Erlang Inets

xushiweizh@gmail.com

2007-11-21

Outline

- inets
- http(d)
- ftp(d)
- httpd modules
- mod_esi
- example

inets

- Internet clients and servers
  - Currently a FTP client and a HTTP client and server has been incorporated in Inets.

inets http(d)

- **httpd**
  - An implementation of an HTTP 1.1 compliant Web server, as defined in RFC 2616.
  - Helper Modules
    - httpd_conf
    - httpd_socket
    - httpd_util
- http
  - An HTTP/1.1 client.
  - Helper Modules
    - http_base_64

inets ftp(d)

- tftp
  - Trivial FTP.
- ftp
  - A File Transfer Protocol client.

httpd modules

- mod_alias
  - URL ailasing, like apache url_rewrite module.
- mod_auth
  - User authentication.
- mod_cgi
  - The WWW Common Gateway Interface Version 1.1
- **mod_esi**
  - Erlang Server Interface (ESI). Which is a more efficient way of writing erlang scripts for your Inets webserver than writing them as common CGI scripts.
- ...

httpd mod_esi

- Conf: ErlScriptAlias
  - Format
    - ErlScriptAlias fakename realname allowed_modules
      - **fakename: Url to access.**
      - **realname: An erlang callback module.**
      - **allowed_modules: modules can be called by the erlang callback module.**
  - eg.
    - ErlScriptAlias /erl test_esi io
- Callback API
  - See OTP document: 'ESI Callback Functions' section of mod_esi.

example

- conf
  - See " /var/tmp/server_root/conf/8888.conf "
    - Port 8888
    - ServerName localhost
    - **ErlScriptAlias /erl test_esi io**
  - Yes, easy to understand. Its format looks like apache conf!
- test_esi.erl
  - An example of erlang callback module.
  - Yes, easy to write. See next slide.

test_esi.erl

```
-module(test_esi).
-export([yahoo/2, test/2]).

yahoo(_Env, _Input) ->
    "Location: http://www.yahoo.com\r\n\r\n".

test(Env, Input) ->
[
    "Content-type: text/html\r\n\r\n",
    "<HTML><HEAD><TITLE>Example</TITLE></HEAD>\r\n",
    "<BODY>\n",
    "<B>Environment:</B> ", io_lib:format("~p",[Env]), "<BR>\n",
    "<B>Input:</B> ", Input, "<BR>\n",
    "<B>Parsed Input:</B> ", io_lib:format("~p",[httpd:parse_query(Input)]), "\n",
    "</BODY></HTML>\n"
].
```

start web server

- httpd:start().
  - That's all. httpd will lookup and load conf file at '/var/tmp/server_root/conf/8888.conf'.
  - You also can give a conf file by calling httpd:start(Conf).

output

- http://localhost:8888/erl/test_esi/yahoo
  - Will go to 'http://www.yahoo.com'

- http://localhost:8888/erl/test_esi/test?a=1&b=2
  - Ouput:

```
Environment: [{server_software,"inets/4.7.11"}, {server_name,"xsw"},
   {gateway_interface,"CGI/1.1"}, {server_protocol,"HTTP/1.1"},
   {server_port,8888}, {request_method,"GET"},
   {remote_addr,"127.0.0.1"},
   {script_name,"/erl/test_esi/test?a=1&b=2"}, {http_accept,"image/gif,
   image/x-xbitmap, image/jpeg, image/pjpeg, application/x-
   shockwave-flash, */*"}, {http_accept_language,"en-us"},
   {http_accept_encoding,"gzip, deflate"},
   {http_user_agent,"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
   5.1; SV1)"}, {http_host,"localhost:8888"}, {http_connection,"Keep-
   Alive"}]
Input: a=1&b=2
Parsed Input: [{"a","1"},{"b","2"}]
```

Thanks