

Обзор преимуществ и недостатков концепции MVC

Круглик Роман Игоревич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье рассматриваются основные понятия MVC. Подробно расписаны все важные составляющие данной концепции. Выявлены преимущества и недостатки использования.

Ключевые слова: MVC, программирование, обзор.

Overview of the advantages and disadvantages of the MVC concept

Kruglik Roman Igorevich

Sholom-Aleichem Priamursky State University

Student

Abstract

In article discusses the basic concepts of MVC. Details are painted all the important components of this concept. Identified the advantages and disadvantages of using.

Keywords: MVC, programming, review.

На сегодняшний день огромное влияние в мире веб-разработки имеют паттерны проектирования. Самым популярным является MVC. Его используют не только для серверной части, но и при проектировании клиентской. На MVC создан JavaScript фреймворк AngularJs.

Цель исследования: анализ концепции Model View Controller для реализации проектов на PHP.

Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. Если взять популярные фреймворки yii и laravel, то там логику пишут в контроллерах, а модели являются экземплярами ORM. Существуют 2 вида MVC:

1. Page Controller;
2. Front Controller.

Первый является более простым и понятным, но не всегда более простой подход будет являться правильным. Каждый запрос к серверу направляется на собственный контроллер (чем больше страниц, тем больше контроллеров). Во втором виде все запросы на открытие страниц поступают на один контроллер, а он перенаправляет уже туда, куда нужно пользователю

Рассмотрим концептуальную схему шаблона MVC: (см. рис. 1).

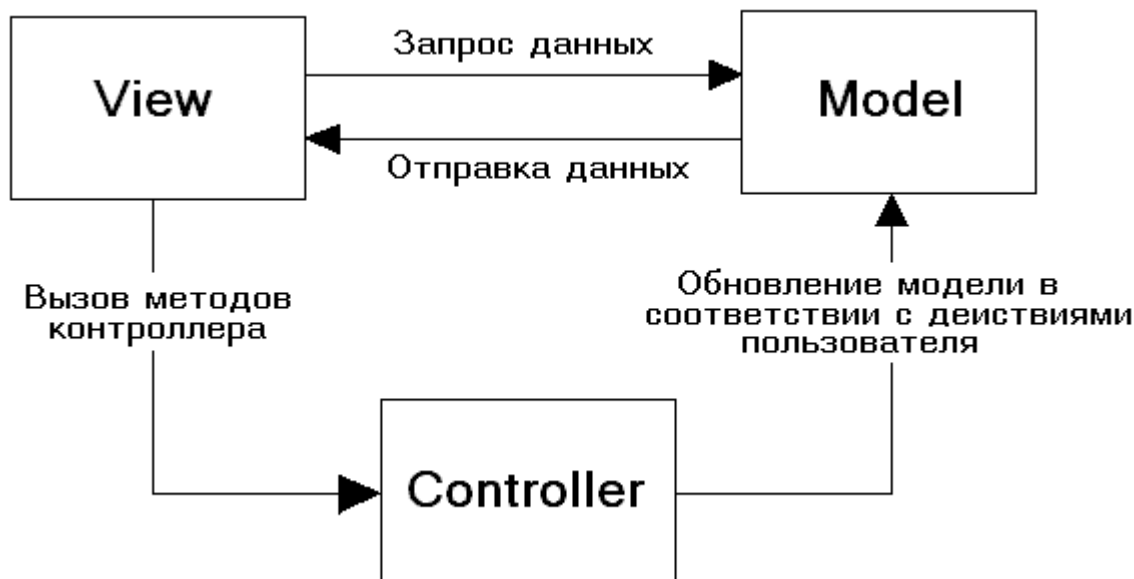


Рисунок 1. Концептуальную схему шаблона MVC

В архитектуре MVC модель предоставляет данные и правила бизнес-логики. Представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Модель - содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки и предоставления конкретных данных. Модель никогда не взаимодействует с пользователем напрямую и тем более не работает с внешним отображением, так как взаимодействует она только с контроллером.

Вид - используется для задания внешнего отображения данных, полученных из контроллера и модели. Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных. Они не обращаются напрямую к БД, не работают с данными, полученными по запросу клиента.

Контроллер - связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. Контроллер отвечает за обработку запросов пользователя. Контроллер не содержит и не обрабатывает SQL-запросы, не содержит HTML код.

Исследования в области применения концепции MVC не заканчиваются и по сей день. В статье К.А. Жерденко [1] рассмотрены современные подходы к разработке клиентской части веб-приложений(MVC). С.М. Гардейчик, А.И. Шербаф [2] приводят краткое описание PHP-фреймворка Laravel, реализующего шаблон MVC. В статье О.Н. Симонова [3] рассказывается о шаблонах проектирования MVC как эффективное средство построения архитектуры программной системы. Р.И. Ибраимов [4] создаёт тесты для spring mvc контроллеров.

Исходя из выше сказанного можно сформировать преимущества и недостатки использования такой структуры.

Преимущества:

1. Легко масштабируется;
2. Тестируется;
3. Сопровождается;
4. Реализуется.

Все эти преимущества обусловлены тем, что программист всегда знает где находится тот или иной кусок кода, который ему нужен. При исправлении багов, сопровождении и модификации не составит труда найти то, что ему нужно не зависимо от количества файлов и кода внутри.

Ярких недостатков у MVC нет, всё зависит от цели и задач, которые программист будет реализовывать. Если проекту не нужна модификация со временем, то и использовать достаточно сложную архитектуру не целесообразно. Недостатков у такого подхода к программированию, как и говорилось ранее нету, есть только преимущества одних шаблонов MVC над другими.

Начальная структура папок с использованием данной концепции может выглядеть вот так (см. рис. 2).

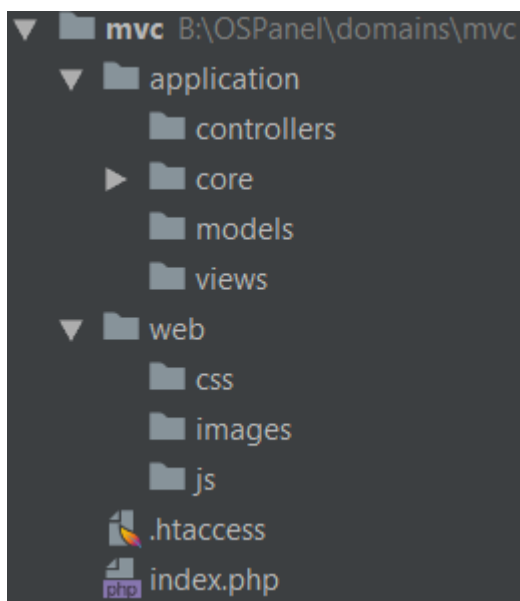


Рисунок 2. Структура папок

При входе на сайт пользователь автоматически делает запрос приложению на открытие определённой страницы, далее скрипт определяет какой контроллер и действие ему выбрать. Создаётся экземпляр контроллера и запускается нужный метод. Данный подход делает созданную систему устойчивее и снижает риск провала проекта на стадии его проектирования. В каждой второй вакансии веб-программиста указывается о необходимости знаний паттернов проектирования.

Библиографический список

1. Жерденко К.А. Применение mvc-паттерна во front-end разработке веб-

- программ // Синергия Наук. 2018. № 22. С. 653-657.
2. Гардейчик С.М., Шербаф А.И. РНР-фреймворк laravel с использованием архитектурной модели MVC // Перспективные направления развития отечественных информационных технологий Материалы III межрегиональной научно-практической конференции. Научный редактор Б.В. Соколов. 2017. С. 133-135.
 3. Симонова О.Н. Шаблон проектирования mvc как эффективное средство построения архитектуры программной системы // Студенческий научный форум -2014 VI Международная студенческая электронная научная конференция: Электронное издание. 2014.
 4. Ибраимов Р.И., Джемалетдинов А.Б., Шевченко А.А. Spring boot: создание тестов для spring mvc контроллеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. № 4 (18). С. 104-111.