



Java EE 6. Programowanie aplikacji WWW

Krzysztof Rychlicki-Kicior

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki:

"Java EE 6. Programowanie aplikacji WWW"

Jest to lista błędów znalezionych po opublikowaniu książki, zgłoszonych i zaakceptowanych przez naszą redakcję. Pragniemy, aby nasze publikacje były wiarygodne i spełniały Twoje oczekiwania. Zapoznaj się z poniższą listą. Jeśli masz dodatkowe zastrzeżenia, możesz je zgłosić pod adresem

<https://ebookpoint.pl/user/erraty>

| Strona | Linia | Jest | Powinno |
|--------|---------------------|---|--|
| 31 | | <pre>Cookie lastVisit = null; for (Cookie c: request.getCookies()) if (c.getName().equals("obecność")){ lastVisit=c; break; }</pre> | <pre>Cookie lastVisit = null; if(request.getCookies()!=null) for (Cookie c: request.getCookies()) if (c.getName().equals("obecność")){ lastVisit=c; break; }</pre> |
| 35 | 12 | Parametry serwletu są określane w podobny sposób jak w przypadku serwletów: | Parametry kontekstu są określane w podobny sposób jak w przypadku serwletów: |
| 49 | Listing 4.2 linia 7 | daneDoZahaszowania | daneDoHaszowania |
| 50 | 5-8 | <pre><p>Wynik haszowania to: <%= request.getAttribute("md5hash") %></p></pre> | <pre><p>Wynik haszowania to: \${requestScope.md5hash}</p></pre> |

| | | | |
|-----|------------------------------|---|--|
| 139 | 18 | ..ustawień serwera MySQL zapisanych w dodatku A, ... | W dalszych przykładach będę korzystał z domyślnych ustawień serwera MySQL, a także z bazy danych o nazwie javaee. |
| 144 | 7 | Obie encje | Obie adnotacje |
| 151 | 6 | dodaj.przychodnie.xhtml | dodajprzychodnie.xhtml |
| 152 | 11 | | import javax.faces.event.ActionEvent; |
| 152 | kod źródłowy | lekarz | przychodnia |
| 152 | Listing 8.10 wiersz 21,22,26 | lekarz.setId(null); em.persist(lekarz); em.getTransaction().commit(); em.close(); this.lekarz = new Lekarz(); | przychodnia.setId(null); em.persist(przychodnia); em.getTransaction().commit(); em.close(); this.przychodnia= new Przychodnia(); |
| 171 | ostatni | ..., a dokładnie generycznego typu listy. | ..., a dokładnie generycznego typu zbioru. |
| 181 | 18/19 | lekarz i pacjent mogą istnieć bez bazy... | lekarz i pacjent mogą istnieć bez wizyty... |

Poniżej znajduje się lista błędów znalezionych przez czytelników, ale jeszcze nie potwierdzonych przez Redakcję:

| Strona | Linia | Jest | Powinno |
|--------|-----------------------|---|---|
| 49 | 7 wiersz listingu 4.2 | daneDoZahaszowania | daneDoHaszowania |
| 116 | 7 | <h:commandButton label="Oblicz" ... | <h:commandButton value="Oblicz" ... |
| 151 | 6 | dodaj.przychodnie.xhtml | dodajprzychodnie.xhtml |

| | | | |
|-----|------------------------|---|--|
| 172 | 2gi pełny akapit | zastosuj wartość FetchType.EAGER | zastosuj wartość FetchType.LAZY |
| 172 | 2gi pełny akapit | zastosuj wartość FetchType.EAGER | zastosuj wartość FetchType.LAZY |
| 175 | 8-9 | Jak zawsze, rekordy potomne będziemy pobierać dopiero w momencie odwołania się do listy | Jak zawsze, rekordy potomne będziemy pobierać od razu, przy utworzeniu obiektu klasy Lekarz, a nie w momencie odwołania się do zbioru (nie listy) |
| 175 | 9 od dołu | lekarz i pacjenci stanowią automatyczne jednostki | lekarz i pacjenci stanowią autonomiczne jednostki |
| 199 | 21 | <managed-bean- scope>session</managed- bean-scope> | <managed-bean- scope>request</managed- bean-scope> |