



Guest lecture from ZIP-Q

Barcode, networking and JSON

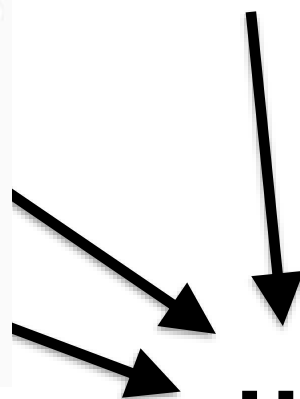
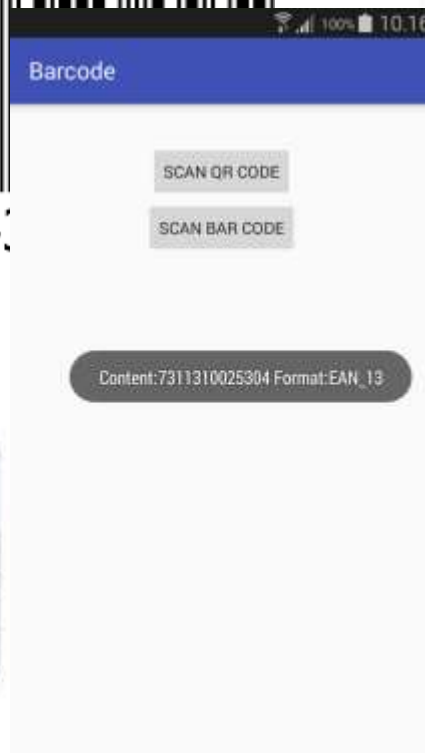
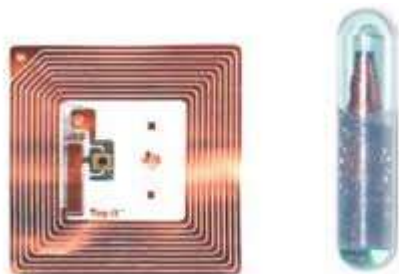
Second Mandatory Assignment

Exercises

# Barcodes



3912345




**unique id**

# Getting product info with the barcode



**Outpan**   [Home](#) [Profile](#)

 7311310025250  
**Santa Maria Malen Koriander 30g**

---

**Images and Videos**

---

No photo or video added yet.

**Attributes**

<b>Brand</b>	Santa Maria
<b>Net Weight</b>	30 g
<b>Title</b>	Malen Koriander
<b>Amount</b>	30g

Shortcut:

[https://www.outpan.com/view\\_product.php?barcode=7311310025250](https://www.outpan.com/view_product.php?barcode=7311310025250)

# Http communication from Android



Similar to Java,  
but must be done in a background thread

Chapter 23 in the textbook

```
public class NetworkFetcher {  
    public byte[] getProductInfo(String urlSpec) throws IOException {  
        URL url = new URL(urlSpec);  
        HttpURLConnection connection = (HttpURLConnection)url.openConnection();  
        ...  
        return out.toByteArray();  
    } finally { connection.disconnect(); }  
}
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

# Getting product info



```
private class FetchOutpanTask extends AsyncTask<String, Void, byte[] > {
    @Override
    protected byte[] doInBackground(String... params) {

        try {
            result = new NetworkFetcher()
                .getProductInfo("https://www.outpan.com/ ... ");
        } catch (IOException ioe) {... }
        return result;
    }

    @Override
    protected void onPostExecute(byte[] result) {
        ...
    }
}
```

# Getting product info



```
private class FetchOutpanTask extends AsyncTask<String, Void, byte[] > {
    @Override
    protected byte[] doInBackground(String... params) {

        try {
            result = new NetworkFetcher()
                        .getProductInfo("https://www.outpan.com/ ... ");
        } catch (IOException ioe) {... }
        return result;
    }

    @Override
    protected void onPostExecute(byte[] result) {
        ...
    }
}
```

# Robust networking



## 1. URL wrong, not responding, ...

```
try {  
    ... getUrlBytes("https:// ...");  
} catch (IOException e) {  
    return "Unable to retrieve web page. URL may be invalid.";  
}
```

## 2. Missing Network connection

```
ConnectivityManager connMgr = (ConnectivityManager)  
    getSystemService(Context.CONNECTIVITY_SERVICE);  
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
if (networkInfo != null && networkInfo.isConnected()) {  
    // fetch data  
} else {  
    // display error  
}
```

<http://developer.android.com/reference/android/content/Context.html>

# Robust networking (2)



## 3. Network slow

```
URLConnection conn= (URLConnection)url.openConnection();
conn.setReadTimeout(10000 /* milliseconds */);
conn.setConnectTimeout(15000 /* milliseconds */);
conn.connect();
int response = conn.getResponseCode();

is = conn.getInputStream();
...
```

Read these carefully:

- <http://developer.android.com/training/basics/network-ops/connecting.html>
- <http://developer.android.com/training/basics/network-ops/managing.html>



# Getting product info



Professional service available from GS1

<http://gepir.gs1.org/v32/xx/gtin.aspx>

Free service available from outpan

<http://www.outpan.com>

[https://www.outpan.com/  
view\\_product.php?barcode=7311310025250](https://www.outpan.com/view_product.php?barcode=7311310025250)



# Outpan

## Product information platform

Outpan is a free service that allows you to share product information with your supply chain network.

### Login

☒ Remember Me

### Create Account

We strongly recommend you agree to our [Terms of Use](#) and [Privacy Policy](#).

## Outpan

[Home](#) [Profile](#) [Notifications](#) [Settings](#)

### Settings

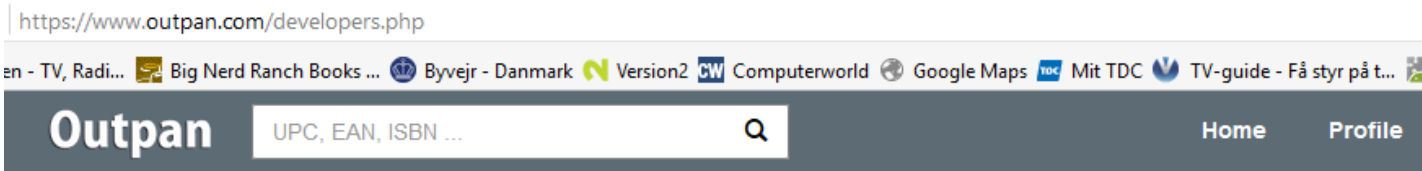
[Profile](#)  
[API Key](#)  
[Change Password](#)

### Profile

Name

About

# Using Outpan's API



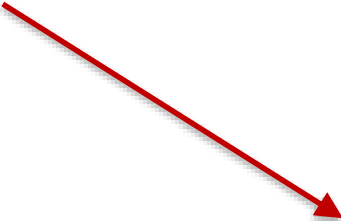
## API Documentation

### Get Product Information

GET `https://api.outpan.com/v2/products/[GTIN]?apikey=[YOUR API KEY]`

`https://api.outpan.com/v2/products/7311310025250/?apikey=0d...a31`

use your own key



```
{
  "gtin": "7311310025250",
  "outpan_url": "https://www.outpan.com/view_product.php?barcode=7311310025250",
  "name": "Santa Maria Malen Koriander 30g",
  "attributes": {
    "Amount": "30g",
    "Brand": "Santa Maria",
    "Net Weight": "30 g",
    "Title": "Malen Koriander"
  },
  "images": [
  ],
  "videos": [
  ],
  "categories": [
  ]
}
```

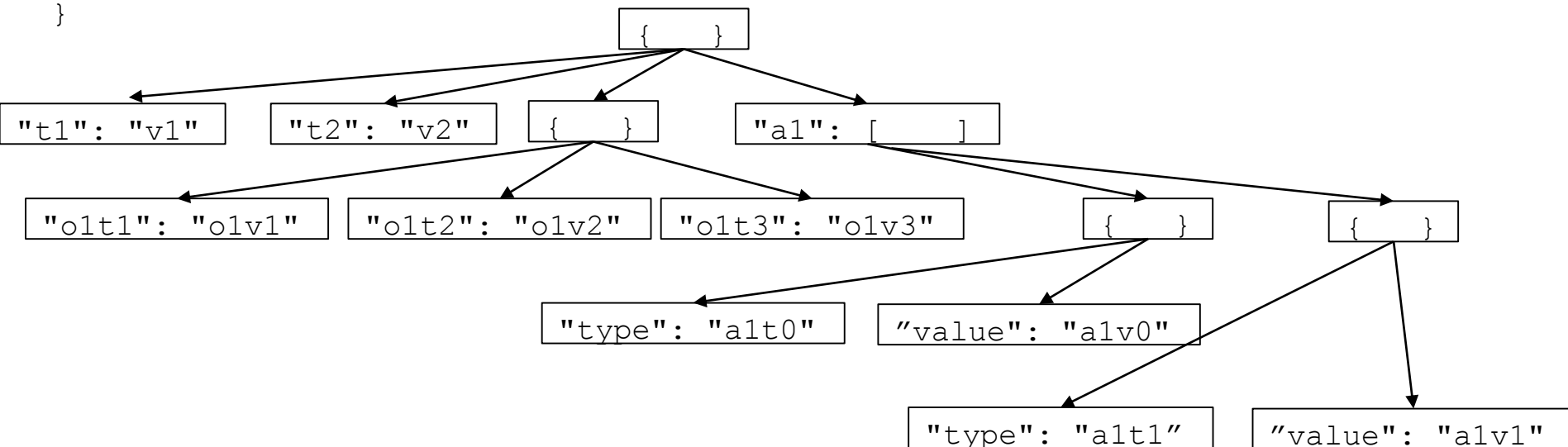


```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

# JSON Object Model



```
{ "t1": "v1",  
  "t2": "v2",  
  "o1": { "o1t1": "o1v1",  
          "o1t2": "o1v2",  
          "o1t3": "o1v3",  
        },  
  "a1": [  
    { "type": "alt0",  
      "value": "a1v0"  
    },  
    { "type": "alt1",  
      "value": "a1v1"  
    }  
  ]  
}
```



# JSON – parsing (object model)



**JSONObject JsB:**

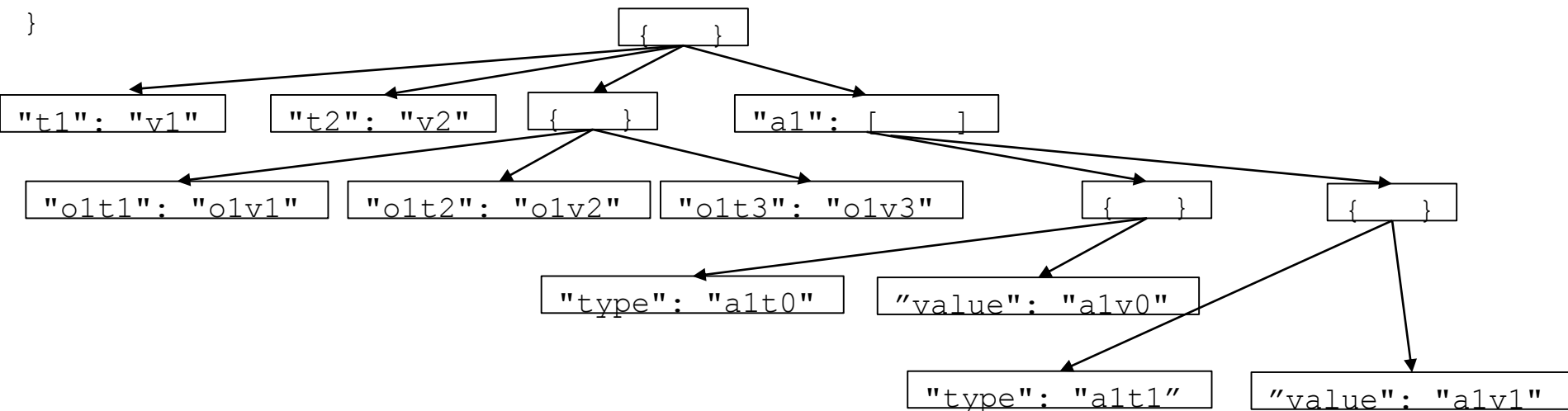
```
{ "t1": "v1",  
  "t2": "v2",  
  "o1": { "o1t1": "o1v1",  
          "o1t2": "o1v2",  
          "o1t3": "o1v3",  
        },  
  "a1": [  
    { "type": "alt0",  
      "value": "alv0"  
    },  
    { "type": "alt1",  
      "value": "alv1"  
    }  
  ]  
}
```

JsB.getString("t1") is "v1"

JsB.getJSONObject("o1") is  
{  
 "o1t1": "o1v1",  
 "o1t2": "o1v2",  
 "o1t3": "o1v3"  
}

JsB.getJSONObject("o1").getString("o1t1") is "o1v1",

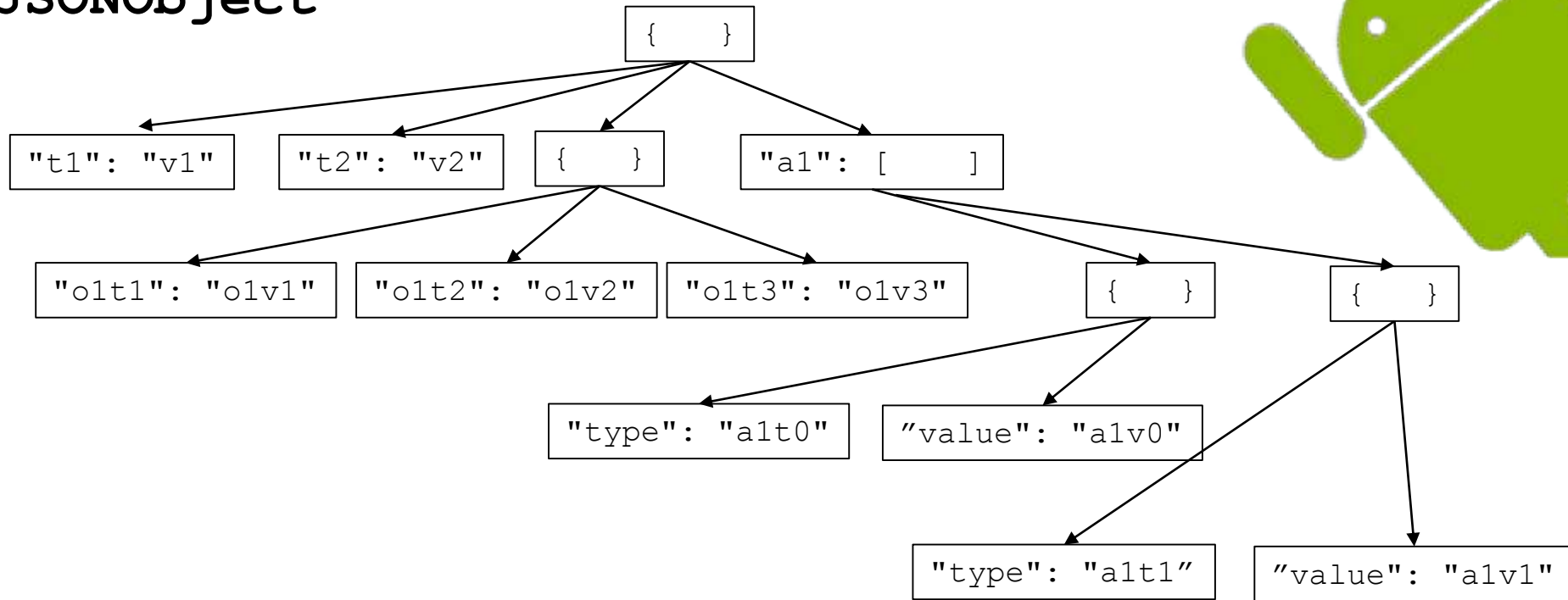
JsB.getJSONArray("a1")  
.getJSONObject(0).getString("type") is "alt0"



# JSON Object versus Streaming model



## JSONObject



## JsonReader

<http://developer.android.com/reference/android/util/JsonReader.html>

```
{"t1": "v1", "t2": "v2", "o1": {"o1t1": "o1v1", "o1t2": "o1v2", "o1t3": "o1v3", },  
  "a1": [{"type": "a1t0", "value": "a1v0"}, {"type": "a1t1", "value": "a1v1"}]}
```



... `nextName().equals("t2")` ...

# Using Outpan's API



https://www.outpan.com/developers.php

en - TV, Radi... Big Nerd Ranch Books ... Byvej - Danmark Version2 CW Computerworld Google Maps Mit TDC TV-guide - Få styr på t...

Outpan

UPC, EAN, ISBN ...



Home

Profile

## API Documentation

### Get Product Information

GET https://api.outpan.com/v2/products/[GTIN]?apikey=[YOUR API KEY]

https://api.outpan.com/v2/products/7311310025250/?apikey=0d...a31

use your own key



```
{
  "gtin": "7311310025250",
  "outpan_url": "https://www.outpan.com/view_product.php?barcode=7311310025250",
  "name": "Santa Maria Malen Koriander 30g",
  "attributes": {
    "Amount": "30g",
    "Brand": "Santa Maria",
    "Net Weight": "30 g",
    "Title": "Malen Koriander"
  },
  "images": [
  ],
  "videos": [
  ],
  "categories": [
  ]
}
```



# Copenhagen City Open Data



The screenshot shows the Copenhagen City Open Data website. The browser address bar displays 'data.kk.dk'. The website header includes the 'COPENHAGEN DATA' logo, navigation links for 'Datasæt', 'Organisationer', 'Grupper', and 'Om', and a search bar. A large search box in the center is titled 'Søg Data' and contains the text 'fx. Cykel / Biler'. Below the search box are 'Populære tags' (Popular tags) including 'WEEE', 'parkering', 'p-plads', 'område', 'vej', 'unge', 'trafik', 'børn', 'alder', and 'Klima'. The background of the website features an aerial view of Copenhagen with a network of lines connecting various points. Below the search area, there are two sections: 'Åbne data fra København' (Open data from Copenhagen) and 'Open Data København statistik' (Open Data Copenhagen statistics).

**Åbne data fra København**

På data.kk.dk kan du finde en række datasæt fra Københavns Kommune. Det er eksempelvis trafik, parkeringsforhold, byens fysiske infrastruktur, aktuelle aktiviteter i byens rum og meget andet.

**Open Data København statistik**

217	17	5
datasæt	organisationer	grupper

<http://pro.jsonlint.com/>

# Second Mandatory Assignment



**You decide on the user interface and functionality !!**

# Possible Tingle extensions



Crowdsourcing of registrations  
separating private and public things?

Efficient handling of images  
searching images?

Ordering search results

Synchronisation with server database(s)

Using location (indoor tracking)

...