

Laboration - Webbtjänster

2015

Överblick

Projektet går ut på att ni ska exponera delar av er ECommerceManager som webbtjänster (JAX-RS). Projektet är betygsgrundande och har två svårighetsgrader **godkänd** och **väl godkänd**.

Nivån för **godkänd** gör ni tillsammans som grupp medan nivån för **väl godkänd** gör ni individuellt.

Projektet kommer att genomföras i grupper om max **fem** personer.

Laborationen kommer att redovisas **onsdagen 11 mars 2015** med början från **klockan 09:30** och nivån för **väl godkänd** kommer att redovisas **fredagen den 13 mars** med början från **klockan 13:00** (detta sker individuellt).

Arkitektur

Allmänt

Laborationen kommer att kräva två projekt (minst). Ett projekt för er ECommerceManager och ett projekt för webbtjänstlagret (JAX-RS). *Om du vill* kan du även bryta ut din Model (User, Product, Order, etc) till ett separat projekt. *Alla* beroenden i de olika projekten ska hanteras med hjälp av Maven. Detta betyder att du kommer att behöva installera din ECommerceManager i din lokala Maven-repository och sedan definiera detta beroende i ditt webbtjänst-projekt.

ECommerceManager

Denna ska följa den design som ni fick i tidigare laboration. All data ska lagras i en databas med hjälp av JDBC.

Webbtjänstlager (JAX-RS)

Detta lager kommer att exponera delar av er ECommerceManager som webbtjänster. Webbtjänsterna kommer att ta emot och leverera JSON. När webbtjänsterna anropar ECommerceManagern kommer JSON att översättas till model-klasser (Product, User, Order, etc) och när ECommerceManagern levererar resultat i form av model-klasser kommer dessa att översättas till JSON för att kunna skickas vidare till klienten.

Krav för Godkänd

(Denna nivå görs som grupp och är obligatorisk)

Applikationen ska via webbtjänsten erbjuda möjligheten att:

- ☒ Hämta alla produkter
- ☒ Hämta en produkt med ett visst id
- ☒ Skapa en ny produkt - detta ska returnera en länk till den skapade produkten i Location-headern
- ☒ Uppdatera en produkt
- ☒ Ta bort en produkt (eller sätta den som inaktiv)
- ☒ Hämta en användare med ett visst id
- ☒ Skapa en ny användare - detta ska returnera en länk till den skapade användaren i Location-headern
- ☒ Uppdatera en användare
- ☒ Ta bort en användare (eller sätta den som inaktiv)
- ☒ Skapa en order för en användare
- ☒ Hämta en användares alla order
- ☒ Hämta en viss order för en användare
- ☒ Uppdatera en order för en användare
- ☒ Ta bort en order för en användare

Krav för Väl Godkänd

(Denna nivå görs individuellt och är frivillig)

- ☒ Det ska via webbtjänsten gå att autentisera en användare - du kommunicerar med statuskoder huruvida autentiseringen gick bra eller inte
- ☒ Du använder dig uteslutande av *MessageBodyReader* och *MessageBodyWriter* för att serialisera och deserialisera - detta betyder att ingen JAX-B får användas
- ☒ Du har minst ett eget Exception som är subklass till *WebApplicationException* som kastas på passande ställe
- ☒ Du har minst ett Exception som mappas till en Response via en *ExceptionMapper*
- ☒ Klienten är uppdaterad så att den nya funktionaliteten kan verifieras

Generella krav

All data ska sparas i en databas som exponeras via ECommerceManagern. Webbtjänsterna ska kommunicera med hjälp av HTTP. Detta betyder att om något gick fel när användaren anropade webbtjänsten ska passande status-code returneras och eventuellt ett meddelande i response-body. Viktigt är också att webbtjänsternas URLer följer en bra design. Här följer ett exempel:

Bra: <http://coredev.se/customer-service/customer/1>

Dålig: http://coredev.se/customer-service/customer?customer_id=1

*Lycka till,
Anders och Pablo*