

Wybrane elementy praktyki projektowania oprogramowania

Zestaw 5

node.js - framework Express

28-11-2016

Liczba punktów do zdobycia: **10/50**

Zestaw ważny do: 13-12-2016

1. (**1p**) Nauczyć się korzystać z debuggera warstwy HTTP, Burpa (<https://portswigger.net/burp/>).
Pokazać jak przechwytywać ruch z przeglądarki do internetu - jak podglądać zawartości żądań i odpowiedzi i jak stawiać pułapki i za ich pomocą **modyfikować** żądania i/lub odpowiedzi (zakładka Proxy/Options, Proxy/Intercept).
Pokazać jak za pomocą Burpa symulować żądania typu GET i żądania typu POST do własnej strony (czyli jak wyklikać w Burp żądanie, które do naszej aplikacji wyśle Burp, a nie przeglądarka, zakładka Repeater!).
2. (**1p**) Pokazać jak przekazywać parametry do widoków wywoływanych z poziomu kodu oraz do widoków włączanych (**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 ciastka w kodzie po stronie serwera. Jak sprawdzić czy przeglądarka obsługuje ciastka?
4. (**2p**) Nauczyć się dodawać, odczytywać i usuwać wartości w kontenerze sesji po stronie serwera. Przejrzeć 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**).
5. (**1p**) Ucząc się języka HTML przyzwyczajamy się do tego, że dla żądań typu GET z przeglądarki przeznaczony jest odsyłacz (link)

```
<a href="strona.aspx?p1=v1&p2=v2">żądanie GET</a>
```

zaś dla żądań typu POST - przycisk

```
<form method="post">  
  <input type="text" name="p1" />  
  <input type="text" name="p2" />  
  <input type="submit" value="żądanie POST" />  
</form>
```

Zadanie polega na tym, żeby pokazać jak to zrobić odwrotnie - to odsyłacz (link) powinien spowodować żądanie typu POST a przycisk żądanie typu GET.

6. (2p) Przygotować aplikację node.js Express, która pozwala wprowadzić i wydrukować standardowy "pasek zgłoszenia zadań".

Aplikacja ma składać się z dwóch widoków: formularza zgłoszenia (widok domyślny `\`) i widoku wydruku (`\print`).

Na formularzu zgłoszenia użytkownik aplikacji powinien mieć możliwość wpisania imienia i nazwiska, daty, nazwy zajęć i numeru zestawu oraz kompletu wyników kolejnych deklarowanych 10 zadań z odpowiednią liczbą punktów.

Program powinien kontrolować poprawność wpisywanych danych - w przypadku braku kluczowych wartości lub niepoprawnego formatu, formularz jest zwracany do przeglądarki do uzupełnienia.

Po zaakceptowaniu formularza zgłoszenia, użytkownik powinien w przeglądarce zobaczyć formularz wydruku: pasek zgłoszenia w postaci możliwej do wydrukowania.

Użyć dowolnej, wybranej przez siebie metody przekazywania danych między widokami (ciastka, sesja na serwerze, parametry GET)

7. (2p) Wykorzystać obiekt sesji po stronie serwera do następującego **naiwnego** rozwiązania problemu autentykacji: w każdej stronie aplikacji w wypadku stwierdzenia w kodzie po stronie serwera braku informacji o użytkowniku w sesji, następuje przekierowywanie (`redirect`) do strony `\login`, na której po poprawnym potwierdzeniu tożsamości (login i hasło) informacja o tożsamości zapamiętywana jest w kontenerze sesji.

Dodatkowo punkt wejścia do aplikacji powinien być poprawnie przywrócony po autentykacji (użytkownik kieruje żądanie do którejkolwiek strony aplikacji, jest przekierowany do strony logowania, a następnie aplikacja sama powraca do strony, od której użytkownik chciał rozpocząć nawigację).

Wiktor Zychla