



Wiederholungstäter

Er hatte es wieder getan. Er konnte einfach nicht anders. Er musste es lernen, dieses Neue, dieses Neuland, dieses React.

Als C#-Entwickler genoss er den Umgang mit Visual Studio, dem Framework und seiner Liebessprache. Doch es fehlte ihm der Kick. Er musste mal wieder raus aus der Komfortzone und endlich wieder zeigen, was für ein toller Hecht er war. Vorspulen bis zum Ende der Geschichte: Hering wäre der passendere Ausdruck. Wieder zurückspulen. Beginnen wir mit

`npx create-react-app my-App --template typescript`

Mit diesem Befehl an der Kommandozeile legte er problemlos und innerhalb einer Minute eine nigelnagelneue React-App an. Aber halt, war das wirklich der Anfang? Noch weiter zurückspulen.

Nachdem er nach der Lektüre diverser Webseiten endlich den Fehler gefunden hatte, der dafür sorgte, dass NPM immer in der falschen Version startete, er die alte Node-Version runtergeworfen und die aktuelle installiert hatte, konnte er resümieren: „Hello World“ innerhalb von nur wenigen Stunden. Wie ermutigend.

Dann traten JavaScript respektive TypeScript auf den Plan. Während unter WPF eine Komponente aus ein bisschen XAML und einer Portion Code besteht, ist das unter React komplett genauso – nur eben absolut anders.

Da müssen Dinge exportiert werden, wie man das früher in der allseits beliebten Sprache C gemacht hat, und das Rendern der Komponente muss in einer Klasse über die Funktion *Render* definiert werden – oder Sie lassen diese doofe Klasse einfach weg und schreiben die Komponente als Funktion. Oder als Konstante. Geht alles. Sie haben die Qual.

Viele Wege, die zum Ziel führen, sorgen für Freiheit – jedoch auch für Zweifel, Entscheidungsprobleme und Fragen. Der Neuling kommt bei solchen Alternativen komplett ins Straucheln, besonders, wenn jedes Tutorial es anders macht als alle anderen.

Lange Geschichte, kurzes Fazit: Im Prinzip kam er am Ende des Tages tatsächlich zu einem Produkt, das sich am ehesten als aufgemotztes „Hello World“ bezeichnen ließe. Im Ausgleich zum geringen Fortschritt hatte die Entwicklung ein Vielfaches an Zeit in Anspruch genommen – ja, hätte er es nur mit C#, WPF und .NET Core und mit Blazor gebaut.

„Siehste“, sagte er sich, „Web ist halt doch Mist. Unter .NET hat es so eine Bremse noch nie gegeben. Hier war die Produktivität immer hoch und Projekte wurden immer in der veranschlagten Zeit ausgeführt.“ Doch dann fiel ihm das Jahr 2002 und der Übergang vom klassischen Visual Basic 6 hin zum .NET Framework und zu C# ein, und mit glasigem Blick öffnete er sich das dritte Bier.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro



Bernhard Pichler

weiß, wie Sie zu richtigen Entscheidungen kommen (S. 14)



Tobias Richling

verwendet Terraform im Team und legt dafür den State in Azure ab (S. 48)



Patrick A. Lorenz

verkauft seine Softwareprodukte automatisiert und ist sich sicher: Sie können das auch (S. 118)