

# Tehnici Avansate de Programare

TPL și Securitatea în ASP.NET Core MVC

---

Petru Rebeja, Marius Apetrii

7 Mai 2020

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

# Introdurre

---

# Recapitulare

- **MVC** este un șablon de proiectare utilizat pentru a decupla interfața grafică (view), datele (model) și logica aplicației (controller).

# Recapitulare

- **MVC** este un șablon de proiectare utilizat pentru a decupla interfața grafică (view), datele (model) și logica aplicației (controller).
- **ASP.NET Core MVC** este o platformă open-source care permite dezvoltarea de aplicații Web pe baza convențiilor asociate șablonului MVC.

# Agenda

- Task Parallel Library
- Middleware
- Securitatea în ASP.NET Core MVC

# Task Parallel Library

---

## Execuție sincronă vs asincronă

Exemplu: Încărcarea unei pagini web de către server.

Utilizatorii A și B deschid pagina web din navigatorul web. Pentru a încărca datele serverul trece prin următorii pași.



## Execuție sincronă

1. Trimite interogare la baza de date pentru A
2. Așteaptă răspunsul
3. Primește răspuns de la baza de date pentru A
4. Generează HTML-ul
5. Trimite răspuns lui A
6. Trimite interogarea la baza de date pentru B
7. Așteaptă răspunsul
8. Primește răspuns la interogarea pentru B
9. Generează HTML-ul pentru B
10. Trimite răspuns lui B

## Execuție asincronă

1. Trimite interogare la baza de date pentru utilizatorul A
2. Trimite interogare la baza de date pentru utilizaotul B
3. Primește răspuns la interogarea pentru A
4. Generează HTML-ul pentru A
5. Trimite răspuns lui A
6. Primește răspuns la interogarea pentru B
7. Generează HTML-ul pentru B
8. Trimite răspuns lui B

## Task Parallel Library<sup>1</sup>

O mulțime de tipuri și funcții pentru a simplifica integrarea execuției paralele și concurente într-o aplicație.

---

<sup>1</sup><https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/task-parallel-library-tpl>

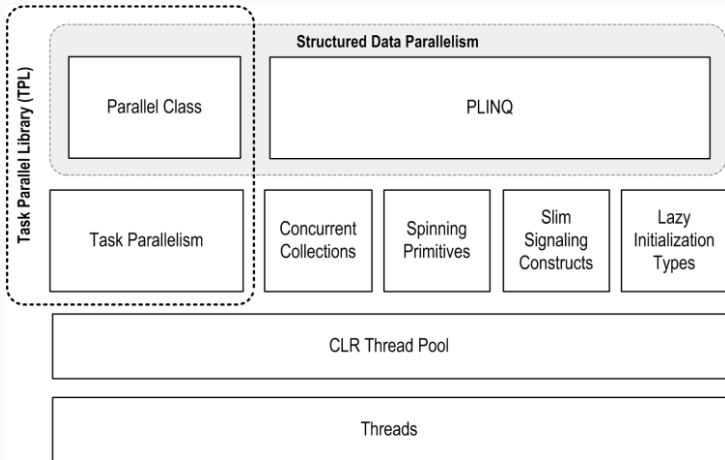
## Beneficiile TPL<sup>2</sup>

- Utilizarea eficientă a procesoarelor.
- Distribuie eficient munca între firele de execuție.
- Gestionează automat detalii precum:
  - Alocarea timpului de procesor,
  - Tranziția între stări etc.
- Oferă suport pentru anulare.
- Reduce codul de umplură.

---

<sup>2</sup><https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/task-parallel-library-tpl>

# Imaginea de ansamblu<sup>3</sup>



<sup>3</sup><http://www.albahari.com/threading/part5.aspx>

# Ce este un Task

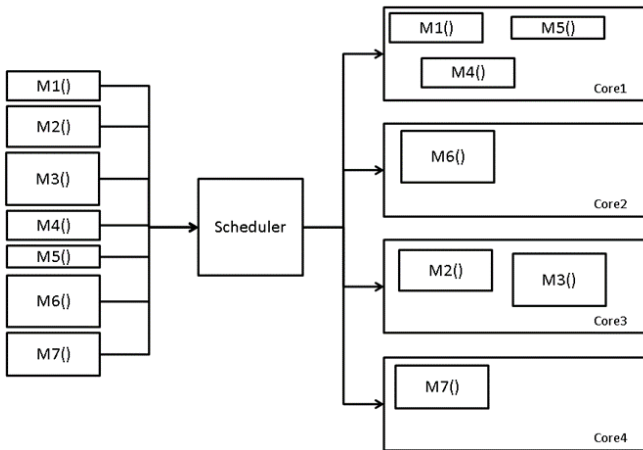
- Unitatea de bază din TPL.
- Reprezintă o operație care este executată asincron<sup>4</sup> și poate întoarce un rezultat.

---

<sup>4</sup><https://docs.microsoft.com/en-us/dotnet/api/system.threading.tasks.task>

- Un fir de execuție poate executa mai multe Taskuri.
- Un Task poate întoarce un rezultat (prin proprietatea `Result`); un fir de execuție nu.

## Task vs Thread<sup>5</sup>



<sup>5</sup>Diagrama este simplificată la un singur fir de execuție per nucleu (core) .



- `async` este aplicat în declarația metodei pentru a semnala că metoda se execută asincron.
- Metodele declarate cu `async` trebuie să întoarcă un `Task` sau un `Task<T>`.
- `await` este un operator care suspendă execuția metodei marcată cu `await` până la terminarea execuției operandului. Apoi metoda își reia execuția.

# Middleware

---

# Ce este middleware?

## Middleware

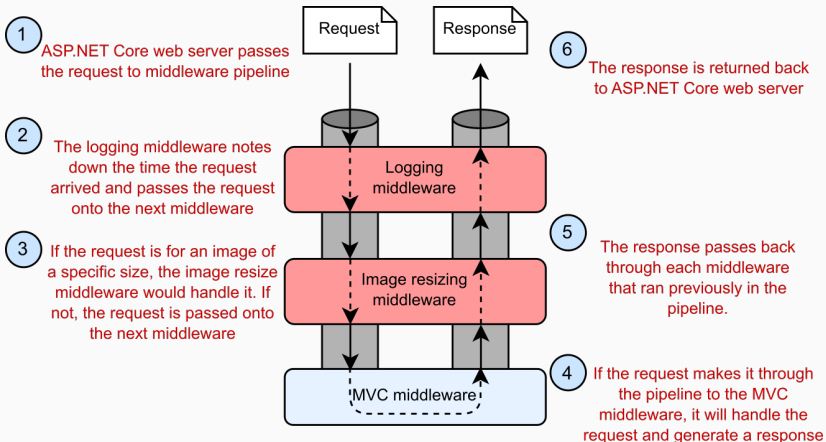
Componente software asamblate într-un sistem pentru a defini un flux de lucru care să proceseze interogările venite de la utilizatori și răspunsurile la aceste interogări<sup>6</sup>.

---

<sup>6</sup>[https:](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/)

[//docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/)

# Middleware<sup>7</sup>



<sup>7</sup>[https:](https://andrewlock.net/asp-net-core-in-action-what-is-middleware/)

[//andrewlock.net/asp-net-core-in-action-what-is-middleware/](https://andrewlock.net/asp-net-core-in-action-what-is-middleware/)

## Securitatea în ASP.NET Core MVC

---

## Autentificare

Procesul de preluare a datelor care atestă identitatea unui utilizator și verificarea acestora.

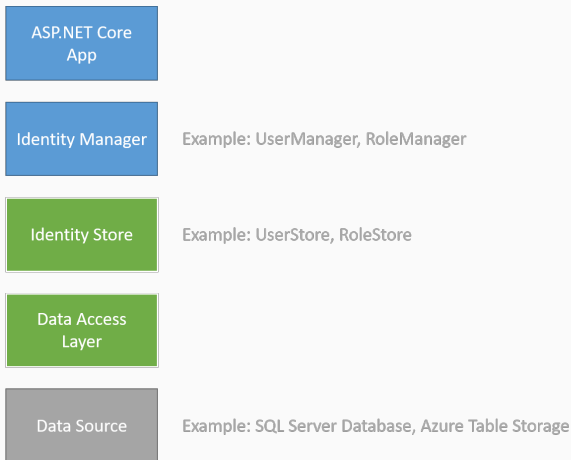
## Autorizare

Procesul de verificare a drepturilor de acces a unui utilizator în vederea accesării unei anumite resurse.

- Bibliotecă pentru implementarea autentificării.
- Gestionează utilizatori, parole, roluri ș.a.



# Componente<sup>8</sup>



---

<sup>8</sup><https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity-custom-storage-providers>

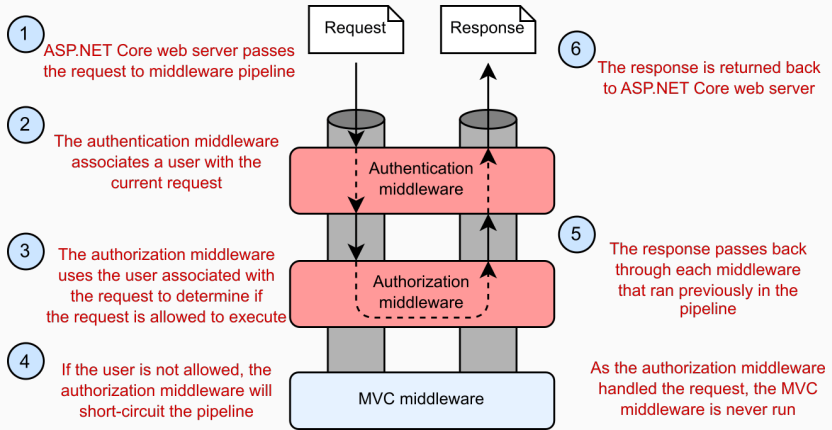
- Se face prin adnotarea controllerului sau metodelor din controller cu atributul `AuthorizeAttribute`.
- Poate fi de mai multe feluri.

- Simplă: `[Authorize]`
- Pe bază de roluri: `[Authorize(Roles="Admin;Manager")]`
- Pe bază de reguli: `[Authorize(Policy="EmployeeOnly")]`

---

<sup>9</sup><https://docs.microsoft.com/en-us/aspnet/core/security/authorization/introduction>

# Fluxul de lucru pentru autentificare și autorizare<sup>10</sup>



<sup>10</sup><https://andrewlock.net/asp-net-core-in-action-what-is-middleware/>

# Demonstrații

---

# Înceiere

---

# Recapitulare

- TPL simplifică integrarea execuției paralele și asincrone în aplicație lăsând în grija sistemului detaliile de nivel jos a.î. programatorul să se poată concentra asupra logicii aplicației.
- Middleware sunt componentele care fac parte din linia de procesare a interogărilor și răspunsurilor la interogări.
- Două componente de acest fel sunt middleware pentru autentificare și autorizare care se ocupă de atestarea identității utilizatorului și respectiv verificarea drepturilor de acces.
- Putem folosi `ASP.NET Core Identity` pentru a implementa autentificarea.
- Autorizarea se face cu ajutorul `AuthorizeAttribute`.

# Vă mulțumesc!

Mulțumesc pentru atenție!