

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

Deadline: 14/2 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. Snygg kod går före mycket funktionalitet.

Betygskriterier:

- Integrationer mellan system
- XML, Json och Protobuf
- Integrera javasystem med andra system
- Skapa en klient som konsumerar tjänster
- Skapa en webbtjänst som klienter kan anropa
- Identifiera integrationsrelaterade krav
- Skapa ett specificerat interface

Litteratur: Föreläsningsmaterial vecka 5 och 6.

Uppgift:

G-nivå

Arbeta i grupp

Projektet ska versionshanteras med 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.

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 url parametrar tillsammans med GET och som body text tillsammans med POST förfrågan.

Den inskickade informationen ska lagras i en **databas** och kunna skickas tillbaka som ett **json dokument** när det efterfrågas via en GET mot rätt URL. Provkör gärna med html formulär för inskickningen av information.

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

Servern ska vara multitrådad och kunna hantera flera förfrågningar samtidigt.

VG:

Utöver kraven för G ska:

Webb servern kunna hantera pluginklasser skrivna i Java som kan laddas in dynamiskt.
Konfiguration av routing m.m. som behövs för pluginen ska göras med runtime annotations.

<https://docs.oracle.com/en/java/javase/15/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 ett plugin 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 databastabell.