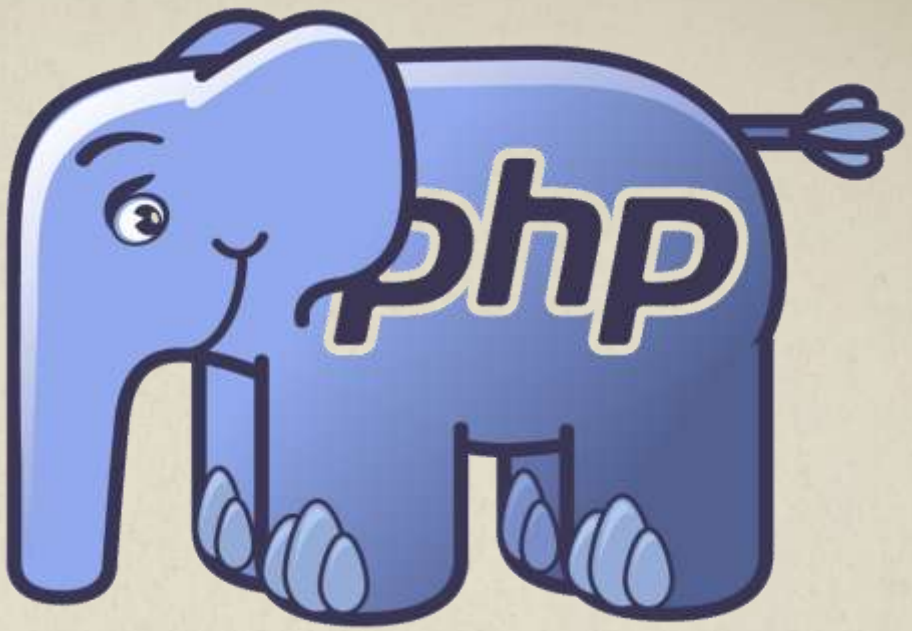


PHP: ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

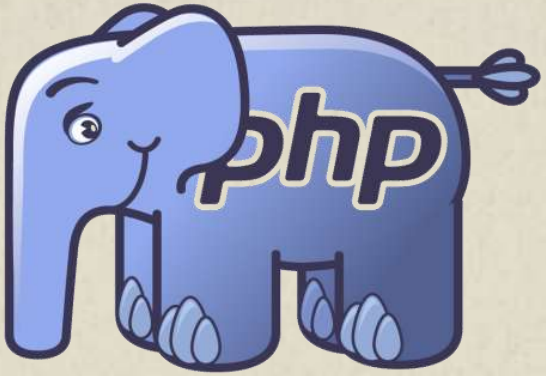
УРОК 4. КОНТРОЛЛЕРЫ И ФРОНТ-КОНТРОЛЛЕР

ОБРАЩЕНИЕ К «СКЛАДЧИКАМ»

Я считаю складчины – полной ерундой. Невозможно научиться чему-то, просматривая видео. Без домашних заданий, без общения с преподавателями и коллегами. Покупая в складчину видеозаписи курсов, вы вредите прежде всего самим себе, создавая иллюзию «обучения». И поддерживаете каких-то мутных личностей-«организаторов», имеющих свой процент.

Впрочем, дело ваше.

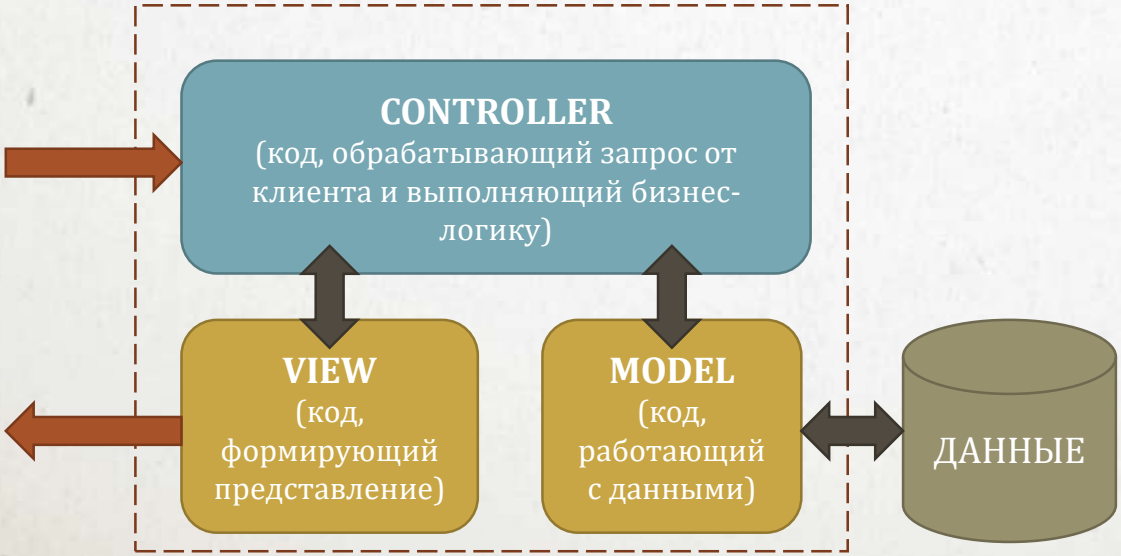
Однако, если вы хотите по-настоящему учиться – приходите. Адрес есть на слайдах. Напишите в поддержку, мол «я складчик, но я хочу учиться». Скидку гарантирую ☺



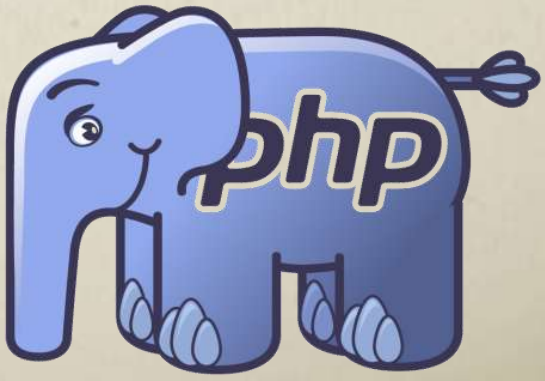
«КОНТРОЛЛЕР» В MVC

Контроллер – это точка входа!

- Его задача – принять запрос от клиента и понять, что хочет клиент
- Контроллер работает с данными через модели
- Контроллер готовит данные для представления и передает их ему
- Контроллер отвечает за выдачу ответа клиенту (возможно – используя слой View)



КОНТРОЛЛЕР

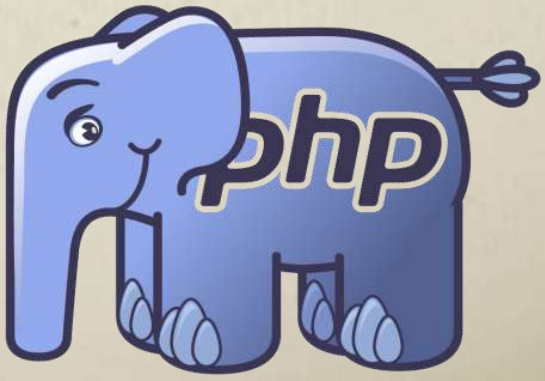


Контроллер не должен быть

- **Толстым**
- **Тупым**
- **Уродливым**

Чем меньше кода в контроллере – тем правильнее вы выбрали архитектуру проекта.

КОНТРОЛЛЕР



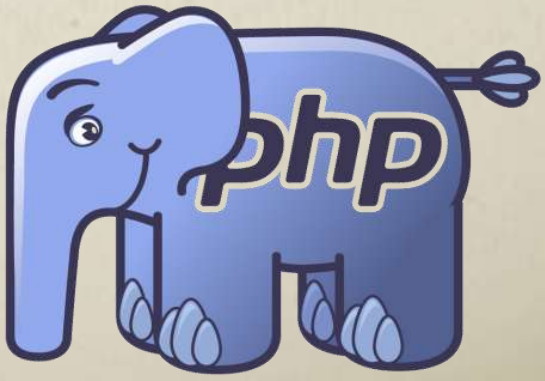
КЛАСС контроллера нам нужен, чтобы:

- Контроллер был объектом, и, значит, мы получили все возможности ООП
- Чтобы использовать преимущества наследования

КОНТРОЛЛЕР

```
namespace App\Controllers;

class News extends \App\Controller
{

  public function actionIndex();

  public function actionOne();

}
```

Вы видите выделение «действий», или “actions” – это и есть аналог «страницы сайта»



Что еще можно полезного сделать в контроллере?

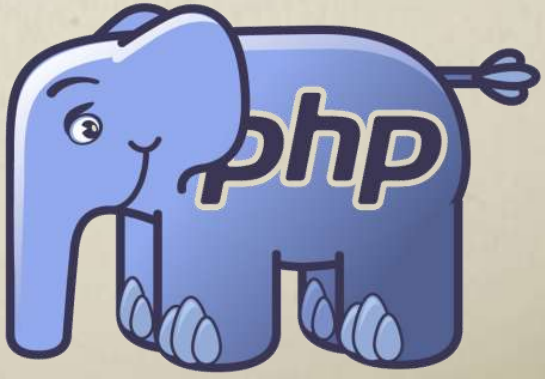
Многое!

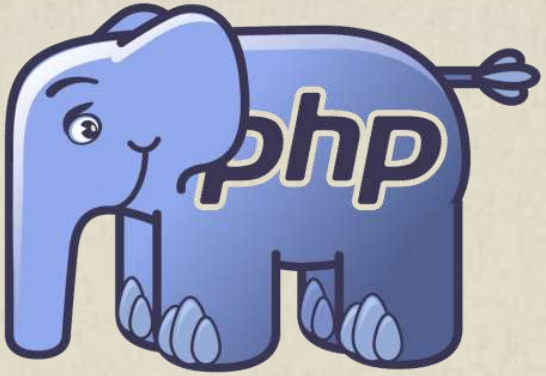
- В конструкторе создать нужные служебные объекты, например View
- Или получить там ссылки на объекты, например – текущего пользователя
- Создать метод beforeAction(), чтобы выполнять какие-то действия до действия ☺
- Написать метод access($action), чтобы проверять права доступа

И так далее!

N.B. *Главная дилемма – вызывать ли View в явном виде в action, либо это делать позже? Каждый фреймворк решает это по-своему…*

КОНТРОЛЛЕР





ФРОНТ-КОНТРОЛЛЕР И РОУТЕР

Но самый главный вопрос – а где же создается контроллер?

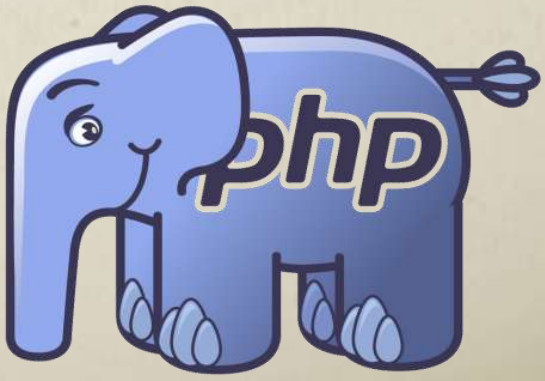
В фронт-контроллере ☺

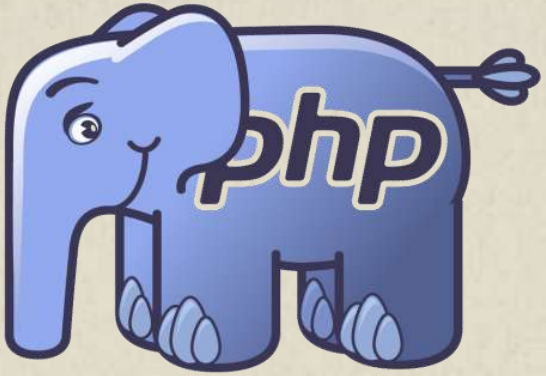
Этим термином обозначают ту часть, вашей программы, которая:

- Инициализирует приложение (выполняет некие начальные действия)
- В зависимости от запроса пользователя определяет, какой контроллер и экшн нужно вызвать

  (это называется «роутинг», а часть программы, соответственно, «роутер»)
- Вызывает их
- Возможно – обрабатывает ответ от действия

ФРОНТ-КОНТРОЛЛЕР





НЕМНОГО МАГИИ

Избавляемся от некрасивых адресов

1. Создаем в корне сайта файл `.htaccess`, пишем в него правила преобразования адресов:

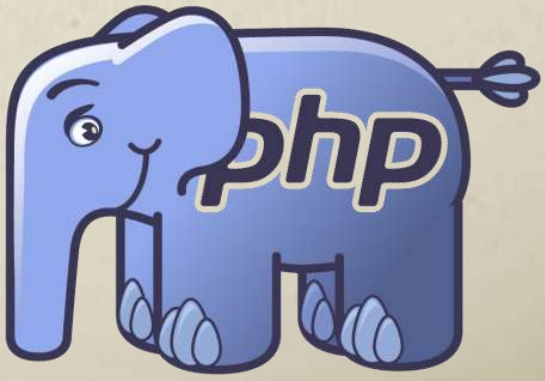
```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-l
RewriteRule ^(.*)$ index.php [QSA]
```

2. В результате теперь ЛЮБОЙ запрос к нашему сайту отправляется на index.php. Он и будет нашим фронт-контроллером

3. Внутри index.php используем переменную `$_SERVER['REQUEST_URI']` чтобы узнать какой адрес набирал пользователь

NB. *Только сервер Apache!*

MOD_REWRITE



ДО ВСТРЕЧИ НА СЛЕДУЮЩЕМ УРОКЕ!

**ВИДЕОЗАПИСЬ, СЛАЙДЫ, ПРЕЗЕНТАЦИЯ
И ДОМАШНЕЕ ЗАДАНИЕ
БУДУТ ВЫЛОЖЕНЫ ДО 10 УТРА СЛЕДУЮЩЕГО ДНЯ**

ProfIT