

# Inlämning 1 - API-utveckling och DB normalisering

## Deadline

Fredag den 25/4 kl 23:59

## Inlämning

Github repo länk som skall innehålla ER-diagram, sql-fil på er databas och Express API:et  
Github repot skall vara privat, och ni skall bjuda in mig som en collaborator via mitt github alias: **sibala**

Lägg till testdata i tabellerna, så att det kommer med i SQL-export filen. Så att man kan testa hur API:et funkar.

## Uppgiftsbeskrivning och krav

Ni skall göra ett normaliserat ER-diagram och ett API för en produktkatalog i en e-shop  
Det gäller specifikt produkter och tillhörande kategorier

Tabellen "**products**" skall innehålla följande information

- title
- description
- stock
- price
- image
- created date

Tabellen "**categories**" skall innehålla följande information

- name

*Obs, ytterligare fält för PK och FK få ni lägga till på egen hand, så att DB tabellerna funkar korrekt enligt beskrivning nedan*

## Uppgiftskrav på G-nivå

- Skapa ett korrekt normaliserat ER-diagram efter följande beskrivning
  - En e-shop har produkter och kategorier
  - Produkter kan tillhöra **en** kategori i taget
- Skapa tabellerna med relevanta **PK fält och nycklar** i PHPMyAdmin
- Tilldela fält till tabell(er) som möjliggör korrekt relation mellan tabellerna i PHPMyAdmin
- Skapa CRUD för kategorier, med följande endpoints
  - Hämta alla kategorier med GET: <http://localhost:3000/categories>
  - Hämta alla produkter tillhörande en viss kategori med GET: <http://localhost:3000/categories/:id/products>
  - Skapa ny kategori med POST: <http://localhost:3000/categories>
  - Uppdatera befintlig kategori med PATCH: <http://localhost:3000/categories/:id>
  - Radera befintlig kategori med DELETE: <http://localhost:3000/categories/:id>
- Skapa CRUD för produkter, med följande endpoints
  - Hämta alla produkter med GET: <http://localhost:3000/products>
  - Hämta enskild produkt med GET: <http://localhost:3000/products/:id>
  - Skapa ny produkt med POST: <http://localhost:3000/products>
  - Uppdatera befintlig produkt med PATCH: <http://localhost:3000/products/:id>
  - Radera befintlig produkt med DELETE: <http://localhost:3000/products/:id>

## Uppgiftskrav på VG-nivå

Allt som G-nivån innefattar, med följande skillnader/tillägg

- Skapa ett korrekt normaliserat ER-diagram efter följande beskrivning
  - En e-shop har produkter och kategorier
  - Produkter kan tillhöra **flera** kategorier
- Lägg till **FK nycklar/index** som säkerställer att produkter kan endast tilldelas kategorier som finns
- Utöver endpoints på G-nivå, skall även följande filtrering/sökning skapas för:
  - Hämta alla produkter med GET: <http://localhost:3000/products>
    - Skall kunna söka produkter efter produkt titel
    - Skall kunna sortera produktlistan efter pris, både (asc/desc)
- Error Hantering med korrekta HTTP status koder och lämpliga felmeddelanden
- Validera produkter vid POST & PATCH. Det skall inte gå att skapa/uppdatera produkter, samt passande felmeddelande skall visas, om något av följande obligatoriska fält saknas:
  - title, description, stock, price