ARTIGO: 11363
Configurando o Veloster - Arquivo de propriedades
Introdução

Neste artigo vamos ver como configurar o Framework Veloster via arquivo de propriedades**.** A configuração por arquivo de propriedades é muito simples, bastando apenas criar um arquivo chamado *resource.properties* dentro da pasta de recursos da aplicação como mostra a figura abaixo .

Lista das configurações disponíveis:

A lista é apresentada com a chave da configuração, o valor, se é usado em ambiente Android e se é usada em ambiente desktop.

Chave	Valor	Android	Desktop
br.com.mobilemind.db.android	[java.lang.Boolean] - Identifica se o Veloster está configurado para embiente Android	Obrigatório	Obrigatório
br.com.mobilemind.db.driver	[java.lang.String] - Identifica o driver do banco de dados. Ex.: org.sqlite.JDBC	Não é usado	Obrigatório
br.com.mobilemind.db.name	[java.lang.String] - Nome do banco de dados. Ex.: MeuBancoDeDados	Obrigatório	Obrigatório
br.com.mobilemind.db.testName	[java.lang.String] - Nome do banco de dados de teste. Ex.: MeuBancoDeDadosTeste	Opcional	Opcional
br.com.mobilemind.db.path	[java.lang.String] - Caminho do banco de dados, ou nome da pasta. PastaDoSistemaVeslorter	Não é usado	Obrigatório
br.com.mobilemind.db.patdEnv	[java.lang.String] - Variável de ambiente java que contém o caminho do banco de dados. Ex.: java.io.tmpdir ou user.home	Não é usado	Obrigatório
br.com.mobilemind.db.host	[java.lang.String] - IP que aponta para o servidor de banco de dados. Ex.: 127.0.0.1	Não é usado	Opcional
br.com.mobilemind.db.port	[java.lang.Integer] - Porta do banco de dados. Ex.: 3306	Não é usado	Opcional
br.com.mobilemind.db.backupPath	[java.lang.String] - Nome da pasta para backup. Ex.: MinhaPastaBackup	Obrigatório	Não é usado
br.com.mobilemind.db.user	[java.lang.String] - Usuário do banco de dados. Ex.: root	Não é usado	Opcional
br.com.mobilemind.db.password	[java.lang.String] - Senha do banco de dados. Ex.: secret	Não é usado	Opcional
br.com.mobilemind.db.ddl	[java.lang.String] - Estratégia para a criação das tabelas do banco de dados. [create | update | none]. Ex.: create	Obrigatório	Obrigatório
br.com.mobilemind.android.applicationPakage	[java.lang.String] - Pacote base da aplicação. O mesmo configurado em AndroidManifest.xml para o valor package. Ex.: br.com.minhaempresa.appname	Obrigatório	Não é usado
br.com.mobilemind.android.dataBaseVersion	[java.lang.Integer] - Versão do banco de dados. Ex.: 1	Obrigatório	Obrigatório
br.com.mobilemind.android.db.Location	[java.lang.String] - Localização do banco de dados. Valor padrão: /data/{0}/databases/{1}	Obrigatório	Não é usado
br.com.mobilemind.db.backupSufixFormat	[java.lang.String] - Sufix usado para nome do arquivo de backup. Deve ser compatível com formato de data (Veja SimpleDateFormat). Ex.: dd-MM-yyyy_HH-mm-ss	Obrigatório	Não é usado
br.com.mobilemind.defaultDateFormat	[java.lang.String] - Formato de data usado para gravação no banco de dados. Deve ser compatível com formato de data (Veja SimpleDateFormat). Ex.: yyyy-MM-dd hh:mm:ss.SSS	Obrigatório	Não é usado

Detalhes

Como é formado o caminho de acesso ao banco de dados, no caso do uso em desktop?

Quando o Veloster é excutado em uma aplicação desktop, de vemos seguir algumas regras para que o banco de dados seja encontrado. A tualmente, o sistema supora MySQL e SQLite. As regras mais especificas se aplicam ao banco de dados SQL ite por se tratar de um banco emba rcado. Vamos as detalhes para cada banco.

SQLite

Para configurarmos o caminho do banco de dados SQLite, devemos fornecer as seguintes configurações:

- br.com.mobilemind.db.android=false
- br.com.mobilemind.db.driver=org.sqlite.JDBC
- br.com.mobilemind.db.name=meubanco.db
- br.com.mobilemind.db.path=minha_app
- br.com.mobilemind.db.pathEnv=user.home
- br.com.mobilemind.defaultDateFormat=yyyy-MM-dd hh:mm:ss.SSS

Para criação da localização do banco, o seguinte código será aplicado:

JavaCode

```
String path = VelosterResource.getProperty("br.com.mobilemind.db.path");
String dbPathEnv = VelosterResource.getProperty("br.com.mobilemind.db.pathEnv");
if (!MobileMindUtil.isNullOrEmpty(dbPathEnv)) {
    path = path = dbPathEnv + File.separator + path;
}
```

Para esse banco de dados devemos omitir os valores referente a host, porta, usuário e senha.

MySQL

Para configurarmos o acesso ao banco de dados MySQL, devemos fornec er as seguintes configurações:

- br.com.mobilemind.db.android=false
- br.com.mobilemind.db.driver=com.mysql.jdbc.Driver
- br.com.mobilemind.db.password=minha_senha
- br.com.mobilemind.db.user=root
- br.com.mobilemind.db.port=3306
- br.com.mobilemind.db.host=127.0.0.1
- br.com.mobilemind.db.name=meu_banco
- br.com.mobilemind.defaultDateFormat=yyyy-MM-dd hh:ss:mm

Essas configurações já seriam o suficiente para que o Veloster acesse o banco de dados informado. Abaixo está disponívem um arquivo de configuração completo. Você apenas deve alterar os valores para as configurações de sua aplicação.

JavaCode

```
br.com.mobilemind.db.android=false
br.com.mobilemind.db.driver=org.sqlite.JDBC
br.com.mobilemind.db.name=veloster.db
br.com.mobilemind.db.testName=veloster_test.db
br.com.mobilemind.db.path=veloster
br.com.mobilemind.db.pathEnv=java.io.tmpdir
br.com.mobilemind.db.host=
br.com.mobilemind.db.port=
br.com.mobilemind.db.backupPath=veloster
br.com.mobilemind.db.user=root
br.com.mobilemind.db.password=123456
#ddl can be: [create | update | none]
br.com.mobilemind.db.ddl=update
br.com.mobilemind.android.applicationPakage=br.com.mobilemind.veloster
br.com.mobilemind.android.dataBaseVersion=1
br.com.mobilemind.android.db.Location=/data/{0}/databases/{1}
br.com.mobilemind.db.backupSufixFormat=dd-MM-yyyy_HH-mm-ss
br.com.mobilemind.defaultDateFormat=yyyy-MM-dd hh:mm:ss.SSS
```

Após definidas as configurações podemos executar o código que, através das configurações definidas, vai preparar o amb iente do Framework.

Java Code

```
//Para desktop
VelosterConfig.build();
//Para Android
VelosterDroid.build(getApplication());
```

Por padrão, o Framework já está pré configurado para trabalhar com o banco de dados SQLite, mas p odemos mudar o driver para ele trabalhar com outro banco:

Java Code

```
//Por padrão trabalha com SQLite
new VelosterConfig().setDriver(new SQLiteDriver()).buildMe();
```

Ou

Java Code

```

//Configura o Veloster para trabalhar com MySQL
new VelosterConfig().setDriver(new MySqlDriver()).buildMe();
```

E com isso, temos o ambiente do Veloter configurado e pronto para uso.