

# LEKCE 09

HTTP Klient, Asynchronní programování



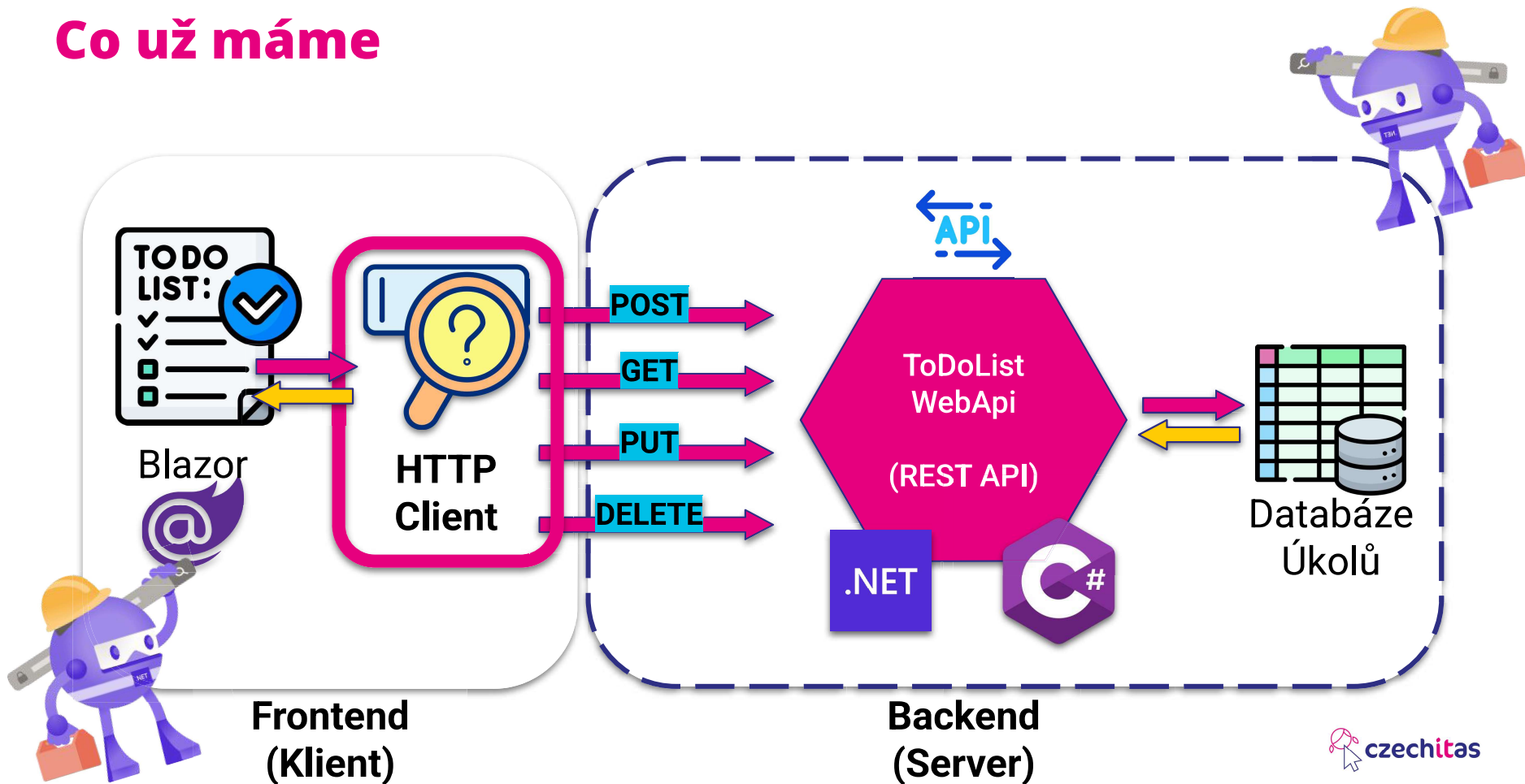
# Lekce 08

1. **HTTP Client**
2. **Asynchronní** programování
3. Závěrečný **projekt**



# HTTP Client

## Co už máme



**Pojďme do kódu**

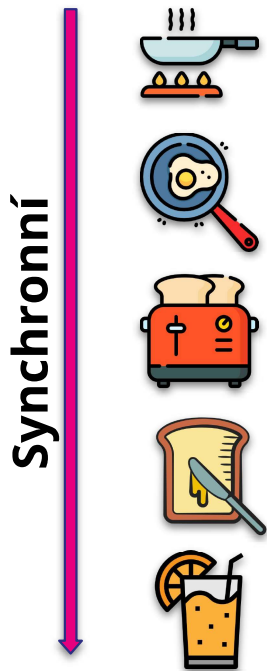
# **Asynchronní Programování**

The background is a solid magenta color. It features several faint, light-colored geometric patterns: a grid of small dots in the upper left, a large arrow pointing up and to the right in the upper right, a large 'X' shape in the lower right, and a circular shape with internal lines in the lower left.

# Synchronní a Asynchronní práce



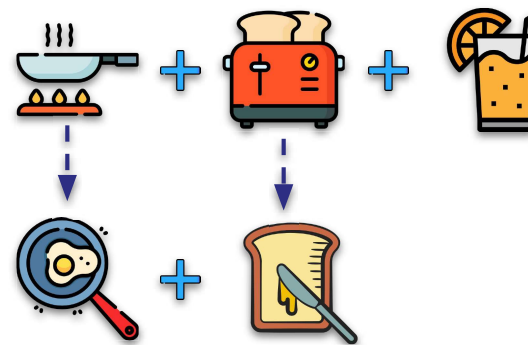
1 kuchařka (Thrída)



30 minut

Asynchronní

20 minut



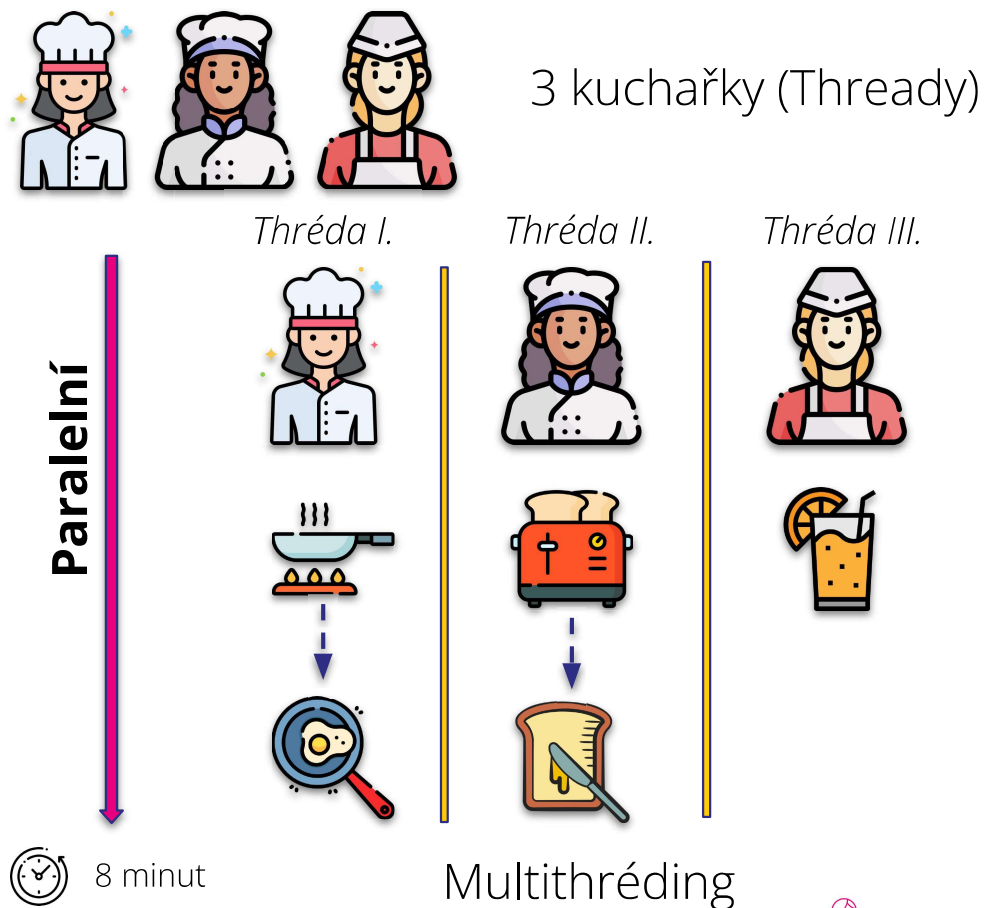
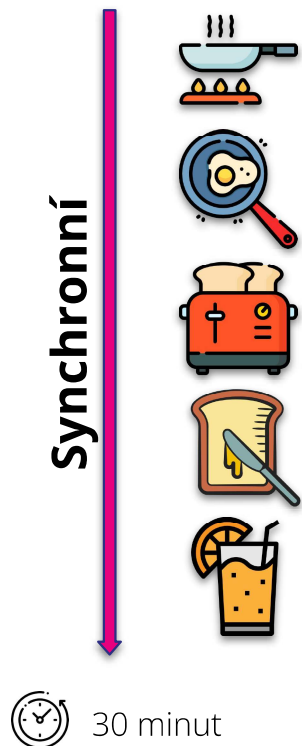
kuchař **nemusí aktivně čekat** na konec každé operace pokud to není potřeba

## Další příklady asynchronní práce

- Zaslání emailu nebo pozvánky na meeting
- Veškeré rezervace
- Stahování souborů
- Nakupování na internetu
- Objednání jídla v restauraci
- Praní prádla v pračce
- Vaření vody v konvici
- Nečekání na zrající ovoce



# Paralelní práce

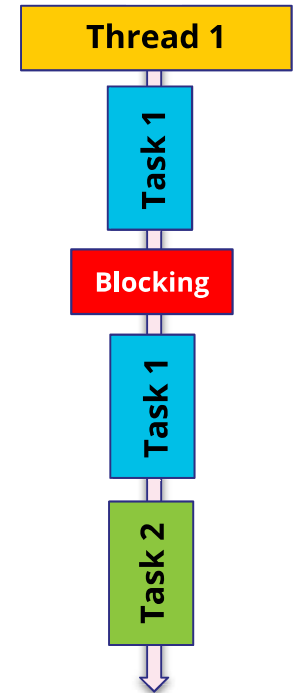


## Další příklady paralelní práce

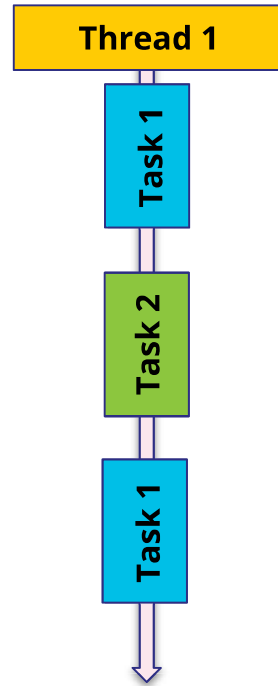
- Orchestr
- Kuchyně fast-foodového řetězce
- Natáčení filmu
- Výuka v celé škole dle rozvrhu
- Oprava silnice?



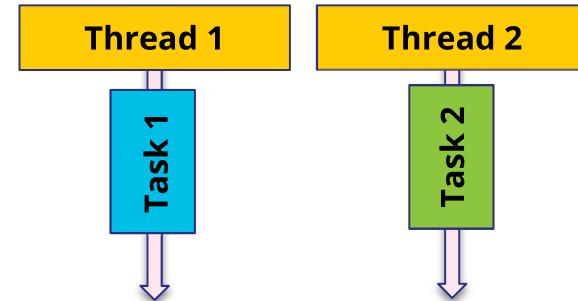
# Synchronní vs Asynchronní vs Multithreading



Jeden kuchař,  
**čekající** na každý  
krok

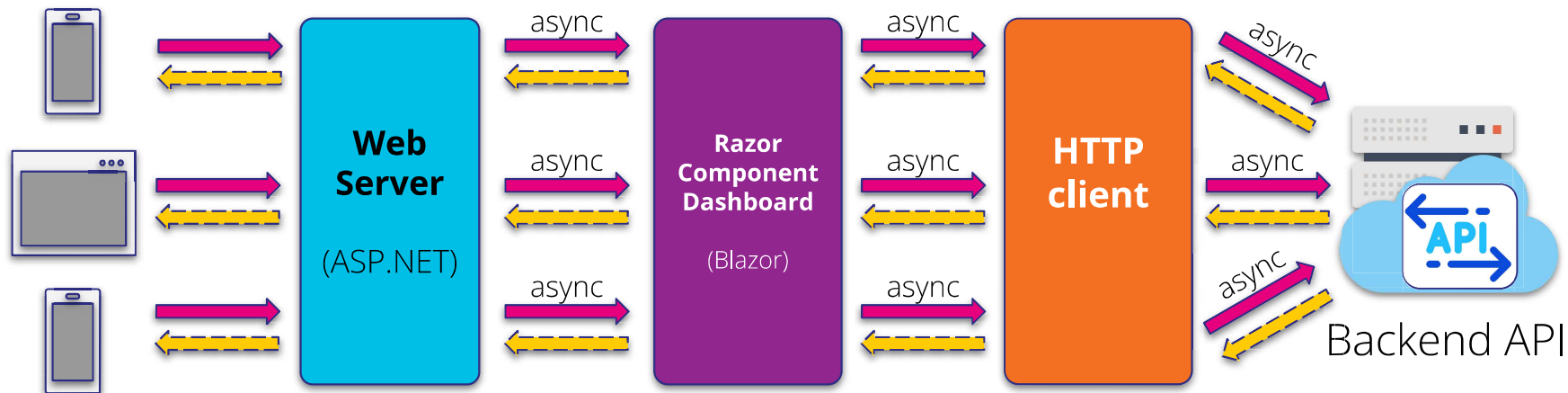


Jeden kuchař, **nečekající**  
na dokončení každého  
kroku. Místo aby čekal,  
dělá něco dalšího



Dva kuchaři,  
dokážeme jim rozdělit práci,  
pracují paralelně.

# Asynchronní programování v ASP.NET Core



## Benefity

- **Zlepšení Performance:** Vyhnete se blokování volání, thread ze které se metody volají se může věnovat dalším úkolům.
- **Zlepšení Scalability:** Naše aplikace takto může zvládat více requestů a uživatelů současně.
- **Zjednodušení Kódu:** Asynchronní kód je jednodušší na napsání díky Task objektům a async/await keywords

# Nová klíčová slova

- **Task<ReturnType>**
- **Task** (*ReturnType je void*)
- **async**
- **await**

# **Závěrečný projekt**

# Závěrečný projekt - Co očekávat

## ToDoList ToDoItem Categories

- Do ToDoList.Models.ToDoItem přidáme nový **string?** field
- **Category** je nový field, který musí být podporován
  - Na backendu (Databáze + EF + Repository + WebAPI + DTOs)
  - Na frontendu
    - Dashboard component
    - EditItem component (v příští lekci ;))



# Závěrečný projekt - Motivace

## ToDoList ToDoItem Categories

- Umíme samostatně pracovat s:
  - Modely
  - DTOs
  - WebAPI
  - Databází
  - Repository patternem
  - Testy
  - Blazorem





# Závěrečný projekt - Flow Odevzdávání

ToDoList Categories  
**Backend**

**2.12.**

Úprava backendu,  
aby podporoval kategorie

- Models
- Entity + Context
- Database Schema
  - Migrations
- DTOs
- Endpoint Controller
- Testy

ToDoList Categories  
**Frontend**

**9.12.**

Úprava frontendu,  
aby podporoval kategorie

- Dashboard component
- EditItem component
- Opravy BE na základě zpětné vazby

ToDoList Categories  
**Final**

**16.12.**

Prostor pro vychytání chyb a  
vylepšení programu a kódu.

- Vylepšení vzhledu pro uživatele frontendu
- Opravy FE na základě zpětné vazby
- Refactoring
  - jednoduchost
  - přehlednost
  - pojmenování

The background is a solid magenta color with several faint, light-pink geometric patterns. In the top-left, there is a grid of small dots. In the top-right, there is a large, stylized arrow pointing upwards and to the right. In the bottom-left, there is a line-art illustration of a plant with a large leaf and a smaller one. In the bottom-right, there is a large 'X' shape and a circular area with diagonal stripes.

# Bootstrap CSS

# Bootstrap

- Frontendový toolkit pro moderní styl stránek
- Responzivní, podporující grid systém
- CSS
- [Cheatsheet · Bootstrap v5.2](#)

<https://getbootstrap.com>