

Schemahantering LTU EA- modellering/Systemintegration

D0031N EXAMINATIONSUPPGIFT 4
ALEXANDER TARBERG

Innehållsförteckning

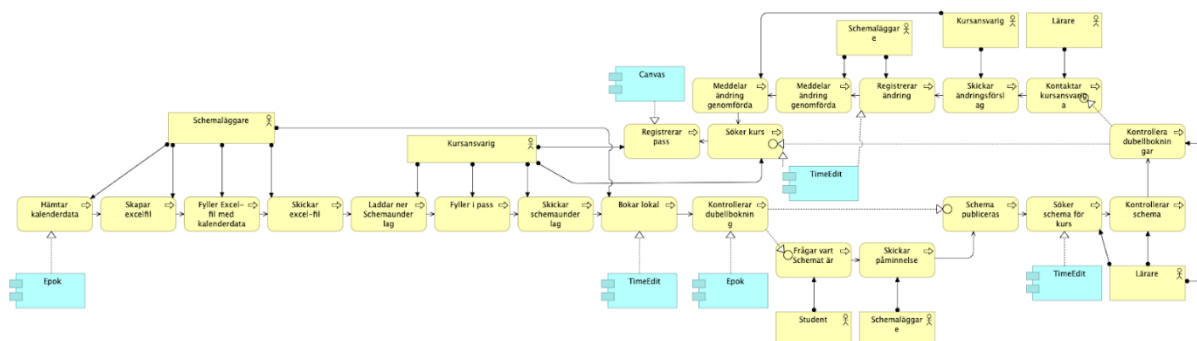
| | |
|--------------------------------|---|
| Inledning..... | 2 |
| Resultat..... | 2 |
| Nuläge..... | 2 |
| Börläge..... | 3 |
| Förklaring Implementering..... | 4 |

Inledning

Uppgiften som gavs var att underlätta Luleås schemahantering. I dagsläget ingår flertalet manuella steg och detta upplevs enligt anställda som tidskrävande. Vi studenter fick i uppgift att genomföra en nulägesbeskrivning samt en bör lägesbeskrivning för LTUs Schemaläggning. En liten del i slutet av verksamhetsprocessen skulle även implementeras/underlättas med programkod, att hämta schema från TimeEdit och registrera passen i Canvas. Detta genomfördes med hjälp av en JSON-parser(eftersom vi inte hade tillgång till TimeEdit API) och POST-api requests till Canvas.

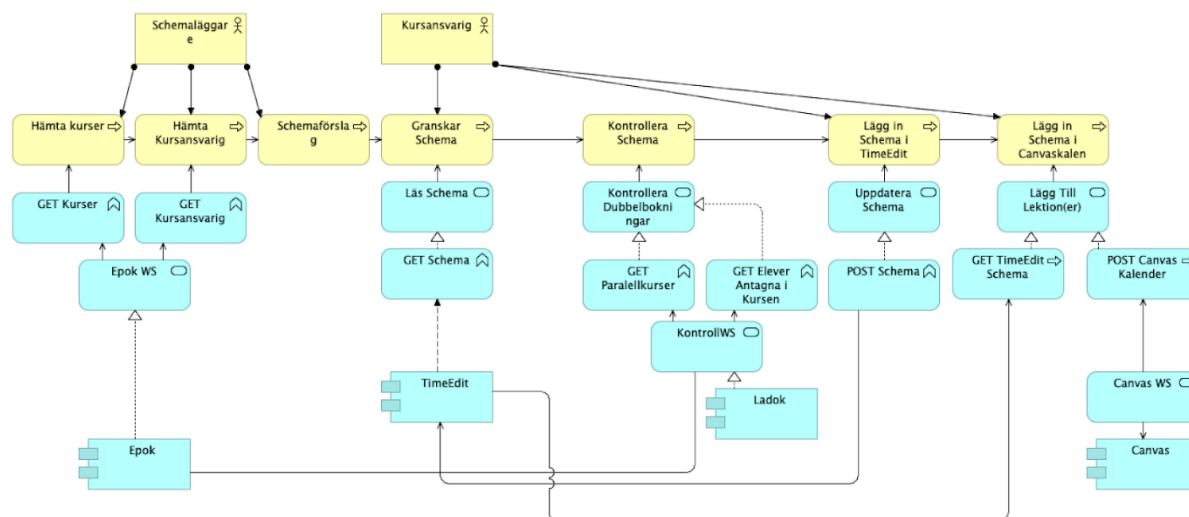
Resultat

Nuläge



Schemaläggare hämtar kalenderdata. Schemaläggare skapar excel fil. Schemaläggare fyller excelfil. Schemaläggare skickar excel fil via e-post till kursansvariga. Kursansvariga laddar ner schema underlaget. Kursansvariga fyller i pass. Kursansvarig skickar fil med schemaunderlag till schemaläggare. Schemaläggare bokar lokal i TimeEdit. Schemaläggare kontrollerar dubbelbokning på parallella kurser via Epok. Student skickar(eventuellt) förfrågan om vart schema är(om schema saknas). Schemaläggare skickar påminnelse till kursansvariga. När alla scheman är registrerade publiceras de i TimeEdit. Lärare söker schema för kurs i TimeEdit. Lärare kontrollerar schemat med underlag. Lärare kontrollerar att de inte finns dubbelbokningar i sin kalender. Lärare kontaktar(eventuellt) kursansvariga. Kursansvariga skickar förslag till ändringar till schemaläggaren. Schemaläggaren registrerar ändringar i TimeEdit. Schemaläggaren meddelar att ändringarna genomförs till kursansvariga. Kursansvariga meddelar lärare att de är genomförda. Kursansvariga söker fram schema för en kurs i TimeEdit. Kursansvariga registrerar pass i Canvas kalender.

Börläge

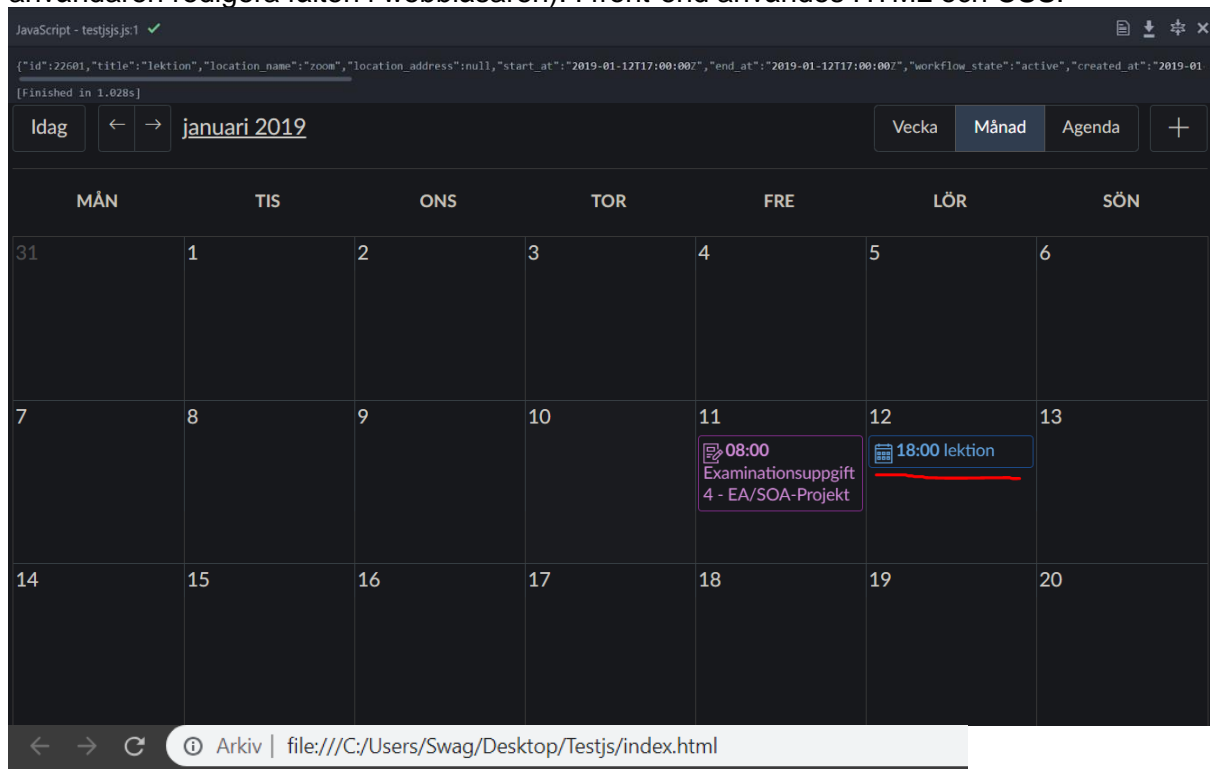


Schemaläggare hämtar kurser med hjälp av en GET Kurser metod. Denna metod använder sig av Epok WS vilket kommunicerar med Epok. Sedan hämtar schemaläggaren kursansvariga med hjälp av GET kursansvarig vilket söker efter kursansvarig på kursen. Detta används som underlag till kalenderdata. I schemaläggningen anger inte beskrivningen vad exakt som hämtas, så undertecknad antar att de är detta. Baserat på denna info sammanställer schemaläggaren ett schemaförslag.

Detta schemaförslag granskas av kursansvarig. Schemat kontrolleras automatiskt för dubbelbokningar med hjälp av WS som hämtar antagna elever i kursen och parallella kurser, sedan kontrolleras bokningarna för duplicates (om två styckna parallella kurser överskrider varandra). Denna WS använder sig av TimeEdit, Epok och Ladok för att slutföra detta. Kursansvariga lägger in schemat i TimeEdit med hjälp av POST-request till TimeEdit (efter att den har kontrollerats av kursansvarig). Till sist lägger läraren enkelt till schemat i canvas-kalender med hjälp av ett interface. Detta interface stödjer hämtning av TimeEdit schema med hjälp av API GET Requests från TimeEdit. Om vi hade fått tillgång till detta API hade GET-requests varit optimalt, men eftersom de inte var tillgängligt behövde vi hantera de med en JSON- eller XML-parser. Infon importerades sedan till interfacet (en HTML-sida i mitt fall) där läraren kan ändra schema info om så behövs, eller bara klicka skicka. När läraren eller kursansvarig klickat "skicka" så skickas en POST-request till Canvas-kalendern. Schemat finns nu tillgängligt för lärare och studenter i både TimeEdit och Canvas.

Förklaring Implementering

Implementeringen genomfördes med hjälp av JavaScript med ramverket NodeJS i back-end. Använde även lite jQuery/Ajax för att hämta data från user input (text i redigera kalendern, om användaren redigera fälten i webbläsaren). I front-end användes HTML och CSS.



Importera Kalender

[Öppna TimeEdit](#)

TimeEdit Url:

Importera

Canvasuppgifter

User ID:

Canvas API Token:

Skicka

Användaren kan enkelt hämta URLen genom TimeEdit genom att klicka på Öppna URL och sedan kopierar de URL:en och lägger in i fältet och klickar på "importera". När användaren sedan har importerat fälten skall de skriva sina USER-id och Canvas API-token, eftersom detta behövs för att genomföra POST-requesten genom Canvas-API.

← → ↻ Arkiv | file:///C:/Users/Swag/Desktop/Testjs/index.html

Redigera kalender

ID:

509926

Kurs:

Enterprise Architecture using SOA, Enterprise Architecture using SOA

Aktivitet:

tutoring

Lärare:

Ingemar Andersson, Svante Edzen

Service:

Zoom

Datum:

2018-12-18

Start Tid:

10:15

Slut Tid:

11:45

Plats:

Zoom

Beskrivning:

Lektion idag

Skicka

Dessa fält ovan populeras efter användaren har klistrat in länken och klickat på "importera". Som vi ser i bilden ovan har jag importerat en Zoom-handledning från den 18:e december mellan 10:15-11:45. Fälten kan även manipuleras om sådant behov finns innan de skickas. När man klickar på "skicka" knappen skapas en länk med inputens från fälten. Denna länk skickar sedan en POST-request till Canvas API att lägga upp eventet i kalendern. Kalendern är sedan synlig för studenterna.

Implementeringen förklaras mer genomgående i kommentarerna inuti programmet.

Områden som skulle kunna förbättras med min lösning är bland annat säkerhet. API-nyckeln bör till exempel gömmas/krypteras för som de ser ut i dagsläget kan folk enkelt sniffa nyckel eftersom requesten sker via HTTP. Eller om de inspekterar programkoden kan de också enkelt hitta känslig info(såsom bearer token i plaintext-form). Så säkerheten bör förbättras i en verklig implementering. Jag kunde heller inte få de att fungera med att lägga in User-id och token i länken genom fälten, vet inte riktigt varför de inte fungerade. Men själva POST-requesten fungerar, om man lämnar dessa fält tomma(koden innehåller mina Canvas Uppgifter). En till del som skulle kunna förbättras är användar interfacet, så att de blir mer estetiskt och användarvänligt.