TDP003

Föreläsning 2

Emma Enocksson Svensson

LINKÖPINGS UNIVERSITET

1. **Kursinformation**
2. Dokument
3. Projektplan
4. Frågor
5. Genomgång av projekt
6. Vad är ett API?
7. Kom ihåg

LINKÖPINGS UNIVERSITET

Vad händer härnäst?

V36
Fö: Projektinformation
Gruppkontraktet

V37
Se: Hur ska en projektplan se ut?
Projektplanering, LoFi-prototyp

V38
Projektplan och Installationsmanual

V39
Fö: Flask och Jinja2
Datalagret godkänt, kompletteringar, Git-bidrag

Kom ihåg

- Webreg: skriv upp er
- Skapa repository i GitLab
- Dagbok
- Planeringen över Projekten

1. Kursinformation
2. **Dokument**
3. Projektplan
4. Frågor
5. Genomgång av projekt
6. Vad är ett API?
7. Kom ihåg

LINKÖPINGS UNIVERSITET

Leverabler

- **Förberedelse:** undersök existerande system och fundera på hur ni vill ha det.
  - Leverabler: skisser på systemet (LoFi-prototyp), projektplan
- **Konstruera:** design och implementation av systemet
  - Två delsystem ska byggas: presentation och data
  - Leverabler: fungerande system och dokumentation
- **Överlämna:** färdigställande och inlämning av projektet
- **Uppföljning:** reflektion, testning av varandras system, utvärdering av hur projektet har gått.

Projektplaneringen

- Grund för tidsplanen i projektplanen
- Skaffa en förståelse för innehållet i kursen
  - Vad ska göras, när behöver det göras
- Använd deadlines i schemat, kurshemsidan, sök upp information
- Hur presentera på tydligt sätt?

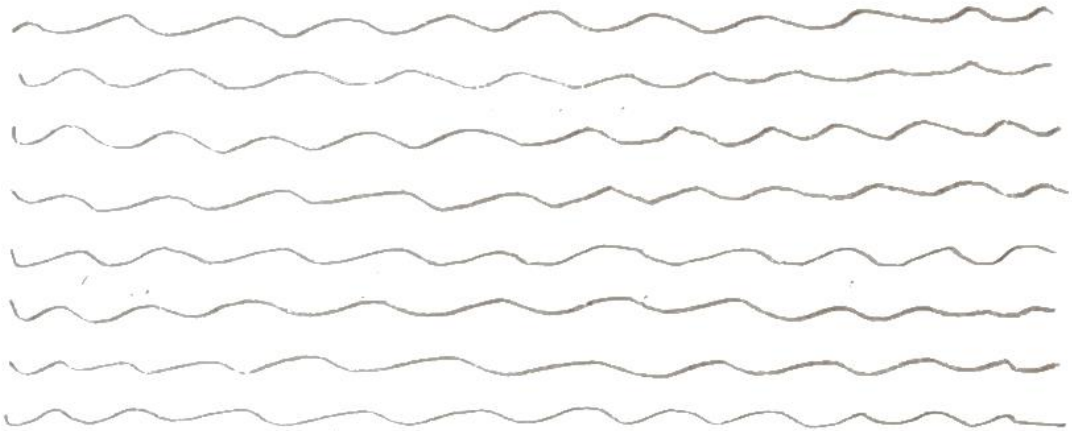
Vecka 38
Deadlines: Första utkast till gemensamma installationsmanualen, första utkast till projektplanen - 21 september
Aktiviteter:
Måndag: Färdigställa installationsmanualen – 3 timmar
Tisdag: Färdigställa projektplanen – 4 timmar
Onsdag: Läsa på om Flask – 2 timmar
Onsdag – Fredag: Jobba med datalagret – 8 timmar

LoFi-prototyp

- Systemskiss
- Pappersprototyp
  - Rita, klippa, klistra
- Rita upp på papper hur ert system ska se ut
- Torrkör med varandra och se om ni har missat något
- Se till att relevanta krav syns

HOME PROJECTS TECHNIQUES

ABOUT ME



Installationsmanual

- Beskriv hur man sätter upp utvecklingsmiljön
  - Ni måste göra det själv!
- Ta reda på vilka verktyg som behövs
- Installation, testa om det fungerar, felhantering
- Tänk på målgrupp
- Gruppvis dokumentation först, sedan ska klassen tillsammans skapa världens bästa installationsmanual

Systemdokumentation

- Lämnas in på slutet av projektet
- Beskriver hur systemet är uppbyggt
- Underlätta för andra programmerare att förstå systemet och underhålla kodbasen
- Både översiktligt och i detalj

Testdokumentation

- Acceptanstestning
  - Testa att systemet fungerar enligt specifikation
- Beskriv tester som täcker alla funktionella krav
- Specificera tester noggrant för repeterbarhet
  - Hur testet görs
  - Indata, utdata
  - Normalfall, undantagsfall
- Daterad testlogg

Individuell dagbok och reflektion

- Kom ihåg att skriva programmerardagbok!

- En skriftlig reflektion görs baserat på dagbok och Code Complete 2 (del 1, 3, och 4)
  - Välj ut teman i Code Complete 2 som du känner att ni kommit i kontakt med under projektet, eller som intresserar dig. Arbeta med dessa teman extra noga.
  - Följ krav och riktlinjer på hemsidan
  - Viktig examinationsform på flera kurser!

1. Kursinformation
2. Dokument
3. **Projektplan**
4. Frågor
5. Genomgång av projekt
6. Vad är ett API?
7. Kom ihåg

LINKÖPINGS UNIVERSITET

Vitsen med en projektplan

- Beskriver hur arbetet ska uppnås
  - Hur ska ni samarbeta
  - Vilka verktyg ska användas
  - etc.
- Överenskommelse mellan er och kunden
  - ...samt inom gruppen

Projektplan

- En projektplan ska (åtminstone):
  - Vara nedskriven i förväg
  - Beskriva
    - *Vad* som ska uträttas
    - *När* det ska uträttas
    - *Hur* det ska uträttas
    - Men *ingen* design
  - Vara välskriven, strukturerad, kortfattad och begriplig
  - Ta hänsyn till tänkbara olyckshändelser
  - Utvecklas i takt med projektet (ett levande dokument)

En projektplan **kan** innehålla

- **Översikt**
  - Intro till arbetet, kunden, gruppens organisation
- **Fasplan/tidsplan**
  - Vilka utvecklingsfaser, vilka produkter, vilka datum, hur lång tid?
- **Organisationsplan**
  - Vilka team, vems ansvar?
- **Testplan**
  - Vem? Hur? Verktyg?
- **Hantering av artefakter**
  - Var och hur ska källkod och dokument lagras?
- **Dokumentationsplan**
  - Vilka, när, till vem? Vem godkänner?

En projektplan kan innehålla

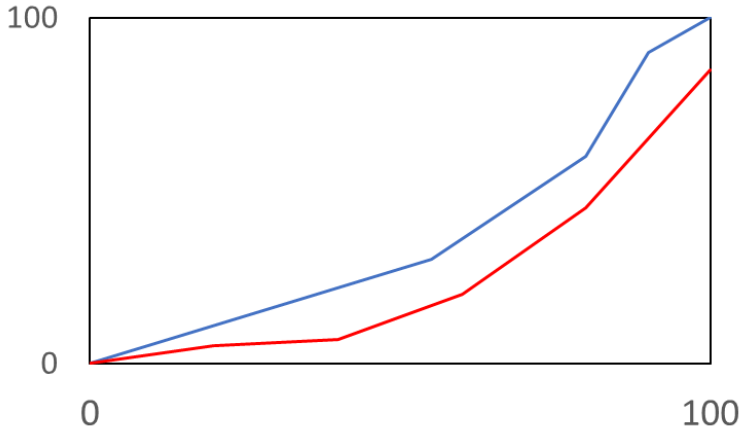
- **Utbildningsplan**
  - Intern, extern. Vem, när, resurser?
- **Plan för rapportering och granskningar**
  - Vad, till vem, när?
- **Installationsplan**
  - Vilka procedurer krävs för att komma igång?
- **Plan för kvalitetssäkring**
  - Standarder som ska användas?
- **Varuplan**
  - Vad ska levereras, när? Delleveranser?
- **Resursplan**
  - Persontid, datortid. Summering av milstolpar!

Milstolpar

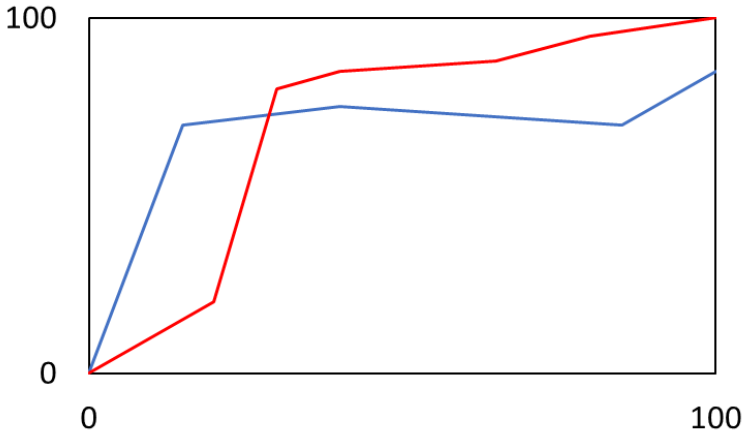
Milstolpe	**Datum**
LoFi-prototyp klar	2017-09-10
Projektplan inlämnad	2017-09-11
Projektplan klar	2017-09-15
load och get_project_count fungerar	2017-09-16
Datalagret klart	2017-09-20
Datalagret godkänt av assistent	2017-09-22
Vi kan visa en enkel sida med Flask och Jinja2	2017-10-01
Projektsidan fungerar	2017-10-05

Tidsplanering

Nybörjare



Professionell



- Arbeta i början!
- Dela upp problemet
- Sikta på professionell kurva

Arbetsinsats Leverabler

Seminarium om projektplan

- Tisdag 14 september
- Att göra innan:
  - Gör och skicka in projektplaneringen till mig
  - Läs igenom exempelprojektplanerna innan
  - Läs igenom all info om projektplanen

- På seminariet:
  - Vi diskuterar exemplen och hur man skriver en bra projektplan

1. Kursinformation
2. Dokument
3. Projektplan
4. **Frågor**
5. Genomgång av projekt
6. Vad är ett API?
7. Kom ihåg

LINKÖPINGS UNIVERSITET

1. Kursinformation
2. Dokument
3. Projektplan
4. Frågor
5. **Genomgång av projekt**
6. Vad är ett API?
7. Kom ihåg

LINKÖPINGS UNIVERSITET

Skapa en webbportfolio

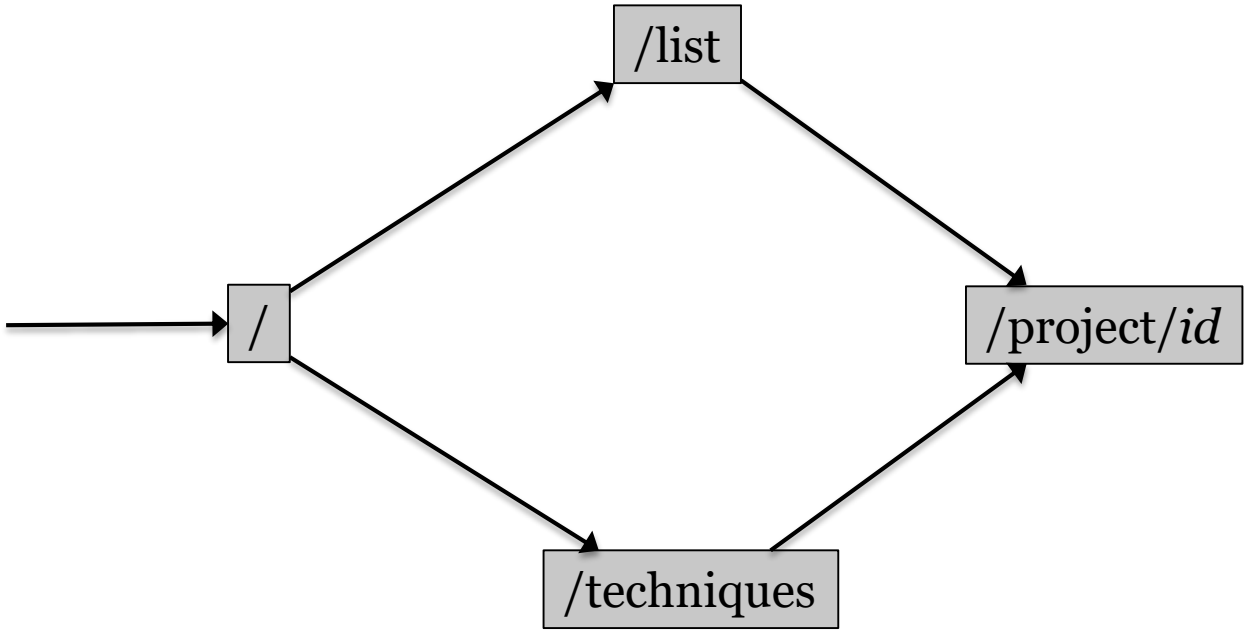
- Ett webbaserat system för att visa upp gjorda projekt
- Listar de projekt ni har gjort och egna projektsidor med information om projektet
- Ett verktyg åt er själva
- Ni är själva beställare av systemet
- Ni och andra som visar era projekt är användare
- Kursledningen är projektledare som dragit upp riktlinjer för arbetsordning (som ni har att acceptera som givna krav)
- Inom givna ramar har ni fria händer

Idé

- Vi har en samling av projekt som vi vill visa
- Varje projekt har ett namn, en beskrivning, en samling tekniker
- Vi vill kunna hitta alla projekt genom att bläddra, söka
- Vi vill också kunna hitta alla projekt som involverar en viss teknik

Systemskiss

- Fyra webbsidor med URL:er:

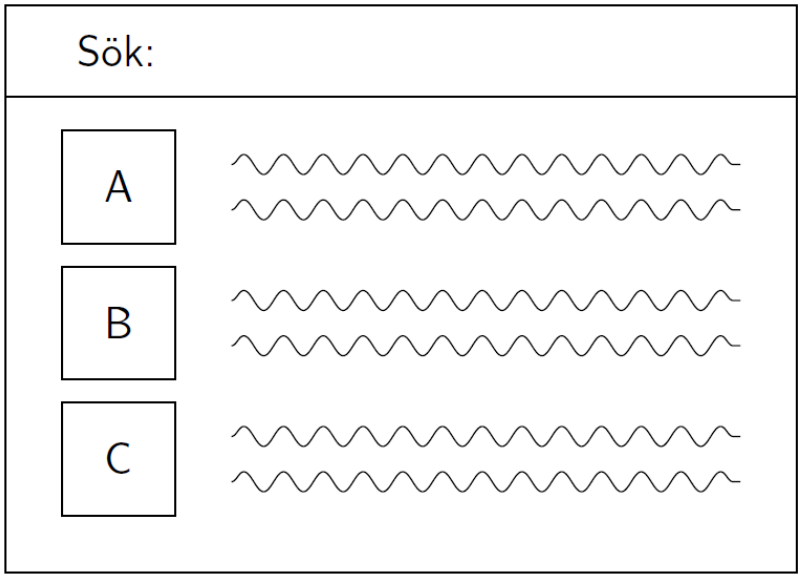


Startsidan

- **URL:** http://foo/
- Statisk eller dynamisk
- Ska ha bilder

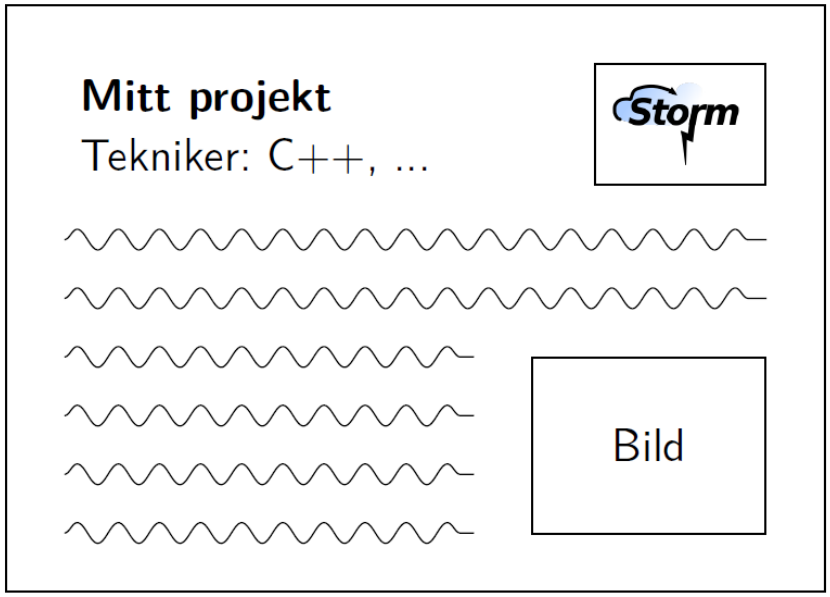
Listsidan

- **URL:** http://foo/list
- Dynamisk lista över projekt med liten bild och kort info för varje
- Ska vara möjligt att sortera och söka



Projektsidan

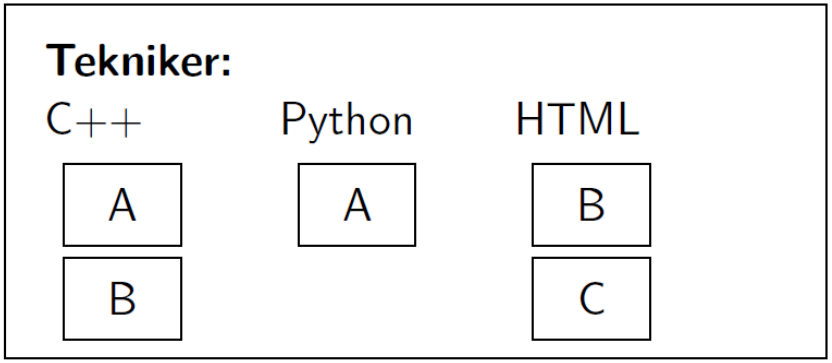
- **URL:** http://foo/project/*project-id*
- Visar fullständig information om ett projekt
- Ska ha en stor bild för projektet



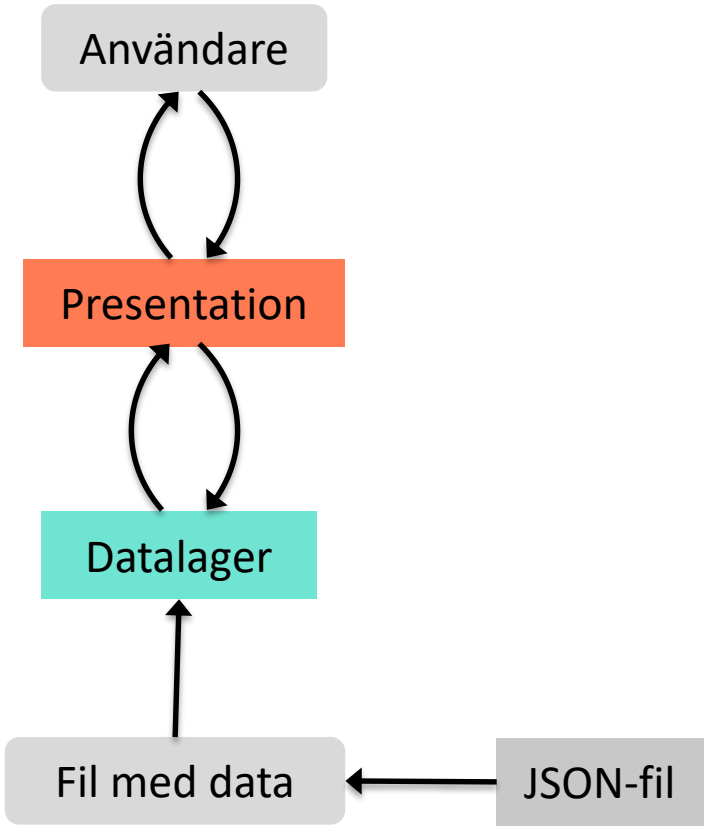
Tekniksidan

URL: http://foo/techniques

- Lista över era projekt, baserat på använda tekniker
- Helt OK att lägga till en extra sida (exempelvis http://foo/techniques/html) som visar alla projekt som innehåller en viss teknik



Arkitektur



- Två delsystem med olika ansvar
  - Presentation: användarhändelser
  - Data: datahantering

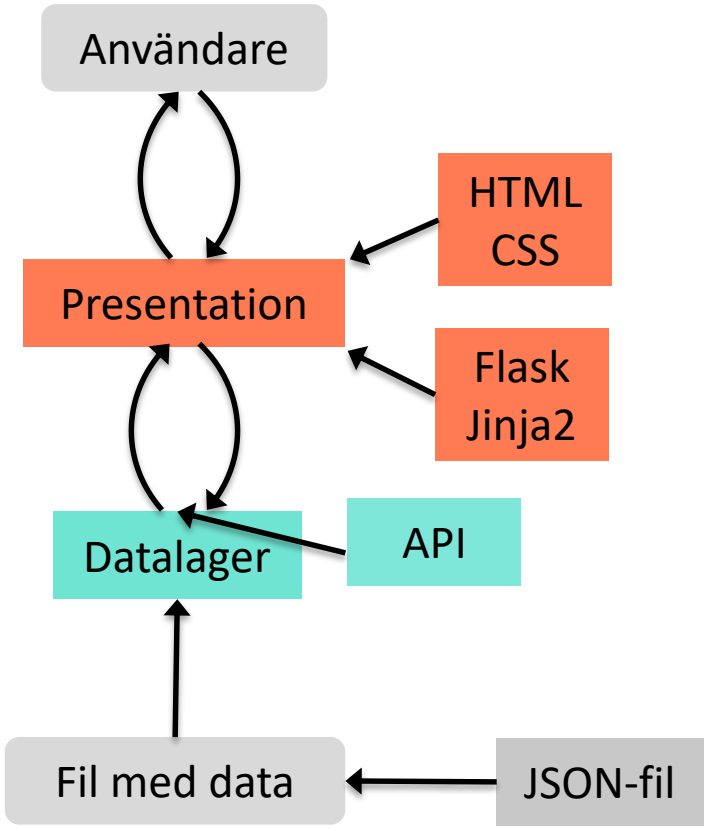
JSON

- JavaScript Object Notation (JSON)
- Datalagring och datautbyte
- Läsbart/skrivbart för människor
- Påminner om dictionary och andra datatyper i Python
- Många programmeringsspråk kan parsa JSON

JSON - exempel

```
[
  {
    "Art": "Hund",
    "Namn": "Idefix"
  },
  {
    "Art": "Katt",
    "Namn": "Sebbe"
  }
]
```

Arkitektur



- Två delsystem med olika ansvar
  - Presentation: användarhändelser
  - Data: datahantering
- Standardiserat källkodsgränssnitt (API) till delsystemet data.
- Vi tillhandahåller test och testdata

1. Kursinformation
2. Dokument
3. Projektplan
4. Frågor
5. Genomgång av projekt
6. **Vad är ett API?**
7. Kom ihåg

LINKÖPINGS UNIVERSITET

API (Application Programming Interface)

- Man vill ofta strukturera kod så att man samlar relaterad kod i en enhet
  - Funktioner, klasser, moduler
  - Abstraktion
- Behövs ett sätt att kommunicera mellan moduler!
- API: Strikt överenskommelse mellan moduler
- Att använda ett API är ett viktigt sätt att hantera komplexitet och isolera olika delar för att få dem utbytbara

”API” på en restaurang

- Köket har bestämt vad som kan serveras
- Köket har koll på hur man tillagar varje rätt
- Kunden kan titta i menyn och beställa det den vill ha
  - Menyn fungerar som ett API
- Kunden behöver INTE gå till köket och förklara hur rätten ska tillagas
- Vilken kund som helst kan gå till restaurangen och beställa det den vill ha
- Kunden kan skicka med extra information

Dyrt att ändra API

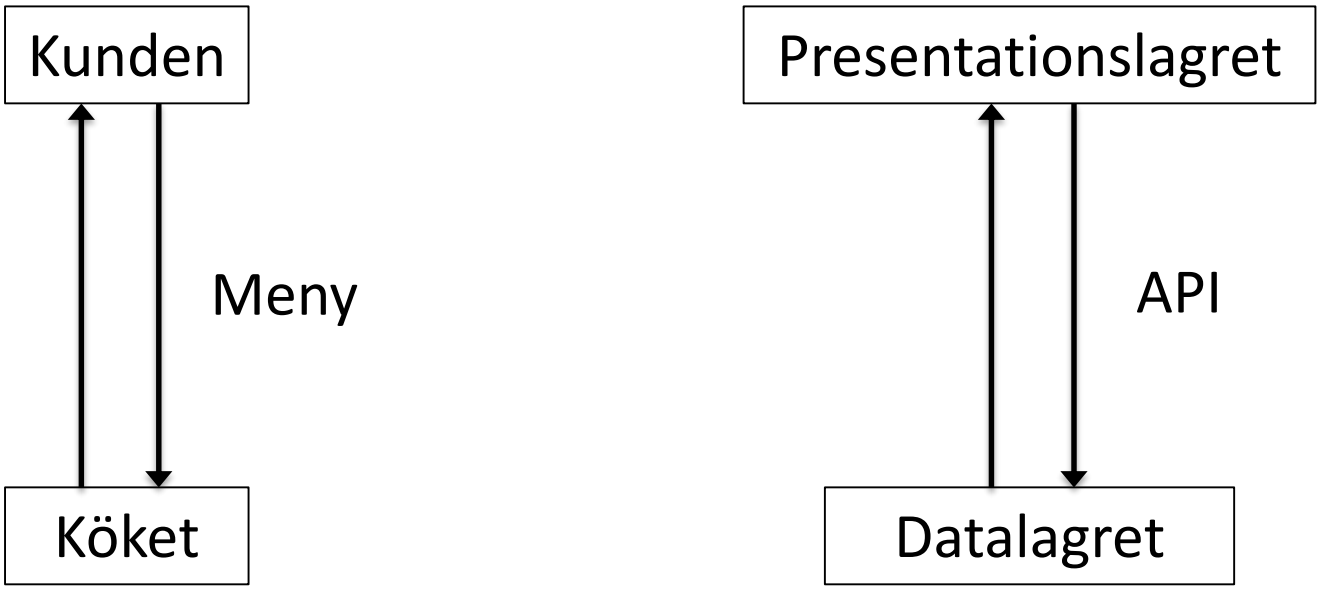
- När ett API väl är satt och börjat användas är det dyrt att ändra
- Tänk om alla restaurangkunder hade fått en kopia av menyn och ringde in sin beställning (”En nr 17 tack!”)
- Om menyn ändras fungerar inga gamla kunders beställning längre
  - Varje gammal kund måste ”uppdateras”

- Tung övergångsperiod från gammalt till nytt API
  - Både gammalt och nytt API måste stödjas

Windows-API

- Microsoft Windows har ett API som alla windowsprogram använder för att visa sina fönster
- Så länge en ny version av Windows behåller samma API kommer gamla program att fortsätta fungera (bakåtkompatibilitet)
- Om en ny version av Windows byter API måste alla gamla program skrivas om!

API till datamodulen



API till datamodulen

- Datamodulen har ett API bestående av 6 välspecificerade funktioner som presentationsmodulen får använda för att komma åt data
- Så länge API:t (specifikationen av funktionerna) inte ändras kan vilken presentationsmodul användas på vilken datamodul som helst
- API:ets specifikation finns på kurshemsidan

API till datalagret

- `load(filename)`
  - Läser en JSON-fil, returnerar ett databasobjekt
- `get_project_count(db)`
  - Hämtar antalet projekt i databasobjektet
- `get_project(db, id)`
  - Hämtar projektet *id* från databasobjektet
- `get_techniques(db)`
  - Hämtar en sorterad lista med alla tekniker
- `get_technique_stats(db)`
  - Hämtar statistik över de tekniker som används

Sök i datalagret

-
```
search( db,
        sort_by='start_date’,
        sort_order='desc’,
        techniques=None,
        search=None,
        search_fields=None)
```

Testa datalagret

- Enhetstestning
  - Testar en specifik, avgränsad del av koden i taget
  - Testerna ska kunna köras igen och igen -> testfil
  - Testbibliotek för att underlätta körandet och sammanställa resultat -> Pythons unittest-modul
- Alla API:ets 6 funktioner ska testas noggrant
- Ni får en grundläggande testfil
- Klassen ska tillsammans förbättra testfilen
- Krav för datalagret att alla tester blir godkända

1. Kursinformation
2. Dokument
3. Projektplan
4. Frågor
5. Genomgång av projekt
6. Vad är ett API?
7. **Kom ihåg**

LINKÖPINGS UNIVERSITET

Kom ihåg

- Anmäl er i Webreg (sista dag idag)!
- Göra klart Gruppkontraktet (deadline på torsdag)
- Kolla på planeringen av projektet (projektplaneringen)
- Skriv dagbok
- Börja kolla på installationsmanualen
- Skissa på er LoFi-prototyp

Emma Enocksson Svensson

www.liu.se

LINKÖPINGS UNIVERSITET