Sesje i ciasteczka w Node.js + Express — ZADANIA PRAKTYCZNE

INF.04.7.1(19), INF.04.7.3(2)

Zadanie 1 — Proste ciasteczko

Cel: Utworzenie ciasteczka `language` i jego odczytanie.

Wymagania:

- Endpoint `GET /set-cookie?lang=pl|en` — ustawia ciasteczko `language` z czasem życia 1h.
- Endpoint `GET /get-cookie` — zwraca aktualnie ustawiony język (lub `brak`, jeśli nie istnieje).

Podpowiedź: Użyj pakietu `cookie-parser` i metody `res.cookie()`.

Test (cURL):

```
curl -i "http://localhost:3000/set-cookie?lang=pl" -c cookies.txt
curl -i "http://localhost:3000/get-cookie" -b cookies.txt
```

Zadanie 2 — Sesja użytkownika

Cel: Implementacja prostej sesji użytkownika.

Wymagania:

- Endpoint `POST /login` — przyjmuje JSON `{ "login": "admin", "password": "zaq1@WSX" }`.
- Po poprawnym logowaniu zapisuje w sesji `req.session.user = { login: "admin" }`.
- Endpoint `GET /me` — zwraca dane zalogowanego użytkownika (lub `401`, jeśli brak sesji).
- Endpoint `POST /logout` — niszczy sesję.

Podpowiedź: Użyj pakietu `express-session`.

Test (cURL):

```
curl -i -X POST "http://localhost:3000/login" \
  -H "Content-Type: application/json" \
  -d '{"login":"admin","password":"zaq1@WSX"}' -c cookies.txt

curl -i "http://localhost:3000/me" -b cookies.txt
curl -i -X POST "http://localhost:3000/logout" -b cookies.txt
```

Zadanie 3 — Koszyk w sesji

Cel: Przechowywanie koszyka w sesji.

Wymagania:

- Endpoint `POST /cart/add` — dodaje produkt do koszyka `{ "productId": 101, "qty": 2 }`.
- Endpoint `GET /cart` — zwraca aktualny koszyk i sumę ilości.
- Endpoint `POST /cart/remove` — usuwa produkt po `productId`.
- Endpoint `POST /cart/clear` — czyści koszyk.

Test (cURL):

```
curl -i -X POST "http://localhost:3000/cart/add" \
  -H "Content-Type: application/json" \
  -d '{"productId":101,"qty":2}' -b cookies.txt

curl -i "http://localhost:3000/cart" -b cookies.txt
curl -i -X POST "http://localhost:3000/cart/remove" \
  -H "Content-Type: application/json" \
  -d '{"productId":101}' -b cookies.txt

curl -i -X POST "http://localhost:3000/cart/clear" -b cookies.txt
```

Zadanie 4 — Middleware sprawdzający logowanie

Cel: Ochrona wybranych tras przed dostępem niezalogowanych użytkowników.

Wymagania:

- Utwórz middleware `requireLogin`, który sprawdza `req.session.user`.
- Jeśli brak sesji → `401 Unauthorized`.
- Jeśli sesja istnieje → przepuszcza do dalszego kodu.

Test:

- Zaloguj się przez `/login`.
- Spróbuj wejść na `/secret` chronione przez `requireLogin`.

Zadanie 5 — Panel administratora (role)

Cel: Rozszerzenie sesji o role użytkownika.

Wymagania:

- Użytkownik `admin` ma rolę `"admin"`.
- Dodaj middleware `requireAdmin`, który sprawdza rolę.
- Utwórz trasę `/admin/dashboard`, która dostępna jest tylko dla admina.

Test:

- Zaloguj się jako `admin` → wejście na `/admin/dashboard` zwraca `"Panel admina"`.
- Zaloguj się jako inny użytkownik → wejście na `/admin/dashboard` zwraca `403 Forbidden`.

Zadanie 6 — Atrybuty bezpieczeństwa

Cel: Wzmocnienie ochrony sesji i ciasteczek.

Wymagania:

- Ustaw ciasteczka sesji z atrybutami: `HttpOnly`, `SameSite=Lax`, `Secure` (dla HTTPS).
- Dodaj limit czasu życia sesji `maxAge: 10 * 60 * 1000` (10 minut).

Zadanie 7 — CSRF (Cross-Site Request Forgery)

Cel: Zabezpieczenie wybranych endpointów przed atakiem CSRF.

Wymagania:

- Generuj token CSRF i zapisuj go w sesji.
- Endpoint `GET /csrf` → zwraca token.
- Endpointy mutujące (np. `/cart/add`) wymagają nagłówka `X-CSRF-Token`.
- Brak/niepoprawny token → `403 Forbidden`.

Kryteria oceny (30 pkt)

- Obsługa ciasteczek (`set-cookie`, `get-cookie`) — **5 pkt**
- Logowanie, sesje i wylogowanie — **7 pkt**
- Koszyk w sesji (pełna obsługa CRUD) — **8 pkt**
- Middleware (requireLogin, requireAdmin) — **4 pkt**
- Atrybuty bezpieczeństwa ciasteczek i sesji — **3 pkt**
- CSRF (generacja + weryfikacja tokenu) — **3 pkt**

Podsumowanie

Dzięki tym zadaniom przećwiczysz:

- podstawową obsługę **ciasteczek** w Express,
- zarządzanie **sesjami** użytkowników,
- implementację koszyka zakupowego,
- tworzenie **middleware** do kontroli dostępu,
- zabezpieczenia: flagi ciasteczek i **CSRF**.