

## ARTIGO: 11378

### Backup/Restore

### Introdução

Nesse artigo veremos como usar a rotina de backup/restore disponível na API do Veloster Framework

## Informações e Configurações

A API Veloster possui uma rotina de backup e restore para Android embutida. As configurações são definidas no arquivo *resource.properties*:

### Java Code

```
br.com.mobilemind.db.backupPath=aaplicacaoNome/backup
br.com.mobilemind.android.applicationPakage=br.com.mobilemind.aplicacaonome
br.com.mobilemind.android.db.Location=/data/{0}/databases/{1}
br.com.mobilemind.db.backupSufixFormat=dd-MM-yyyy_HH-mm-ss
```

Através dessas quatro configurações são definidas as diretrizes de backup.

**br.com.mobilemind.db.backupPath:** Define a pasta onde o backup será salvo e recuperado

**br.com.mobilemind.android.applicationPakage:** Define o pacote da aplicação. Usado para recuperar o banco de dados

**br.com.mobilemind.android.db.Location:** Essa configuração é padrão. É usada para recuperar o caminho do banco de dados.

**br.com.mobilemind.db.backupSufixFormat:** Usado para gerar um sufixo para o arquivo de backup;

## Usando a rotina de backup/restore

Para usarmos a rotina de backup/restore temos que recuperar uma instância de `br.com.mobilemind.veloster.extra.DatabaseBackupHelper`:

### Java Code

```
public interface DatabaseBackupHelper {
    /**
     * do database backup. configure options in resources.properties
     */
    void doBackup();

    /**
     * do restore of dababase backup. configure options in resources.properties
     *
     * @param info backup info to restore
     */
    void doRestore(BackupInfo info);

    /**
     * remove database
     */
    void deleteDatabase();

    /**
     * remove database test
     */
    void deleteTestDatabase();

    /**
     * list old backups
     *
     * @param ignoreInvalidsFormats. if invalids formats in backup folder should be ignored.
     * @return list of old backups
     */
    List<BackupInfo> listOldBackups(boolean ignoreInvalidsFormats);
}
```

## Realizando o backup

### Java Code

```
DatabaseBackupHelper backupHelper = VelosterConfig.getConf().getDatabaseBackupHelper();
StringBuider message = new StringBuider();
try {
    backupHelper.doBackup();
    message.append("Backup realizado com sucesso!");
    error = false;
} catch (BackupException e) {
    message.append(e.getMessage());
} catch (Exception e) {
    message.append("Erro ao fazer backup. Mensagem: ").append(e.getMessage());
}
```

```
Toast.makeText(context, message, Toast.LENGTH_SHORT)
```

## Listando as backups para restore

### Java Code

```
List<BackupInfo> infos = null;
List<String> displayList = new LinkedList<String>();

try {
    infos = backupHelper.listOldBackups(true);
} catch (Exception e) {
    AppLogger.error(getClass(), e);
    return;
}

for (int i = 0; i < infos.size(); i++) {
    displayList.add(DateUtil.timestampToStr(infos.get(i).getDate())); // datas dos backups
}
```

## Restaurando um backup

### Java Code

```
String selectedBackupDate = displayList.get(0); // recuperar backup selecionado
BackupInfo selectecBackup = null;
List<BackupInfo> infos = null;
StringBuider message = new StringBuider();

try {
    infos = backupHelper.listOldBackups(true);
} catch (Exception e) {
    AppLogger.error(getClass(), e);
    return;
}

String b = null;
for (BackupInfo it : infos) {
    b = DateUtil.timestampToStr(it.getDate());
    if (b.equals(value)) {
        selectecBackup = it;
        break;
    }
}

if(selectecBackup != null){
    backupHelper.doRestore(selectecBackup);

    try {
        backupHelper.doRestore(selectecBackup);
        message.append("Restauração da base de dados realizada com sucesso!");
    } catch (BackupException e) {
        message.append(e.getMessage());
    } catch (Exception e) {
        message.append("Erro ao restaurar base de dados. Mensagem: ").append(e.getMessage());
    }
    Toast.makeText(context, message, Toast.LENGTH_SHORT)
}
```

Sabemos que uma rotina de backup é indispensável para qualquer tipo de aplicação que use banco de dados. E como podemos ver é bem simples fazer o backup/resrore usando a API do Veloster Framework.