Factory: methods

Method types: select, selectfirst, count, update, delete, php

```
<method type="select">
    <parameter />
    <conditions />
    <order />
    <limit />
</method>
```

Parameter take the form of:

```
<parameter name="login" default="foo"/>
```

Conditions can take the form of:

```
<conditions logic="AND/OR">
    <eq property="foo" value="" />
    <neq property="foo" value="" />
    <lt property="foo" value="" />
    <gt property="foo" value="" />
    <lteq property="foo" value="" />
    <gteq property="foo" value="" />
    <like property="foo" value="" />
    <isnull property="foo"/>
    <isnotnull property="foo"/>
    <in property="foo" value="" />
    <notin property="foo" value="" />
</conditions>
```

jDao: data access

Getting Data

```
* $dao = jDao::get('foo~bar')
* $rec = $dao->get(1)
* $rec = $dao->findAll()
```

Creating data

```
* $rec = jDao::createRecord('foo~bar')
* $dao->insert($rec)
```

Updating data

```
* $dao->update($rec)
```

Deleting data

```
* $dao->delete(1)
```

Conditions

```
$conditions = jDao::createConditions();
$conditions->addCondition('label','=',$un_nom);
$conditions->addCondition('status','=',5);
$list = $myFactory->findBy($conditions);
```

jTpl

Instanciate

```
$tpl = new jTpl()
```

Assign

```
$tpl->assign($name, $value);
$tpl->assignIfNone($name, $value);
$tpl->assignZone($name, $zoneSelector, $params);
$tpl->assignZoneIfNone($name, $zoneSelector,
$params);
```

Get content

```
$content = $tpl->fetch('mymodule~mytemplate');
```

Using templates

```
{$j_basepath}, {$j_jelixwww}, {$j_themepath},
{$j_datenow}, {$j_timenow}

{$myvariable}
{$myvariable * 3}
{$onevariable|upper}
{$onevariable|upper|escxml}
{$myDate|jdatetime:"db_datetime","lang_date"}

{@foo~bar.baz@}

{if condition_1}
{elseif condition_2}
{else}
{/if}

{foreach $foos as $key => $foo}
{/foreach}

{for expression}
{/for}

{while condition}
{/while}

{meta_html css '/styles/my.css'}
{meta_html js 'fooscript.js'}
{meta_html bodyattr array('onload'=>'alert("load")')}
```


Authors are members of the Jelix Team :
- Olivier Denah <foxmask@foxmask.info>
- Loic Mathaud <loic@mathaud.net>

Complete Jelix documentation available on :
- http://jelix.org/

Thanks to the Jelix Team for help and review and Guy Philippe Uberos for images and design advices

JELIX Memento

for jelix 1.1.x

An open source php5 framework

Jelix-scripts

`jelix.php` is in `lib/jelix-scripts/` directory

`php jelix.php help` for available commands

Commands start by `php jelix.php` followed by :

```
--myapp createapp
--myapp createmodule moduleName
--myapp createctrl moduleName controllerName
--myapp createdao moduleName daoName tableName
--myapp createform moduleName formName daoName
--myapp createzone moduleName zoneName
--myapp initadmin entrypointName
```

Application directories

Standard application structure is :

```
myproject/
  |- lib/        jelix framework
  |- myapp/      your application
  |- temp/       temporary directory
```

Application directories :

```
myapp/
  |- modules/          modules of the application
  |- responses/        common responses
  |- plugins/          plugins (db, coord, tpl, ...)
  |- var/
    |- config/ configuration files
    |- log/            log files
  |- www/              document root of the application
```

Files naming conventions

- Classes	`classes/myclass.class.php`
- Controllers	`controllers/mycontroller.classic.php`
- jDao	`daos/foo.dao.xml`
- jForm	`forms/foo.form.xml`
- Locales	`en_EN/foo.UTF-8.properties`
- Templates	`templates/foo.tpl`
- Zones	`zones/foo.zone.php`
- jEvent	`classes/mylistener.listener.php`
- jResponse	`reponses/myHtmlResponse.class.php`

Global variables and constants

Global variables

```
$gJConfig
$gJCoord
```

Constants

```
JELIX_APP_PATH
JELIX_APP_TEMP_PATH
JELIX_APP_VAR_PATH
JELIX_APP_LOG_PATH
JELIX_APP_CONFIG_PATH
JELIX_APP_WWW_PATH
JELIX_APP_CMD_PATH
```

Selectors

They are used a lot in Jelix. They indicate a resource without having to know its path :

action, class, dao, locale, template, zone

Format : "`aModule~aResource`"

Example : jDao will look for a DAO file with the name myDao.dao.xml in the myModule module :

```
jDao::create('myModule~myDao')
```

Controller and action

Controller file

```
fooModule/controllers/bar.classic.php
```

Controller code

```
class barCtrl extends jController {
    public function index(){
        $resp = $this->getResponse('html');
        $resp->title = 'This is my first action';

        $user_name = $this->param('name');

        $tpl = new jTpl();
        $tpl->assign('user_name', $user_name);

        $resp->body->assign('MAIN',
                            $tpl->fetch('hello'));

        return $resp;
    }
}
```

Responses

```
$resp = $this->getResponse('response_type');
```

Response types

`atom1.0, cmdline, binary, css, html, htmlfragment,json, jsonrpc, lt2pdf, rdf, redirect, redirectUrl, rss2.0, soap, tcpdf, text, xml, xmlrpc, xul, zip`

Classes

Class file

```
fooModule/classes/bar.class.php
```

Using the class

```
jClasses::inc('fooModule~bar);
$bar = new bar();

$bar = jClasses::create('fooModule~bar');

$bar = jClasses::getService('fooModule~bar');
```

jDao: file

XML format

```
<dao xmlns="http://jelix.org/ns/dao/1.0">
   <datasources> datasources section </datasources>
   <record> properties section </record>
   <factory> methods section </factory>
</dao>
```

Datasources: tables

```
<primarytable name="p" realname="products"
    primarykey="id_product" />
```

Record: fields

```
<property
    name="a simple name"
    fieldname="filedname"
    datatype="" required="true/false"
    minlength="" maxlength="" regexp=""
    sequence="sequence name"
    updatepattern="" insertpattern=""
selectpattern=""
    default=""
/>
```