

Mål: Få en förståelse för hur en webbserver är uppbyggd, hur HTTP-protokollet fungerar och hur vi kan designa en modulär applikation. Hur hanterar vi förfrågningar till en webb tjänst och hur ger vi ett svar i json format.

Deadline: 17/2 23.55

Redovisningen: Laddas upp innan deadline på ithsdistans.se som en länk till projektgruppens github repo. Alla i projektgruppen skickar in samma länk men till länken bifogar ni en egen text som beskriver vad ni bidragit med i projektet och då speciellt med tanke på g och vg kraven.

Betygskriterier:

- Använda web services och ramverk för webbtjänster
- Anropa ett API med SOAP & REST på ett effektivt sätt
- Använda en responsiv design med html, css och javascript

Litteratur: Föreläsningsmaterial vecka 7 och 8.

<https://medium.com/@ssaurel/create-a-simple-http-web-server-in-java-3fc12b29d5fd>
<https://www.baeldung.com/java-spi>

Uppgift:

G-nivå

Arbeta i grupp

Projektet ska versionshanteras i git.

Projektet ska skrivas i Java eller Kotlin

Bygg en enkel HTTP-server som använder version 1.1 av protokollet.

Servern ska kunna hantera enklare GET HEAD och POST förfrågningar inklusive parametrar.

Kunna serva statiska filer från disk av formaten html, css, js, pdf samt ett eller flera bildformat med rätt content-type i svaret.

Det ska finnas minst en url som ger oss möjlighet att skicka in information till webbservern via parametrar. Den inskickade informationen ska då skickas tillbaka som json data.

Om en adress efterfrågas som inte finns ska en 404 sida returneras.

Skapa en eller flera statiska webbsidor som demonstrerar webbserverns möjligheter inklusive access till json informationen.

VG:

Bygga ut webbservern till att bli en applikations server.

Webb servern ska kunna hantera pluginklasser skrivna i Java som kan generera kod dynamiskt baserat på inkommande parameterar. Vilken adress som ska användas för att köra en viss pluginklass kan konfigureras via en inställningsfil för webbservern där även andra parametrar som root-katalog, serverport m.m. med fördel anges eller genom att pluginet tillhandahåller den informationen.

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ServiceLoader.html>

Pluginklasser ska ha ett definierat interface att extenda och använda sig av dependency injection för att få objekt att hämta inkommande information via och skicka utgående information till. Webb servern behöver alltså tillhandahålla någon form av Request och Response objekt.

Skapa en eller flera plugin som man kan begära information ifrån. Responsen ska bestå av dynamiskt skapade html-sidor eller json som svar.

Får gärna finnas en modul som kan visa statistik för webbläsaren. Tex vilka klienter som gjort förfrågningar, hur många gånger en viss sida har begärts m.m. Kan vara lämpligt att spara den informationen i en sqlite databas eller någon filbaserad nosql lösning.