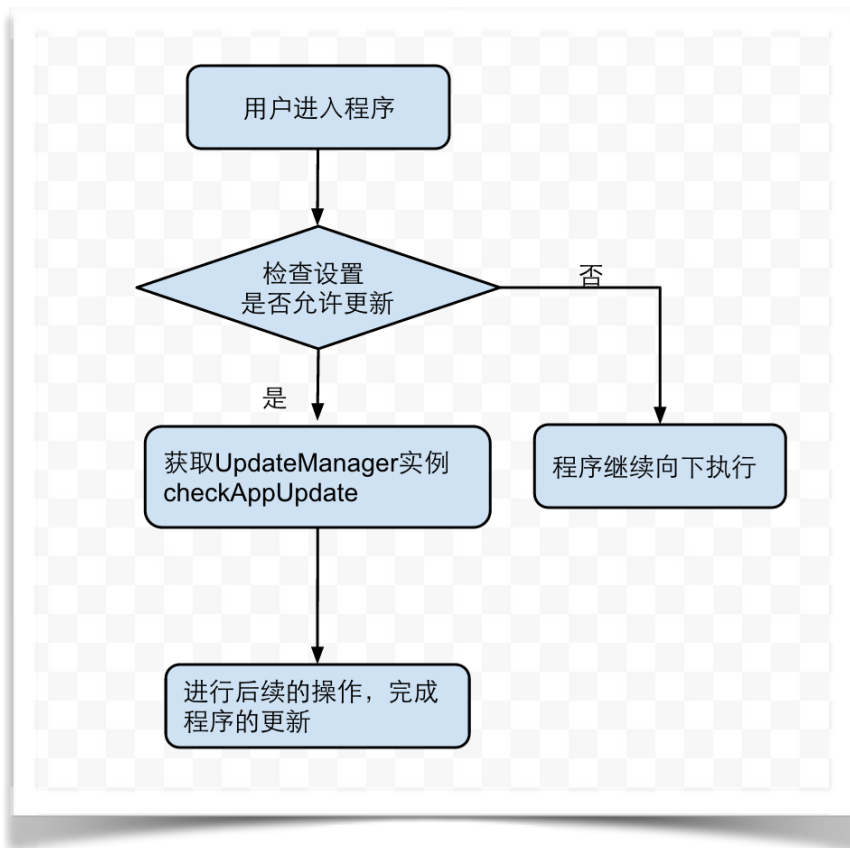


在程序的设置界面中，有一个选项“启动检查更新”，当被勾选后，将会在启动的时候检查服务器是否有新版本来更新。

软件更新流程图：



在程序MainActivity的onCreate()方法中：

```
// 检查新版本
if (mContext.isCheckUp()) {
    UpdateManager.getUpdateManager().checkAppUpdate(this, false);
}
```

接下来会走到UpdateManager中，在这个类中涉及到很多有用的知识点：

1、获取应用当前版本号getCurrentVersion()：

```
/**
 * 获取当前客户端版本信息
 */
private void getCurrentVersion() {
    try {
        PackageInfo info = mContext.getPackageManager().
            getPackageInfo(mContext.getPackageName(), 0);
        curVersionCode = info.versionCode;
    } catch (NameNotFoundException e) {
        e.printStackTrace(System.err);
    }
}
```

## 2、安装apk:

```
/**
 * 安装apk
 * @param url
 */
private void installApk() {
    File apkfile = new File(apkFilePath);
    if (!apkfile.exists()) {
        return;
    }
    Intent i = new Intent(Intent.ACTION_VIEW);
    i.setDataAndType(Uri.parse("file://" + apkfile.toString()),
        "application/vnd.android.package-archive");
    mContext.startActivity(i);
}
```

## 3、下载apk:

```
/**
 * 下载apk
 * @param url
 */
private void downloadApk() {
    downloadThread = new Thread(mdownloadApkRunnable);
    downloadThread.start();
}
```

## 4、对应用更新实体类的封装: Update.java

## 5、使用到的网络部分:

```
/**
 * 获取App更新信息
 * @return
 * @throws AppException
 */
public Update getUpdateInfo() throws AppException {
    return ApiClient.getUpdateInfo(this);
}
```

a. 监测程序新版本信息, 获取APP更新信息: ApiClient.getUpdateInfo(this)

b. 下载程序新版本

网络涉及到的类:

ApiClient.java、HttpRequestor.java、Update.java、AppContext.java

接下来, 就从程序更新这块来, 切入网络部分的学习!