

Sujet d'épreuves des Finales Nationales de la 47^e Compétition des Métiers

MÉTIER N°09 SOLUTIONS LOGICIELLES EN ENTREPRISE

Module 2 - Annexe

Soumis par :

Paul NGO, Expert WorldSkills France

Briac DELAIGUE, Expert Adjoint WorldSkills France

ANNEXE - USAGE DE HTTP CLEARTEXT

Depuis la version 8 d'Android, l'accès de l'application que l'on développe a été modifié pour davantage de sécurité. La navigation en clair, c'est-à-dire sans aucune sécurité est par défaut interdite. Si vous sollicitez une URL non protégée, vous aurez alors au moment de l'exécution de votre application ce message "Cleartext HTTP traffic to * not permitted". Il existe plusieurs solutions pour résoudre ce souci.

Si vous êtes dans le cas pour lequel il n'existe pas d'URL sécurisée, il existe différentes solutions qui permettent de contourner ce problème. On peut ajouter dans une application Android le fichier XML `network_security_config.xml`. On peut ajouter dans ce fichier des domaines pour lesquels on souhaite autoriser la navigation en clair. Pour effectuer cette manipulation, créez le fichier dans le dossier "res/xml" et ajoutez-y le code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">IP ou URL</domain>
  </domain-config>
</network-security-config>
```

L'attribut "includeSubdomains" permet d'ajouter tous les sous-domaines d'un domaine, comme par exemple `monblog.monsiteinternet.fr`. Vous devez ensuite modifier le fichier `AndroidManifest.xml`, à la racine de votre application, pour indiquer la présence du fichier de configuration.

```
<uses-permission android:name="android.permission.INTERNET" />
<application
    android:networkSecurityConfig="@xml/network_security_config"
```

L'avantage de cette option est que l'on n'autorise que les requêtes à destination des domaines que l'on choisit. Cela permet de garder un niveau de sécurité important, surtout si l'on est celui qui gère le ou les domaine(s) cible(s).

Une autre option consiste à modifier directement le fichier `AndroidManifest.xml`. On peut, lors de la déclaration de l'application, ajouter le paramètre "android:usesCleartextTraffic" avec comme valeur "true" pour indiquer que le trafic réseau vers les sites non sécurisés est autorisé.

Vous devez faire attention avec cette modification, car il n'y a plus de contrôle de sécurité, quel que soit le site indiqué, le trafic sera autorisé.

Lorsque vous modifiez le fichier `AndroidManifest.xml`, faites également attention à la présence du paramètre `android:targetSandboxVersion`. S'il est présent, sa valeur influe directement sur l'option `usesCleartextTraffic`. Sa valeur doit être fixée à 1, sinon les autres modifications ne seront pas prises en compte, le trafic en clair vers des sites en HTTP étant interdit par défaut dans les autres niveaux.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:targetSandboxVersion="1">
    <uses-permission android:name="android.permission.INTERNET" />
    ...
</manifest>
```

ANNEXE - USAGE DE LA LIBRAIRIE VOLLEY

Mise à jour du `build.gradle` (Module `:app`) pour prendre en compte la librairie « Volley » pour communiquer avec l'API Rest

```
implementation 'com.android.volley:volley:1.2.1'
implementation 'com.google.code.gson:gson:2.8.9'
implementation 'androidx.cardview:cardview:1.0.0'
```

COMPRENDRE REQUESTQUEUE ET TRAVAILLER AVEC VOLLEY DANS ANDROID :

Volley est une bibliothèque réseau gérée par `RequestQueue` et principalement utilisée à des fins de réseautage plus petites sous Android. Pour l'utiliser, vous devez d'abord instancier la `RequestQueue` et plus tard, vous pouvez démarrer ou arrêter la demande, ajouter ou annuler la demande et accéder au(x) cache(s) de réponse.

```
RequestQueue queue = Volley.newRequestQueue(this);
```

Après avoir instancié `RequestQueue`, une requête doit être créée. Les classes de demande par défaut déjà incluses dans la bibliothèque Volley sont la demande de chaîne, la demande JSON et la demande d'image. Vous pouvez également créer une demande personnalisée en étendant la classe de demande de Volley.

Les constructeurs de requêtes utilisés dans Volley prennent 4 paramètres :

```
JsonObjectRequest request = JsonObjectRequest(Request.Method.GET, url, new  
ResponseListener(), new ErrorListener());
```

Premier paramètre : Request.Method.GET – Le GET est utilisé pour lire. Vous pouvez également utiliser POST (pour créer), PUT (pour mettre à jour/remplacer), DELETE (pour supprimer), PATCH (pour mettre à jour/modifier) et plus encore.

Deuxième paramètre : URL - L'URL qui répondra à la requête HTTP.

Troisième paramètre : Écouteur de réponse réussie - Où vos données iront une fois la demande terminée avec succès.

```
private class ResponseListener implements Response.Listener{  
    @Override  
    public void onResponse(JSONObject response){  
  
    }  
}
```

Quatrième paramètre : Écouteur d'erreurs - Ce qui sera dit s'il y a eu un problème avec votre demande. Par exemple, vous pouvez l'afficher dans le journal pour voir l'erreur.

```
private class ErrorListener implements Response.ErrorListener{  
    @Override  
    public void onErrorResponse(VolleyError error){  
  
    }  
}
```

Maintenant, la dernière étape consiste à ajouter votre demande à la file d'attente des demandes et Volley se chargera pour vous.

```
queue.add(request);
```

Ici, vous pouvez également ajouter plus de demandes à la file d'attente que vous souhaitez en même temps et la réponse sera envoyée à leurs classes de réponse/erreur respectives.