

[DocV1.Setupnewdatabase](#)

Adding a support for another database

To add support for a new database you have to modify the eXo Platform code, but that's quite straightforward.

First you have to add the database constant in DatabaseService located at the path

exoplatform/exo-services/base-services/api/src/java/exo/services/database/DatabaseService :

```
public interface DatabaseService { public static final String HSQL =
'hsqldb'; public static final String MYSQL = 'mysql'; public static final
String DB2 = 'db2'; public static final String ORACLE = 'oracle'; public
static final String POSTGRESQL = 'postgresql'; public static final
String UNKNOWN = 'unknown'; public static final String
EXO_DATASOURCE_NAME = 'jdbc/ExoDS'; public static final String
EXO_WORKFLOW_DATASOURCE_NAME = 'jdbc/WorkflowDS'; public DataSource
getDefaultDataSource() ; public DataSource getDataSource(String name) ;
public void dropTable(String tableName, String dsName) ; public void
deleteAll(String tableName, String dsName) ; public boolean
hasTable(String tableName, String dsName) ; public String
getDatabaseType(String dsName) ; }
```

Then in DatabaseServiceImpl

(exoplatform/exo-services/database/impl/src/java/exo/services/database/impl/DatabaseService) you have to modify the public String getDatabaseType(String dsName) method.

```
public String getDatabaseType(String dsName) { String dbType = null ;
Connection conn = null ; try { DataSource ds = getDataSource(dsName) ;
conn = ds.getConnection() ; DatabaseMetaData data = conn.getMetaData() ;
String pname = data.getDatabaseProductName().toLowerCase() ;
if(pname.indexOf(DatabaseService.HSQL) >= 0) { dbType =
DatabaseService.HSQL ; } else if(pname.indexOf(DatabaseService.MYSQL) >=
0) { dbType = DatabaseService.MYSQL ; } else
if(pname.indexOf(DatabaseService.DB2) >= 0) { dbType =
DatabaseService.DB2 ; } else if(pname.indexOf(DatabaseService.ORACLE) >=
0) { dbType = DatabaseService.ORACLE ; } else
if(pname.indexOf(DatabaseService.POSTGRESQL) >= 0) { dbType =
DatabaseService.POSTGRESQL ; } else { dbType = DatabaseService.UNKNOWN ;
} catch (Exception ex) { ex.printStackTrace() ; } finally { if(conn
!=null) { try { conn.close() ; } catch (Exception ex) { } } } return
dbType ; }
```

Finally, in HibernateServiceImpl

(exoplatform/exo-services/database/impl/src/java/exo/services/database/impl/HibernateService
you have to modify the private void setDialect(String dbType) method to register
the hibernate dialect associated with your database type.

```
private void setDialect(String dbType){  
if(DatabaseService.HSQL.equals(dbType)) {  
config_.setProperty(DIALECT_PROPERTY,  
'net.sf.hibernate.dialect.HSQLDialect'); } else  
if(DatabaseService.MYSQL.equals(dbType)) {  
config_.setProperty(DIALECT_PROPERTY,  
'net.sf.hibernate.dialect.MySQLDialect'); } else  
if(DatabaseService.DB2.equals(dbType)) {  
config_.setProperty(DIALECT_PROPERTY,  
'net.sf.hibernate.dialect.DB2Dialect'); } else  
if(DatabaseService.ORACLE.equals(dbType)) {  
config_.setProperty(DIALECT_PROPERTY,  
'net.sf.hibernate.dialect.Oracle9Dialect'); } else  
if(DatabaseService.POSTGRESQL.equals(dbType)) {  
config_.setProperty(DIALECT_PROPERTY,  
'net.sf.hibernate.dialect.PostgreSQLDialect'); } }
```

[DocV1.Setupanewdatabase](#) (en)

Cr ateur: XWiki.benjamin.mestrallet Creation Date: 2005/02/09 14:20

Dernier Auteur: XWiki.benjamin.mestrallet Last Modification Date: 2005/02/15
22:55

Copyright 2004 (c) Auteurs des pages