# Spring Security i Spring Boot

Chodorowski Michał

# whoami

# Sprawy organizacyjne

- Na wszystkie pytania odpowiem po zakończeniu wykładu
- Jak nie dostałeś welcome packa napisz na IG
- Nagrody za poprawne odpowiedzi na pytania w trakcie wykładu :
  - 2x Czysty Kod - Robert C. Martin
  - TDD - Sztuka tworzenia dobrego kodu - Kent Beck
  - Wzorce projektowe GOF
  - Godzinna konsultacja ze mną
  - Ebook “Od czego zacząć przygodę z programowaniem” - you can dev

# Agenda

Sekcja 1: Wprowadzenie

- Czym jest Spring Boot?
- Architektura Spring Boot

Sekcja 2: Tworzenie aplikacji

- Spring Initializr
- Struktura projektu
- Pierwszy kontroler

Sekcja 3: Działanie Spring Boot
- Proces uruchamiania
- Autokonfiguracja

Sekcja 4: Podstawy Security
- Bezpieczeństwo aplikacji
- Spring Security - architektura

Sekcja 5: Konfiguracja Security
- Podstawowa konfiguracja
- Zabezpieczanie endpointów

Sekcja 6: JWT
- JWT - wprowadzenie
- Przykładowa implementacja

# Sekcja 1: Wprowadzenie

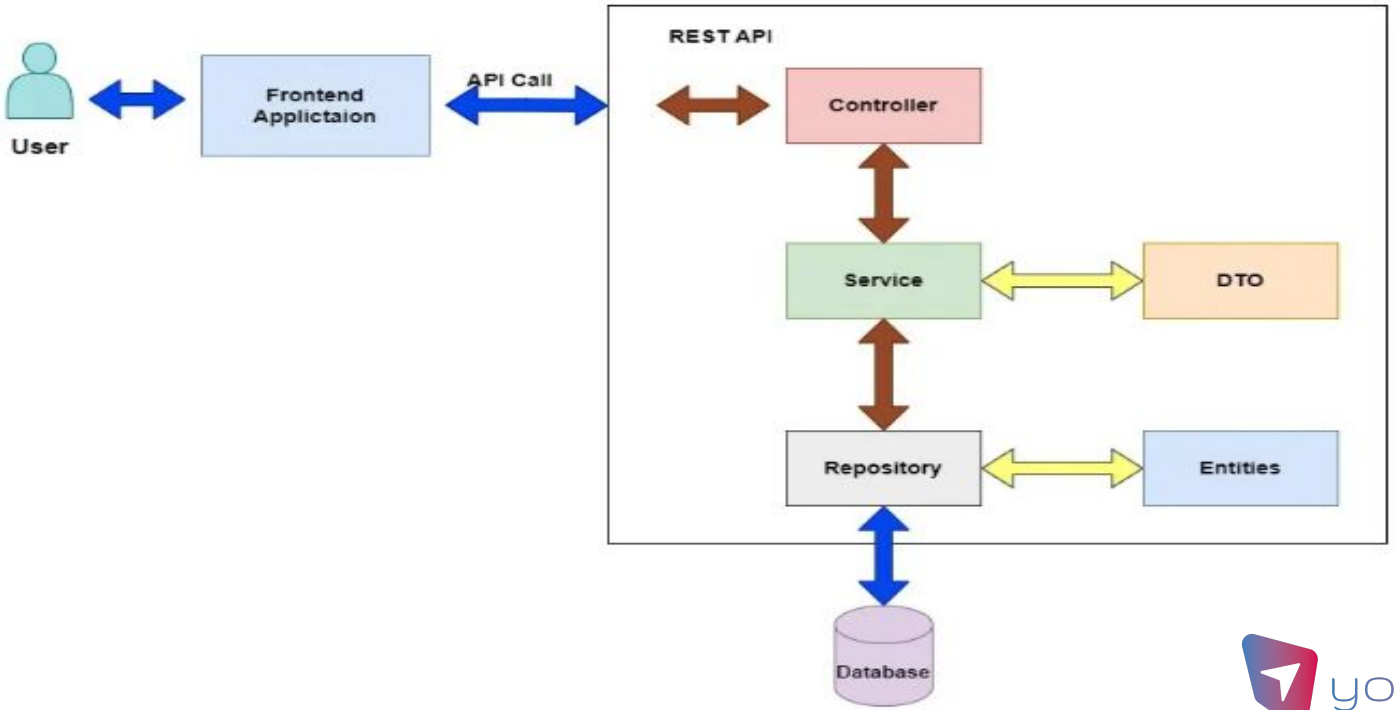
# Czym jest Spring Boot?


you can dev

# Czym jest Spring Boot - podstawy

- Framework rewolucjonizujący sposób tworzenia aplikacji
- Convention over Configuration
- Automatyczna konfiguracja
- Gotowe do użycia komponenty

# Architektura typowej aplikacji w Spring Boot




you can dev

# Uproszczona architektura Spring Boot

Kluczowe elementy:

- Warstwa auto-konfiguracji
- System starterów (Spring Boot starters)
- Wbudowane narzędzia monitorowania stanu aplikacji (Actuators)

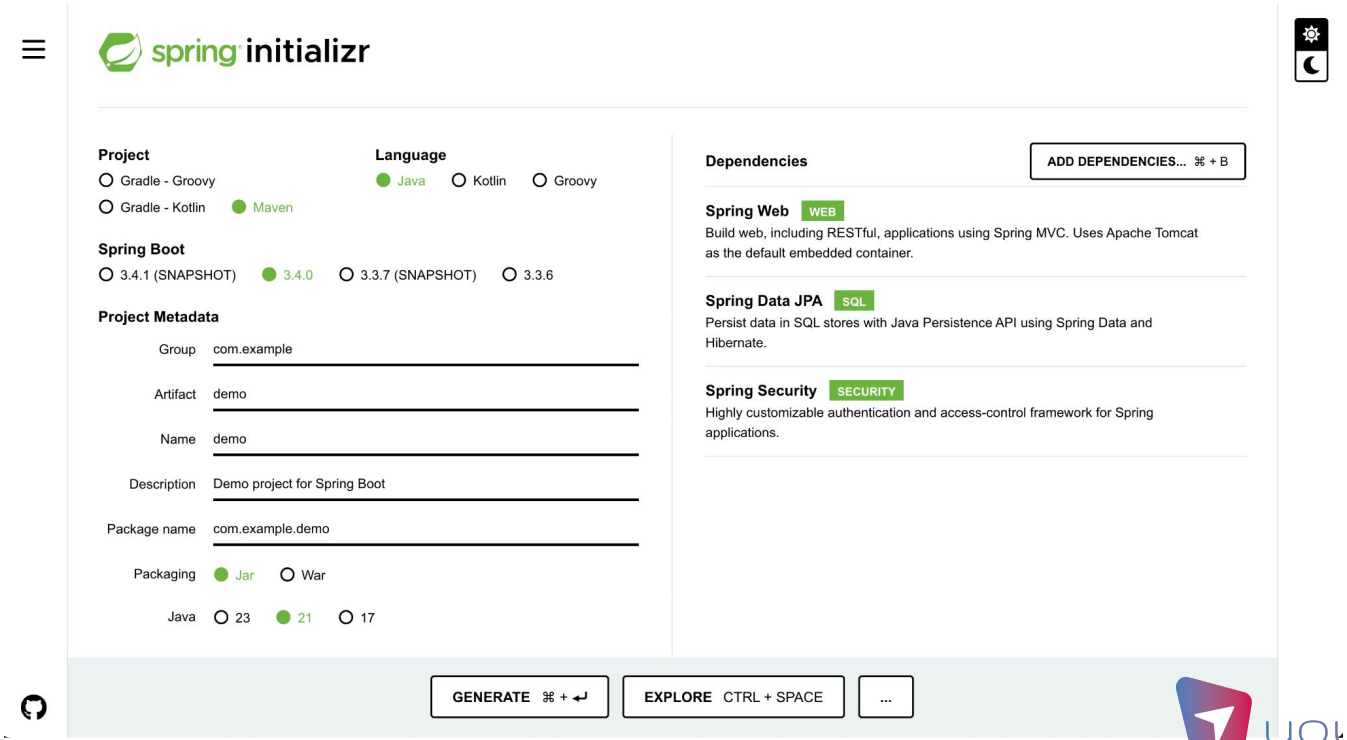
# Sekcja 2: Tworzenie aplikacji

# Spring Initializr

Rozpoczęcie pracy z projektem:

- Narzędzie online (start.spring.io)
- Wybór zależności
- Generowanie struktury projektu

# Spring Initializr

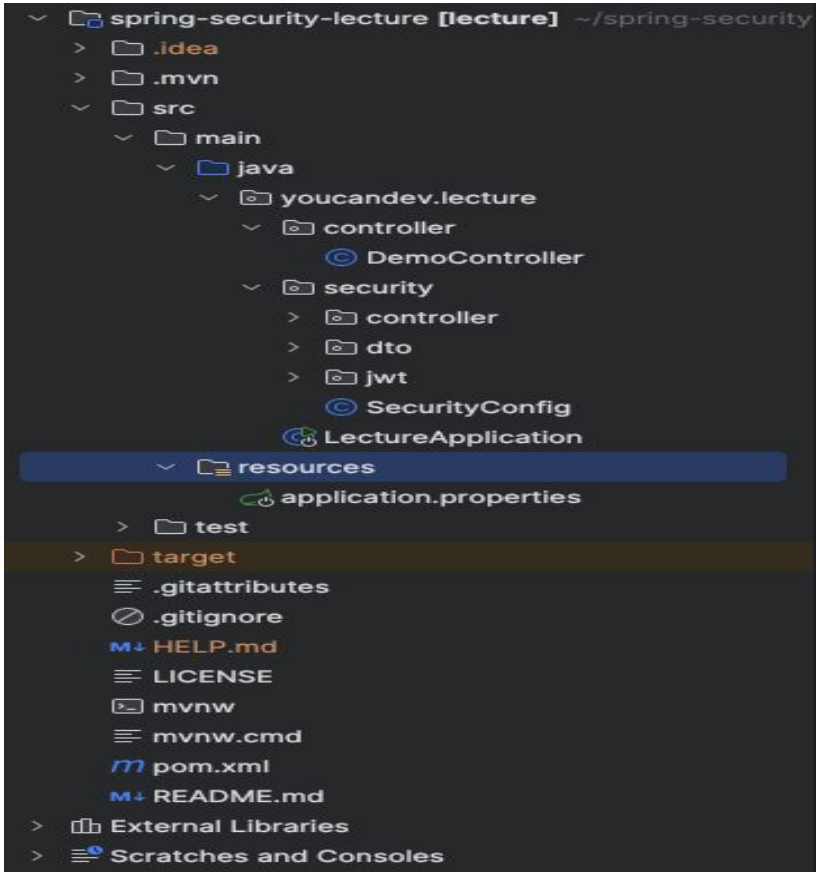



you can dev

# Struktura projektu

- src/main/java - klasy javowe
- klasa główna z @SpringBootApplication oraz metodą main
- katalog resources
- strategie organizacji kodu i pakietów

# Struktura projektu




you can dev

# Pierwszy kontroler

```java
@RestController  youcandev
public class DemoController {

    @GetMapping("/public")  youcandev
    public String publicPage() { return "This is a public page. Accessible by anyone."; }

    @GetMapping("/user")  youcandev
    public String userPage() { return "This is a user page. Accessible by authenticated users."; }

    @GetMapping("/admin")  youcandev
    public String adminPage() { return "This is an admin page. Accessible only by admins."; }
}
```

you can dev

# Sekcja 3: Działanie Spring Boot

# Proces uruchamiania Spring Boot

Kolejność działań:

- Skanowanie classpath
- Wykrywanie @SpringBootApplication
- Budowa ApplicationContext
- Inicjalizacja komponentów
- Uruchomienie serwera aplikacyjnego

# Autokonfiguracja

Mechanizm działania:

- Analiza classpath
- Automatyczna konfiguracja na podstawie zależności
- Możliwość nadpisywania domyślnych ustawień

# Sekcja 4: Podstawy security

# Bezpieczeństwo aplikacji

Główne zagrożenia:

- Nieautoryzowany dostęp
- SQL Injection
- Cross-Site Request Forgery (CSRF)

# Architektura Spring Security

Komponenty:

- Security Filters
- Authentication Manager
- UserDetailsService

# Podstawowa konfiguracja Security

Security Filter Chain:

- Nowoczesne podejście do konfiguracji
- Łańcuch filtrów bezpieczeństwa
- Elastyczność i przejrzystość konfiguracji

# Zabezpieczenie endpointów

Kontrola dostępu:

- Granularna kontrola dostępu
- Różne poziomy dostępu dla różnych ścieżek
- Dynamiczna autoryzacja
- Integracja z zewnętrznymi systemami

# Sekcja 6: JWT

# JWT - Wprowadzenie

Struktura JWT:

- Header (Nagłówek)
- Payload (Ładunek)
- Signature (Podpis)

Algorithm HS256

## Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI
6IkpXVCJ9.eyJzdWIiOiIxMjM0N
TY3ODkwIiwibmFtZSI6IkpvaG4g
RG9lIiwiaWF0IjoxNTE2MjM5MDI
yfQ.SflKxwRJSMeKKF2QT4fwpMe
Jf36POk6yJV_adQssw5c
```

## Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) secret base64 encoded
```

# Implementacja JWT

Komponenty:

- JwtAuthenticationFilter
- SecurityConfig
- Mechanizmy generowania i walidacji tokenów

# Dobre praktyki JWT

Zalecenia:

- Używanie silnych secretów https://randomkeygen.com/
- Odpowiedni czas wygasania tokenów
- Implementacja mechanizmu odświeżania tokenów
- Bezpieczne przechowywanie po stronie klienta

# Slajdy oraz kod źródłowy

you can dev

scouDev

# Dzięki za uwagę!

you can dev

www.youcandev-mentoring.com

scouDev

www.scoudev.pl