

Wybrane elementy praktyki projektowania oprogramowania

Zestaw 7

node.js - framework Express

2025-12-02

Liczba punktów do zdobycia: **7/67**

Zestaw ważny do: 2025-12-16

1. **(1p)** Pokazać działanie formantu `<input type="file" ... />` umożliwiającego wysłanie za pomocą przeglądarki pliku z dysku lokalnego na serwer. Uwaga! Standardowo middleware **body parser** nie obsługuje możliwości przesłania pliku w parametrach POST. Taką możliwość mają bardziej specjalizowane middleware np. **multer**
<https://www.npmjs.com/package/multer>.
Proszę zwrócić uwagę na to jakie ustawienie atrybutu **enctype** taga **form** jest wymagane, żeby poprawnie przesyłać na serwer pliki.
2. **(1p)** Pokazać jak korzystać ze stron layoutowych (**express-ejs-layouts**).
3. **(1p)** Pokazać jak przekazywać parametry do widoków załączanych (**include**) do innych widoków: na podstawie przykładu szablonu listy rozwijalnej (**select-option**) przedstawionego na wykładzie, pokazać szablon dla listy wyboru typu **radio** lub listy wyboru typu **checkbox**.
4. **(1p)** Pokazać jak z przeglądarki zwracać dynamicznie utworzony na serwerze strumień danych, który przeglądarka zinterpretuje jako plik. Formalnie - pokazać jak korzystać z nagłówka **Content-Disposition** z wartością **attachment**, ustawionego w strumieniu odpowiedzi za pomocą **setHeader**.
Efektem wysłania strumienia oznakowanego jako attachment do przeglądarki powinno być w przeglądarce, zamiast wydenderowania strony, standardowe okno Otwórz-Zapisz-Anuluj.
5. **(1p)** Nauczyć się dodawać, odczytywać i usuwać ciasteczka w kodzie po stronie serwera. Jak sprawdzić czy przeglądarka obsługuje ciastka? Czy to jest w ogóle możliwe?
6. **(1p)** Nauczyć się dodawać, odczytywać i usuwać wartości w kontenerze sesji po stronie serwera. Przeglądać listę dostępnych implementacji zasobnika sesji po stronie serwera (<https://github.com/expressjs/session>), wybrać i zademonstrować jedną implementację inną niż domyślna w pamięci (podpowiedź: niektóre z przedstawionych są bardzo łatwe do użycia, np. **session-file-store**).
7. **(1p)** Zapoznać się z dokumentacją podatności aplikacji internetowych publikowanych przez OWASP (OWASP Top 10).

Aktualna wersja: https://owasp.org/Top10/A00_2021_Introduction/

Które z wymienionych zagrożeń dotyczą nawet tak prostych aplikacji jak te które budujemy?

Na spreparowanej aplikacji zademonstrować w praktyce podatność **Web Parameter Tampering**.

Opis: https://owasp.org/www-community/attacks/Web_Parameter_Tampering

Nauczyć się technik przeciwdziałania temu zagrożeniu.

Wiktor Zychla