

Хранение состояния

№ урока: 6 **Курс:** Хранение состояния

Средства обучения: Visual Studio
.NET SDK

Обзор, цель и назначение урока

Систематизировать знания по разным механизмам хранения состояния в ASP.NET Core приложениях.

Изучив материал данного занятия, учащийся сможет:

- Правильно использовать разные подходы для хранения состояния на стороне сервера и на стороне клиента

Содержание урока

1. Хранение состояния в веб приложениях
2. Хранение состояния на клиенте
 - a. Адресная строка
 - b. Cookies
3. Хранение состояние на сервере
 - a. Session
 - b. TempData
 - c. Cache
 - d. HttpContext.Items

Резюме

Механизмы хранения состояния можно разделить на две категории – хранение на сервере и в браузере.

Query String

- Передача ограниченного количества данных в адресной строке
- Удобно использовать для создания ссылок с сохранённым состоянием
- Так как состояние адресной строки открытое, query string не должен использоваться для чувствительных данных

Cookies

- Сохранение данных между разными запросами
- Так как данные отправляются при каждом запросе, cookies должны содержать минимум информации. Большинство браузеров ограничивает размер до 4096 байт.
- Cookies могут быть удалены пользователем или срок их жизни может истечь.
- Часто используются для авторизации пользователя.
- Могут быть изменены на клиенте и требуют проверки на сервере

Session

- Сессия используется для сохранения данных, пока пользователь использует веб приложение.
- Сессия хранится на сервере, но на клиент в cookies передаётся ID сессии.
- Сессия одного пользователя не может разделяться несколькими браузерами.
- ID сессии в cookies удаляется когда заканчивается сессия браузера.

- По умолчанию сессия существует в течении 20 минут, но значение может быть изменено.

TempData

- Свойство TempData MVC контроллера или Razor страницы сохраняет данные до первого чтения.
- TempData удобно использовать для сохранения значения более чем на один запрос.
- Место хранения данных TempData зависит от выбранного TempDataProvider – сессия или cookies
- По умолчанию используется провайдер основанный на хранении данных в cookies

Выбор провайдера

- Если приложение уже использует сессию использование TempData на основе сессии не будет добавлять нагрузку на приложение.
- Если TempData используется не активно и используется для хранения данных до 500 байт, хранение данных можно производить в cookies.

Cache

- Эффективный способ хранения и получения данных. Позволяет повысить эффективность работы приложения сведя к минимуму время на повторную генерацию данных (создание копии данных, которая извлекается из кэша быстрее чем из оригинального источника)
- Данные хранятся как набор ключей/значений.
- Кэш НЕ связан с запросом, пользователем или сессией. Данные являются глобальными для всего приложения. Не сохраняйте данные специфичные для пользователя в кэш, так как они могут стать доступными для других пользователей.
- Код использующий кэш всегда должен иметь fallback опцию для сценария, когда кэш не доступен.
- Для уменьшения объема используемых ресурсов и для работ с актуальными значениями используйте таймауты и зависимости при добавлении записей в кэш.

HttpContext.Items

- HttpContext.Items –коллекция, которая используется для хранения данных во время обработки одного запроса.
- Коллекция очищается после завершения обработки запроса.
- Используется для передачи данных между компонентами или middleware во время обработки одного запроса

Закрепление материала

- Почему в веб приложениях необходимо сохранять состояние класса не в поле, а через специальные инструменты, которые предоставляет Framework?
- Какие инструменты используются для хранения данных на стороне сервера?
- Какие инструменты используются для хранения данных на стороне клиента?
- Как работает сессия?
- Как работают Cookies?
- Как работает TempData?
- Для чего используется кэширование? Какие способы можно использовать для того чтобы хранить актуальные данные в кэш?

Дополнительное задание

Задание

Создайте MVC приложение форму, на которой будет находиться два поля ввода, одно поле ввода для значения, а второе поле ввода для установки даты и времени (можете использовать HTML элемент управления) и кнопка, которая отправит данные на сервер. На сервере

полученная информация должна быть записана в Cookies с установкой даты устаревания равной той, которая была установлена во втором поле ввода в форме. Сделайте страницу, которая будет использоваться для проверки наличия значения в Cookies

Самостоятельная деятельность учащегося

Задание 1

Создайте MVC приложение, которое будет подсчитывать количество пользователей онлайн. Количество пользователей должно отображаться на странице вашего приложения. Для проверки работы приложения, запустите его в несколько браузеров. Если вы откроете приложение в трех браузерах, количество пользователей онлайн должно быть равное трем.

Рекомендуемые ресурсы

Управление состоянием

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/app-state?view=aspnetcore-3.1>

Кэширование

<https://docs.microsoft.com/en-us/aspnet/core/performance/caching/memory?view=aspnetcore-3.1.1>