



Gruppetime

Uke 4

Camilla & Lars



Plan for dagen

- ★ Gjennomgang Oblig 1
- ★ Datahenting/API
- ★ ViewModel og Arkitektur
- ★ Oblig2/Lab

PSA: Utveksling

- ★ Frist 15.feb (Høst 2024)
- ★ [Avtaler ved IFI](#)
- ★ [Avtaler ved MatNat](#)
- ★ Kan trekke søknaden senere!
- ★ studieinfo[a]mn.uio.no



Technische
Universität
München



Oblig 1 - Takeaways

- ★ Navigasjon med NavHost + navController
- ★ Kjøring av enhetstester
- ★ Bruk av @Preview
- ★ Bruk av remember
- ★ Pakker
- ★ LocalSoftwareKeyboardController
- ★ ExposedDropDownMenuBox
- ★ Kodesplitting + kodeprinsipper - Ikke legg alt i MainActivity!

Mappestruktur

```

└─ app
  └─ manifests
  └─ kotlin+java
    └─ no.uio.no.in2000.sondrbw.oblig_1
      └─ ui
        └─ palindrome
          └─ PalindromeScreen.kt
        └─ theme
        └─ unitconverter
          └─ UnitConverterScreen.kt
          └─ Converter.kt
          └─ ConverterUnits
          └─ MainActivity
          └─ Palindrome.kt

```

Datahenting/API

Hvordan henger alt sammen?

HTTP/HTTPS:

HyperText Transfer Protocol /
HyperText Transfer Protocol Secure

Protokollen som tillater
kommunikasjon mellom klienter
(f.eks. nettlesere, telefonen) og
webservere

En måte å overføre data på over
internettet

Både hente og sende data

Vi bruker vanligvis port 80

Vi bruker en HTTP-klient for å
hjelp oss å hente data (GET)

API:

Application programming
Interface

Grensesnitt for en plattform eller
tjeneste

Single Point of Entry – en måte å
aksessere på

Gjør at vi enkelt kan hente data
fra “noen andre”.

Mange APIer er åpne for alle

Vi sender en forespørsel og bryr
oss ikke om hvordan den
håndterer spørringen eller henter
dataen fra databasen - bare vi får
det vi ønsker

REST:

Representational State Transfer

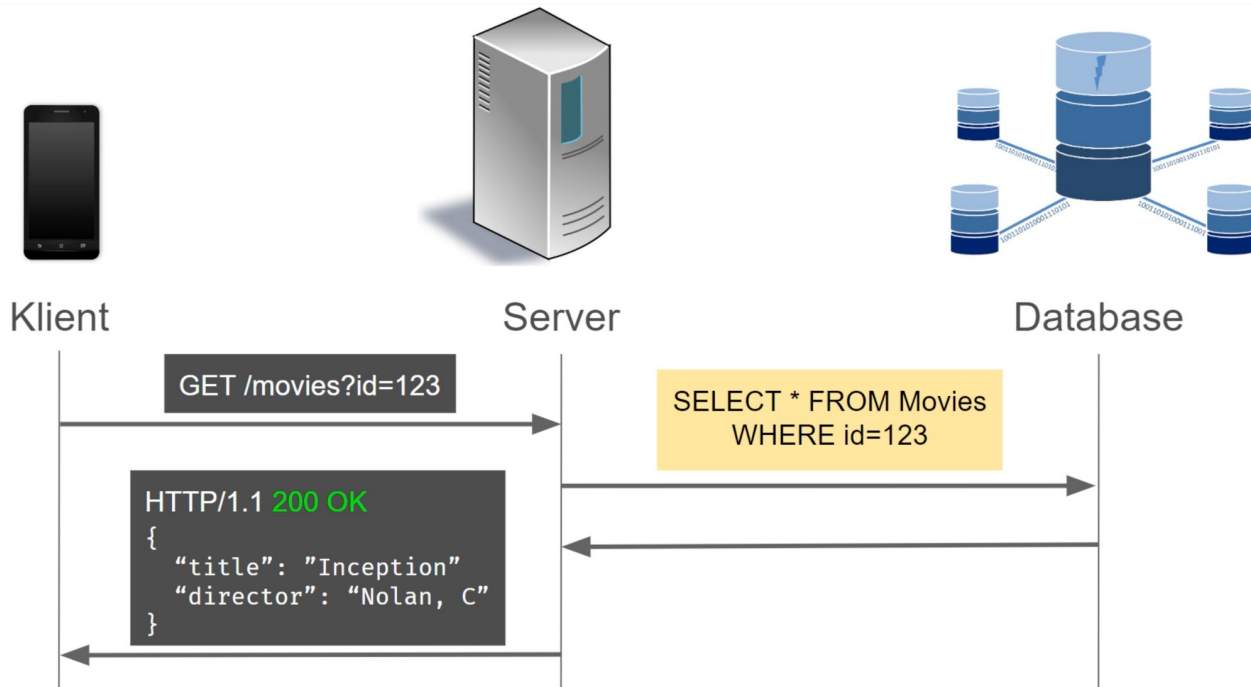
Ikke protokoll, men arkitektur

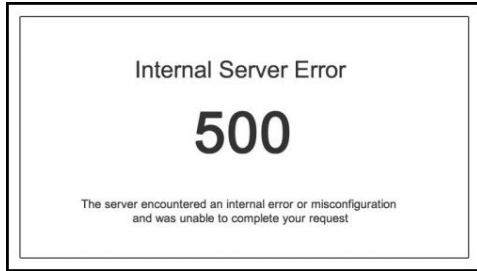
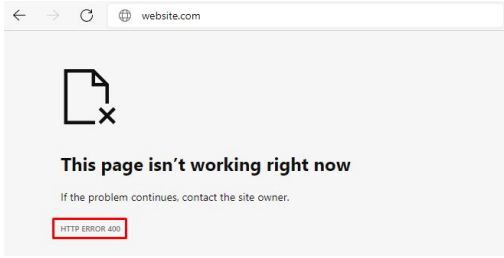
Enda en standardisert måte å hente
og sende data på, men dette
bestemmer selve grensesnittet til
API-et

Et RestAPI består av ressurser som
er adressert med en URI (unik
identifikator for hver ressurs), og
hver ressurs er representert med et
dataformat (f.eks. JSON/XML)

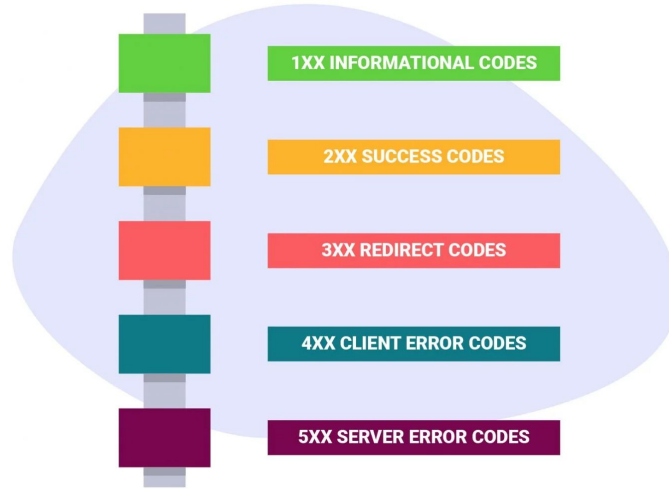
Bruker HTTP for å få tak i ressursen

GET request





HTTP Status Codes



200 OK.

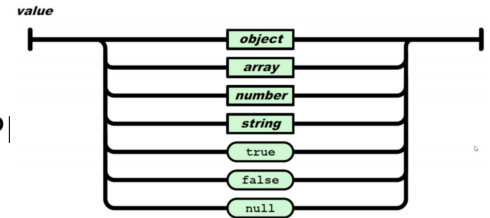


302 Found.

Hvordan henter vi dataen?

- JSON
 - JavaScript Object Notation
 - data i tekstformat
 - en måte å strukturere og organisere data på
 - samling av nøkler og verdier som er enkelt å lese og forstå
 - støttes av mange forskjellige programmeringsspråk
- Postman eller Restman
 - programvareverktøy som gjør det enklere å teste og utforske API
 - kan legge inn ulike parametere og data for å se hvordan API-er

```
{  
  "Students": [  
    {  
      "Name": "Amit Goenka",  
      "Major": "Physics"  
    },  
    {  
      "Name": "Smita Pallod",  
      "Major": "Chemistry"  
    },  
    {  
      "Name": "Rajeev Sen",  
      "Major": "Mathematics"  
    }  
  ]  
}
```



Pass på ved henting

Objekt



```
{  
  "hilsen1": "Hei",  
  "hilsen2": "På",  
  "hilsen3": "Deg"  
}
```

Array



```
[  
  "Hei",  
  "På",  
  "Deg"  
]
```

Arkitektur / MVVM

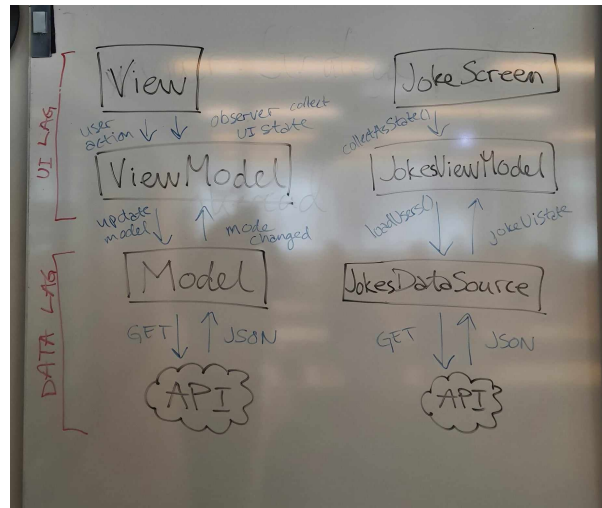
Arkitektur

MVVM - Model, View, View Model.

- Er et strukturelt designmønster for logisk oppbygging mellom Datalaget og UI-laget
- Er standard arkitektur i android apper

MVVM (og designmønstre generelt) fører til god design; bla. lav kobling og høy kohesjon(gjør det lettere å vedlikeholde, utvide og endre kode) DETTE LIKER VI!

- Vi får høy kohesjon ved å skille ulike ansvarsoppgaver i forskjellige klasser/moduler.
- Vi får lav kobling ved at klasser/moduler med ulike ansvarsoppgaver har enkle grensesnitt mellom seg. (er ikke avhengige av mange andre klasser)



View: Observerer UIstate/data i viewmodel og presenter den + Muliggjør brukerinteraksjon

ViewModel: Oppdaterer UI state hvis dataen i model endrer seg. Sier at model skal oppdatere dataen sin (ved f.eks. brukerinteraksjon fra view / eller kall på funksjon som henter ny data)

Model: Har ansvar for å hente, behandle og representere data til viewmodel.

Lab

Jobb med ukesoppgaver eller oblig! :)

camilltk@uio.no

larshaas@uio.no