Easily Transition from Desktop to Web Development with Syncfusion Blazor Controls

The fact that Syncfusion already had production-ready Blazor controls reaffirmed my decision.

Uwe Hein
Freelance Developer, Pedav GmbH

BENEFITS

- Exceptional Blazor controls
- Easy to integrate
- Trusted support team

From Music to Developing

Uwe was a professional guitarist for over a decade before jumping into the world of desktop app development. When he settled down and started a family, he started working at a music school where he eventually reached a management position. This level-up involved Uwe having to handwrite 150 recital invites, which became a much shorter process once he started learning programming.

Fast forward 20 years, and Uwe is still at the same music school, now one of the biggest in Germany, and he’s using Syncfusion’s tools to keep activities running smoothly.

Uwe’s most recent project was to build a room-management app that allowed students, teachers, and administrators to book available space in the school—practice rooms, classrooms, and recital halls. The most important requirement for the app was that it provide an interactive scheduler control, one that clearly defined time slots, appointments, and resource assignments.

Taking to the web with Blazor and Syncfusion

Uwe planned to create it as a web app so it’d be accessible across mobile and desktop devices. The challenge was that he came from a WinForms background, so he wanted to avoid having to enter the vast territory of learning JavaScript. He found a way to sidestep learning JavaScript when he came across Blazor controls by Syncfusion.
“It allowed me to write the app using C# and deploy it without having to write any JavaScript,” said Uwe. “The fact that Syncfusion already had production-ready Blazor controls reaffirmed my decision.” With the ASP.NET Core framework as his development platform and Blazor for his interface, Uwe said he was able to save two to three years on this project.

In addition to the Scheduler control, Uwe used Syncfusion’s CheckedListBox and Calendar controls. He recounted the simplicity of integrating the controls and how they provided feature-rich UIs that elevated the project to the next level: “They helped me create an app that looked professional and mature, as if it had been developed and polished for a very long time.”

**Syncfusion support guides the way**

When problems popped up during development, Uwe relied on the Syncfusion support team and the community forum to help guide him. Sometimes he wasn’t quite sure how to handle certain parts of the web development that were new to him. He was delighted not only by the quick response time from support, but also the samples they provided from working applications.

Uwe admits that when he created the app, Blazor was a unique challenge as Microsoft was still working on it, releasing a series of preview builds in .NET Core 3 updates. The updates included changes to ASP.NET Core and Blazor. But he says through it all, the Syncfusion team worked to ensure all their controls supported the latest advances, often turning around working solutions within days. When Uwe found a bug or an issue, he usually received a response within one day, and fixes were delivered quickly and consistently.

**Looking ahead**

As Blazor continues to develop, its importance for web developers will continue to grow not only with .NET Core 3, but also as Microsoft works on .NET 5 in the coming year. Uwe expects Syncfusion will remain at the forefront of enterprise-ready Blazor controls for a long time and won’t hesitate to use their tools again in his next web project.