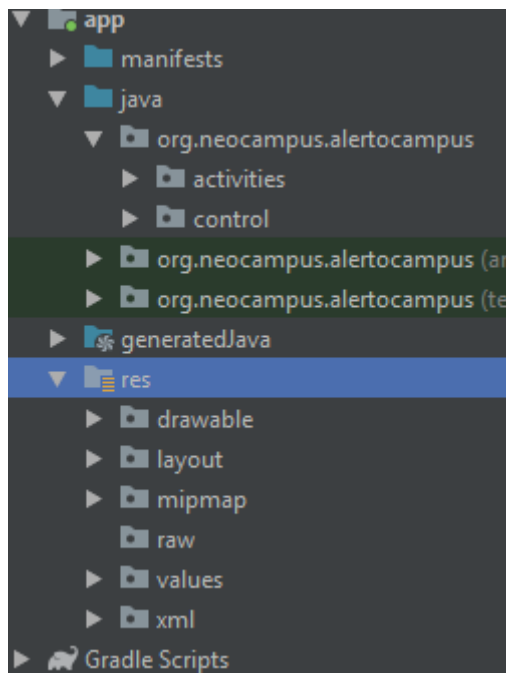
# Guide développeur

AlertOCampus

## Table des matières

I.	Hiérarchie .....	3
II.	Détail activities .....	3
	APropos .....	3
	Autre .....	3
	Déclaration .....	3
	Echec .....	3
	FormulaireInformatique .....	3
	Localisation .....	4
	Main .....	4
	Materiel .....	4
	Menu .....	4
	PbAutres .....	4
	PbConnexion .....	4
	PbMenu .....	4
	QrCode .....	4
	Software .....	4
	Thank .....	4
III.	Détail control .....	5
	Conf .....	5
	EnvoiTicket .....	5
	SessionManager .....	6

## I. Hiérarchie



Côté java nous avons 2 packages : activities qui contient comme son nom l'indique toutes les activités de l'application et control qui contient le fichier de configuration, des shared preferences et la classe d'envoi de mail.

Ensuite dans le dossier res on retrouve dans drawable toutes les icones et logos qui ont servi à l'application, mais aussi les formes de certains fond d'icônes

Layout contient tous les fichiers de layout des activités

Values contient les couleurs les dimensions utilisées fréquemment (pour les marges notamment), les strings utilisés, ainsi que nos styles personnalisés

Gradle Scripts contient toutes les dépendances de l'application (potentiellement à mettre à jour dans les versions futures)

## II. Détail activités

### APropos

Contient la page à propos donc avec les logos des partenaires de l'application la redirection vers leur site, le nom des développeurs et des concepteurs du projet ainsi que les liens indiquant les auteurs des icônes utilisées

### Autre

Contient la page qui redirige l'utilisateur vers le pdf de la liste des gpl (potentiellement à mettre à jour dans le fichier conf)

### Déclaration

Page de déclaration des incidents liés au patrimoine proche de la page formulaire mais pas directement liée à elle. Contient le formulaire bâtiment salle description de l'incident, la prise de photo

### Echec

Page en cas d'échec d'envoi du mail, contenu qui change en cas de problème lié au mail de l'émetteur avec bouton permettant de réessayer l'envoi du mail

### FormulaireInformatique

Contient le choix scan ou saisie manuelle de l'emplacement de la machine, donc jeu sur l'affichage des nombreux composants de cette page, et quelques dispositions différentes en fonction de l'activité qui a appelé le formulaire :

Si on vient pb connexion/si on vient de pbMenu/sinon cas normal

Contient le traitement du retour de scan, et du retour de la prise de photo

Crée un objet EnvoiTicket à la validation

### Localisation

Page qui contient la carte et qui localise l'utilisateur si il a autorisé la localisation, bouton pour centrer la carte sur lui et crée un objet EnvoiTicket à la validation.

### Main

Page de lancement

Page de connexion donc contient tout le traitement lié à la connexion vérification des identifiants, stockage de ceux-ci en cas de réussite.

Verifie si la personne est déjà connectée et redirige vers le menu si c'est le cas.

Bouton pour contacter le poste de sécurité

Contient aussi en commentaire un essai de connexion à [https://auth.test.univ-toulouse.fr/cas\\_login\\_unr/](https://auth.test.univ-toulouse.fr/cas_login_unr/) infructueux mais qui peut potentiellement aider

### Materiel

Contient le formulaire lié aux incidents Materiel

### Menu

Menu principal contenant tous les type d'incidents à déclarer et bouton pour contacter le poste de sécurité

### PbAutres

Contient le formulaire pour les problèmes autres

### PbConnexion

Contient le formulaire lié aux problèmes de connexion

### PbMenu

Menu des types d'incidents informatiques

### QrCode

Page du lecteur de QrCode qui ouvre l'appareil photo et cherche un Qrcode tant qu'on est sur cette activité

### Software

Contient le formulaire pour les problèmes de logiciel

### Thank

Page de remerciement et de confirmation de l'envoi du ticket

### III. Détail control

#### Conf

Fichier de configuration

```
public class Conf {  
  
    public static final String SMTP_SERVEUR = "smtp-scout.univ-toulouse.fr";  
    public static final int PORT_SERVEUR = 465;  
    public static final int requestCodeQrCode = 10;  
    public static final int requestCodePicture = 5;  
    public static final int connexionOK = 1;  
    public static final int connexionLogin = 0;  
    public static final int connexionNOK = -1;  
    public static final String DESTINATAIRE_INFO = "ticket.gun@univ-tlse3.fr";  
    public static final String DESTINATAIRE_INFO_DOUBLON = "philippe.saby@univ-tlse3.fr";  
    public static final String DESTINATAIRE_INFO_TRIPLON = "philippe.lacassagne@univ-tlse3.fr";  
    public static final String DESTINATAIRE_EAU = "esteban.bravo@univ-tlse3.fr";  
    public static final String DESTINATAIRE_AIR = "dorian.azema@univ-tlse3.fr";  
    public static final String DESTINATAIRE_LUMIERE = "dorian.azema@univ-tlse3.fr";  
    public static final String DESTINATAIRE_FENETRE = "dorian.azema@univ-tlse3.fr";  
    public static final String ENTETE_MAIL = "[fsi aoc di] ";  
    public static final String LIEN_PDF = "https://intranet.ups-tlse.fr/medias/fichier/gpl-correspon";  
    public static final boolean fuiteEau = true;  
    public static final boolean fuiteAir = true;  
    public static final boolean lumiere = true;  
    public static final boolean fenetre = true;  
    public static final boolean informatique = true;  
    public static final boolean autres = true;  
}
```

Smtp\_serveur et Port serveur sont les adresses et ports du serveur smtp à utiliser pour la connexion et l'envoi de mail

Les requestCode permettent de définir les codes de retour attendus pour certaines opérations (il faut juste que requestCode soit différent de requestCodePicture)

connexionOK/Login/Nok permettent de différencier les retours de la tentative de connexion

connexionOK : pas de problème

connexionLogin : problème d'identifiants

connexionNOK : problème réseau

Les Destinataires permettent de changer les adresses de destination des différents mails

Lien\_PDF : lien du pdf de la liste des gpl

Entete\_Mail représente le début du sujet du mail (doit correspondre au filtre du GUN)

Et les booleens permettent d'activer ou désactiver aisément les boutons du menu principal

#### EnvoiTicket

Classe s'occupant de l'envoi des mails, gérant aussi l'affichage du chargement pendant l'envoi via le handler fourni dans le constructeur et redirige vers thank si tout s'est bien passé et sinon vers Echec en indiquant si l'échec du au mail de l'expéditeur ou pas

## SessionManager

Classe des sharedPreferences gerant le stockage des identifiants et du statut connecté ou non