Vorbereitungsauftrag – Single Page Applications mit React

|  |  |
| --- | --- |
| Datum | 23.02.2024 |
| NDK | Web Engineer NDK HF |
| Modul | Web-Technologien Aufbau |
| Thema | React |
| Plattform | Selbststudium |

**Buch Pro MERN Stack / Kapitel 4: Event Handling**

|  |  |
| --- | --- |
| **Aufgabe** | **Lösung** |
| 1. Refresh your browser; you'll see that the added issues are gone. How does one persist the change? | Die issues können entweder im local storage im Browser oder im Server gespeichert werden. |
| 1. Remove *e.preventDefault()*. Click the Add button with some values for owner and title. What happens? What do you see in the URL bar? Can you explain this? | Die Seite wird aufgrund einer request neu geladen. In der URL sind die query-Parameter für owner und title ersichtlich:    Die default-Aktion eines Formulars ist eine get-http-request mit den Werten des Formulars. |
| 1. Use the developer console to inspect the table and add a breakpoint on the *<tbody>* element as "break on subtree modification". Now, add a new issue. How many times is the subtree being modified? Compare this with exercise #2 in "Updating State," where you traced the number of *render()* calls in a *IssueRow*. | Der subtree wird nur einmal angepasst. Es wurde lediglich ein child hinzugefügt und keines der bestehenden angepasst.  Verglichen mit der Anzahl *render()* method calls, wurde nur die neue Zeile zur DOM hinzugefügt, obwohl *render()* für jede Zeile aufgerufen worden ist. |