**Membres du groupe :**

|  |  |
| --- | --- |
| NOMS ET PRENOMS | MATRICULES |
| TCHAYA NGONGAG VANNELLE LAURE | 16A018FS |
| ABABA MALLOUM PATCHA KANDJA | 15A482FS |
| ANEMETA DON DIVIN | 16B007FS |
| MBOGNING THOFFO JUNIOR | 16B101FS |
| NASSOUR ANNOUR SAAD | 16A001FS |
| GASSISSO GILBERT | 15B335FS |
| TCHOKOUEGNO DE MOFFO GABBI EVRAD | 16B140FS |
| WADE URBAIN BORIS | 12A660FS |
| SALIHOU AMADOU | 16B132FS |
| NEKUI TIEFFANG BIEDIANT | 16A980FS |
| FOMENI JEAN PAUL | 14A350FS |
| MICHE FRANK | 15A749FS |

**SUJET 3** : SIMULER L’INTER-COMMUNICATION ENTRE APPLICATIONS SUR ANDROID

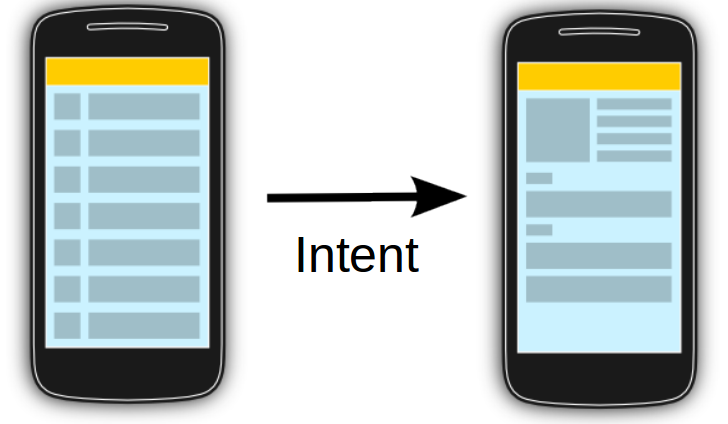
Les objectifs : Expliquer de manière explicite le rôle des « intents » dans le comportement des applications Android.

La méthode utilisée : L’on a fait des captures d’écran de chaque étape lors des différentes étapes qui mènent vers l’activation des « intents ». Et pour chaque types d’« intents », nous avons pris des exemples concrets.

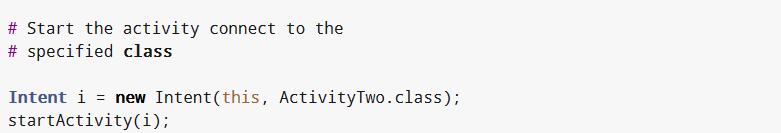
Sous Android, l’on dénomme deux types d’intents : les intents implicites et les intents explicites. Qui ont deux comportements différents vis-à-vis d’Android.

Simulation de l’execution d’un **intent explicite**:

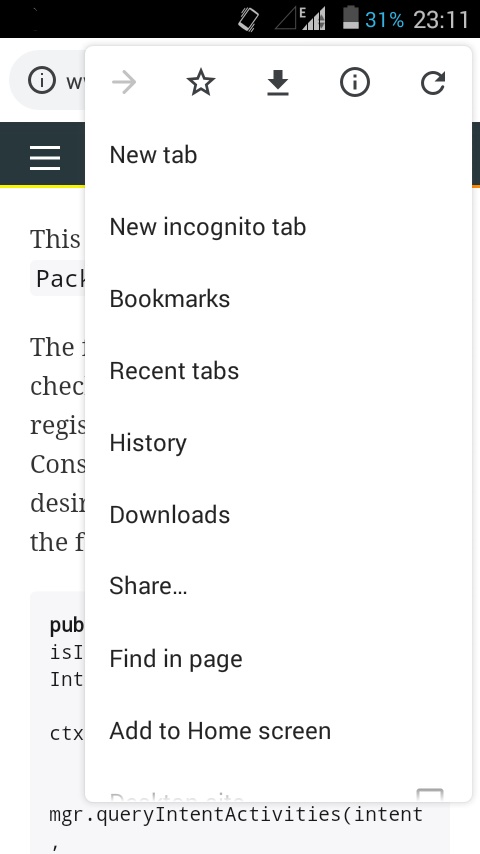
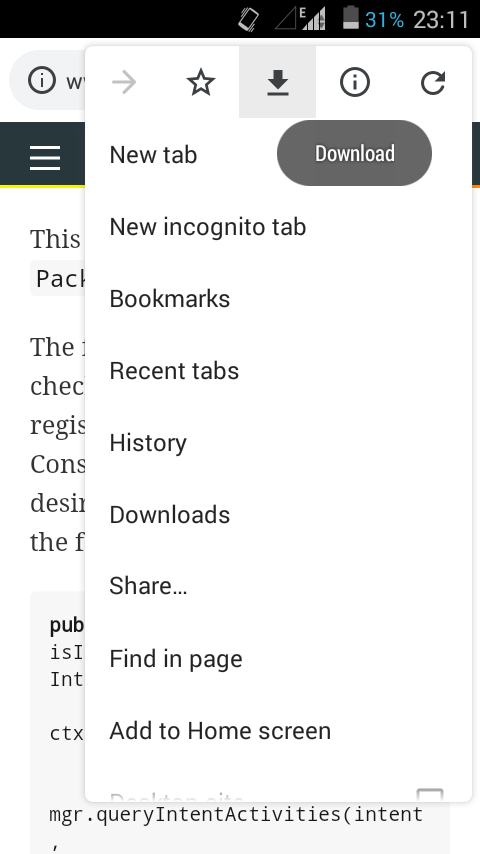
Les intents explicites sont ceux qui en générale permettent de lier les activités d’une même application. Un intent est dit explicite lorsque dans le code source, il spécifie le nom complet de la classe qu’il souhaite instancier.

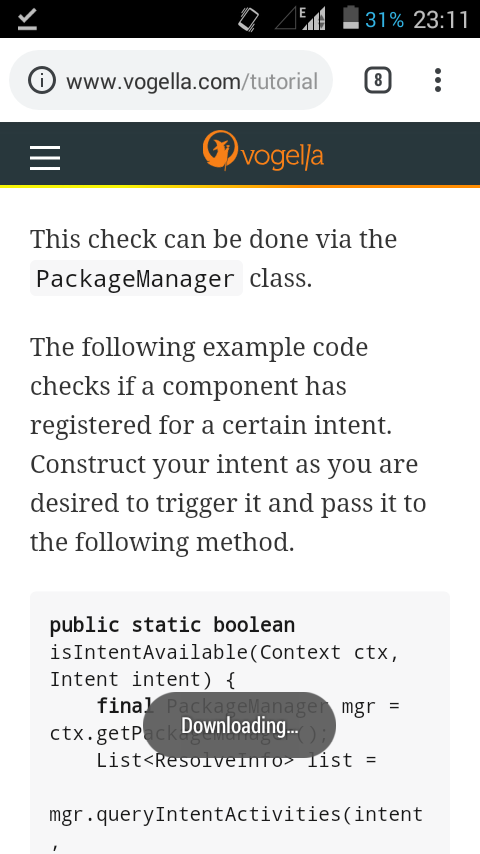


Grace à eux, on peut d’une activité passer à une autre. A l’appui, un exemple de code source de lancement d’une activité grâce à un « intent » ci-dessous :Seulement, les intens explicite peuvent avoir des fonctions différentes de celles d’habitude. Comme, l’exemple illustratif dans notre simulation. Les intents explicite peuvent également permettre d’instancier certaine composant d’Android lui-même.



Nous prenons ici un navigateur sur une page web, et nous décidons d’enregistrer une page en hors-ligne.



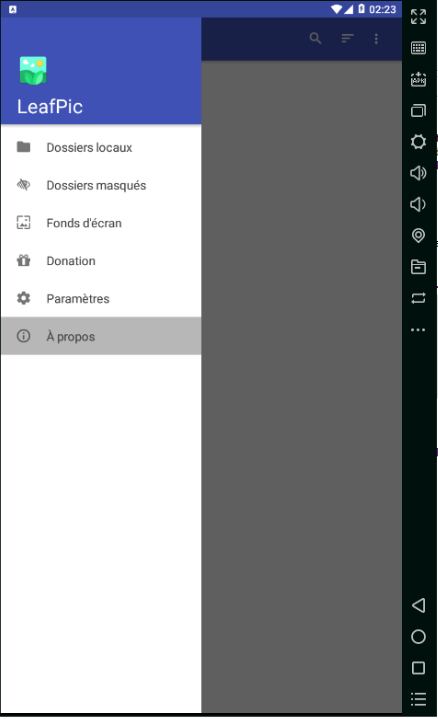


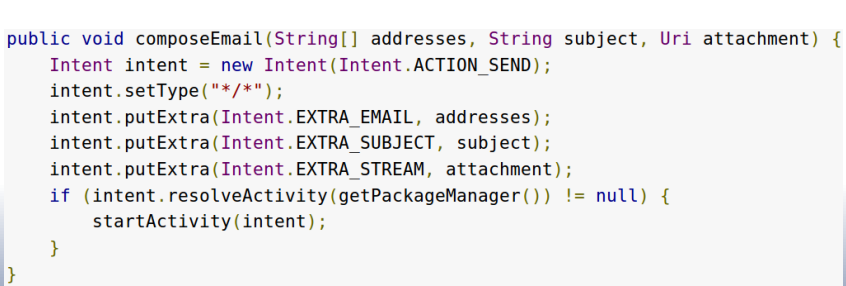
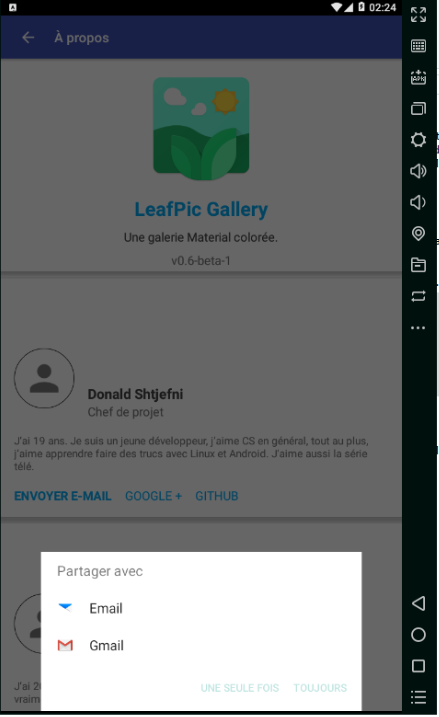
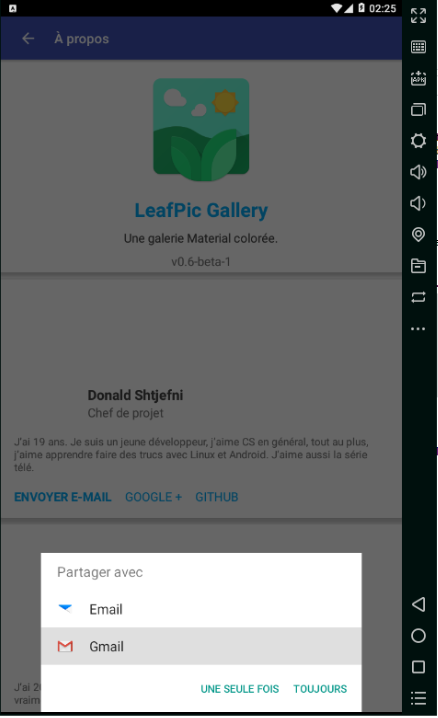
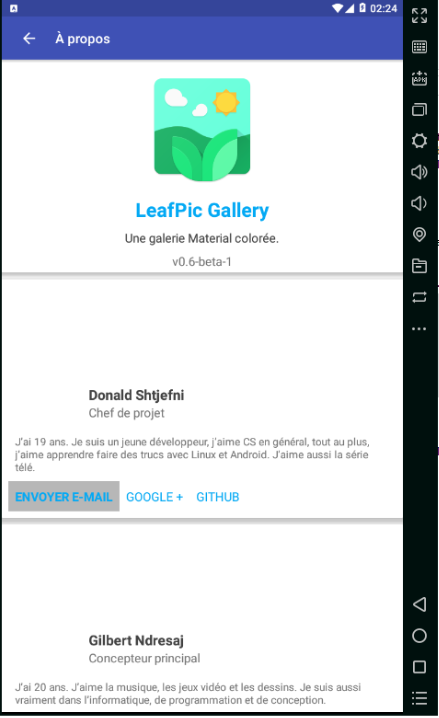
Dans notre cas, l’on peut apercevoir le nom de la classe que le navigateur souhaite instancier afin de pouvoir télécharger ce que nous lui avions demande. L’intent spécifie qu’il s’agit du **DownloadService.class** (la classe du service de téléchargement d’Android).

Simulation de l’execution d’un **intent implicite** :

Les intents implicite quant à eux servent en quelque sorte à laisser certaine tache a d’autre application sans en spécifier un en particulier. En effet il laisse le système choisir l’application qui lui sied parfaitement et lorsque ce dernier en trouve plusieurs qui peuvent effectuer le travail demande, le choix revient donc à l’utilisateur.

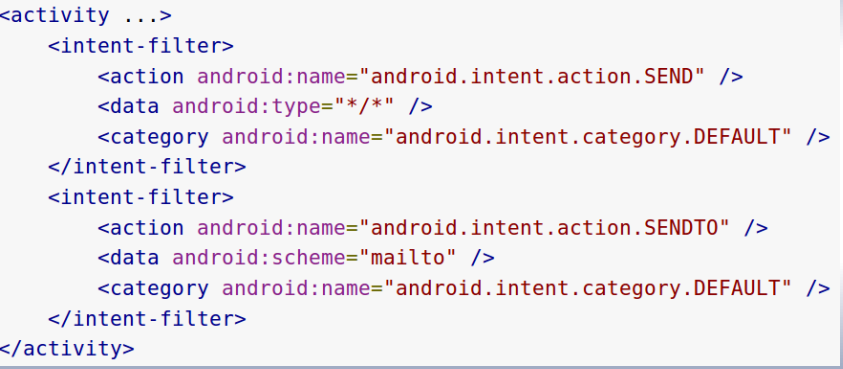
Dans notre simulation, l’on souhaite envoyer un e-mail au concepteur de l’application leafpic,





Lorsque l’on clique sur le bouton **ENVOIYER E-MAIL,** l’intent (dont le code source est ci-dessus) est actif. Comme on peut le constater aucune classe n’est spécifiée, par contre l’action de celui-ci y est. C’est donc au système de trouver l’application pouvant exécuter l’action qui est **ACTION\_SEND** (l’action d’envoyer) avec quelques paramètres supplémentaires qui permettent au système de savoir que c’est un e-mail qu’on souhaite envoyer.

L’étape suivante consiste a rechercher les applications pouvant exécuter l’action spécifier par l’intent. Et pour se faire, Android parcourt les fichier manifeste de toutes les applications afin de trouver les « intents-filter » qui répondent avec les paramètres demander.



Ci-dessus sont des exemples d’intents filters. Le deuxième répond parfaitement à l’action demandée. Dans notre cas, Android a trouver deux applications qui ont des intents-filters correspondant à l’action. Donc le choix de l’application revient à l’utilisateur.

Bibliographie :

-Introduction aux systèmes d’exploitation mobile du Dr.-Ing tchakounte Franklin

-Cours Android de F.Michel

-Les intents de Frederic Guinand