

Lärosätenas förväntningar på Ladok som informationsmaster

Ladok har en unik roll som informationsmaster för hela sektorn för högre utbildning. Nästan samtliga lärosäten tappar förmågan att bedriva utbildning om informationsflödena ut ur Ladok slutar fungera. Lokala integrationer som hanterar Ladok-data har genomgående högsta prioritet på lärosätena, och integratörer och arkitekter har stort lokalt tryck på effektiva och välfungerande informationsflöden.

På samma sätt som Ladoks användargränssnitt är verktyg för studieadministratörer, så är Ladoks API:er verktyg för arkitekterna och integratörerna. Ett svårarbetat, icke-intuitivt eller odokumenterat verktyg ger både ineffektivt arbete, dåliga resultat och har potential att orsaka arbetsmiljöproblem.

Från ATI-gruppen inom ITCF vill IT-arkitekterna på svenska lärosäten förmedla följande förväntningar på Ladok i denna verksamhetskritiska roll.

1. Ladok behandlar API:er för integratörer som en första klassens produkt

Vi förväntar oss att Ladoks gränssnitt för integratörer är enhetliga, med genomtänkta och testade funktioner framtagna iterativt med representanter för integratörer, och har heltäckande och välskriven dokumentation. Vi förväntar oss att Ladok betraktar ny funktionalitet som färdig först då API:er levererats i sin helhet enligt detta.

Det är samma krav som Ladok med självklarhet uppfyller vad gäller funktionalitet och GUI.

Integratörsgränssnitten uppstår idag som bieffekt av övrig utveckling, och är oftast varken kravställda, avstämda, heltäckande, enhetliga eller dokumenterade.

2. Ladok har en dokumenterad informationsmodell och funktionsbeskrivning

Vi förväntar oss att Ladok har en uppdaterad dokumentation som beskriver alla de entiteter som Ladok hanterar åt lärosätena, deras attribut och dessas respektive betydelser, relationer mellan entiteter, möjliga värden och så vidare.

Vi förväntar oss också att Ladok tillhandahåller en funktionsbeskrivning med vars hjälp vi kan förutse normala förändringar av datamängderna. Det innefattar till exempel en beskrivning av möjliga state-förändringar, och noteringar om vilka värden som Ladok anser vara statiska då de väl skapats.

3. Ladok erbjuder dataåtkomst via koherenta synkrona och asynkrona API:er

Vi förväntar oss att Ladok erbjuder ändamålsenlig åtkomst till all den data man håller åt lärosätet via synkrona och asynkrona ändpunkter, bägge modellerade enligt samma publicerade informationsmodell.

Oavsett synkron eller asynkron leverans ser innehåll och representation identisk ut. Skillnaden är att de synkrona ändpunkterna ger klientinitierad åtkomst till valfri instans, medan de asynkrona ändpunkterna självmant levererar ut en instans så snart någon data i den ändrats.

Lärosätena väljer själva hur man implementerar behov av högre tillgänglighet eller sänkt latens, t.ex. genom HTTP-cache framför de synkrona ändpunkterna eller genom att konsumera objekten asynkront och spara ner i lokal lagring.

4. Ladoks API:er representerar dess information, inte dess implementation

Den information Ladok hanterar och hur olika informationsobjekt relaterar till varandra är med nödvändighet hårt knutet till Ladoks funktionalitet, men det finns många olika möjliga implementationer för att göra denna datahantering.

Vi förväntar oss att Ladok:s API:er representerar informationen som hanteras på ett sätt som gör att mottagaren inte behöver någon kunskap om Ladoks valda implementation utöver den publicerade informationsmodellen och funktionsbeskrivningen.

Den största mängden vidareutveckling består i att lägga till, snarare än att plocka bort, information. Vi förväntar oss att API:erna är anpassade efter detta och att det blir lätt att utöka informationsmodellen utan att befintliga integrationer går sönder.

5. Ladok har komposita API:er enligt standarderna LIS och EduAPI

För de behov av komposita (färdigberikade) objekt som lärosätena kan tänkas ha erbjuds sådana objekt enligt LIS och EduAPI. Övriga API:er utgår från informationsmodellen och levererar kompletta instanser med referenser snarare än att berika från tänkta/faktiska användningsfall hos mottagaren. Ladok lägger inte resurser på att försöka ta fram egna komposita objekt utifrån lärosätenas upplevda behov.

Lärosäten med behov av lokalt berikade objekt väljer själva hur detta sker utifrån de enhetliga synkrona och asynkrona API:erna, med eller utan lokal cache på någon nivå.

6. Långsiktighet

Vi förväntar oss att publicerade API:er förblir aktiva och fungerande under lång tid. När ett lärosäte väl byggt en integration mot ett visst API kan det ta mycket lång tid för lärosätet att kunna prioritera en förändring. Lärosätena har inte förmågan att på sex veckors varsel förändra en integration på grund av ändrat API - ett rimligt målvärde är tolv månaders förvarning innan API:er slutar fungera.

Samtliga övriga förväntningar i detta dokument samspelar för att göra det enklare för Ladok att uppfylla denna förväntan, vilket de med dagens API-strukturer har en utmaning att göra. Vi tror att det finns en sund grundstruktur som gör allt detta hanterbart.

7. Arbetssätt

ATI-gruppen vill fungera som en naturlig samarbetspartner och referensgrupp för övergripande frågor om API:erna. Ladoks egen tekniska referensgrupp kan sedan fungera som motpart för detaljfrågor.

Vi har alla ett gemensamt önskemål om goda API:er, och vill aktivt hjälpa Ladok att nå fram dit. Vi tror att vi för de övergripande designfrågorna kan komma fram till en uppsättning principer och mallar som Ladoks team kan följa, och att löpande frågor om API:ets design blir relativt få.