

ParamParser et WebBuilder

Framework de développement
d'application web
utilisé en bioinformatique

Emmanuel Courcelle

LIPM – UMR INRA-CNRS



ParamParser et WebBuilder

ne pas réinventer la roue...

... tout en restant « indépendant » ...

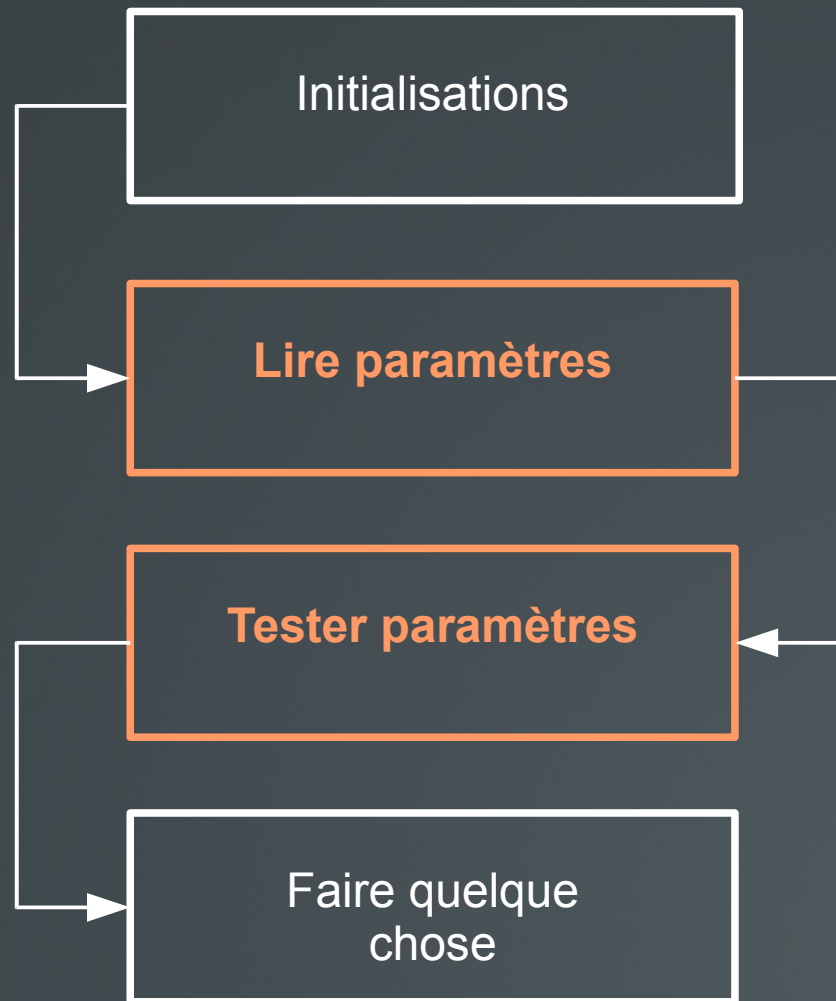
... et « kiss »

ParamParser

Analyser les paramètres



Un squelette de programme



Un programme complet

```
#!/usr/bin/perl
```

```
use strict;  
use warnings;  
use FindBin;  
use lib "$FindBin::Bin/../../corelib";  
use ParamParser;
```

```
my $o_param = New ParamParser( 'GETOPTLONG', \&Usage, 'input=s', 'n=i', 'help' );
```

```
$o_param -> AssertDefined ( 'input' );  
$o_param -> AssertFileExists ( 'input' );  
$o_param -> SetUnlessDefined ( 'n',10);  
$o_param -> AssertInteger ( 'n' );
```

```
PrintHead($o_param);
```

```
sub PrintHead
```

```
{
```

```
    my ($o_param) = @_;  
    my $fh_input = $o_param -> GetStreamIn( 'input' );  
    my $n        = $o_param -> Get ( 'n' );  
    for(my $i=0; $i<$n; $i++)  
    {  
        my $line = <$fh_input>;  
        print $line;  
    }  
}
```

```
sub Usage { print "Usage:\n\t./head.pl --input=some_file.txt [--n=count] [--help]\n"; }
```

D'où viennent les paramètres ?

de la ligne de commande

```
head.pl --input=file.txt --n=20
```

```
head.pl --help
```

```
my $o_param = New ParamParser('GETOPTLONG', \&Usage,  
                                'input=s',  
                                'n=i',  
                                'help') ;  
my $file = $o_param -> Get ('input');
```

D'où viennent les paramètres ?

d'un objet CGI.pm

```
my $o_param = New ParamParser('CGIPM') ;  
...  
my $file = $o_param -> Get ('input');
```


D'où viennent les paramètres ?

d'un fichier

```
input=file.txt  
n=50
```

```
my $o_param = New ParamParser('head.cfg') ;  
...  
my $file = $o_param -> Get ('input');
```


D'où viennent les paramètres ?

des variables d'environnement :

```
export input=file.txt  
export verbose=1
```

```
my $o_param = New ParamParser('ENV') ;  
...  
my $file = $o_param -> Get ('input');
```

Ils viennent de partout !

on définit un ordre de priorité

```
my $o_param = New ParamParser('ENV') ;  
$o_param -> Update ('head.cfg', '0') ;  
$o_param -> Update ('GETOPTLONG', '0',  
                    'input=s',  
                    'n=i',  
                    'help') ;  
my $file = $o_param-> Get ('input');
```

Valeurs par défaut

Fixer des valeurs par défaut

```
$o_param -> SetUnlessDefined('n',10) ;
```

Vérifier la validité

Paramètres requis, choix de valeurs, ...

```
$o_param -> AssertDefined('input') ;  
$o_param -> AssertNonEmptyFile ('input') ;  
$o_param -> AssertInteger ('n');
```

Ouvrir un fichier

Un paramètre contient un nom de fichier

Le fichier peut être compressé

```
input = some_file.txt.xz
```

```
my $fh_file = $o_param -> GetStreamIn ( 'input' );
```



Les behaviours

modifier le comportement de l'objet

```
$o_param -> SetBehaviour('use_substitution_table');
```

Fixer la valeur au runtime

les caractères %X peuvent être substitués

```
input=file-%d.txt
```

```
$o_param -> SetSubstitution('%d', \&GetDate) ;  
...  
my $file = $o_param -> Get ('input');
```


Namespaces

structurer les noms de paramètres

```
MAIN_input_file = some_file.txt
```

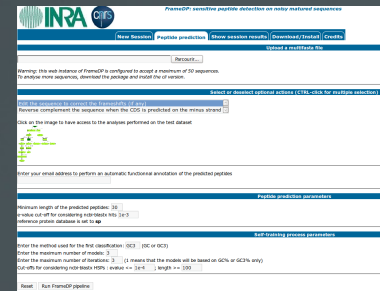
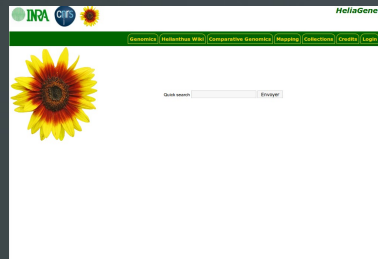
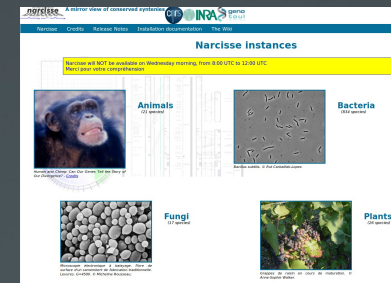
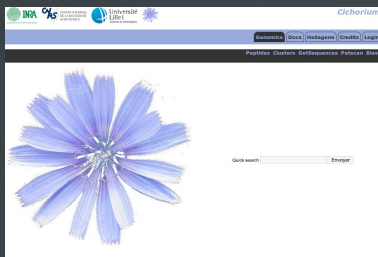
```
$o_param -> SelectNameSpace('MAIN_') ;
```

```
...
```

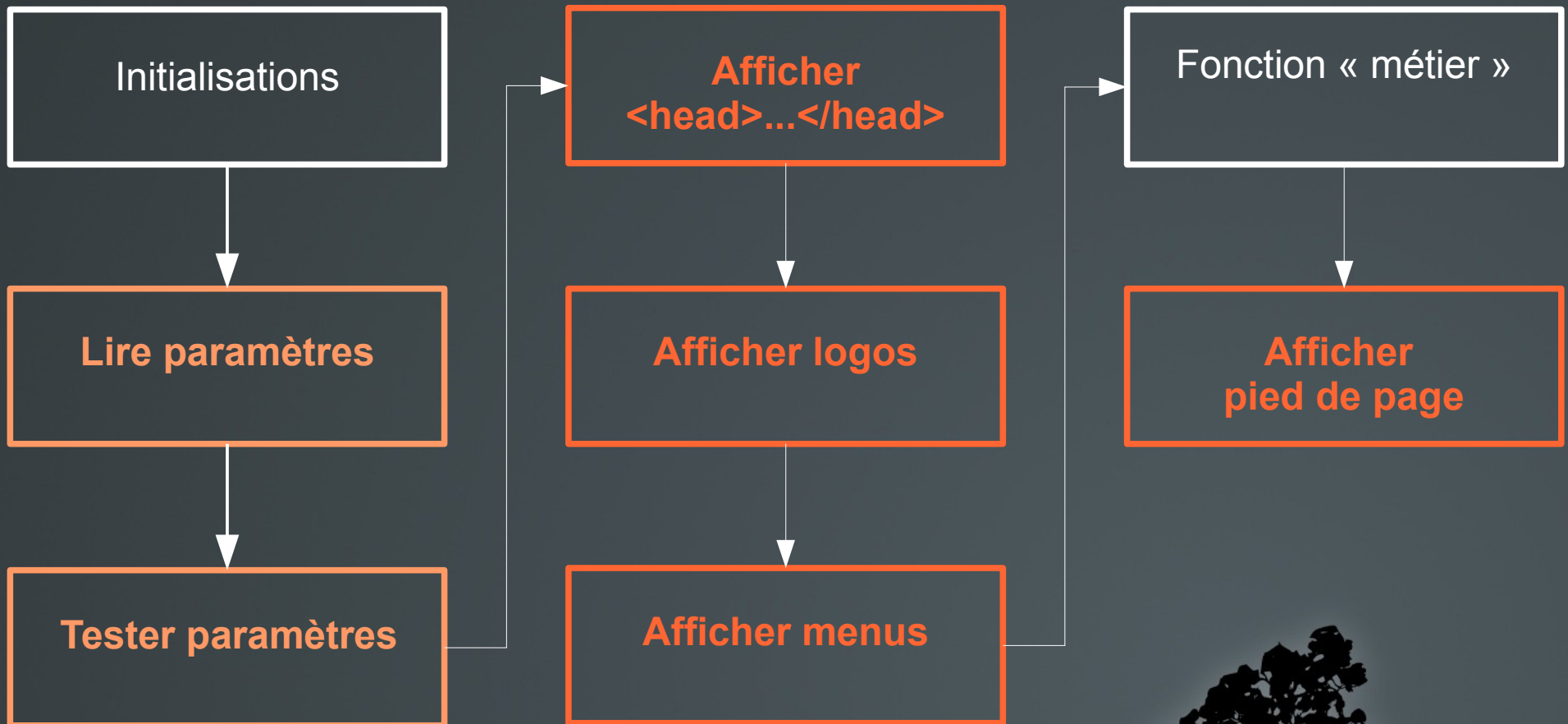
```
my $file = $o_param -> Get ('input_file');
```

WebBuilder

réaliser vite et bien son script cgi

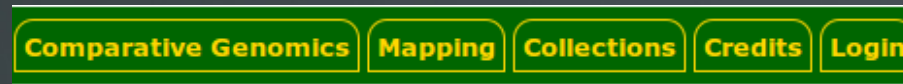
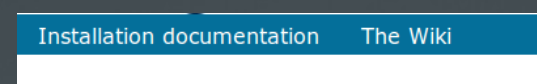
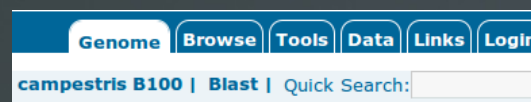
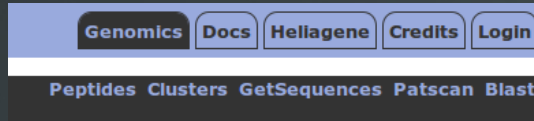


Un squelette de cgi



Décrire son interface

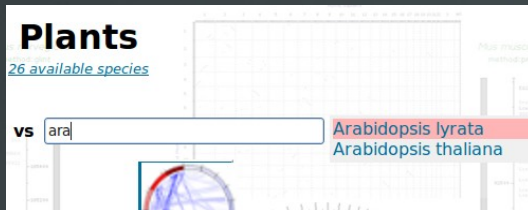
l'apparence...



```
<stylesheet>  
  <css id="A" url="css/WebBuilder_HH.css"/>  
</stylesheet>
```

Décrire son interface

le comportement...



```
<javascript>  
  <js id="A" url="js/autocomplete.js"/>  
</stylesheet>
```

Décrire son interface

les meta-tags...

```
<title>Narcisse rel 2.1.1: a mirror view of conserved syntenies</title>  
<meta name="Description" content="Tools and databases developed ... » />  
<meta name="Author" content="Emmanuel Courcelle, ..." />  
<meta name="Keywords" content="bioinformatique, bioinformatics, bioinfo, ..." />
```

```
<title>Narcisse...</title>  
<authors>Emmanuel...</authors>  
<description>Tools and ...</description>
```


Décrire son interface

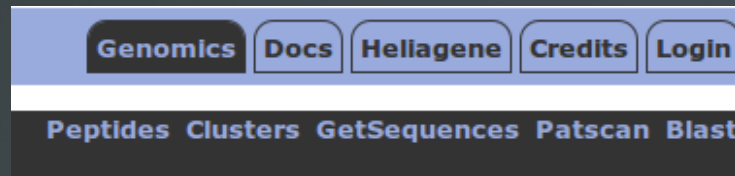
les logos...



```
<logo id="A" alt="inra"  
link="http://www.inra.fr..."  
img_src="http://..." />
```


Décrire son interface

les menus et sous-menus avec leur handler

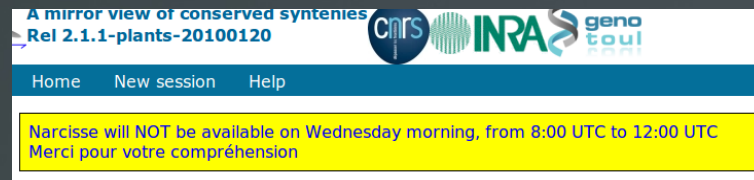


```
<main_menu_item id="M1" function="Genomics" label="Genomics">
  <sub_menu_item id="SM11" function="Peptides" >
    Peptides
  </sub_menu_item>
  <sub_menu_item id="SM12" function="Clusters">
    Clusters
  </sub_menu_item>
  ...
</main_menu_item>
```

Message Of The Day

afficher le message du jour ...

... seulement sur certaines pages



```
<motd id="M1" path="/.../motd.txt"  
      param="script"  
      value="^(HOME)?$" />
```

Désactiver l'application

si le fichier disabled est présent



```
<disabled path="../../../disabled.txt" />
```

Des paramètres persistants

paramètres propagés par menus et sous-menus :

`__wb_function`

`__wb_session`

`__wb_...` (*au choix*)

pas de cookies (sauf si authentification)

Construire WebBuilder

ParamParser → paramètres

Authentic → authentification

Nom de fichier → description xml

```
my $o_wb = New($o_param, undef,  
               './description.xml');
```

Les behaviours

modifier le comportement de l'objet

```
$o_wb -> SetBehaviour('use_substitution_table');
```

Sécurité...

Possibilité d'encoder l'url

```
$o_wb -> SetBehaviour('encoded_url')
```


Sécurité...

Jeu réduit de caractères autorisés

```
$o_param -> SetBehaviour( 'assert_value_secure' )
```

Les sessions

répertoire de session géré par WebBuilder :

- créé si nécessaire à l'initialisation
- `chdir` dans ce répertoire
- nettoyé au bout de quelque temps

```
$o_wb -> SetBehaviour('use_session');  
$o_wb -> SetBehaviour('session_autoclean');
```

Realize

affichage de la page web

- les en-têtes html, logos, menus,...
- appel de la fonction « handler »
- le pied de page

```
$o_wb -> Realize();
```

Ajax

requête ajax → pas de `<head></head>`

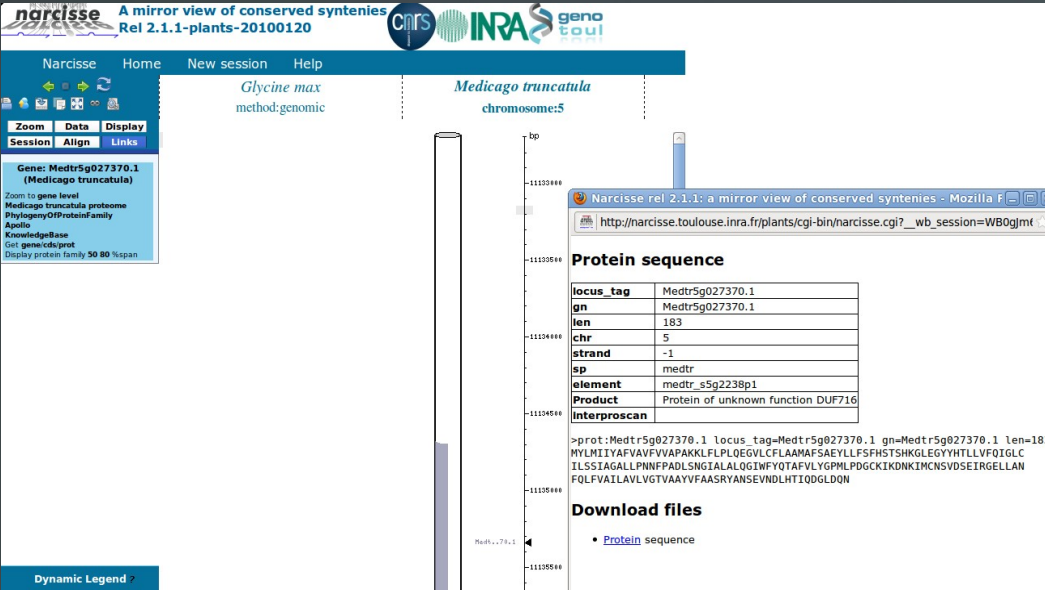
`__wba_ajax` est définie → requête ajax

`__wba_ajax =j` → type mime = `application/json`

Popups

popup → pas de barre de menus

`__wba_popup` est défini → requête de type popup



The screenshot shows the Narcisse web application interface. The main content area displays a comparison between *Glycine max* (method:genomic) and *Medicago truncatula* (chromosome:5). A sidebar on the left provides details for the gene **Medtr5g027370.1** (*Medicago truncatula*), including options to zoom to the gene level, view phylogeny, and access the knowledge base. The main content area shows a protein sequence viewer for the gene, with a popup window open over the sequence. The popup window displays the following information:

Protein sequence	
locus_tag	Medtr5g027370.1
gn	Medtr5g027370.1
len	183
chr	5
strand	-1
sp	medtr
element	medtr_s5g2238p1
Product	Protein of unknown function DUF716
Interproscan	

Below the table, the protein sequence is displayed in a monospaced font:

```
>prot:Medtr5g027370.1 locus_tag=Medtr5g027370.1 gn=Medtr5g027370.1 len=183
MYLMIIYAFVAVFVWAPAKKLLPLQEGVLCFLAAMAFSAEYLLFSFHSTSHKLEGYHTLLVFQIGLC
ILSSIAGALLPNNFPADLSNGIALALQGIWFYQTFVLYGPHLPDGGCKIKDNKIMCNSVDSEIRGELLAN
FQLFVAIHLAVLVTVAAYVFAASRYANSEVNDLHIQDGLDGN
```

The popup window also includes a "Download files" section with a link to the "Protein sequence".

Merci

<http://lipm-bioinfo.toulouse.inra.fr/download/lipmutils>