How to configure Mobyle

Mobyle 0.97

Contents

1 General Configuration

The Mobyle configuration file is write in python, so it must follow the python syntax.

1.1 Link mobyle with a web server

ROOT_URL: the root url and the port of the mobyle project. (mandatory)
HTDOCS_PREFIX : the extra path to the htdocs mobyle project. (mandatory)
CGI_PREFIX : the extra path to the cgi mobyle project. (mandatory)

example of Mobyle configuration using a virtual host: apache configuration:

```
<VirtualHost 192.168.0.3:83>
        ServerName server.domain.ext:83
        ScriptAlias "/cgi-bin" "/var/www/localhost/cgi-bin/mobyle/"
        DocumentRoot "/var/www/localhost/htdocs/mobyle/"
        DirectoryIndex index.html index.xml
```

```
        ErrorLog "/var/log/apache2/mobyle_error_log"
        TransferLog "/var/log/apache2mobyle_acces_log"
</VirtualHost>
```

Mobyle installation

```
python setup.py install --install-htdocs= /var/www/localhost/htdocs/mobyle/ \
                                    --install-cgis=/var/www/localhost/cgi-bin/mobyle/\
                                    --install-core=/any/where
```

Mobyle configuration

```
ROOT_URL = ‘‘http:mobyle.mydomain.ext:83’’
HTDOCS_PREFIX = ‘‘’’
CGI_PREFIX = ‘‘cgi-bin/mobyle’’
```

example of Mobyle configuration using a web subdirectory:
apache configuration:

```
ServerName server.domain.ext
ScriptAlias "/cgi-bin" "/var/www/localhost/cgi-bin/"
DocumentRoot "/var/www/localhost/htdocs/"
DirectoryIndex index.html index.xml
```

Installation

```
python setup.py install --install-htdocs= /var/www/localhost/htdocs/mobyle/ \
                                    --install-cgis=/var/www/localhost/cgi-bin/mobyle/\
                                    --install-core=/any/where
```

Mobyle configuration

```
ROOT_URL = ‘‘http:mobyle.mydomain.ext/mobyle’’
HTDOCS_PREFIX = ‘‘mobyle’’
CGI_PREFIX = ‘‘cgi-bin/mobyle’’
```

1.2 Mail

MAINTAINER: the emails list which will receive alert emails when problems occur in Mobyle. (mandatory)
HELP: the email adress where the users can have help about their jobs. (mandatory)
MAILHOST: the mail transfert agent used by Mobyle to send an email. (mandatory)
SENDER: the email address representing Mobyle which send, long job notification, results etc... (for further details, see MailTemplate section)
MAXMAILSIZE: if the results size is over MAXMAILSIZE only a notification of the end of his job is send to the user, in bytes. (default = 2097152 (2Mo))

1.3 Execution system

the Execution System has been rewrite to be higly flexible so it's configuration has completly changed. See execution_system documentation to learn how to configure it.

1.4 Logging

LOGDIR: the directory where are located the different file loggers. (default = /dev/null)

- access_log: to log the jobs launched
- error_log : to log the MobyleError

- build_log : to log all the step leading to build the command line (when debug >= 2)

ACCOUNTING: if its set to True an account_log will be created to log some statistics about jobs. This log file can be used to tune the execution system.

1.5 Data converter management

You can define through DATACONVERTER which converter(s) you want to use to manage the datatype(s) format.
For each Datatype you want to manage the format, you can provide an ordered list of converter(s) as following:

```
DATACONVERTER = {
'Datatype1': [ converter1_class('/path/to/bin/converter1'),
  converter2_class('/path/to/bin/converter2')     ] ,
'Datatype2': [ converter3_class('/path/to/bin/converter3') ]
                }
```

Basically, two converter classes are provided in Mobyle: squizz_alignment and squizz_sequence to manage respectively Alignment and Sequence formats with squizz program (we strongly recommend squizz).
Converter classes are located in MOBYLEHOME/Src/Mobyle/Converter.

Example to use the provided converters:

```
DATACONVERTER = {
'Sequence': [ squizz_sequence('/path/to/bin/squizz') ] ,
                'Alignment': [ squizz_alignment('/path/to/bin/squizz') ]
                }
```

1.6 Debug

DEBUG allow to set the default debug level in Mobyle which can be overload with PARTICULAR_DEBUG. This feature is used in conjonction of RESTRICTED_ACCESS to test/debug a program or it's interface. To do this turn the PARTICULAR_DEBUG to 2 or 3 (if you want to test the execution and results) for your program and restrict the access at his program to your own machin. Only you can access to the interface in the portal and the build_log register all steps of the command line building which could be useful to debug an interface. (default debug = 0)

example:
DEBUG= 0
PARTICULAR_DEBUG={ 'clustalw' : 2 }
In this example, all services have a debug level set to 0 except clustalw which is set to 2.

- Level 0 , used in production:
  - the command line is build
  - the build log is NOT fill
  - the job is executed
- Level 1 , to test a xml (python syntax in code , precond ...):
  - the command line is build
  - the build log is NOT fill
  - the job is NOT executed
- Level 2 ,to know what's wrong in the xml I wrote:
  - the command line is build

- the build log is fill
- the job is NOT executed

- Level 3, to test the xml and the job execution and the results retrieving:
  - the command line is build
  - the build log is fill
  - the job is executed

1.7 Binary path

BINARY_PATH is a list of strings representing the paths where the binaries could be found. Each element of the list must be a valid path. The order of the element is kept to build the final path. These pathes are add before the canonical PATH. (default BINARY_PATH = [ ]).
example to add 'usr/local/bin' to $PATH :
BINARY_PATH = ["/usr/local/bin"]

1.8 Data banks

DATABANKS_CONFIG describes the locally available databanks to fetch entries from, using the various utilities.

```
DATABANKS\_CONFIG = { name of the bank :{  'dataType' : determine which kind of data manage
                                             this bank, it must correspond to a
                                             Mobyle dataType.
                                          'bioTypes' : determine which kind of biological
                                             entity manage this bank, it must
                                             correspond to a Mobyle bioType.
                                          'label'    : the label show to the user.
                                          'command'  : how to retrieve an entry.
                                        }
                    }
```

example:

```
DATABANKS_CONFIG = { 'WGS' : { 'dataType' : 'Sequence',
                               'bioTypes' : [ 'Nucleic' ],
                               'label' : 'Genbank - Whole Genome Shotgun',
                               'command' : [ 'golden', '%(db)s:%(id)s' ] },
                     'PDB' : { 'dataType' : '3DStructure',
                               'bioTypes' : [ 'Protein' ],
                               'label' : 'Protein Data Bank',
                               'command' : [ 'PDBGet.py', '%(id)s' ] }
                   }
```

1.9 Statistics

USEGA : set this to True to activate Google Analytics support. Google Analytics requires a code unique for the web site. You must be the owner of the web site or an administrator must validate the site for you. By default, Google Analytics is disabled. Statistics will track visits and jobs run by visitors.
GACODE : code for this web site, required is USEGA is set to True.

1.10 Authentication

OPENID : set this to True to activate OpenId authentication. For OpenId, no user registration is required, user can connect directly to the portal, it will automatically create an AUTHENTICATED_SESSION. (default = False)

OPT_EMAIL : set this to True to allow the users to run a job without specify any email. (default = False)
PARTICULAR_OPT_EMAIL : you can override the general OPT_EMAIL option for a specific program.
example :
OPENID = True
OPT_EMAIL = False
PARTICULAR_OPT_EMAIL = { 'golden' : True }
The email is mandatory to run any programs except for golden (a very short program).

Welcome page configuration WELCOME_CONFIG is a dictionnary with 2 entries:

- **'url'** : point toward the document to include in the portal welcome page
- **'format'**: with 2 avalaible values:
  - **'html'** if the url point out an html page.
  - **'atom'** if the url point out an file in atom format.

news_ex illustrating example feed is in Example/Local directory.

You can perform a domain name resolution of the user email and search if this domain has a mail exchanger field (to avoid fake user email address) by setting DNS_RESOLVER = True. In this case dnspython must be installed. (default = False)

Each user have a space to store his data. This space can be temporary (during the working session) or more persistant. The temporary space is created when a user connect to the portal and is accessible during the work session this we call an ANONYMOUS_SESSION. Whereas the AUTHENTICATED_SESSION is created after the user register in the portal. In this case the user can retrieve his data each time he signin into the portal. The both user spaces can cohabit in same time.

anonymous session can take 3 values :

- 'no' : the anonymous sessions are not allowed.
- 'yes' : the anonymous sessions are allowed, without any verification.
- 'captcha' : the anonymous sessions are allowed, but with a captcha challenge (default)

authenticated session can also take 3 values :

- 'no' : the authenticated session are not allowed.
- 'yes' : the authenticated session are allowed and activated without any restriction.
- 'email' : the authenticated session are allowed but an email confirmation is needed to activate it (default).

1.11 Misc

- TIMEOUT:if a job is longer than TIMEOUT we consider this job as "long" job. Then the user is notify by email the end of this jobs. Otherwise the results are only show directly in the portal.(in sec mandatory)
- REFRESH_FREQUENCY: TODO (default = 240 sec)
- FILELIMIT : If a job generate a file exceeding FILELIMIT, the process is killed (default = 2147483648 , 2 Gib)
- SESSIONLIMIT: the user space size limit (in byte) (default = 52428800 , 50 Mib)
- PREVIEW_DATA_LIMIT: the size over which the results will not be displayed directly in the portal (default = 1048576 , 1 Mib)
- RESULT_REMAIN: the time, in days, the jobs are kept on the server. The jobs are cleaned by mobclean tool (see Tools/mobclean and Tools/README) which can be run in a cron. (default = 10 days).

1.12 Disabling services

Some times you need to disable the portal or a service (program , workflow) for maintaining operation etc... if DISABLE_ALL is True no new job could be submit, but the running job keep running.

To disable specifically one service (program or workflow) from any portal, you can append it in DISABLED_SERVICES. Joker can be used, so it's easy to disable all services from a given portal. This portal is call 'local'.
To re-enable services just toggle DISABLE_ALL to False or remove it from the DISABLED_SERVICES list example:

```
DISABLED_SERVICES = [ 'portal1.service1' , # disable the service1 from the imported portal1
                                           # (as defined in PORTALS )
                      'portal2.*' ,        # disable all services from the imported portal2
                      'local.clustalw*'    # disable all services begining by
                                           # clustalw (clutalw-multialign, clustalw-sequence ,
                                           clustalw-profile ) from this server.
                                           ]
```

By default all services are enabled

1.13 Restriction services access

By default all the programs available are usable by all users who can access your web server. But sometimes, due to some license restrictions etc..., you need to restrict the accessibility of some programs to some users. To do that use the AUTHORIZED_SERVICES. The filtering is based on the ip of the requester. AUTHORIZED_SERVICES is a dictionary with the service names of programs to restrict as keys and the list of ip which can access these programs as values.

```
AUTHORIZED\_SERVICES = { serviceURL :[ ip or ip mask , ..] \}
                                        the ip addresses which can use the service

AUTHORIZED_SERVICES = {
   'http://myMobyle.mydomain.fr/data/programs/toppred.xml' : [
             '125.234.60.18' , # only the machine with this ip could access to toppred
             '125.234.60.*'  , # all the machines in subnet could access to toppred
                                                              ]
                      }
```

1.14 Services management

These configuration variables are used to deploy the xml programs descriptions on the Mobyle web part from the Mobyle/Local/Programs and Mobyle/Programs. All the xml from Local/Programs are deployed. The xml from Programs are filtered following the rules below

- LOCAL_DEPLOY_ORDER : The order in which INCLUDE and EXCLUDE directive are evaluated.(default = ['include' , 'exclude'])
- LOCAL_DEPLOY_INCLUDE : The list of programs descriptions to install. (default = ['*'])
- LOCAL_DEPLOY_EXCLUDE : The list of programs descriptions to not install. (default = [ ])

For INCLUDE and EXCLUDE directives shell jokers could be used. By example, 'dna*' refers to all programs descriptions beginning by 'dna'.... By default all xml are deployed (include all , exclude nothing). for example, if you want deploy only "blast family" programs, you can configure mobyle like this: LOCAL_DEPLOY_ORDER = ['exclude' , 'include']
LOCAL_DEPLOY_INCLUDE = ['blast*']
LOCAL_DEPLOY_EXCLUDE = ['*']
but if you want all programs except the blast family programs: LOCAL_DEPLOY_ORDER = ['include' , 'exclude']

LOCAL_DEPLOY_INCLUDE = ['*']
LOCAL_DEPLOY_EXCLUDE = ['blast*']
Use the mobdeploy script which is located in Tools subfolder to deploy programs descriptions (for more details see associated README).

1.15 Grid aspects

TODO

PORTALS: The mobyle Srevr from which you want to import services. for a server you need to specify 6 fields: name, url, help, repository, programs, jobsBase.

- name: is the nickname you give to a Mobyle server. The programs imported from this server will be labeled with this name in the programs panel in the portal.
- url: the url of the portal.
- help: the email adress to send help messages.
- repository: the url where are store the xml interfaces.
- programs: the list of programs you want to import from this server.
- jobsBase: the url of the jobs repository.

EXPORTED_SERVICES: The list of local programs you want to export toward the other Mobyle server.

example:

```
PORTALS={
         'mobyleA': {
            'url': 'http://mobyle.domain.ex/cgi-bin/MobylePortal',
            'help' : 'user@domain.ex',
            'repository': 'http://mobyle.domain.ex/data/programs/',
            'programs': ['clustalw-multialign'],
            'jobsBase': 'mobyle.domain.ex/data/jobs'
            } ,
        'univB': {
            'url': 'http://bio.univB.fr/cgi-bin/',
            'help' : 'mobyle-help@univB.fr',
            'repository': 'http://bio.univB.fr/Mobyle/programs',
            'programs': ['muscle' , 'sspro' ],
            'jobsBase': 'http://bio.univB.fr/Mobyle/Results'
            }
         }

EXPORTED_SERVICES = [ 'protpars' , 'dnapars', neighbor' ]
```

2 MailTemplate

Mobyle use emails in several circumstances. You can tune the contents of each mail by modifying the appropriate template in Local/mailTemplate.py. A template is composed of two parts, the header and the body. the header nust contains the fileds "From" and "Subject" and can contains the fields "Cc", "Bcc", "Reply-To" and "Organization". For each template you can use some variables which will be expanded at runtime. all available variables are explain in Local/mail.template.py.

- CONFIRM_SESSION : if AUTHENTICATED_SESSION = 'email' an email is send to the user to confirm his registration.

- LONG_JOB_NOTIFICATION : when a job is longer than TIMEOUT we consider this job as a long job and a notification is send to the user to inform the user his job keep running.
- RESULTS_FILES : when a job is finished all results are send to the user as a zip archive (if email was specified).
- RESULTS_TOOBIG : if the results size is bigger than MAXMAILSIZE a notification of the end of the job and the url of the results is send to the user.
- RESULTS_NOTIFICATION : if a trouble occur during the results zipping, a notification of the end of the job and the url of the results is send to the user.
- HELP_REQUEST : whereas the previous emails are sent by "mobyle" SENDER to the user, this email is sent by the user to HELP. The user trigger this action by click on the "ask for help" button in the results page.
- HELP_REQUEST_RECEIPT: TODO

a CONFIRM_SESSION email is send to the user to confirm his registration, if AUTHENTICATED_SESSION = 'email'

3 Policy.py

Policy.py contain functions which are called in Mobyle but you can used them to plug your code in them to adapt mobyle to your local policy.

3.1 emailCheck

check if the email is according to the local rules. This function must return either:

- Mobyle.Net.EmailAddress.VALID : if the email is valid
- Mobyle.Net.EmailAddress.INVALID : if the email is rejected
- Mobyle.Net.EmailAddress.CONTINUE : to continue futher the email validation process

This function is called after the "syntax", "black list" and eventually (it depend of your configuration) before the "dns" checking.

3.2 authenticate

This function overload the Mobyle authentication session method. You can put here an authentication based on http, database/openID .... This function take 2 arguments (login, password) and must return either :

- Mobyle.AuthenticatedSession.AuthenticatedSession.CONTINUE: the method does not autentified this login/passwd. The fall back method must be applied.
- Mobyle.AuthenticatedSession.AuthenticatedSession.VALID: the login/password has been authenticated.
- Mobyle.AuthenticatedSession.AuthenticatedSession.REJECT: the login/password is not allow to continue.

4 Black list

This file define 2 structures to store the emails or the ip you want to forbid the access to your server.

4.1 users

The first structure "users" is a list of emails which are unable to run a job on your Mobyle.
users = ['foo@bar.com' , ...]

4.2 host

The second structure “host” is a list of ip which are unable to run a job on your Mobyle. You can use * as a joker to black list a whole subnet.
host = [’192.168.3.1’ , ’192.168.2.*’]