Erlang Academy

Лекция 4

Автор курса: Костюшкин Сергей

План

- Структура каталогов OTP-приложения
- Инструмент сборки erlang.mk
- Инструмент сборки rebar
- Фреймворк для тестирования Eunit

Структура каталогов

src/

ebin/

include/

docs/

test/

c_src/

deps/

apps/

Базовые консольные команды

ls - показать список файлов в текущем каталоге

mkdir {имя папки} - создать папку

rm -rf {имя файла/папки} - рекурсивно удалить

mv {откуда} {куда} - переместить / переименовать

touch {имя файла} - создать пустой файл

Компиляция проекта вручную

```
$ erlc -I include/ -o ebin/ src/*.erl
```

erlang.mk

erlang.mk (Github, Docs)

Makefile для самых маленьких

Создние проекта на базе erlang.mk

1. Создаем папку проекта:
   $ mkdir lesson_2
   $ cd lesson_2
2. Скачиваем erlang.mk:
   $ wget https://erlang.mk/erlang.mk
3. Создаем скелет проекта-библиотеки:
   $ make -f erlang.mk bootstrap-lib
4. Компилируем проект:
   $ make
5. Запускаем:
   $ erl -pa ebin/

rebar

rebar2 (Github, Docs)

rebar3 (Github, Docs)

Создние проекта на базе rebar2

1. Создаем папку проекта:
   $ mkdir lesson_2
   $ cd lesson_2
2. Скачиваем rebar:
   $ wget https://raw.github.com/wiki/rebar/rebar/rebar
   $ chmod u+x rebar
3. Создаем скелет проекта-библиотеки:
   $ ./rebar create-lib [libid=lesson_2]
4. Компилируем проект:
   $ ./rebar compile
5. Запускаем командой:
   $ erl -pa ebin/

Создние проекта на базе rebar3

1. Создаем папку проекта:
   $ mkdir lesson_2
   $ cd lesson_2
2. Скачиваем rebar3:
   $ wget https://s3.amazonaws.com/rebar3/rebar3
   $ chmod u+x rebar3
3. Создаем скелет проекта-библиотеки:
   $ ./rebar3 new lib lesson_2
4. Компилируем проект:
   $ ./rebar compile
5. Запускаем командой:
   $ erl -pa ebin/

Автоматическая пересборка с sync

```
$ cd $HOME
$ git clone git@github.com:rustyio/sync.git
$ cd sync
$ make
$ cd ..
$ sudo mv sync/ /usr/lib/erlang/lib/
$ sudo chown root.root -R /usr/lib/erlang/lib/sync/
$ erl
1> sync:go().
Starting Sync (Automatic Code Compiler / Reloader)
Scanning source files...
```

EUnit

Официальная документация EUnit

Пример модуля с EUnit

```
-module(fib).
-export([fib/1]).

fib(0) -> 1;
fib(1) -> 1;
fib(N) when N > 1 -> fib(N-1) + fib(N-2).
```

Пример модуля с EUnit

```
-ifdef(TEST).
-include_lib("eunit/include/eunit.hrl").
fib_test_() -> [
    ?_assert(fib(0) =:= 1),
    ?_assert(fib(1) =:= 1),
    ?_assert(fib(2) =:= 2),
    ?_assert(fib(3) =:= 3),
    ?_assert(fib(4) =:= 5),
    ?_assert(fib(5) =:= 8),
    ?_assertException(error, function_clause, fib(-1)),
    ?_assert(fib(31) =:= 2178309)].
-endif.
```

Макросы EUnit

%% Assert macros

?assert(BoolExpr)

?assertNot(BoolExpr)

?assertMatch(GuardedPattern, Expr)

?assertEqual(Expect, Expr)

?assertException(ClassPattern, TermPattern, Expr)

?assertError(TermPattern, Expr)

?assertExit(TermPattern, Expr)

?assertThrow(TermPattern, Expr)

Макросы EUnit

%% Macros for running external commands

?assertCmd(CommandString)

?assertCmdStatus(N, CommandString)

?assertCmdOutput(Text, CommandString)