

# 9

## Fonctionnalités diverses

---

Dans ce chapitre, nous allons voir différentes fonctionnalités de Codename One qui n'ont pas trouvé leur place dans les chapitres précédents, mais qui sont tout aussi importantes à la conception d'une application complète et bien fournie.

### 9.1. La classe Display

Cette classe est l'une des plus importantes de Codename One. C'est elle qui gère le [thread principal](#) qui s'occupe du rendu et des événements. De plus, elle inclut de nombreuses fonctions utiles, permettant notamment d'utiliser le vibreur de l'appareil, de récupérer la dimension de l'écran, d'ouvrir une URL dans un navigateur ou encore un document quelconque, de récupérer le nom de la plateforme courante, de basculer l'application du mode portrait au mode paysage et vice versa, d'effectuer une tâche en arrière-plan et beaucoup d'autres choses encore. Je vous invite à consulter la [Javadoc](#) de cette classe pour découvrir les autres fonctionnalités qu'elle propose. Display est un singleton, donc il vous suffit de récupérer son unique instance pour l'utiliser comme c'est le cas avec la notation suivante :

```
Display.getInstance().NOM_DE_LA_METHODE_A_APPELER();
```

- `void execute(String url)` – Exécute l'élément dont le nom ou l'URL est passé en paramètre. Elle peut s'utiliser pour ouvrir un fichier de n'importe quel type (PDF, document Word, fichier audio, fichier vidéo, etc.). Il sera alors ouvert avec l'application appropriée disponible sur l'appareil. Elle s'utilise aussi pour ouvrir une adresse web dans un navigateur internet.
- `void exitApplication()` – Quitte et ferme une application.
- `Form getCurrent()` – Retourne le Form courant (Form affiché à l'écran).
- `int getDisplayWidth()` – Retourne la valeur de la largeur de l'écran.
- `int getDisplayHeight()` – Retourne la valeur de la hauteur de l'écran.
- `String getPlatformName()` – Retourne le nom de la plateforme de l'appareil. Il s'agit d'un mot de deux ou trois lettres : `ios` (pour la plateforme iOS), `and` (pour

la plateforme Android), `rim` (pour BlackBerry OS), `win` (pour Windows Phone), `me` (pour J2ME).

- `boolean isEdt()` – Retourne `true` si l'action en cours se passe dans l'`EDT`.
- `boolean isPortrait()` – Retourne `true` si le téléphone est en mode d'orientation portrait (vertical).
- `boolean isTablet()` – Retourne `true` si l'appareil est une tablette et non un téléphone.
- `void lockOrientation(boolean portrait)` – Bloque l'orientation de l'affichage dans un seul sens. Si le paramètre `portrait` est à `true` alors l'orientation sera bloquée sur le mode portrait (verticalement). Si sa valeur est `false` alors l'orientation sera bloquée sur le mode paysage (horizontalement).
- `void unlockOrientation()` – Permet de désactiver le verrouillage de l'écran dans un seul sens. En bref, elle désactive l'effet de la méthode `lockOrientation()`.
- `boolean minimizeApplication()` – Réduit et renvoie une application en arrière-plan si cette fonctionnalité est supportée par la plateforme. Elle retourne `true` si l'action s'effectue et `false` dans le cas contraire.
- `void restoreMinimizedApplication()` – Permet de restaurer à l'écran l'application réduite en arrière-plan avec `minimizeApplication()`.
- `Object notifyStatusBar(String text, String titleText, String bodyText, boolean vibrate, boolean flashLights, Hashtable args)` – Crée et place une notification dans la barre de statut si cette fonctionnalité est disponible sur l'appareil. Le premier paramètre est le texte de la notification qui défilera sur la barre de statut, le deuxième est le titre de la notification, le troisième est le texte du contenu de la notification, le quatrième permet de faire vibrer l'appareil si sa valeur est à `true`. Le cinquième permet d'activer l'éclairage interne de l'écran du téléphone si sa valeur est à `true`. Le sixième et dernier permet d'ajouter des arguments divers à la notification.
- `void scheduleBackgroundTask(Runnable r)` – Permet d'exécuter une action en tâche de fond. En paramètre à cette méthode, un thread créé par `Runnable`. La priorité de ce thread est faible.
- `void setScreenSaverEnabled(boolean screenSaver)` – Active et désactive l'affichage de l'écran de veille de l'appareil. Le paramètre `screenSaver` doit être à `false` pour le désactiver et à `true` pour l'activer.
- `void vibrate(int duration)` – Fait vibrer l'appareil pendant la durée passée en paramètre. Le paramètre `duration` est en millisecondes.

## 9.2. Appel, e-mail, SMS

Les méthodes qui permettent d'effectuer les trois actions que nous allons voir font aussi partie de la classe `Display` vue à la section précédente.

- `void dial(String number)` - Ouvre l'application native de l'appareil qui permet de lancer un appel avec le numéro de téléphone passé en paramètre sous forme de chaîne de caractères.
- `void sendSMS(String number, String message)` - Envoie un SMS avec le message passé en deuxième paramètre au numéro de téléphone passé en premier paramètre.
- `void sendMessage(String[] recipients, String subject, Message msg)` - Utilise l'application d'envoi d'e-mail par défaut de l'appareil pour envoyer un e-mail aux destinataires dont les e-mails sont passés en premier paramètre. Même s'il s'agit d'un seul destinataire, il faut quand même passer un tableau de chaînes de caractères à ce premier paramètre. En deuxième paramètre, on a le sujet de l'e-mail et en troisième paramètre, le message à envoyer. Ce message doit être créé par la classe `Message`. Voici un exemple d'utilisation :

```
Message message=new Message("Bonjour, juste pour avoir de tes nouvelles.  
Tu me manques") ;  
Display.getInstance().sendMessage(new String[]  
{"dodericg@gmail.com"}, "Salutation", message);
```

La méthode `sendMessage()` peut aussi être appelée directement avec la classe `Message` sans passer par `Display` parce que cette méthode est statique à l'intérieur de la classe `Message`.

La classe `Message` contient des méthodes intéressantes comme celle permettant d'attacher une pièce jointe ou celle pour définir le type MIME de l'e-mail (format texte ou HTML).

**Note** > Les abonnés payants au cloud de Codename One disposent également avec `sendMessage()` d'une possibilité d'envoi d'e-mails par le cloud sans passer par les fonctionnalités natives de l'appareil. La méthode pour le faire est `sendMessageViaCloudSync()` et sa version asynchrone est `sendMessageViaCloud()`. Un message envoyé par l'une de ces deux méthodes aura pour expéditeur Codename One. C'est fait exprès pour protéger des spams.