Velkommen til gruppetime i IN2000!

Gruppe 1 med Ida, Jamila & Jørgen

Viktige beskjeder!

- Underkjent oblig? :(
- Dobbeltsjekk feedback og status + evt. ny frist

Oppsummering oblig 1

Det viktigste

- Navigasjon med NavHost + navController
- Kjøring av enhetstester
- Bruk av @Preview
- Bruk av remember
- LocalSoftwareKeyboardController
- ExposedDropdownMenuBox
- Pakker
- Kodesplitting + kodeprinsipper

Figma / Material Theme

Material 3 / Figma revansj

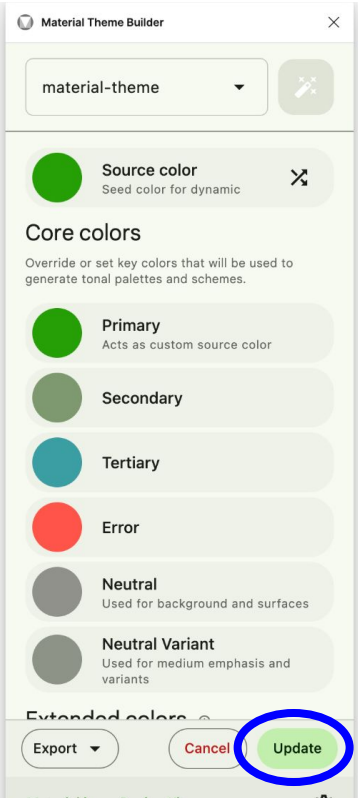
Jeg prøver meg igjen!

Nå har jeg satt meg inn i hva problemet var sist, og siden jeg ikke får deltatt i flere gruppetimer lager jeg slides som Jørgen og Jamila må klare å formulere for meg :)

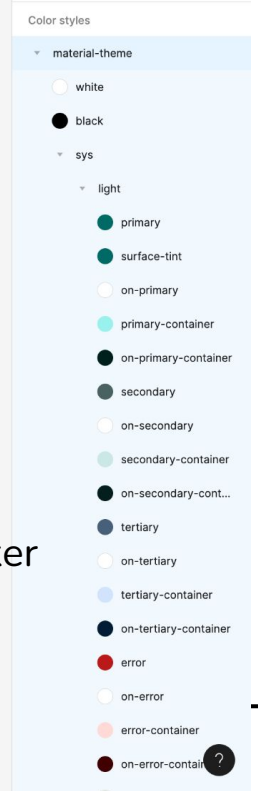
Håper dette gir mer mening, og er mer oversiktlig enn da jeg prøvde meg live. Nok en gang har jeg innsett at det sjeldent er en god idé...

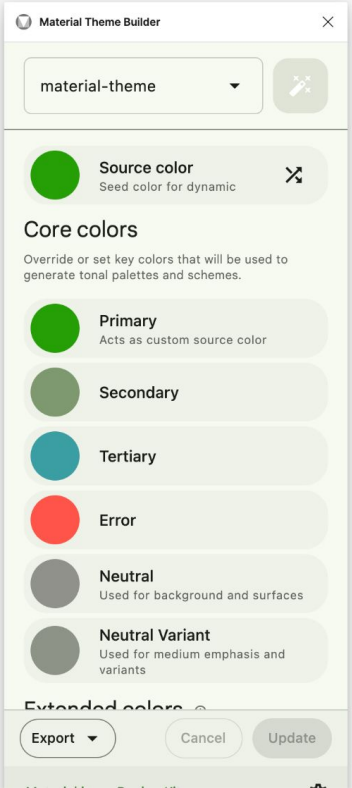
Lage tema i Figma

Temaet du får laget ved hjelp av “Material Theme Builder” lagres i Figma, som gjør det lett å bruke fargene når man skal designe!

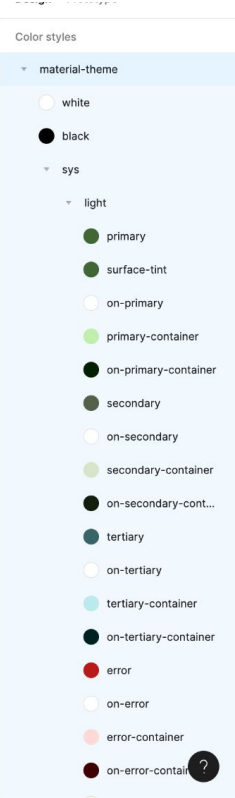


Når du har valgt/fått nytt fargepalett trykker du update



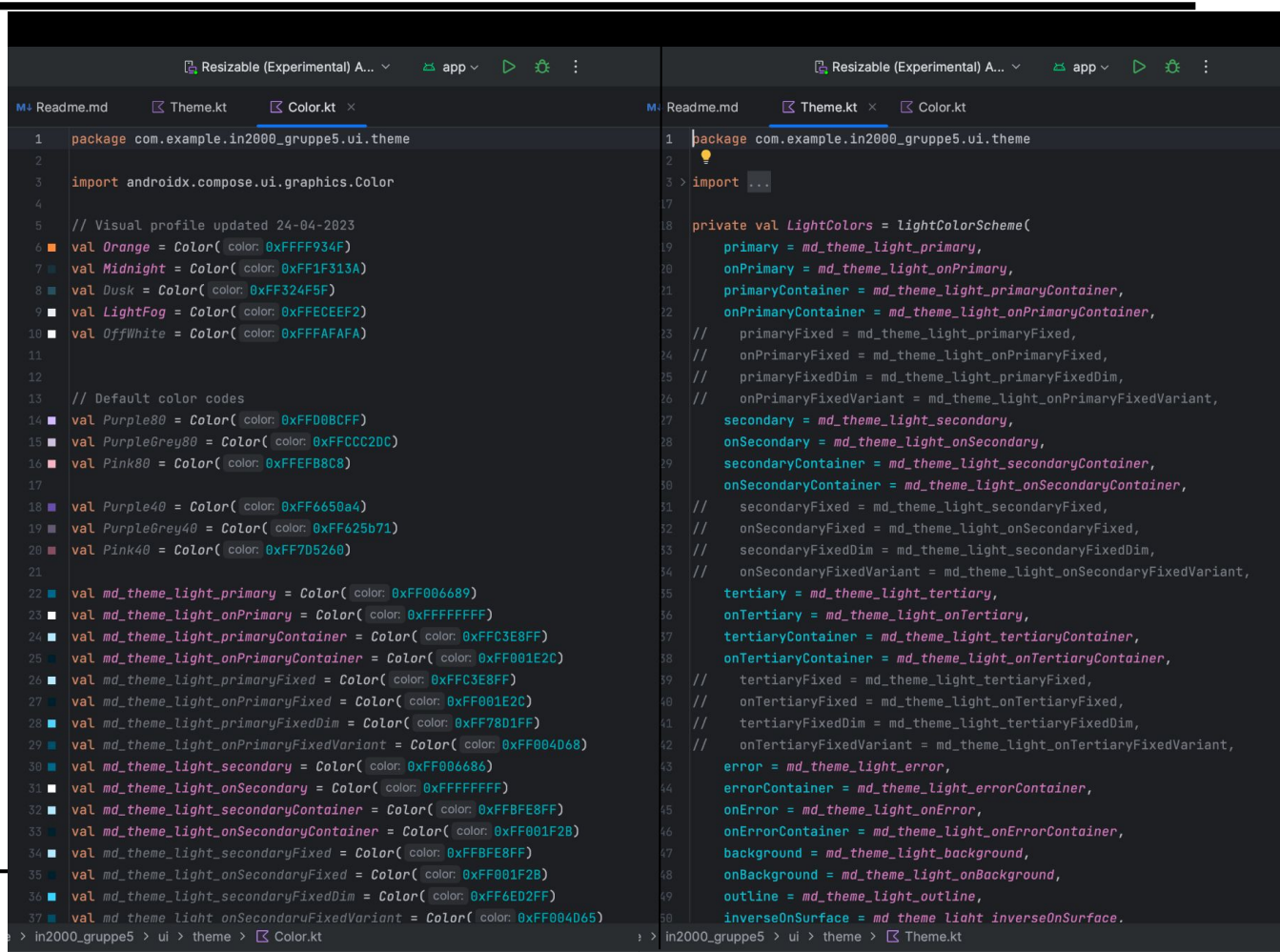


Og fargene i “material-theme” oppdaterer seg!



Laste inn i Android Studio

Kodefilene man får eksportert fra figma er ulike fra hva Android oppretter automatisk.. Dette gjør det litt kløynete å skulle legge over første gangen (som dere så sist), men her er hvordan det så ut i våre filer i prosjektet vårt i fjor



Eksempel på hvordan man tar det i bruk

```
val costStrings = stringArrayResource(id = R.array.cost_strings)

val screenConfig = LocalConfiguration.current
val scope = rememberCoroutineScope()
val bottomSheetState =
    rememberModalBottomSheetState(initialValue = ModalBottomSheetValue.Hidden)



ModalBottomSheetLayout(
    sheetState = bottomSheetState,
    sheetContent = { AppDrawer( header: "Hva koster det å...", costStrings[0]) },
    sheetBackgroundColor = Midnight,
    sheetShape = RoundedCornerShape( percent: 5)
) {
    Scaffold(
        topBar = {
            TopAppBar(
```

Resizable (Experimental) A...

Readme.md | Theme.kt | Color.kt

```
1  package com.example.in2000_gruppe5.ui.theme
2
3  import androidx.compose.ui.graphics.Color
4
5  // Visual profile updated 24-04-2023
6  val Orange = Color( color: 0xFFFF934F)
7  val Midnight = Color( color: 0xFF1F313A)
8  val Dusk = Color( color: 0xFF324F5F)
9  val LightFog = Color( color: 0xFFECEEF2)
10 val OffWhite = Color( color: 0xFFFAFAFA)
11
```

```
Column(
    modifier = Modifier.weight(4f),
) { this: ColumnScope
    Text(
        text = stringResource(id = "Hva koster det å...?"),
        style = Typography.headlineMedium
    )
}
```

Material 3 / Figma remote revansj - oppsummering

Det er altså ikke så farlig at det ikke er helt likt oppsett på filene.

Dersom dere ønsker at appen skal være 100% kompatibel med

Tips:

Datahenting / API

Hvordan henger alt sammen?

HTTP/HTTPS:

HyperText Tansfer Protocol / HyperText Tansfer 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 post 80

Vi bruker en HTTP-klient for å hjelpe 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:

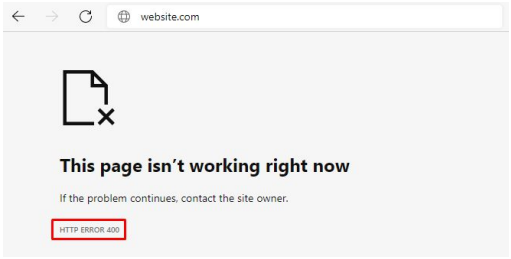
Representational State Transfer

Ikke protokoll, men arkitektur

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

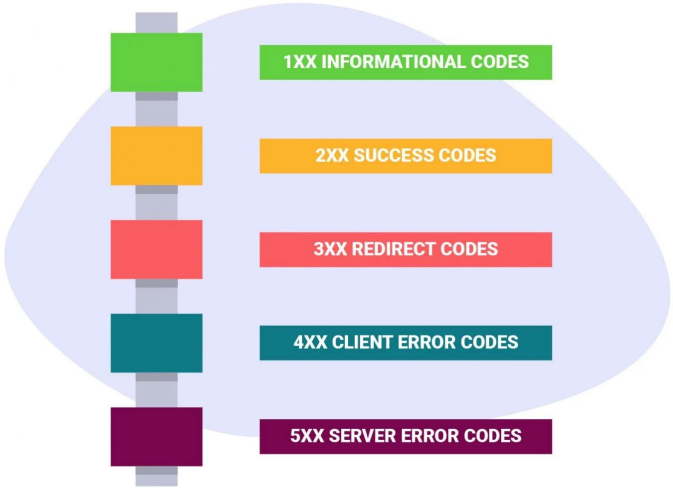
Et RestAPI består av ressurser som er adressert med en URI (unik identifikator for hver ressurs), og hver ressurs er representert med et dataformat (f.eks. JSON/XML)

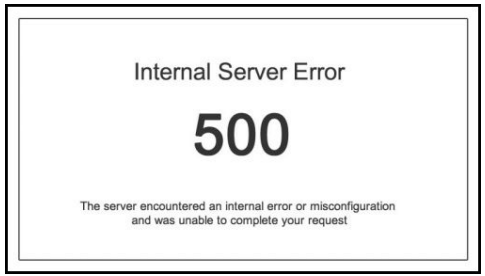
Bruker HTTP protokollen for å få tak i ressursen



200 OK.

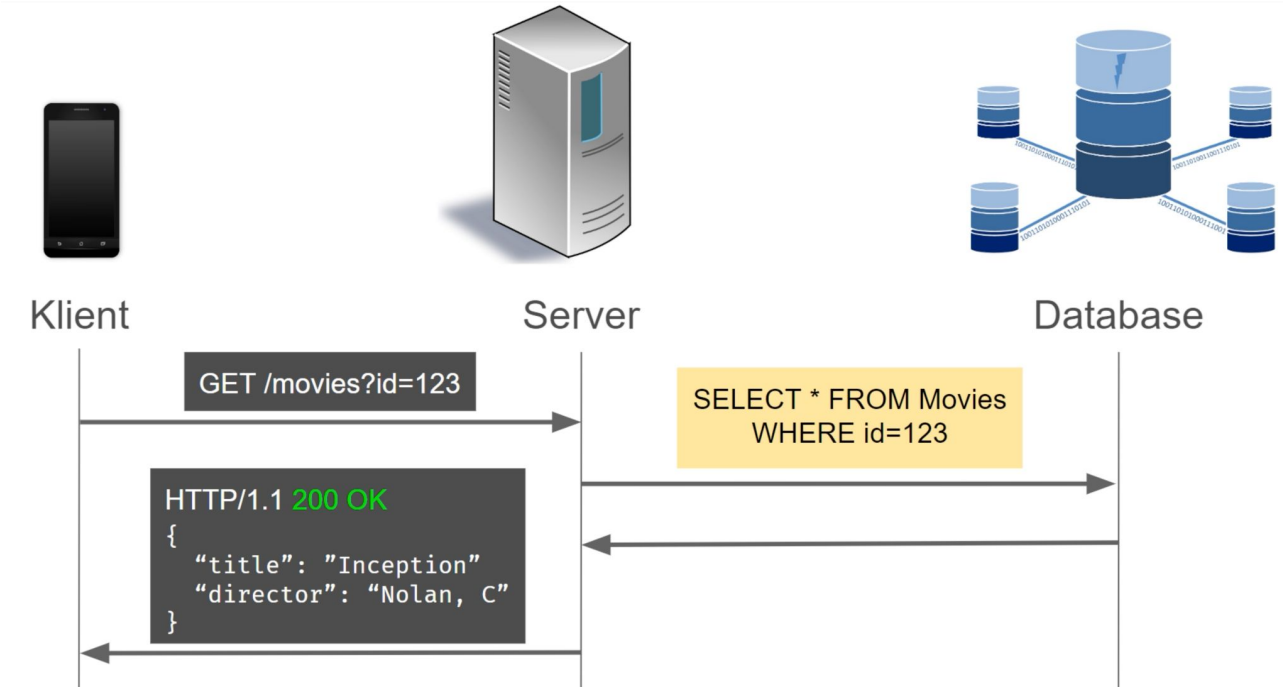
HTTP Status Codes





302 Found.

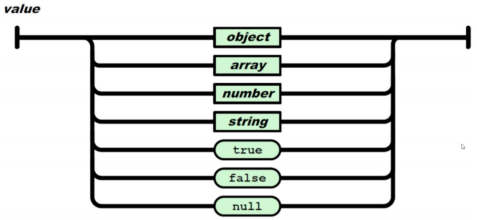
GET request


Klient
Server
Database
GET /movies?id=123
SELECT * FROM Movies
WHERE id=123
HTTP/1.1 200 OK
{
"title": "Inception"
"director": "Nolan, C"
}

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

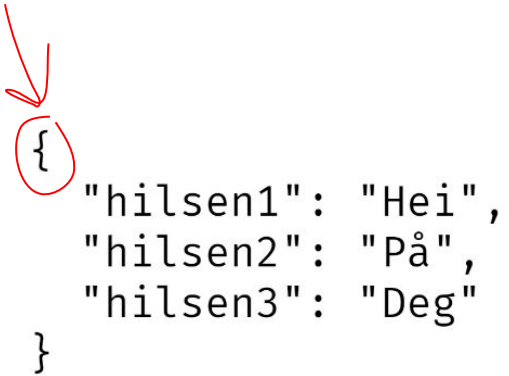
```
{
  "Students": [
    {
      "Name": "Amit Goenka",
      "Major": "Physics"
    },
    {
      "Name": "Smita Pallod",
      "Major": "Chemistry"
    },
    {
      "Name": "Rajeev Sen",
      "Major": "Mathematics"
    }
  ]
}
```



- Postman eller Restman
  - programvareverktøy som gjør det enklere å teste og utforske API-er
  - kan legge inn ulike parametere og data for å se hvordan API-er

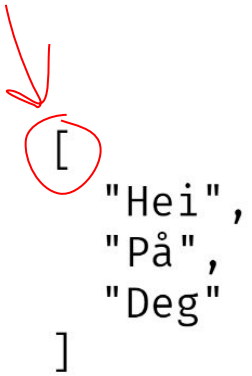
Pass på ved henting

Objekt



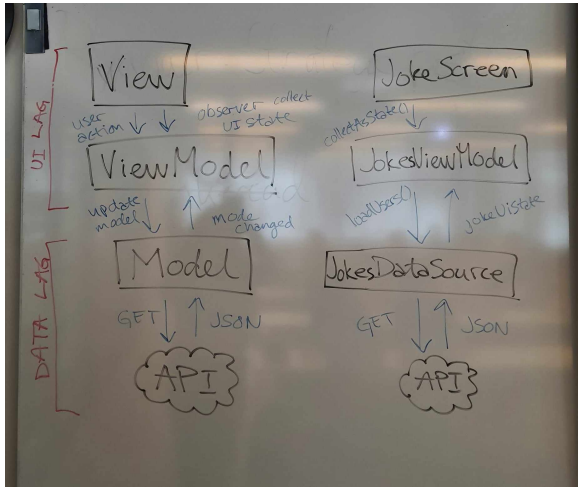
```
{
  "hilsen1": "Hei",
  "hilsen2": "På",
  "hilsen3": "Deg"
}
```

Array



```
[
  "Hei",
  "På",
  "Deg"
]
```

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.

Spørsmål om oblig 2?

Send oss en mail:

idakene@uio.no
jamilakm@uio.no
jorgenao@uio.no