



Blazor WebAssembly

Verktygspresentation



C# i webbläsaren

Marketgrowth
Written by Albin Stenhoff
<https://github.com/OtrevligAbbe/MarketGrowth>

20
25

VAD ÄR BLAZOR

Blazor WebAssembly är ett modernt ramverk från Microsoft som låter utvecklare bygga interaktiva webbgränssnitt (Single Page Applications) med C# och .NET istället för JavaScript. Detta innebär att utvecklare kan använda samma språk på både servern och klienten.

Tekniken bygger på WebAssembly (WASM), en öppen webbstandard som gör det möjligt att köra binär kod direkt i webbläsaren med nästan samma prestanda som inbyggda applikationer (Native apps). Inga plugins eller tillägg krävs.

Fördelen är enorm för .NET utvecklare Du får tillgång till hela .NET ekosystemet, stark typning och verktyg som Visual Studio direkt i frontend utvecklingen. Det eliminerar behovet av att byta kontext mellan olika programmeringsspråk.

VARFÖR VÄLJA BLAZOR?

Enhetlig Kodbas (Fullstack C#) Den största styrkan är möjligheten att använda C# och .NET över hela stacken. Istället för att växla mellan JavaScript på klienten och C# på servern, kan utvecklare dela kod, datamodeller och valideringslogik sömlöst. Detta minskar risken för buggar och ökar utvecklingstakten markant.

Kraftfullt Ekosystem Blazor drar nytta av Microsofts mogna .NET ekosystem. Med tillgång till verktyg som Visual Studio, stark typning och miljontals färdiga NuGet paket får utvecklare en robust och produktiv arbetsmiljö direkt från start, något som ofta kräver komplex konfiguration i JavaScript världen.

Komponentbaserad Arkitektur Precis som React eller Vue bygger Blazor på återanvändbara komponenter ("Razor Components"). Detta gör det enkelt att kapsla in både utseende (HTML/CSS) och logik (C#) i små byggstenar, vilket leder till skalbara och underhållbara applikationer för stora företagssystem.

FÖRDELAR & NYTTA

PRODUKTIVITET 🚀	Genom att använda C# över hela stacken kan utvecklare återanvända kod, modeller och logik. Detta minskar dubbelarbete och snabbar upp utvecklingsprocessen markant.
PRESTANDA ⚡	Tack vare WebAssembly körs koden direkt i webbläsaren med nära "native" hastighet. Detta avlastar servern och ger en blixtsnabb upplevelse för användaren.
SÄKERHET & TYPNING 🛡️	C# är ett starkt typat språk, vilket fångar många fel redan vid kompilering. Detta leder till stabilare kod och färre oväntade buggar jämfört med JavaScript.
EKOSYSTEM 📦	Direkt tillgång till hela .NET ekosystemet och NuGet paket. Utvecklare kan använda branschstandardiserade bibliotek och verktyg som de redan behärskar.



ROLLER & SCENARIO

Det här är Lisa, helt vanlig person.

Nu ska vi demonstrera Kod till Användare.

Utvecklaren (JAG)

Genom att välja Blazor kunde jag stanna i .NET miljön. Jag slapp skriva dubbel kod för validering och kunde använda samma C# modeller på både server och klient. Det sparade enormt med tid

Lisa med mobil

För Lisa känns MarketGrowth inte som en vanlig hemsida, utan som en installerad app. Tack vare Blazor WebAssembly sker all interaktion blixtsnabbt direkt i hennes webbläsare, utan onödiga omladdningar.

Lisa på tåget

När Lisa åker tåg och nätet svajar, fortsätter appen fungera. Eftersom Blazor laddat ner logiken till klienten (PWA) kan hon analysera sina grafer offline. Servern belastas minimalt och Lisa kan sova direkt efter det utan att vara frustrerad över att det inte fungerar.

C# i webbläsaren

SAMMANFATTNING

Slutsats Blazor förändrar spelplanen för hur vi bygger webbapplikationer. Det ger oss möjligheten att skapa snabbare, säkrare och mer robusta lösningar för användaren, samtidigt som vi förenklar arbetet "under huven". Det innebär en modernare upplevelse utan onödig komplexitet.

Lärdomar från MarketGrowth Arbetet med MarketGrowth har bevisat teknikens värde i praktiken. Genom att kunna återanvända samma "byggstenar" i hela systemet har vi sparat värdefull tid och minskat risken för fel. För organisationer som redan använder Microsofts teknik är detta det mest effektiva och framtidssäkra valet.

Namn: Albin Stenhoff

Roll: Lead Developer / Grundare

Länk: github.com/OtrevligAbbe/MarketGrowth