Static Files

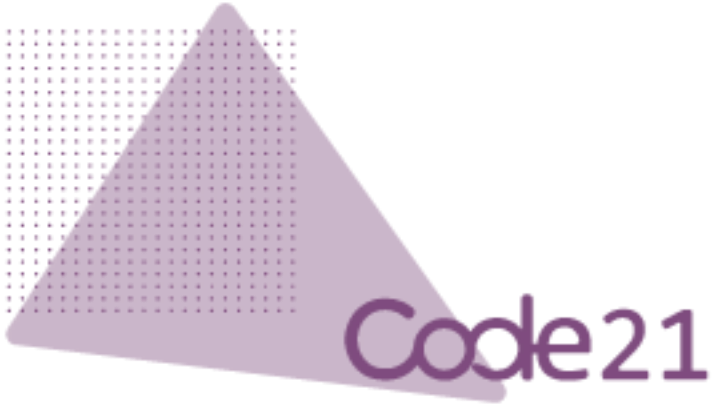
- Em uma aplicação ASP.NET Core precisamos explicitamente definir o uso de arquivos estáticos
- Utilizando o método “**UseStaticFiles**()”
- Por padrão o diretório para os arquivos estáticos será **wwwroot**
- Pode ser alterado pela classe **StaticFilesOptions**

```
app.UseStaticFiles(new StaticFileOptions
{
    FileProvider = new PhysicalFileProvider(Directory.GetCurrentDirectory())
});
```

State Management

O protocolo HTTP por padrão não armazena o estado da aplicação. Existem então algumas abordagens para que a aplicação consiga manter essas informações entre requisições.

- Cookies
- Session State
- TempData
- Query string
- Hidden Fields
- Cache



Controle de Sessão

O estado da sessão pode ser armazenado na memória do servidor enquanto o usuário navega pela aplicação. É vinculado ao cliente, ou seja, ao navegador, que cria grava uma chave(Session ID) no cookie local para identificar o usuário.

Algumas APIs e algumas aplicações como SignaR(WebSockets) não suportam o armazenamento se sessão no servidor.

Normalmente os dados de sessão expiram com o tempo, padrão é de 20 minutos.

TempData

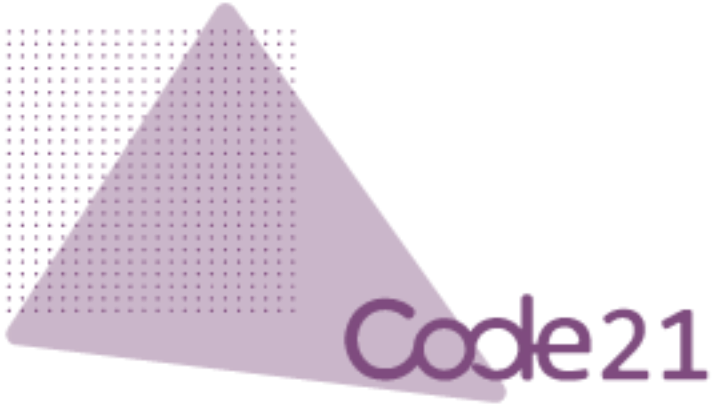
É um recurso que armazena um valor na memória do servidor, por usuário, até a próxima requisição.

- Util para passar valores após o redirecionamento
- Utiliza a mesma estrutura do controle de sessão
- Pode-se usar os métodos **Peek** e **Keep** para mantar as informações para a próxima requisição
- Também precisa habilitar os recursos pelo Services na inicialização

Hidden Fields

Os campos de formulário do tipo Hidden, permitem armazenar informações sem a exibição ao usuário.

- O servidor pode enviar informações para o formulário no navegador
- Dentro do campo hidden esses dados não aparecem
- Ao postar o formulário as informações são reenviadas ao servidor
- Os dados ficam, porém, visíveis no código fonte do HTML



Cache de Servidor

Processo eficiente de armazenamento e retorno de dados entre requisições.

- Implementa a interface *IMemoryCache*
- Mais rubusto e melhor estruturado que a Session
- Armazena os dados na memória do servidor
- Aumenta a performance e a escalabilidade da aplicação
- Mantém os dados para toda a aplicação, não apenas para a sessão

Cache de Servidor

- Rotina não deve depender da existência do cache
- Se o processo do servidor Web reiniciar o cache é excluído
- Em modelos de Web Form o cache fica onde foi criado
- Evitar inserir dados externos no cache para evitar consumo
- Sempre utilizar os parâmetros de expiração do cache

```
var cacheEntryOptions = new MemoryCacheEntryOptions()
    .SetSlidingExpiration(TimeSpan.FromSeconds(3));

_memoryCache.Set(CacheKeys.Entry, cacheValue, cacheEntryOptions);
```

```
var cacheEntry = _memoryCache.Get<DateTime?>(CacheKeys.Entry);
```

Exercício 11

Implementando um CRUD completo utilizando Cache