



Описание консольного микрофреймворка

Вокруг PHP – экосистема
веб-приложений



Описание консольного микрофреймворка

Для того, чтобы его запустить нужно ввести команду:

```
Unset
php runner -c some_event
```

Эта команда создаст экземпляр класса в namespace **App\Commands** с именем класса **SomeEventCommand** и запустит метод `run()`. Если класс не найден будет выдано исключение **App\Exceptions\CommandNotFound**.

В конструктор команды будет передан экземпляр класса **App\Application**.

С помощью класса `Application` мы можем обратиться к `.env` файлу.

Для этого достаточно вызвать метод `env()` объекта класса `Application`:

```
Unset
// Файл .env

SOME_KEY=bot

// Код приложения

$app->env('SOME_KEY'); // Вернет строку 'bot'
```

Для подключения к БД используется `SQLite`.

Для его подключения задайте поле `SQLITE_DATABASE` в `.env` файле, куда нужно передать полный путь к файлу `sqlite`:

```
Unset
SQLITE_DATABASE=/home/occur/PhpstormProjects/reminger-tg-bot/database.sqlite
```

Все модели базы данных создаются в namespace **App\Models** и наследуют абстрактный класс **Model**, который требует, чтобы в конструктор был передан экземпляр класса **App\Database\Db**, который, по сути, является драйвером для соединения с СУБД.

Так как доступен только **App\Database\SQLite** рекомендую использовать его.

Для того, чтобы ваша модель заработала, нужно переназначить переменную `$table`, куда записать название базы данных, с которой связана модель:

```
Unset
Class Foo extends Model
{
    protected string $table = 'bar';
}
```

У модели есть 2 метода `insert` и `select`:

- **insert** – это надстройка над стандартным INSERT языка SQL. Принимает массив названий колонок и массив их значений
- **select** – это надстройка над стандартным SELECT языка SQL. Можно передать ему строку `$conditions`, куда передать нужные WHERE, ORDER BY, LIMIT и пр., но не обязательно.

К примеру, код может выглядеть так:

```
Unset
// вставка

$model->insert(['col1', 'col2'], ['text1', 'text2']); // Запишет строку где
col1 равно text1 а col2 равно text2.

// select

$model->select('WHERE col1 = text1 AND col2 = text2 LIMIT 1'); // вернет
[['col1'=>'text1', 'col2'=>'text2']]
```

Имя таблицы автоматически подтягивается из переменной `$table`.