

Catalyst

MVC web framework

Мой опыт

1

Что такое фреймворк?

1

Что такое фреймворк?

Фреймворк = рабочая среда

1

Что такое фреймворк?

Фреймворк = рабочая среда

Фреймворк \approx библиотека

1

Что такое фреймворк?

Фреймворк = рабочая среда

Фреймворк \approx библиотека

Не дает доступ к функции
`main()`

2

Что такое MVC?

2

Что такое MVC?

M — Model

(Данные, с которыми работаем)

2

Что такое MVC?

M — Model

(Данные, с которыми работаем)

Создавать, читать,
редактировать, удалять.

2

Что такое MVC?

V — View
(Представление)

2

Что такое MVC?

V — View
(Представление)

Система отображения данных
XML, HTML, JSON, etc...

2

Что такое MVC?

C — Controller
(Логика)

3

Что такое Catalyst?

Catalyst — MVC-фреймворк

Что такое Catalyst?

Catalyst — MVC-фреймворк

- а) Мощный
- б) Масштабируемый
- в) Гибкий
- г) «Зрелый»
- д) Популярный

4

Немного о гибкости

Model

DBI, DBIx::Class,
Rose::DB::Object,
CouchDB, XML::Feed,
YouTube, SVN,
etc...

4

Немного о гибкости

View

TT, HTML::Mason, XML,
TD, PDF, JSON, GD,
JavaScript, PHP
etc...

5

УСТАНОВИТЬ

```
$ sudo cpan Catalyst::Runtime \  
Catalyst::Devel
```

5.5

* `File::ChangeNotify`

`libfile-changenotify-perl`

6

Создать проект

6

Создать проект

```
$ catalyst.pl Hello
```

6

Создать проект

```
$ catalyst.pl Hello
```

(Проект Hello создан)

7

Запустить проект

```
$ cd Hello/script
```

```
$ ./hello_server.pl -r
```

8

Посмотреть на результат

<http://localhost:3000/>

9

Немного про урлы

action == «звено» URL
/hello/world => Hello::world()

9

Root.pm — «корневой»
контроллер

/ => Root::index()

/hello => Root::hello()

10

Контекст

10

КОНТЕКСТ

$\$c$ — КОНТЕКСТ

10

КОНТЕКСТ

$\$c$ — КОНТЕКСТ

$\$c \rightarrow \text{response} \rightarrow \text{body}$

10

КОНТЕКСТ

$\$c$ — КОНТЕКСТ

$\$c \rightarrow \text{response} \rightarrow \text{body}$

$\$c \rightarrow \text{request} \rightarrow \text{param}$

10

КОНТЕКСТ

$\$c$ — КОНТЕКСТ

$\$c \rightarrow \text{response} \rightarrow \text{body}$

$\$c \rightarrow \text{request} \rightarrow \text{param}$

$\$c \rightarrow \text{authenticate}$

11

Создаем экшен

12

Шаблон

12

Шаблон

HTML::Mason

\$./hello_create.pl view Web Mason

12

Редактируем экшен:

```
## $c->res->body("Hello!");
```

12

Шаблон рендерится
автоматически

(Почти)

12

Шаблон рендерится автоматически

(Почти)

Экшен

`end : ActionClass('RenderView')`

12

имя экшена = имя шаблона

13

Модель

13

Модель

DBI

```
$ ./hello_create.pl model DBI DBI
```

13.5

База данных

13.5

База данных

```
$ sqlite3 my.db
```

13.5

База данных

```
sqlite> create table `users` ( `id` integer primary key, `name` text );  
sqlite> insert into `users` ( `id`, `name` ) values (1, 'Bob');  
sqlite> insert into `users` ( `id`, `name` ) values (2, 'Bill');  
sqlite> insert into `users` ( `id`, `name` ) values (3, 'Barak');
```

14

Инкапсуляция запросов

Запрос = метод

14

Инкапсуляция запросов

Запрос = метод

```
SELECT * FROM `users`
```

14

Инкапсуляция запросов

Запрос = метод

```
SELECT * FROM `users`  
get_all_users()
```

15

Универсальный метод

`_execute_request()`

— Получает запрос

15

Универсальный метод

`_execute_request()`

- Получает запрос
- Выполняет его

15

Универсальный метод

`_execute_request()`

- Получает запрос
- Выполняет его
- Возвращает результат
построчно

15

$\$res \rightarrow [\dots] \rightarrow \{\dots\}$

Универсальный метод

```
my ( $self, $stm, $bind ) = @_;  
  
my $sth = $self->dbh->prepare( $stm );  
$sth->execute(@$bind);  
  
my @res;  
if( $stm =~ /^select/i ) {  
    while( my $ref = $sth->fetchrow_hashref ) {  
        push @res, $ref if keys %$ref;  
    }  
    $sth->finish;  
}  
  
return \@res;
```

16

get_all_users

```
my ( $self ) = @_;
```

```
my $res = $self->_execute_request(qq{  
    SELECT *  
    FROM `users`  
}, []);
```

```
return $res;
```

16

get_user_by_id

```
my ( $self, $params ) = @_;
```

```
my @bind = ( $params->{id} );
```

```
my $user = $self->_execute_request(qq{  
    SELECT `name`  
    FROM `users`  
    WHERE `id`=?  
}, \@bind );
```

```
return $user->[0];
```

17

Снова идем в контроллер

```
my $params = $c->req->params;
my $user =
    $c->model('DBI')
        ->get_user_by_id( $params );

$user ||= {
    name => 'Default User'
};
```

18

stash

```
$c->stash->{user} = $user;
```

18

stash

```
<%args>  
  $user,  
</%args>  
<h1>Hello, <% $user->{name} %></h1>
```

Полезные плагины

Catalyst::Plugin::Session

a) Store

b) State

Catalyst::Plugin::Authorization

Catalyst::Plugin::Authentication