

Uppgift 1: Bygg ett Eventplaneringssystem med Node.js

Gruppuppgift med skriftlig redovisning

- **Betyg:** IG/G
- **Försök:** 2 försök (1 omexamination.)
- **Kursmål:**

1. (kunskap) Kommunikation via API.
2. (kunskap) Att konfigurera en Node.js-server för att köra JavaScript i olika operativsystem samt vilka konsekvenser valet av server har för en applikation samt översikt över pakethantering.
3. (kunskap) En grundläggande förståelse för ekosystemet kring backend för den moderna webben. Känna till begrepp som exempelvis webbshop, open source, CMS, headless, REST, m.fl.
4. (färdighet) Utföra installationer av en Node.js-instans som möjliggör att JavaScript körs på server och lägga till utvalda tilläggspaket.

Gruppuppgift

Bakgrund

Ni ska arbeta i grupp om 3-4 personer och tillsammans utveckla ett komplett API för att skapa ett avancerat eventplaneringssystem.

Systemets syfte är att göra det möjligt för eventorganisatörer att planera, skapa och hantera olika event. Samt för deltagare att ta reda på information och anmäla sig till event.

Tanken är att skapa ett fungerande alternativ för olika organisatörer, exempelvis offentliga- och ideella, som inte vill eller bör använda Facebook. Event som skapas i systemet bör därför vara enkla och lättillgängliga för allmänheten och vara förfarande vara anpassade för att dela i sociala medier och andra sammanhang.

Systemets kravlista

Allmäninformation: Systemet BÖR inkludera de flesta av de funktionaliteter som finns i Facebook, eller ha motsvarande alternativ. Systemet MÅSTE innehålla tillräckligt mycket event-information för att vara ett användbart alternativ till Facebook Event.

Deltagarinformation: Systemet SKA möjliggöra att viss information i ett event endast blir tillgängligt för deltagare som har anmält sig till eventet.

Godkänd anmälan: Eventorganisatörer SKA kunna låsa ett event så att anmälningar måste godkännas av eventorganisatörerna först.

Betald anmälan: Eventorganisatörer SKA kunna låsa ett event så att en anmälan måste betalas innan den godkänns.

Valfri anmälan: Eventorganisatörer SKA kunna välja för varje event om anmälan krävs, om anmälan måste godännas, om anmälan måste betalas. Något eller flera alternativ är möjliga.

Max antal anmällda: Eventorganisatörer BÖR kunna ange ett max antal deltagare, efter det går det ej att anmäla sig till eventet.

Hemliga event: Eventorganisatörerna SKA kunna välja att event är publikt eller hemligt.

Användargränssnitt: Systemet SKA med ett lämpligt SPA gränssnitt kunna visa och uppdatera information på sidan utan att behöva laddas om manuellt.

Lista efter organisatör: användargränssnittet SKA kunna visa alla event en organisatör har skapat som är publika.

Lista efter plats: användargränssnittet SKA kunna visa alla event på en viss plats som är publika.

Uppgiftsbeskrivning

Ert mål är att under kursens gång skapa detta System, vilket kommer bestå av ett Webb API samt ett användargränssnitt i ett SPA verktyg, exempelvis React. Ni kommer jobba med projektet under kursens gång, från teoretisk planering till genomförande.

Arbetsuppdelning

Gruppen ansvarar gemensamt för allas delaktighet, alla i gruppen ska dock bidra med commits.

Vecka 1:

Projektstruktur och konfiguration: Ni har påbörjat ert projekt, gjort ett gitrepo, samarbetskontrakt etc. Ni har börjat planera erat API och svarat på frågan om vilka REST resurser ni behöver ha.

Förslag på resurser efter vecka 1, erat projekt ser liknande ut (men alla resurserna här måste inte finnas):

```
/events
/users
/participations
/payments
/links
/organisations
/locations
```

Vecka 2:

Implementering: Ni skapar nu ett NodeJS projekt som ni kan köra från terminalen. Projektet startar och svarar på anrop till alla resurserna, även om vissa anrop orsakar fel så finns dem där. Ni har kommit olika långt med er funktionalitet.

Ni har börjat med ett grafiskt gränssnitt (GUI) som kan öppnas i webbläsaren och som kan göra anrop till REST API:et. GUI:t hanterar alla ev. fel från Rest API:et.

Förslag på arbetsuppdelning följer vecka för vecka...