

# Instalacja i konfiguracja serwera WEB



# Wprowadzenie



- Implementacja serwera w oparciu o:
  - Moduł http
  - Moduł expres
- Moduły i klasy pomocnicze

# Express



- Jest modułem, który upraszcza wykorzystanie modułu http oraz rozszerza jego możliwości
  - Upraszcza obsługę tras, odpowiedzi, informacji cookie oraz statusu żądań
- Instalacja: `npm install express@4.0.0`
- Możliwe jest dodanie do package.json
- Przykład:

```
var express = require('express');  
var app = express();
```

# Konfiguracja komponentu



- Env – definiuje środowisko: development, testing i production
  - Trust proxy – obsługa proxy
  - Json callback name – nazwa wywołania zwrotnego: ?callback=
  - Json replacer – definicje funkcję wywołania zwrotnego (null)
  - Json spaces – liczba spacji w podczas formatowania odpowiedzi
  - Case sensitive routing – rozróżnianie /home a /Home (disabled)
  - Strict routing – rozróżnia /home oraz /home/
  - View cache – włącz/wyłącza buforowanie widoków szablonu
  - View engine – określa domyślne rozszerzenie
  - Views – określa ścieżkę dla mechanizmu szablonów
- 
- Ustawiane poprzez funkcję set(ustawienia,wartość) lub enable(ustawienie), disable(ustawienie)

# Implementowanie szablonów



- Zamiast budować strony uzupełnia się szablon o informacje
  - Prostota
  - Szybkość
- Przedstawiono dwa mechanizmy: Jade i EJS (Embedded JS)
- Do obsługi modułów szablonów zastosowano:
  - Npm install [jade@1.3.1](#)
  - Npm install [ejs@1.0.0](#)

# Definiowanie mechanizmów

- Zdefiniowanie domyślnego mechanizmu obsługi oraz katalogu z szablonami
    - `App.set('views', './views');`
    - `App.set('view engine', 'jade');`
  - Powiązanie rozszerzeń plików z silnikiem:
    - `App.engine('jade', require('jade').__express)`
    - `App.engine('ejs', require('ejs').__express)`
- UWAGA. Funkcje domyślne działają tylko dla domyślnych rozszerzeń:
- `App.engine('html', require('ejs').renderFile)`

# Przykłady



- Zapoznaj się z przykładami załączonymi przykładami: skrypty a\_... – o\_ ...
- Uruchom przykłady oraz doinstaluj brakujące moduły (jeżeli wymagane)
- Następnie na podstawie przykładów zrealizuj następujące zadania:

# 1.http



- Za pomocą modułu url:
  - Dokonaj podziału 3 adresów URL wybranych z sieci i rozpoznaj:
    - ✦ ich typ (pl, edu, com),
    - ✦ parametry query,
    - ✦ Ścieżkę oraz hash.
- za pomocą klienta http połącz się z dowolną stroną i wyświetl w konsoli wynik
- Zaimplementuj serwer http obsługujący statycznie 2 strony z katalogów /nowa i /stara (treści dowolne) (wykorzystaj strumienie)
- \* zainicjuj komunikację pomiędzy klientem a serwerem w ramach jednej aplikacji: co 1 sekundę klient wysyła zapytanie o datę a serwer zwraca dane w pliku json
- \* pobierz dane o pogodzie co 5 minut i udostępnij na stronie.



## 2. http i net



- Dodaj funkcję HTTPS dla utworzonych już stron /stara i /nowa
- Dodaj funkcjonalność dynamicznego generowania tablicy na stronie (na podstawie parametrów: znak, x, y)
- Za pomocą gniazda TCP prześlij plik pomiędzy klientem i serwerem
- \* zaimplementuj możliwość jednoczesnego wysyłania pliku do 5 osób

### 3. Express



- Zbuduj serwer http, który reaguje na 2 z 4 sposobów konfigurowania trasy,
- \* zastosuj wszystkie sposoby konfigurowania tras
- Udostępnij stronę /nowa i /stara za pomocą serwera Express
- Skorzystaj z szablonów do wygenerowania strony