Контроллеры и методы действия

№ урока:	3	**Курс:**	ASP.NET Core

Средства обучения:	Visual Studio 2019 .NET SDK

Обзор, цель и назначение урока

Научиться создавать контроллеры. Научиться получать данные контекста в контроллере. Изучить типы данных, представляющие результаты методов действий.

Изучив материал данного занятия, учащийся сможет:

- Создавать контроллеры и методы действий.
- Создавать контроллеры, используя базовый класс Controller.
- Использовать основные свойства класса Controller.
- Получать данные запроса, отправленные в теле запроса или адресной строке.
- Передавать информацию из контроллера в представление. Использовать базовые методы класса Controller для формирования разных результатов метода действий.
- Использовать перенаправления.
- Использовать в качестве ответа файлы.
- Использовать JSON в качестве результата метода действия.

Содержание урока

1. Что такое контроллер
2. Получение данных контекста
3. Чтение данных запроса
4. Передача данных из контроллера в представление
5. Перенаправления
6. Возврат JSON в ответе
7. Возврат содержимого разных типов
8. Возврат файлов в качестве ответа
9. Возврат статус кодов

Резюме

- **Контроллер** – C# класс, который содержит логику для получения запроса, обновления модели и выбора ответа. Контроллер — это центральная часть MVC приложения.
- Для создания контроллера необходимо создать класс с модификатором доступа public. Имя класса контроллера должно заканчиваться словом Controller. Любой открытый метод контроллера называется методом действия. Методы действия могут быть запущены через HTTP запрос, при условии наличия соответствующего маршрута в настройках маршрутизации.
- Класс контроллера не обязательно должен наследоваться от других классов, но наличие

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

базового класса Controller, дает доступ к разным свойствам, которые содержат информацию о запросе, обрабатываемом контроллером, и методам, позволяющим подготовить разные формы ответа пользователю.

- Важные свойства контроллера:
  - **Request** – Свойство возвращает объект HttpRequest. Описывает запрос.
  - **Response** - Свойство возвращает объект HttpResponse. Применяется для создания ответа.
  - **HttpContext** - Свойство возвращает объект HttpContext. Источник для многих объектов. Предоставляет доступ ко всем свойствам HTTP.
  - **RouteData** - Свойство возвращает объект RouteData. Доступ к значениям, полученным системой маршрутизации.
  - **User** - Свойство возвращает объект ClaimsPrincipal. Описывает текущего пользователя Контроллера. Используется для обработки запроса пользователя. При этом в запросе обычно находится информация, которую прислал пользователь.
- Получить доступ к свойствам контроллера можно разными способами. Наиболее удобный способ – использование привязки модели, механизма, который автоматически заполняет параметры методов действий, на основе данных существующих в запросе. Кроме того, что данные могут быть представлены в метод действия через привязку модели, их можно получить напрямую из запроса.
- Reqest.Form[key] – получение данных из **тела запроса**, например, когда информация отправляется на сервер через HTML элемент form
- RouteData.Values[key] – получение значений из **переменных сегментов запроса** (значения разделенные / в адресе запроса)
- Request.Query[key] – значения, взятые из **Query String** – значений после символа ? в адресе запроса)
- Упрощенный процесс обработки запроса:
  1) Запрос поступил на сервер, определен контроллер, который должен обработать запрос
  2) На контроллере запускается метод действия, в процессе выполнения формируются данные, которые нужно вставить в HTML разметку.
  3) Метод действия указывает, какое представление (HTML разметку) нужно вернуть и передает данные, которые в представлении будут использоваться.
- Есть несколько вариантов передачи данных из контроллера в представление:
  - ViewData
  - ViewBag
  - Модель и представление
  - Модель и строго типизированное представление
- ViewData это коллекция данных ключ-значение, а ViewBag - это обертка над ViewData, которая предоставляет удобный синтаксис обращения к значениям коллекции. Оба подхода позволяют без лишних затрат на написание кода передать информацию из контроллера в представление. Но этот подход будет полезен для небольших порций данных и незначительных изменений в интерфейсе, например - для установки Title страницы.
- Наиболее безопасный с точки зрения типов подход – использование строго типизированного представления и определения экземпляра модели в методе действия, возвращающего представление.
- Для создания строго типизированного представления, в начале cshtml файла необходимо указать **@model тип_модели,** при этом свойство Model данного представления будет того типа, который указан после @model. Такой подход дает дополнительные преимущества в виде проверки на этапе компиляции, возможности переименования идентификаторов во всем проекте, а также позволяет пользоваться другими возможностями редактора кода.

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Для удобства выбора результата работы метода действия используйте базовый класс Controller – он предоставляет ряд методов, которые позволяют удобно создать экземпляры классов результатов.
- Для того, чтобы метод действия мог возвращать разные результаты, используйте возвращаемый тип IActionResult.
- ActionResult – реализация по умолчанию для интерфейса IActionResult.
- Интерфейс IActionResult содержит метод ExecuteResultAsync, который асинхронно выполняет операцию создания ответа. Данный метод вызывает MVC для обработки запроса
- Для перенаправления пользователя на другой веб ресурс используется тип RedirectResult и методы из базового класса Redirect и RedirectPermanent (Для 301 перенаправления. Чаще всего используется для целей SEO оптимизации.)
- Для перенаправления пользователя на другой метод действия текущего приложения используется тип результата RedirectToActionResult и метод базового класса RedirectToAction.
- Для перенаправления пользователя на другой метод действия текущего приложения, но определяя по отдельности значения для сегментов, используется тип результата RedirectToRouteResult и метод базового класса RedirectToRoute.
- Для отправки статус кода, в ответ пользователю, используются методы NotFound() – 404, BadRequest() – 400, Unauthorize(), Ok() – 200, StatusCode(int) – возврат указанного кода.
- Для отправки файла, в качестве ответа пользователю, используются типы данных – FileContentResult, FileStreamResult, PhysicalFileResult, для всех этих типов используется один метод File с разными перегрузками.
- JSON – JavaScript Object Notation. Используется как альтернатива XML при передаче данных в сети.
- С помощью метода Json можно создать JsonResult и вернуть пользователю сериализованный в JSON объект, используемый в приложении?

Закрепление материала

- Что такое контроллер?
- Опишите принципы создания и именования классов контроллеров.
- Как получить доступ к данным, которые были отправлены через элемент <form> на странице?
- Как получить в методе действия контроллера значение id при обработке запроса, пришедшего по такому адресу http://localhost:50234/home/index?id=10
- Что такое ViewBag?
- Что такое строго типизированное представление?
- Когда следует использовать ViewBag, а когда строго типизированное представление?
- Какой интерфейс используют классы результатов?
- Какой класс является реализацией результата по умолчанию?
- С помощью каких методов можно сделать перенаправление, опишите их отличия?
- С помощью какого метода можно вернуть файл в ответе пользователю, если содержимое файла находится в виде массива в базе данных?
- Что такое JSON, где он используется?
- Как можно вернуть ответ из метода действия в виде JSON?

CyberBionic systematics

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Дополнительное задание

Задание
Создайте веб приложение. Добавьте модель, которая будет представлять коллекцию объектов со свойствами Id, Price, Name. Заполните коллекцию случайными значениями.
Сделайте контроллер Products с методом доступа Index, который возвращает представление со всеми данными из модели. Сделайте передачу данных в представление двумя способами, через ViewBag и через строго типизированное представление. С какими сложностями Вы столкнулись во время реализации представлений двумя разными способами?

Самостоятельная деятельность учащегося

Задание 1
Создайте веб приложение с пользовательским интерфейсом, представляющим простой калькулятор. Приложение должно иметь два поля ввода для числовых значений и несколько кнопок для арифметических действий. Реализуйте контроллер, который будет обрабатывать запросы и выполнять арифметические действия.

Задание 2
Создайте контроллер, который отображает пользователю информацию о его IP адресе и браузере, которым тот выполнил запрос.

Задание 3
Модифицируйте приложение SampleApp из домашнего задания к первому уроку таким образом, чтобы скачивание файла по ссылке на главной странице происходило с помощью выполнения метода действия, который возвращает файл.

Задание 4
Доработайте JsonController и его представления таким образом, чтобы на сторону клиента возвращался массив экземпляров Client в виде JSON, а также они соответствующим образом отображались в пользовательском интерфейсе.

Рекомендуемые ресурсы

Контроллеры в ASP.NET Core
https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/actions?view=aspnetcore-3.1

Получение доступа к HttpContext
https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/controller-methods-views?view=aspnetcore-3.1

Action Results в ASP.NET Core
https://www.tutorialspoint.com/asp.net_core/asp.net_core_action_results.htm

Action Results в Web API
https://docs.microsoft.com/en-us/aspnet/core/web-api/action-return-types?view=aspnetcore-3.1

CyberBionic Systematics ®
19 Mariny Raskovoy Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com