Password Encryption

Current Status

Currently, whether the database password is stored inside `nuxeo.conf` or inside an environment variable, the configuration variable will be visible into the files generate by the configuration template system.

Basically :

```
nuxeo.conf
  nuxeo.db.password=snoopy
```

or

```
java -Dnuxeo.db.password="snoopy"
```

Will end up generating :

```
config/datasources-config.xml

[...]
 <datasource name="jdbc/nuxeo"
   [...]
   password="snoopy"
   accessToUnderlyingConnectionAllowed="true" >
</datasource>
[...]
```

To the question *"why is it not handled in a more secure way ?"*, then answer may be because there is no real clean solution to really make it secure.

See [tomcat wiki on "Why are plain text passwords in the config files"] for reference

Workaround : *"double escape"*

There is a trick to avoid the configuration files to contain the actual password.

Step 1

configure nuxeo.cong to forward to an other variable

```
nuxeo.conf
   nuxeo.db.password=${super.secret}
```

Then the configuration template will end up generating :

```
config/datasources-config.xml

[...]
 <datasource name="jdbc/nuxeo"
   [...]
   password=${super.secret}
   accessToUnderlyingConnectionAllowed="true" >
```

```
</datasource>
[...]
```

Then you can run Nuxeo with the `super.secret` environment variable set.

Symmetrical Cypher

To do better, we need to encrypt the password with a key.

This encryption needs to be symmetrical.

As always, the difficult part is to know where we store the key, but let's see that later.

Encrypted values

A simple solution is to say that encrypted values and prefexed :

A clean text value :

```
nuxeo.db.password=snoopy
```

An encrypted value :

```
nuxeo.db.password=enc:kjkiudisdufo
```

Decrypt

Provided we have a key, we could add the decryption logic inside `Framework.expandVars` :

- resolve values
- check prefix
- decrypt values having a prefix

Encrypt

Dedicated command line

```
> nxcrypt key=XXX value=snoopy
kjkiudisdufo
>
```

Integrated into nuxeoctl

```
> bin/nuxeoctl config-crypt key=XXX values=nuxeo.db.password,...
nuxeo.conf has been encrypted !
```

But, where is the key ?

That's actually where the "it's turtles all the way down" expression comes into play. we need to hide the key to be able to hide the password.

There are several options :

- we can use a simple java property
    - could be provided by env
    - could be provided as parameter of `nuxeoctl` (this is just a shortcut)
- we could use the Connect CLID as key

The first solution is probably the best option.