

# Краткое описание архитектуры приложения.

Приложение разработано на паттерне проектирования MVC.

Реализованные модели:

1. Model – общий компонент. Одноимённый класс реализован в файле Modules/Model.php. Имеет ряд методов и свойств для работы с базой данных. Предназначен для дальнейшего наследования моделями конкретных сущностей (в данной версии только Film).
2. Film – модель конкретного фильма. Класс реализован в файле Models/Film.php. Статические методы этого класса разработаны для работы с массивом фильмов (выгрузка, поиск, валидация). Все остальные методы и все свойства предназначены для работы с 1 конкретным фильмом, экземпляром данного класса.

Реализованные представления:

1. Шаблоны (реализованы в папке views/templates):
  - 1.1. film.php – общий шаблон для всех представлений контроллера FilmController (см. далее). Здесь реализована навигационная панель, а так же форма поиска фильмов.
  - 1.2. pagination.php – общий шаблон пагинации.
2. Компоненты представлений контроллера FilmController (views/film)
  - 2.1. 404.php – представление ошибки (страница не найдена).
  - 2.2. create.php – представление страницы добавления фильма (ов).
  - 2.3. index.php – главная страница приложения, здесь отображаются все фильмы, отсортированные по названию.
  - 2.4. view.php – страница отображения подробной информации о фильме.
  - 2.5. zero\_result.php – страница, сообщающая пользователю, об отсутствии результата поиска.

Представления устроены таким образом, чтобы класс View (см. далее) выгружал данные из переданного ему файла шаблона, затем – из файла-компонента и поставлял значение последнего в соответствующее ему место в шаблоне (`<?=$this->content?>`), перед этим сохранив эти данные в свойство content конкретного экземпляра этого класса.

Реализованные контроллеры:

1. Controller – общий компонент. Одноимённый класс реализован в файле Modules/Controller.php. Имеет перечень свойств, предназначенных для передачи их в другие функции/методы (при создании экземпляра этого класса свойства, в основном, базируются на информации о запросе, таких как массивы `$_GET`, `$_POST` и строке запроса URI). Все свойства этого

класса статичны и предназначены для парсинга параметров запроса, получения метода запроса и тому подобное.

2. **FilmController** – контроллер предназначенный для выполнения глобальных задач, таких как просмотр всех фильмов, поиск, удаление и просмотр конкретного фильма. Наследуется от **Controller**, получая, в основном, только его свойства, необходимые ему для выполнения операций в соответствии с полученными данными.

Подробнее о методах и свойствах контроллеров и моделей ищите в комментариях в соответствующих файлах.

## Запуск приложения.

1. Откройте командную строку.
2. Перейдите в пустой каталог, в который хотите копировать репозиторий.
3. Введите команду: `git clone https://github.com/GeminiCross/films.git`

После окончания копирования файлов из репозитория можно приступить к конфигурации приложения.

В первую очередь следует развернуть базу данных. Для рассмотрения возьмём пример СУБД MySQL.

Далее, по порядку:

1. Создайте новую базу данных (рекомендуется использовать кодировку `utf8_general_ci`).
2. Импортируйте информацию о её структуре из файла `films.sql`, что находится в одном каталоге с папкой `films` (в корне репозитория).

Перед запуском приложения рекомендуется просмотреть файлы конфигурации и комментарии к ним. Из параметров, которые необходимо изменить – только файл `dbconfig.php`.

Файлы конфигурации (большинство находятся в папке `config`):

1. `app.php` – содержит общую конфигурацию для приложения.
2. `dbconfig.php` – конфигурация базы данных.
3. `routes.php` – (находится в корне приложения). Хранит информацию о существующих контроллерах, их действиях и параметрах, необходимых для этих действий.

Каждый параметр описан подробнее в комментариях соответствующих файлов.

После конфигурации можно запустить проект. Инструкция для запуска на `php local server`:

1. В уже открытом терминале введите `cd films/films`.
2. Следующая команда `php -S localhost:8000` для запустит локальный сервер `php`.
3. Далее в браузере введите url: `localhost:8000/`

Для загрузки тестовых фильмов кликните по ссылке Add на верхней навигационной панели.

Под надписью «Загрузить фильм(ы) из файла» будет форма загрузки файла. Нажмите «Browse», в открывшемся окне выберите файл `testfilms.txt`, который находится в корне проекта рядом с `films.sql` и нажмите «Отправить файл».

После этого данные из файла, пройдя валидацию попадут в базу данных.

Теперь вы можете перейти на главную страницу и увидеть весь перечень добавленных фильмов.

## **Функционал:**

1. Добавление фильма: на главной странице по ссылке Add в верхней навигационной панели.
2. Удаление фильма: кнопка удаления фильма находится на главной странице, странице поиска и странице подробной информации о фильме.
3. Показать информацию о фильме: кнопка подробнее возле каждого фильма на странице просмотра всех фильмов и странице поиска.
4. Показать список фильмов, отсортированных по названию в алфавитном порядке: главная страница и страница поиска.
5. Найти фильм по названию: форма ввода находится на навигационной панели справа.
6. Найти фильм по имени актёра: в форме поиска сверху выбрать «by actor».
7. Импорт фильмов с текстового файла: страница добавления фильма, внизу есть 2 формы. Первая для загрузки подготовленного файла, вторая для получения файла примера, в котором находится шаблон для добавления указанного количества фильмов.