

Wybrane elementy praktyki projektowania oprogramowania

Zestaw 6

node.js - framework Express

2020-12-01

Liczba punktów do zdobycia: **5/50**

Zestaw ważny do: 2020-12-15

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 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**.
3. (**1p**) Nauczyć się dodawać, odczytywać i usuwać ciasteczka w kodzie po stronie serwera. Jak sprawdzić czy przeglądarka obsługuje ciastka? 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**).
4. (**2**) Zapoznać się z dokumentacją podatności aplikacji internetowych publikowanych przez OWASP (OWASP Top 10 2017). Które z wymienionych zagrożeń dotyczą nawet tak prostych aplikacji jak te które budujemy? Na spreparowanej aplikacji zademonstrować w praktyce następujące podatności: **Web Parameter Tampering** oraz **Cross-site Request Forgery**. Nauczyć się technik przeciwdziałania tym zagrożeniom.

Wskazówka: zagrożeniu CSRF można przeciwdziałać za pomocą dedykowanego middleware, np. **csurf**. Należy więc nauczyć się go używać i objaśnić jego działanie.

W przypadku zagrożenia Web Parameter Tampering istnieją co najmniej dwa dobre sposoby przeciwdziałania - szyfrowanie/podpisywanie query string i/lub dodatkowa walidacja po stronie serwera. Opowiedzieć o obu tych możliwościach, a jedną z nich zademonstrować w praktyce.

Wiktor Zychla