

Internettechnologie: Practicum 2

Web-APIs consumeren op mobiele platformen

Academiejaar 2014–2015

dr. ir. Ruben Verborgh

Nabespreking Practicum 1

moeilijkheidsgraad?

beschikbare tijd?

kennisbijdrage?

Overzicht

REST APIs gebruiken

Ontwikkelen voor Android

Opgave

Overzicht

REST APIs gebruiken

Ontwikkelen voor Android

Opgave

Interactie Client – Web API

Client 1



GET /event/1

Accept: text/html

200 OK; Content-Type: text/html

Client 2



GET /event/1

Accept: application/json

200 OK;
Content-Type: application/json

Client 3



GET /event/1

Accept: application/json

200 OK;
Content-Type: application/json

Web API

Hypermedia drijft de applicatie



GET /event/1
Accept: application/json

Web API

```
{  "title": "test",  
    "description": "test",  
    "start": "2014-03-11T16:18:00.000Z",  
    "end": "2014-03-11T16:18:00.000Z",  
    "created_at": "2014-03-11T16:18:39.170Z",  
    "updated_at": "2014-03-11T16:18:39.170Z",  
    "url": "http://localhost:3000/events/1",  
    "index": "http://localhost:3000/events",  
    "confirmations": "http://localhost:3000/events/1/confirmations",  
    "messages": "http://localhost:3000/events/1/messages"  
}
```

Client krijgt alle nodige informatie voor volgende stappen van de Server in **één antwoord**.

Server applicatie onthoudt niets (stateless)

Hypermedia: statelessness

```
{ "confirmation": {  
    "event_id": 1,  
    "going": true,  
    "person_id": 1  
}}
```



POST <http://localhost:3000/events/1/confirmations>



Web API

302 Found;

Location: <http://localhost:3000/events/1/>

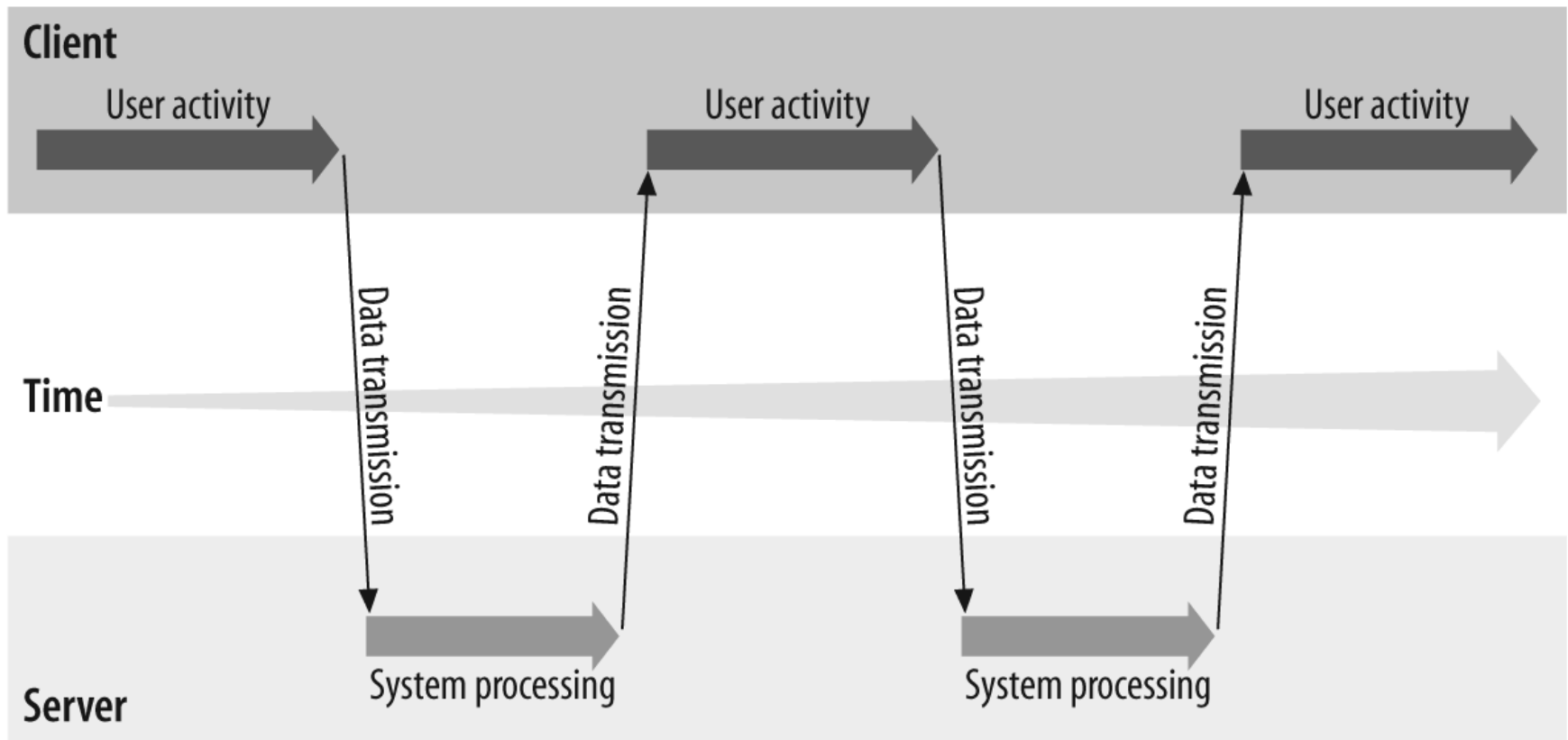
Client volgt link en zendt nodige info.

Krijgt opnieuw link voor volgende actie

Synchroon programmeren

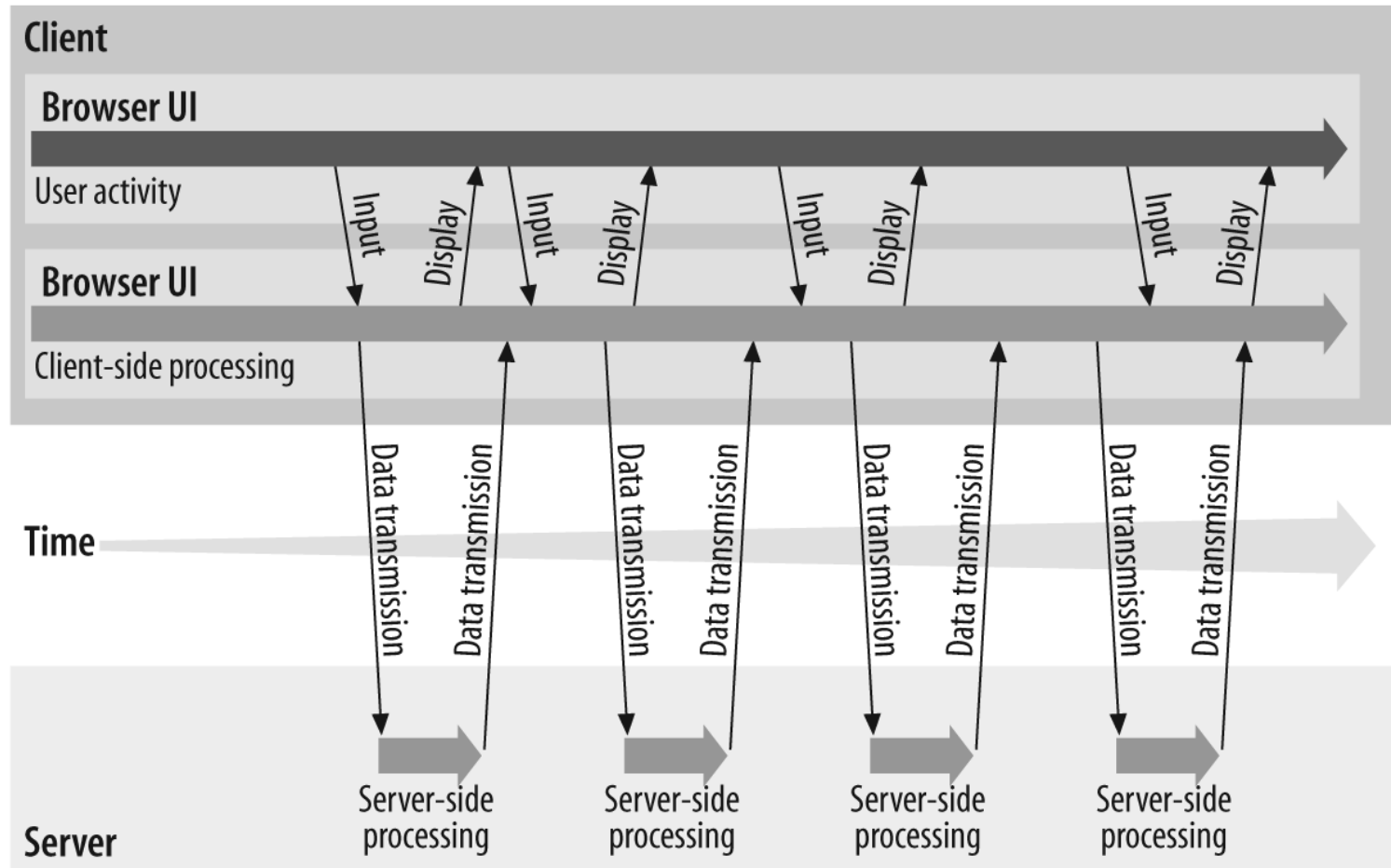
Vooral desktop applicaties

Assumptie ontwerp klassieke programmeertalen



Asynchroon programmeren

Web applicaties (AJAX) + Moderne script talen



Asynchroon programmeren

Geen zekerheden bij HTTP-requests

Komt er een antwoord?

Wanneer komt het antwoord?

Wat is het antwoord? Een fout? Een time-out?

Enige weg tot een responsieve interface

Vermijdt dat de toepassing hangt

Eventgebaseerde I/O (en meer)

Ondersteuning in alle talen voor webontwikkeling

Overzicht

REST APIs gebruiken

Ontwikkelen voor Android

Opgave

Android ontwikkelproces

Setup ontwikkelomgeving (SDK, Eclipse)

SDK: compiler, debugger, device emulator

Multiplatformondersteuning: Windows, Mac, Linux

Java-programmeertaal

heeft een eigen Java Virtual Machine en speciale byte code

Android project met java bestanden en bronbestanden

Pack project in debuggable *.apk

Installeren, uitvoeren en debuggen op emulator of apparaat

Publiceer app in Android Market

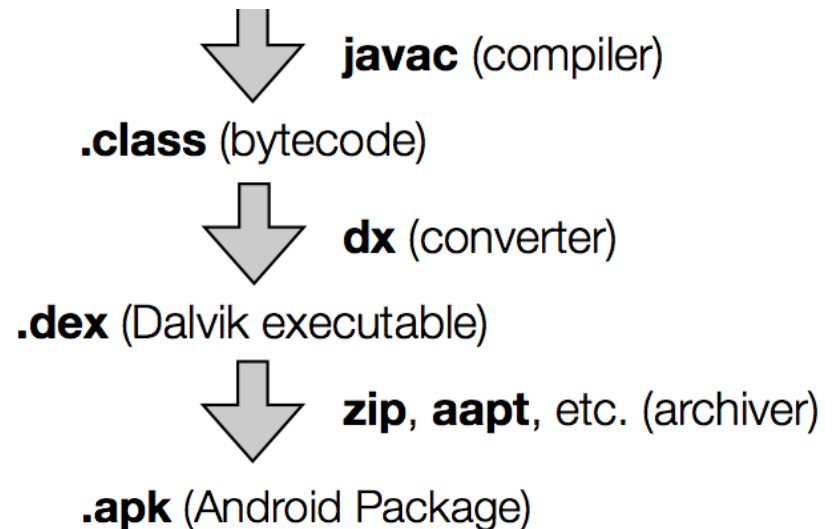
Android Application Framework

Loopt in zijn eigen virtuele machine & proces

→ Isolatie tussen apps

Bestaat uit basiscomponenten

Componenten kunnen worden geactiveerd wanneer een van de componenten dient te worden uitgevoerd



Basisconcepten voor Android-toepassingen

Activity

een specifieke gebruikerstaak

vb.: een lijst van personen, detail van een persoon

Layout

concrete weergave van een activity

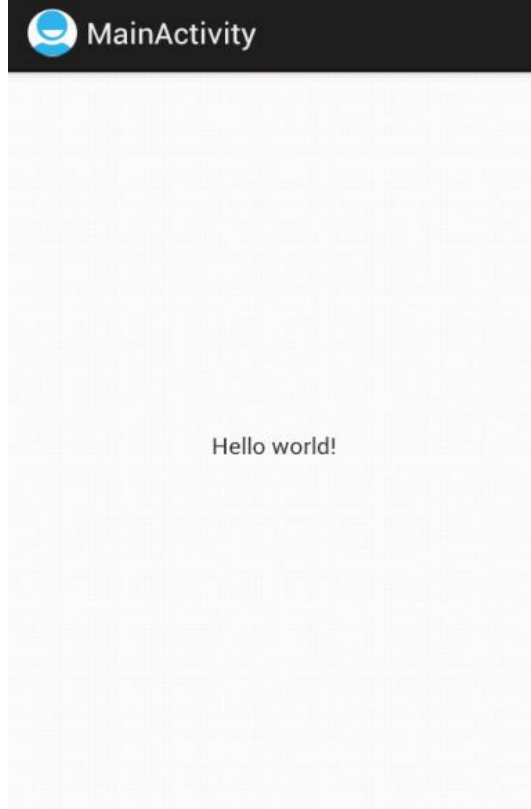
View

besturingselement in een layout

Activity

```
MainActivity.java
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

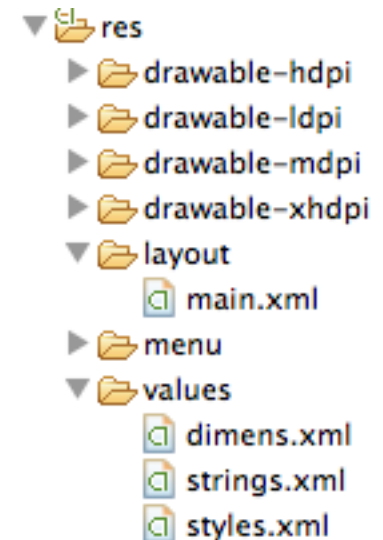


Layouts

Bepalen hoe Views worden ingedeeld: LinearLayout, TableLayout, RelativeLayout

Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</LinearLayout>
```



Views

Views zijn de bouwstenen van de UI

Twee types:

Leaf: TextView, EditText, Button, Form, TimePicker, ListView

Composite (ViewGroup): LinearLayout, RelativeLayout, ...

View gedefinieerd in XML

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
}
```

```
...  
TextView myTextView =
```

```
(TextView) findViewById(R.id.myTextView);
```

main.xml

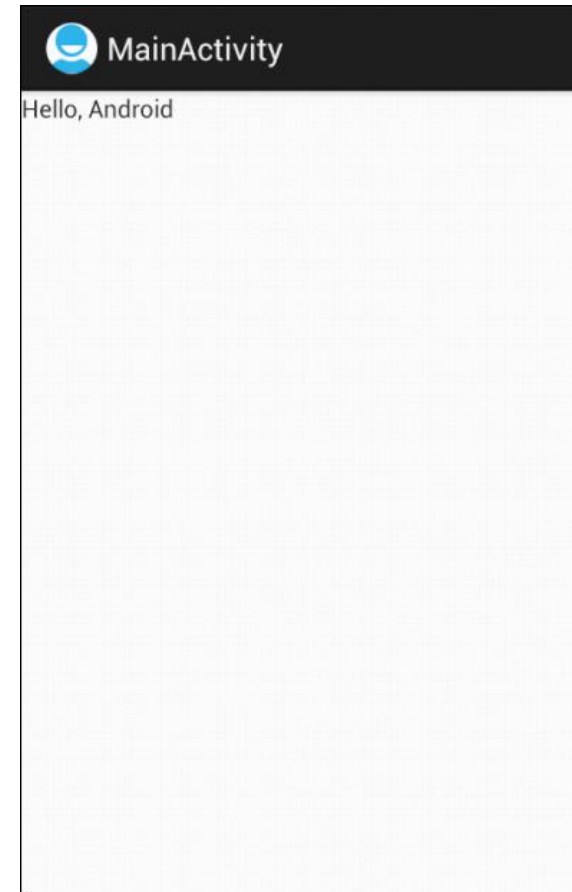
```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.  
com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
    <TextView  
        android:id="@+id/myTextView"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="Hello World,  
HelloWorld"  
    />  
</LinearLayout>
```

View gedefinieerd in Java

Views kunnen ook in code aangemaakt worden

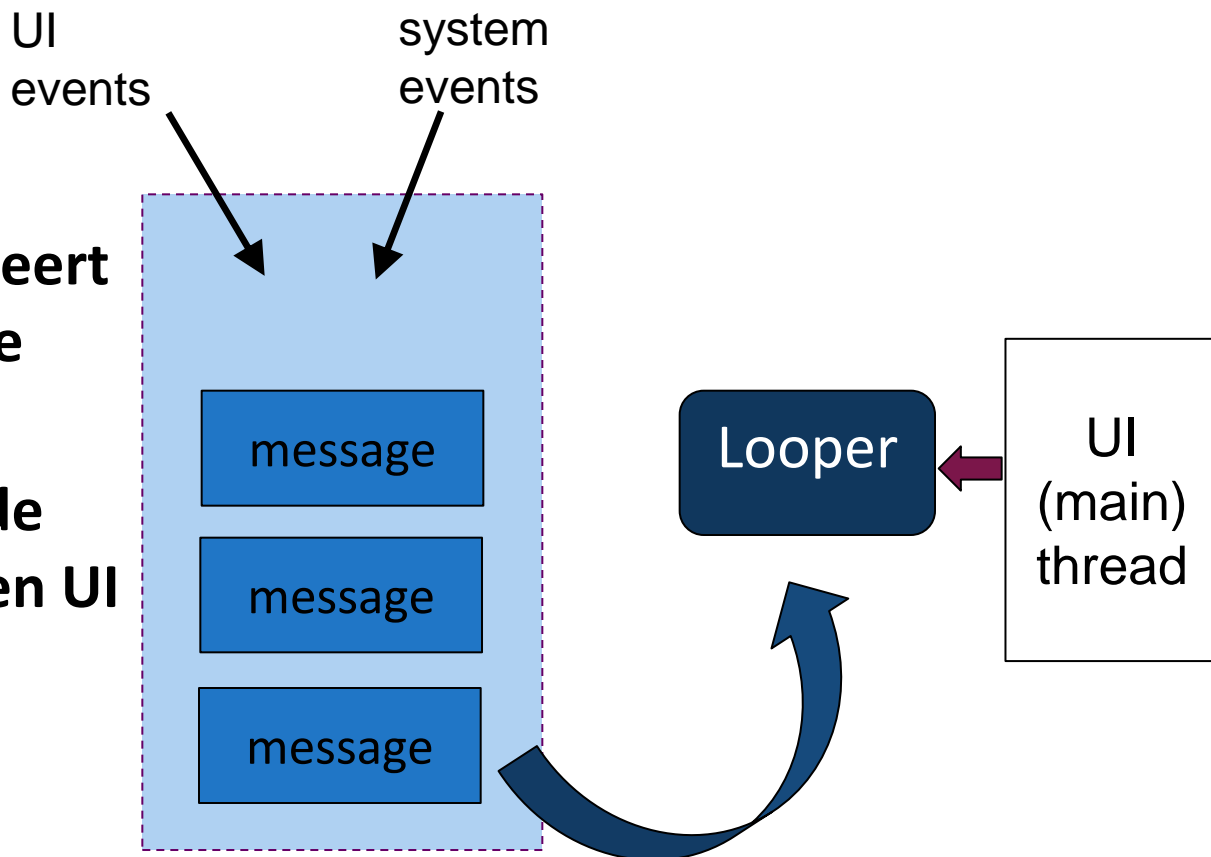
MainActivity.java

```
public class MainActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        //setContentView(R.layout.main);  
        TextView tv = new TextView(this);  
        tv.setText("Hello, Android");  
        setContentView(tv);  
    }  
}
```



Event Handler en Responsiviteit

**Event handler blokkeert gebeurtenissen in de *message* wachtrij
=> Langzaam lopende handler leidt tot geen UI reactie**



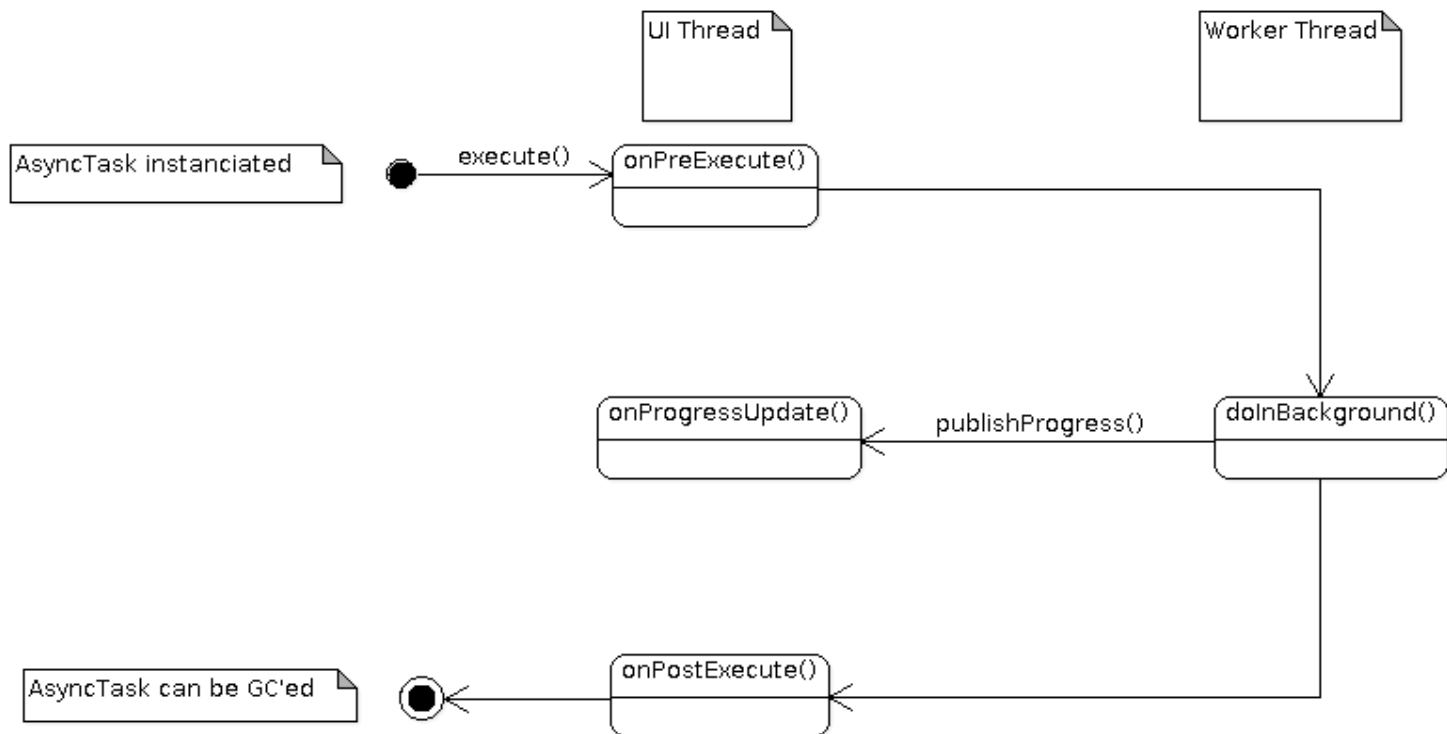
<http://developer.android.com/guide/practices/responsiveness.html>

AsyncTask klasse

```
abstract class AsyncTask {  
    doInBackground(String... params) {  
        return null;  
    }  
  
    protected void onPreExecute() {  
    }  
}
```

```
private class ListEventsTask  
    extends AsyncTask<Void, Void, Void> {  
    @Override  
    protected Void doInBackground(Void... args) {  
        //Doe HTTP request hier  
        return null;  
    }  
    @Override  
    protected void onPostExecute(Void arg) {  
    }  
}
```

AsyncTask : levenscyclus



AsyncTask : kenmerken

Eenvoudig te implementeren, eenvoudige taken

Een `AsyncTask` hangt vast aan de `Activity`

Wisselen van App pauseert de `Activity`, dus ook de `AsyncTask`

=> resultaten komen niet meer binnen!

Intent: kenmerken

Moeilijker te implementeren, complexere taken

Een `Intent` staat los van de `Activity`

Een `Intent` kan, indien gewenst opgevangen worden door elke App die zich daarvoor registreert.

Wanneer de `Activity` pauseert, wordt de `Intent` verder afgehandeld

Networking

Net APIs

Standaard Java networking APIs

Twee HTTP clients:

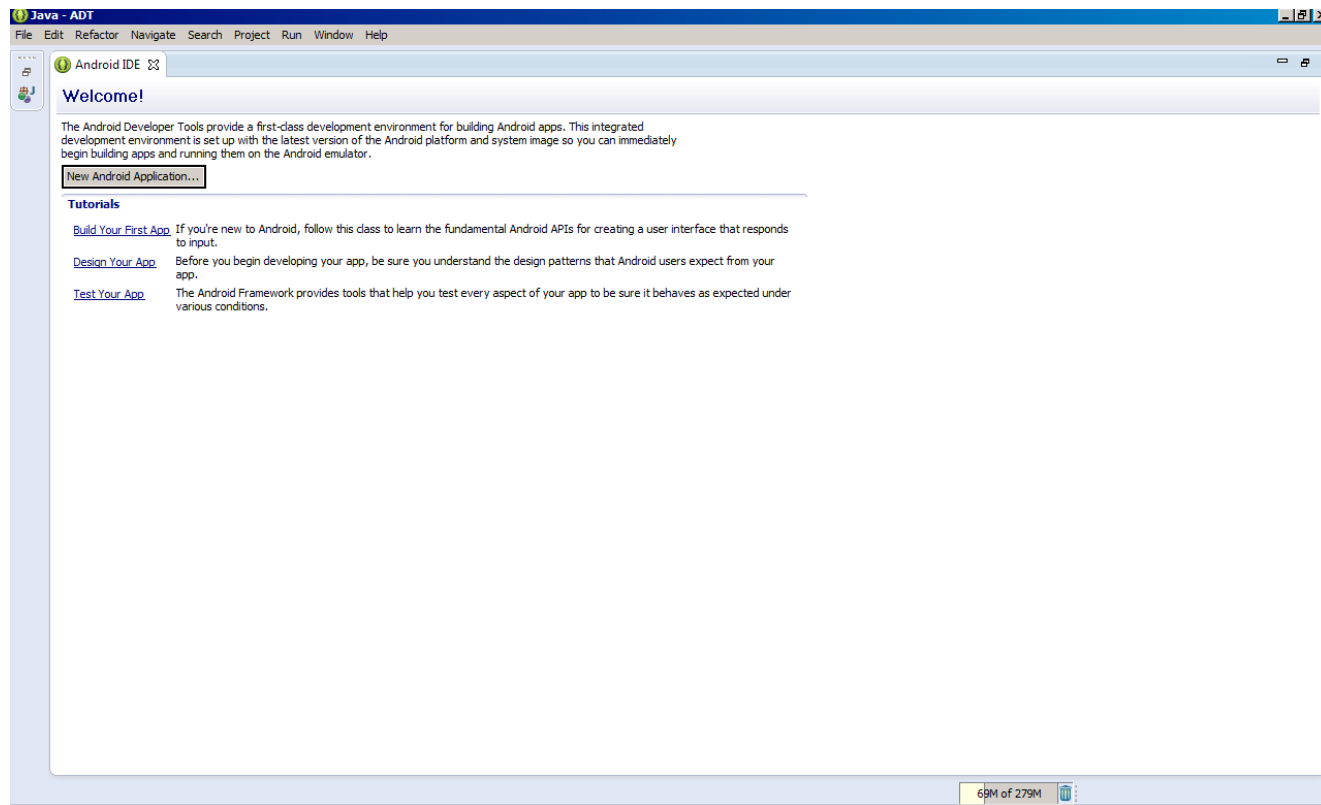
URLConnection en **Apache Http client**

JSON Java API

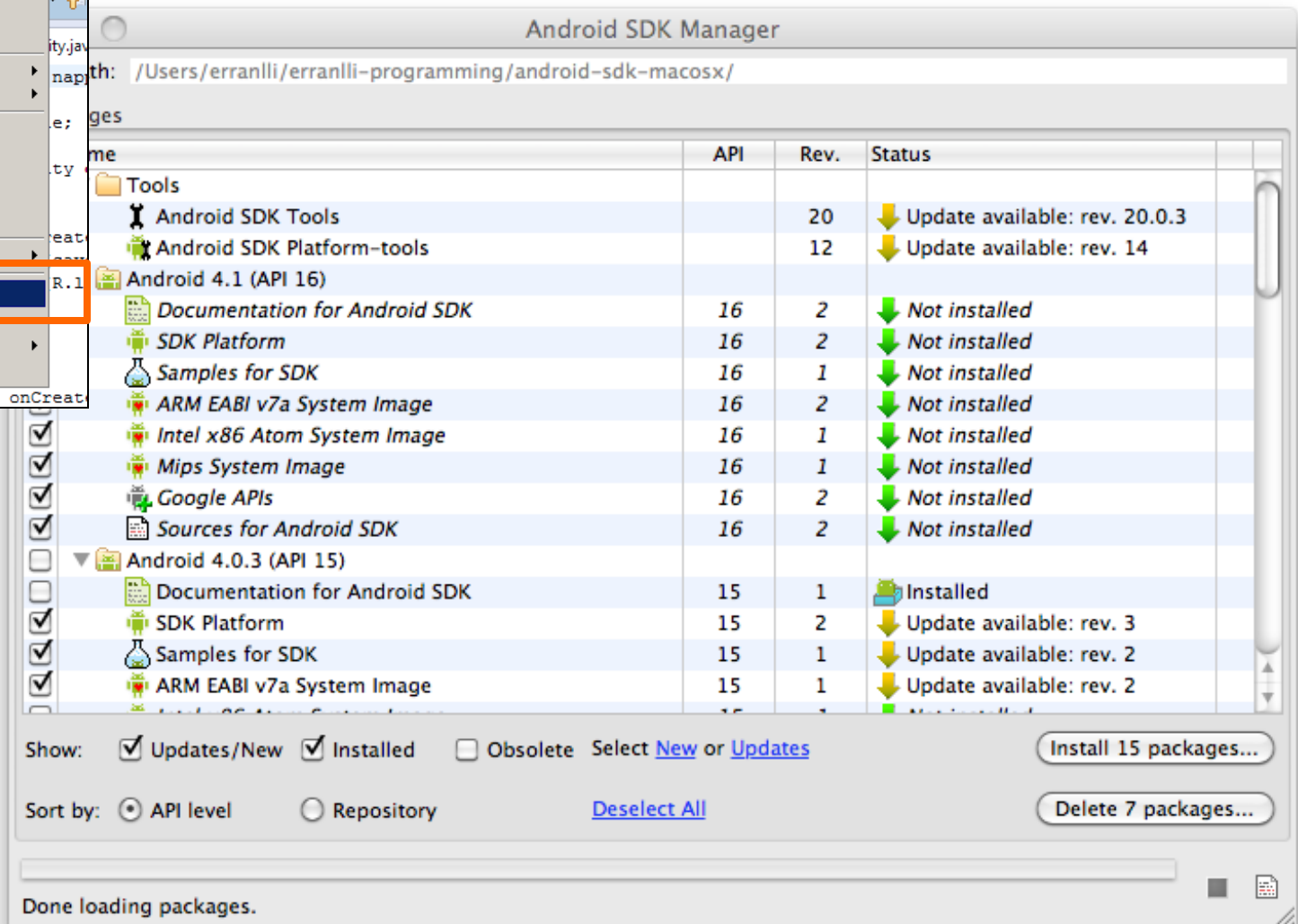
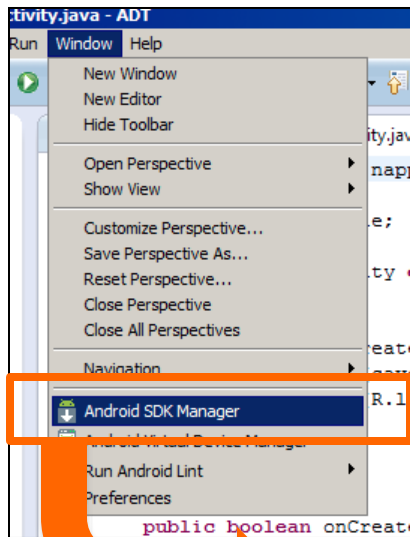
Server response parsen

Android Developer Tools

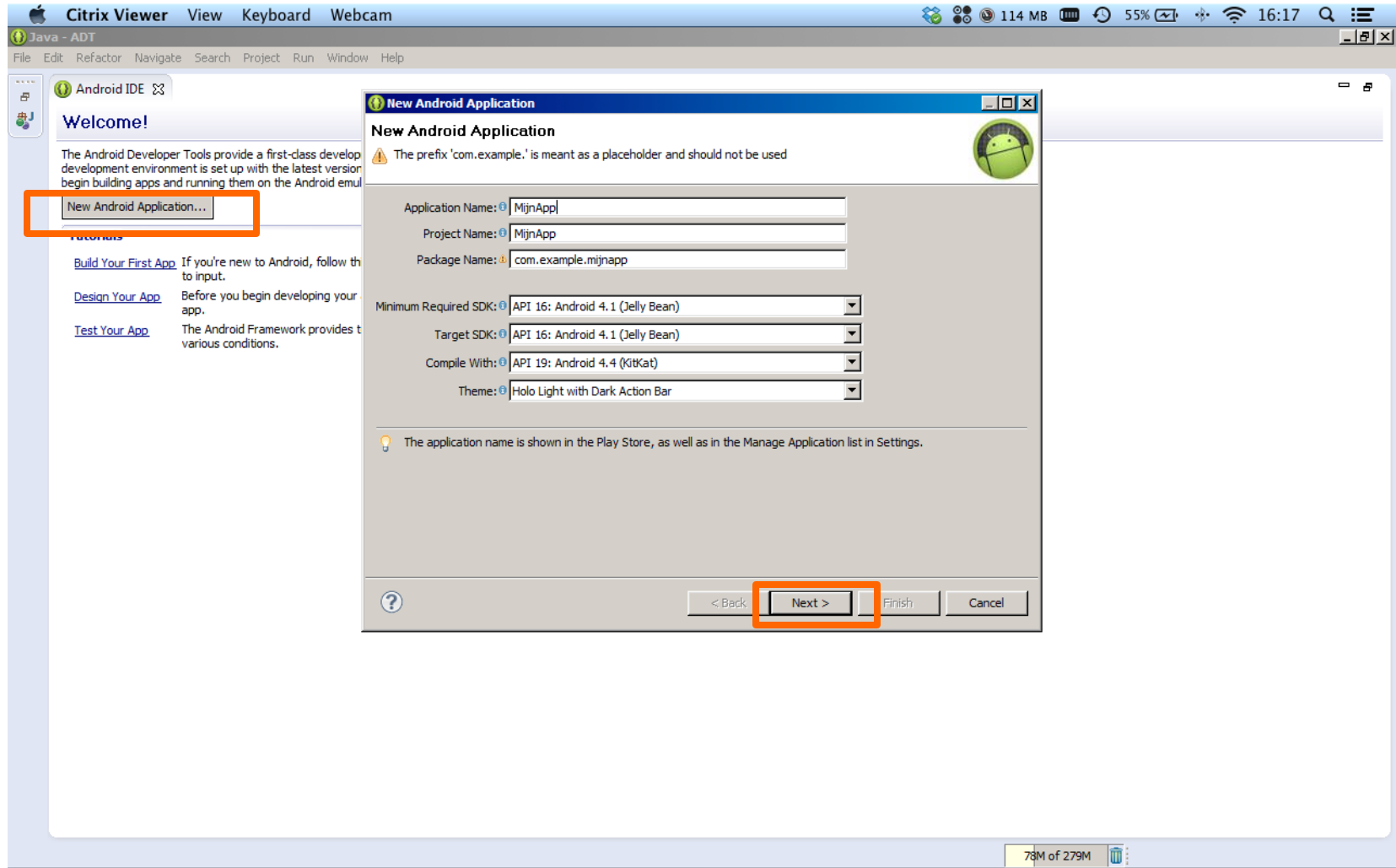
Gebaseerd op Eclipse omgeving, zoals Aptana
Beschikbaar op Athena en lokaal “Konrad Zuse”



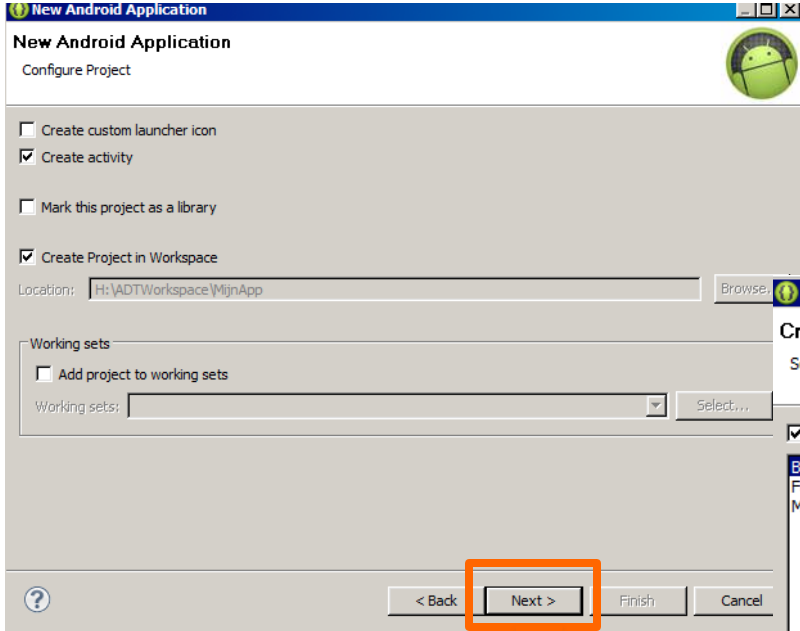
Android SDK Manager



Nieuw project aanmaken



Nieuw project aanmaken

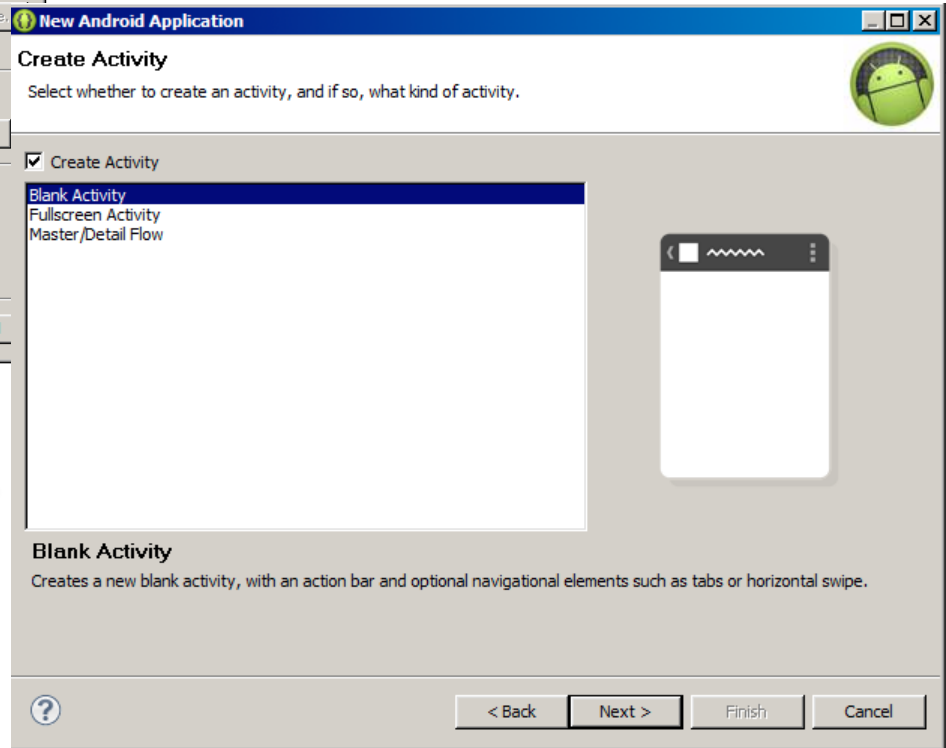


New Android Application
Configure Project

☐ Create custom launcher icon
☒ Create activity
☐ Mark this project as a library
☒ Create Project in Workspace

Location:

Working sets
☐ Add project to working sets
Working sets:



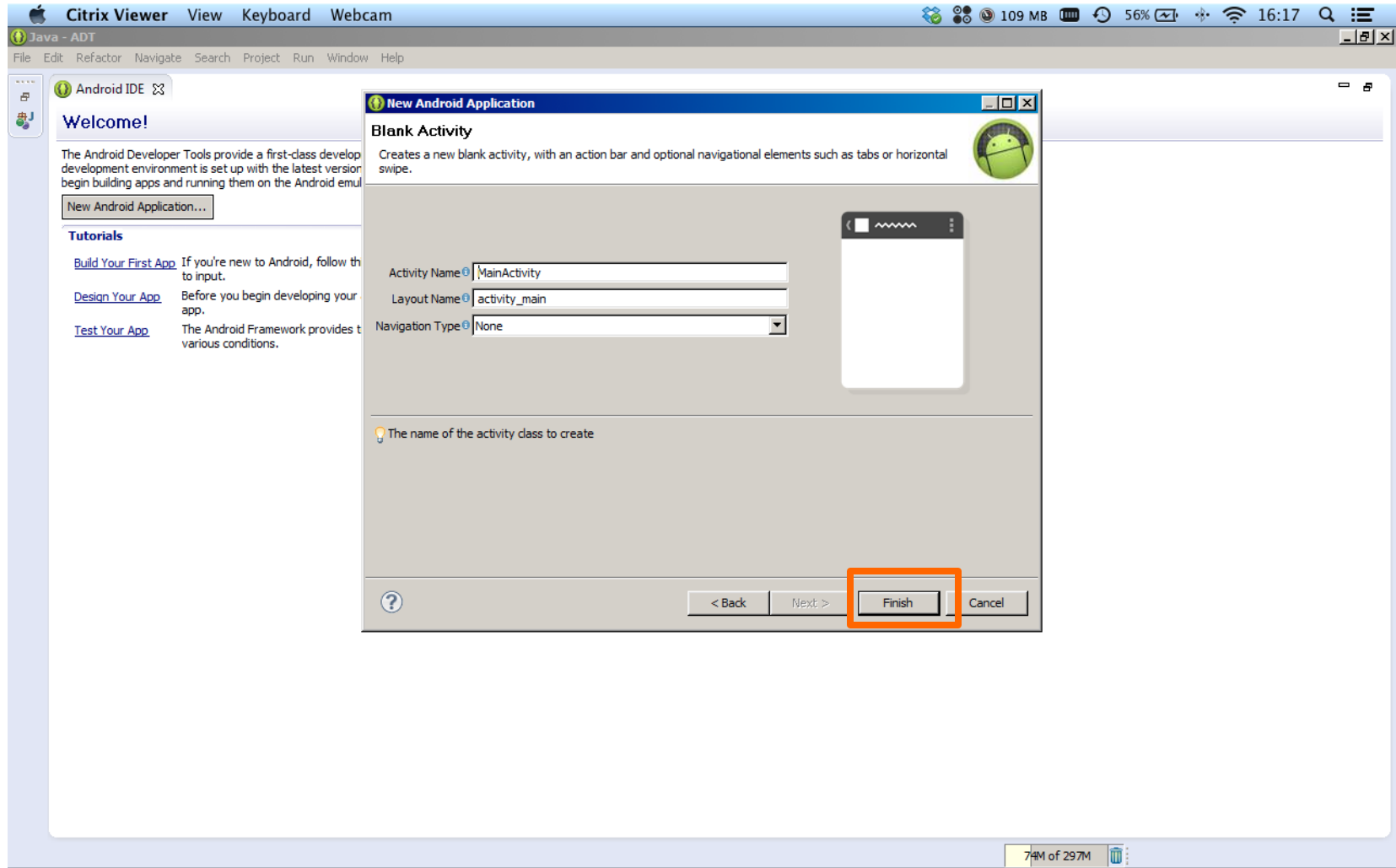
Create Activity
Select whether to create an activity, and if so, what kind of activity.

☒ Create Activity

Blank Activity
Fullscreen Activity
Master/Detail Flow

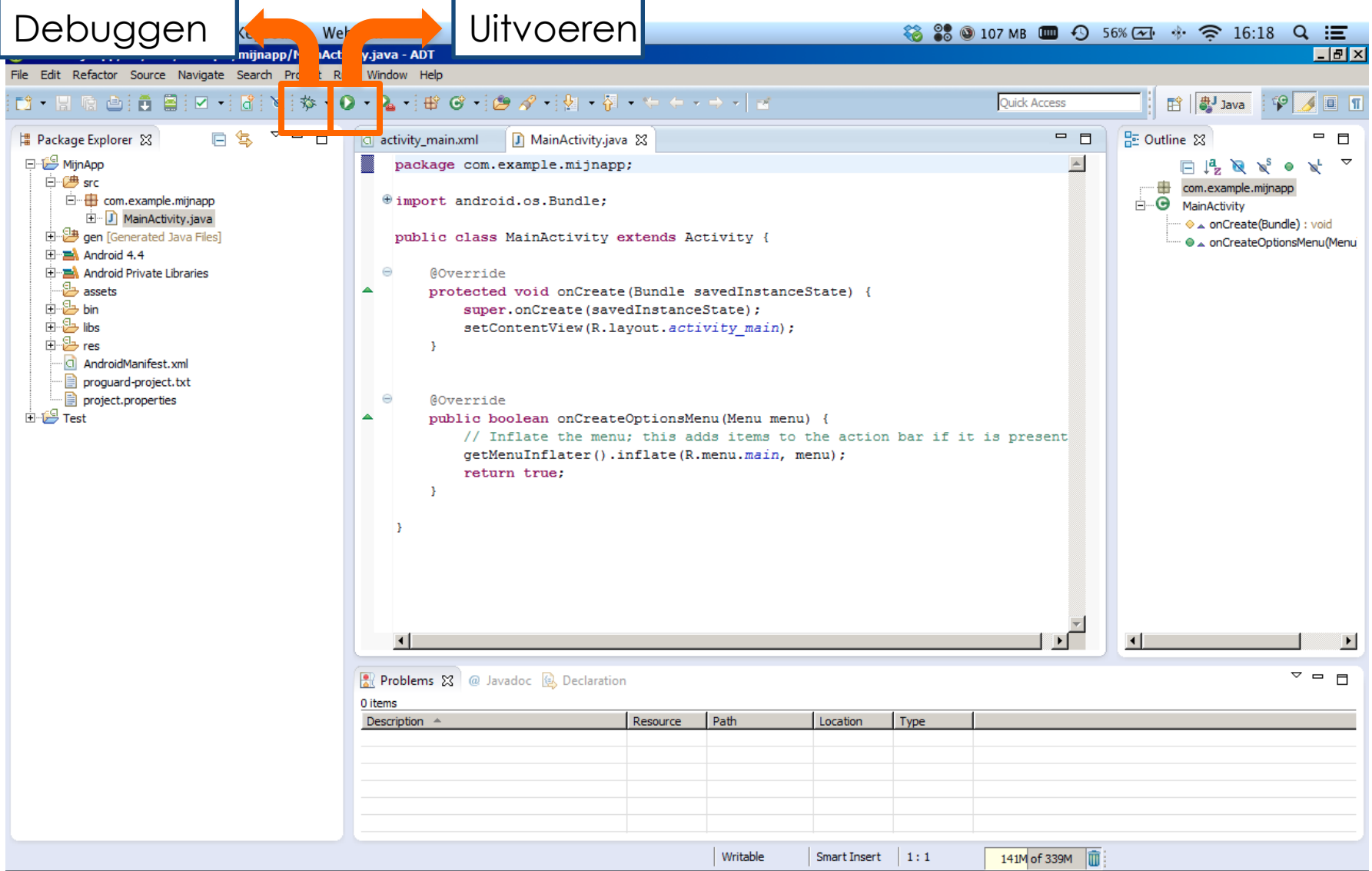
Blank Activity
Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Nieuw project aanmaken



Debuggen

Uitvoeren



The screenshot shows the Android Studio IDE with the following components:

- Package Explorer:** Shows the project structure for 'MijnApp', including 'src', 'gen', 'Android 4.4', 'Android Private Libraries', 'assets', 'bin', 'libs', 'res', 'AndroidManifest.xml', 'proguard-project.txt', 'project.properties', and 'Test'.
- MainActivity.java:** The main activity class is open, showing the following code:


```
package com.example.mijnapp;

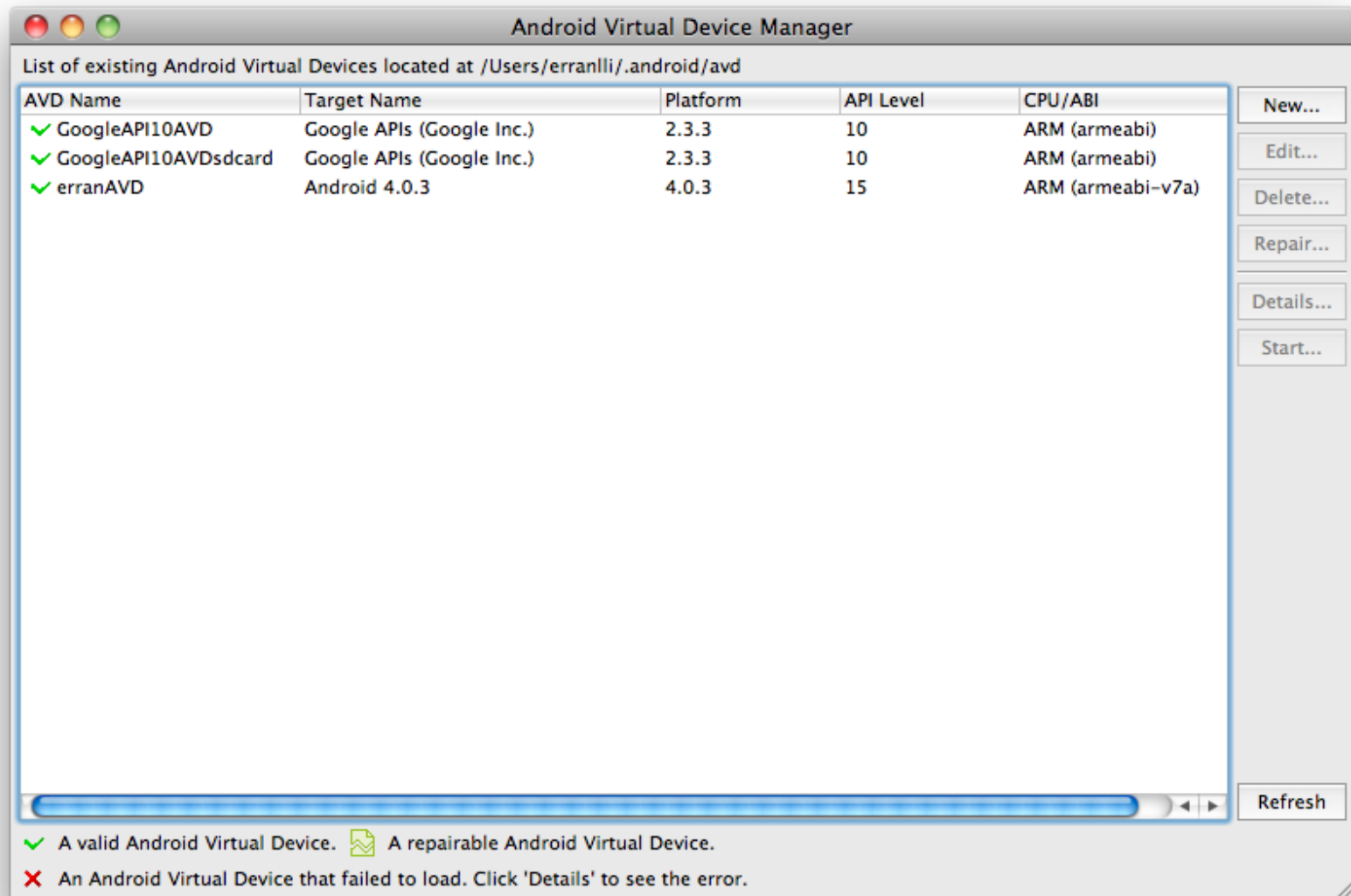
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```
- Outline:** Shows the class hierarchy for 'com.example.mijnapp' and 'MainActivity', with methods 'onCreate(Bundle) : void' and 'onCreateOptionsMenu(Menu)' listed.
- Problems:** Shows 0 items, indicating no errors or warnings.
- Run Button:** A green play icon in the toolbar, highlighted with a red box. Two orange arrows point from it to the 'Debug' button (a green bug icon) and the 'Uitvoeren' button (a green play icon).

Uitvoeren 1: Gebruik een Emulator



Uitvoeren 2: Gebruik een fysiek toestel

Installeer de stuurprogramma's voor het apparaat

Sluit het apparaat aan op een computer via USB-kabel

Zorg ervoor ingeschakeld USB-foutopsporing op het apparaat
(Instellingen -> Application -> Ontwikkeling -> USB debugging)

Apparaat zal binnen ADT worden erkend (DDMS view)

Doe dit enkel indien je tijd over hebt.

Overzicht

REST APIs gebruiken

Ontwikkelen voor Android

Opgave

Opgave practicum 2

Android-toepassing ontwikkelen

Functionaliteit analoog aan Practicum 1

Events API : <http://events.restdesc.org/>

Hypermedia drijft de toepassing

Responsieve interface via *AsyncTask*

Reflectie: Web vs. Native Apps

Opgave practicum 2

Opgavetekst
op Minerva

Deadline: donderdag 23 april om 14u

Oplossingen maken
PC-klas 'Konrad Zuse'