

I denna laborationsuppgift göra följande:

Då Blazor WebAssembly inte har stöd för ett eget backend ska du i denna laboration skapa ett RESTFUL API och använda det som backend. Det går att göra REST API med både ASP.NET CORE WEB API eller med Azure Functions i Azure. Laborationerna framöver kommer använda sig av Azure Functions som lösning, men i denna övning ska du göra sakerna med både och.

CRUD with a .NET 6 Web API & Entity Framework Core

https://www.youtube.com/watch?v=Fbf_ua2t6v4

Intro to Azure Functions - What they are and how to create and deploy them

<https://www.youtube.com/watch?v=zlfkub7CLY>

Läs mer om hur du kan använda olika triggers och binding i Azure Function

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings?tabs=csharp>

Läs mer om hur du kan köra Javascriptsfunktioner i Microsoft Blazor

<https://www.c-sharpcorner.com/article/blazor-wasm-calling-javascript-from-c-sharp-vice-versa/>

Del 1 – Spara temperaturdata

I föregående laboration fick du reda på vilken data som ska visas för en enhet. Denna data vill vi kunna lagra någonstans och sedan hämta upp. Du ska därför bygga en lagringslösning som just sparar in dem värdena som du har designat upp för tidigare. I denna laboration ska du spara dessa värden och hämta värdena med både en Azure Functions och en ASP.NET Core Web Api. Men som tidigare nämnt så kommer vi bara använda oss av Azure Functions framöver sen.

Del 2 – Varningsnotiser

Ibland kan det hända att en sensor går utanför sina rekommenderade värden. Såsom att temperatursensor klarar av en temperatur på mellan 0–50°C. När sensorn börjar närma sig dessa min- och maxvärden kommer den skicka ett varningsmeddelande. Detta varningsmeddelande vill vi kunna logga på något sätt. Du ska därför i denna laboration se till att spara ner sådan typ av information till en databas av något slag.