

---

# RESTful Android

Na co zwrócić uwagę przy tworzeniu aplikacji  
klienckich REST na Androidzie



Jacek Laskowski

[jacek@japila.pl](mailto:jacek@japila.pl)

<http://www.JacekLaskowski.pl>

# O prelegencji - Jacek Laskowski

---

- **Entuzjasta** Java EE, OSGi, SCA oraz programowania funkcyjnego (Clojure)
- Założyciel i lider **Warszawa JUG**
- Organizator konferencji **Javarsovia, Confitura, warsjava**
- Członek zespołu **NetBeans DreamTeam**
- Blogger na **<http://JacekLaskowski.pl>**
- Blogger na **<http://blog.japila.pl>**
- Twittuje jako **@jaceklaskowski**
- Członek zespołów **Apache Geronimo** i **Apache OpenEJB**
- **Specjalista** produktów **IBM WebSphere** w IBM Polska





konferencja społeczności jadowej w Polsce

# Confitura 2012

<http://confitura.pl>

6-7 lipiec

Warszawa

# Problem?

---

```
package pl.japila.android;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;

public class ANRDemoActivity extends Activity implements OnClickListener
{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        View button = findViewById(R.id.button1);
        button.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        try {
            Thread.sleep(30*1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

# 30 sekund bezruchu

---

```
package pl.japila.android;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;

public class ANRDemoActivity extends Activity implements OnClickListener
{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

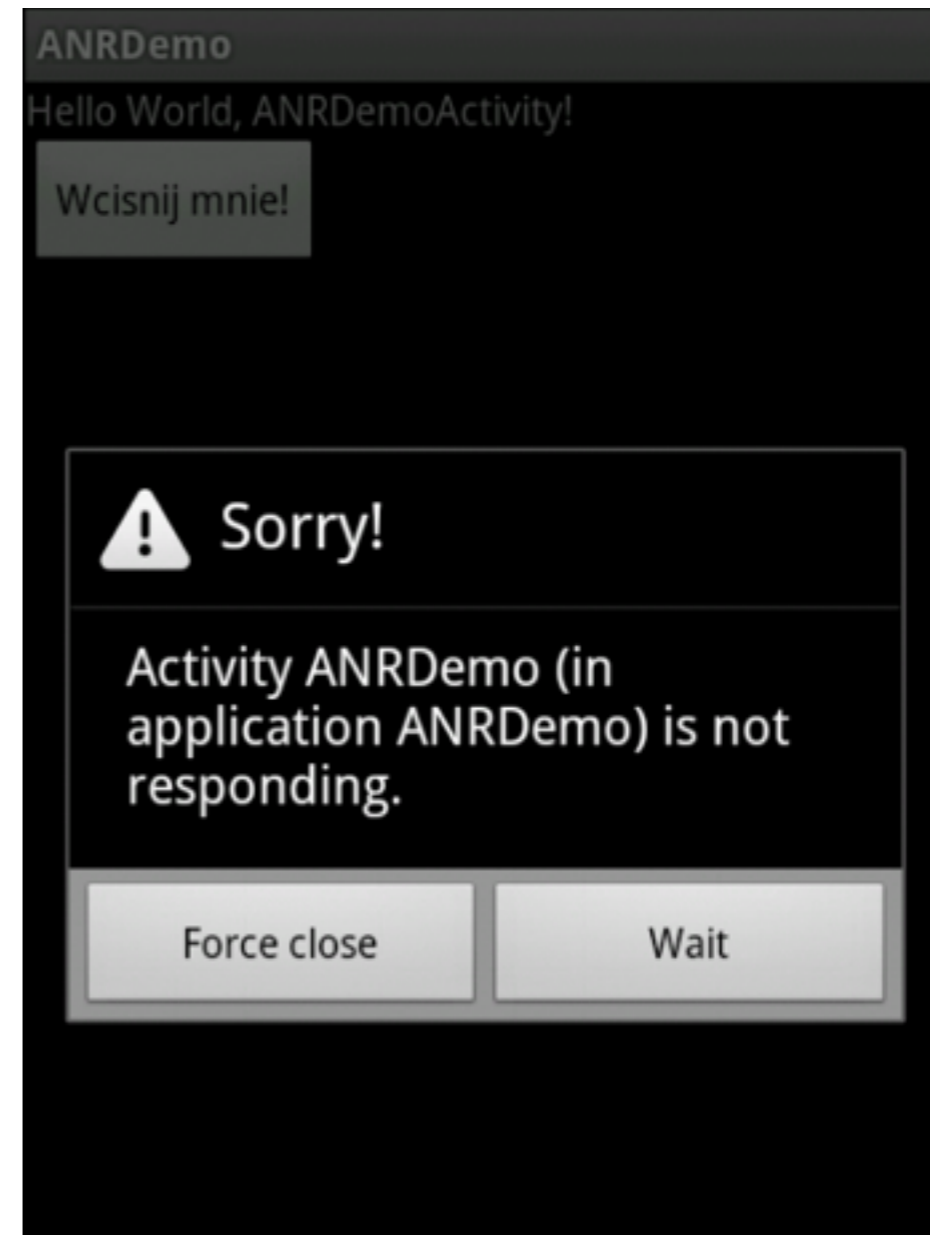
        View button = findViewById(R.id.button1);
        button.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        try {
            Thread.sleep(30*1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

# Application Not Responding (ANR)

---

- Wątek główny obsługujący UI - "main" lub "thread UI"
- Android chroni przed niewłaściwie (nie)aktywnymi aplikacjami
- Brak odpowiedzi po 5 sek.
- Długotrwałe operacje I/O (sieciowe)



# Z Androida po HTTP

---

- java.net.**HttpURLConnection**
- Apache HTTP Client - **org.apache.http.client**
- android.net.http.**AndroidHttpClient**
- *“For Gingerbread and better, HttpURLConnection is the best choice. New applications should use HttpURLConnection; it is where we will be spending our energy going forward.”*  
<http://android-developers.blogspot.com/2011/09/androids-http-clients.html>

# URLConnection w użyciu

---

```
@Override
public void onClick(View v) {
    HttpURLConnection urlConnection = null;
    try {
        URL url = new URL(REST_SERVICE);
        urlConnection = (HttpURLConnection) url.openConnection();
        InputStreamReader isr = new InputStreamReader(urlConnection.getInputStream());
        BufferedReader br = new BufferedReader(isr);
        try {
            String line;
            while ((line = br.readLine()) != null) {
                Log.i("ANRDemo", line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (urlConnection != null) {
            urlConnection.disconnect();
        }
    }
}
```

# URLConnection w użyciu...złym

---

```
@Override
public void onClick(View v) {
    HttpURLConnection urlConnection = null;
    try {
        URL url = new URL(REST_SERVICE);
        urlConnection = (HttpURLConnection) url.openConnection();
        InputStreamReader isr = new InputStreamReader(urlConnection.getInputStream());
        BufferedReader br = new BufferedReader(isr);
        try {
            String line;
            while ((line = br.readLine()) != null) {
                Log.i("ANRDemo", line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (urlConnection != null) {
            urlConnection.disconnect();
        }
    }
}
```

Application Not Responding

# Wątki w Androidzie

---

- Długotrwałe operacje wykonywane w osobnym wątku
- `java.lang.Thread/Runnable`
- Specjalizowane klasy
  - `android.os.Handler`
  - `android.app.IntentService`
  - `android.os.AsyncTask`

```
@Override
public void onClick(View v) {
    try {
        Thread.sleep(30*1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

# Poprawnie z Thread/Runnable

---

```
@Override
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            Log.d(TAG, Thread.currentThread().getName());
            HttpURLConnection urlConnection = null;
            try {
                URL url = new URL(REST_SERVICE);
                urlConnection = (HttpURLConnection) url.openConnection();
                InputStreamReader isr = new InputStreamReader(urlConnection.getInputStream());
                BufferedReader br = new BufferedReader(isr);
                try {
                    String line;
                    while ((line = br.readLine()) != null) {
                        Log.d(TAG, line);
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            } catch (Exception e) {
                e.printStackTrace();
            } finally {
                if (urlConnection != null) {
                    urlConnection.disconnect();
                }
            }
        }
    }).start();
}
```

# Aktualizacja aktywnego ekranu

---

- Specjalizowane metody do aktualizacji danych w UI
  - `Activity.runOnUiThread(Runnable)`
  - `View.post(Runnable)`
  - `View.postDelayed(Runnable, long)`

```
@Override
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            final Bitmap bitmap = loadImageFromNetwork("http://example.com/image.png");
            mImageView.post(new Runnable() {
                public void run() {
                    mImageView.setImageBitmap(bitmap);
                }
            });
        }
    }).start();
}
```

# Czym jest REST?

---

- **RE**presentational **S**tate **T**ransfer
- Styl architektoniczny
  - Powrót do zasad HTTP - protokół transportowy
  - przesyłanie stanu między klientem a aplikacją
  - reprezentacja zasobu - XML, JSON
- **C**reate = POST /prezentacja + dane (stan)
- **R**ead = GET /prezentacja/1430
- **U**psdate = PUT /prezentacja/1430 + dane (stan)
- **D**elete = DELETE /prezentacja/1430

# RESTClient - GET

---

```
@Override
public void onClick(final View v) {
    new Thread(new Runnable() {
        public void run() {
            // ...
            URL url = new URL(REST_SERVICE);
            conn = (HttpURLConnection) url.openConnection();
            // ...
            v.post(new Runnable() {
                @Override
                public void run() {
                    TextView textView = (TextView) findViewById(R.id.editText1);
                    textView.setText(response);
                }
            });
        }
    }).start();
}
```

# RESTClient - POST

---

```
@Override
public void onClick(final View v) {
    new Thread(new Runnable() {
        public void run() {
            // ...
            URL url = new URL(REST_SERVICE);
            conn = (HttpURLConnection) url.openConnection();

            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);

            conn.setRequestMethod("POST");
            // ...
            v.post(new Runnable() {
                @Override
                public void run() {
                    TextView textView = (TextView) findViewById(R.id.editText1);
                    textView.setText(response);
                }
            });
        }
    }).start();
}
```

# RESTClient - PUT

---

```
@Override
public void onClick(final View v) {
    new Thread(new Runnable() {
        public void run() {
            // ...
            URL url = new URL(REST_SERVICE);
            conn = (HttpURLConnection) url.openConnection();

            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);

            conn.setRequestMethod("PUT");
            // ...
            v.post(new Runnable() {
                @Override
                public void run() {
                    TextView textView = (TextView) findViewById(R.id.editText1);
                    textView.setText(response);
                }
            });
        }
    }).start();
}
```

# RESTClient - DELETE

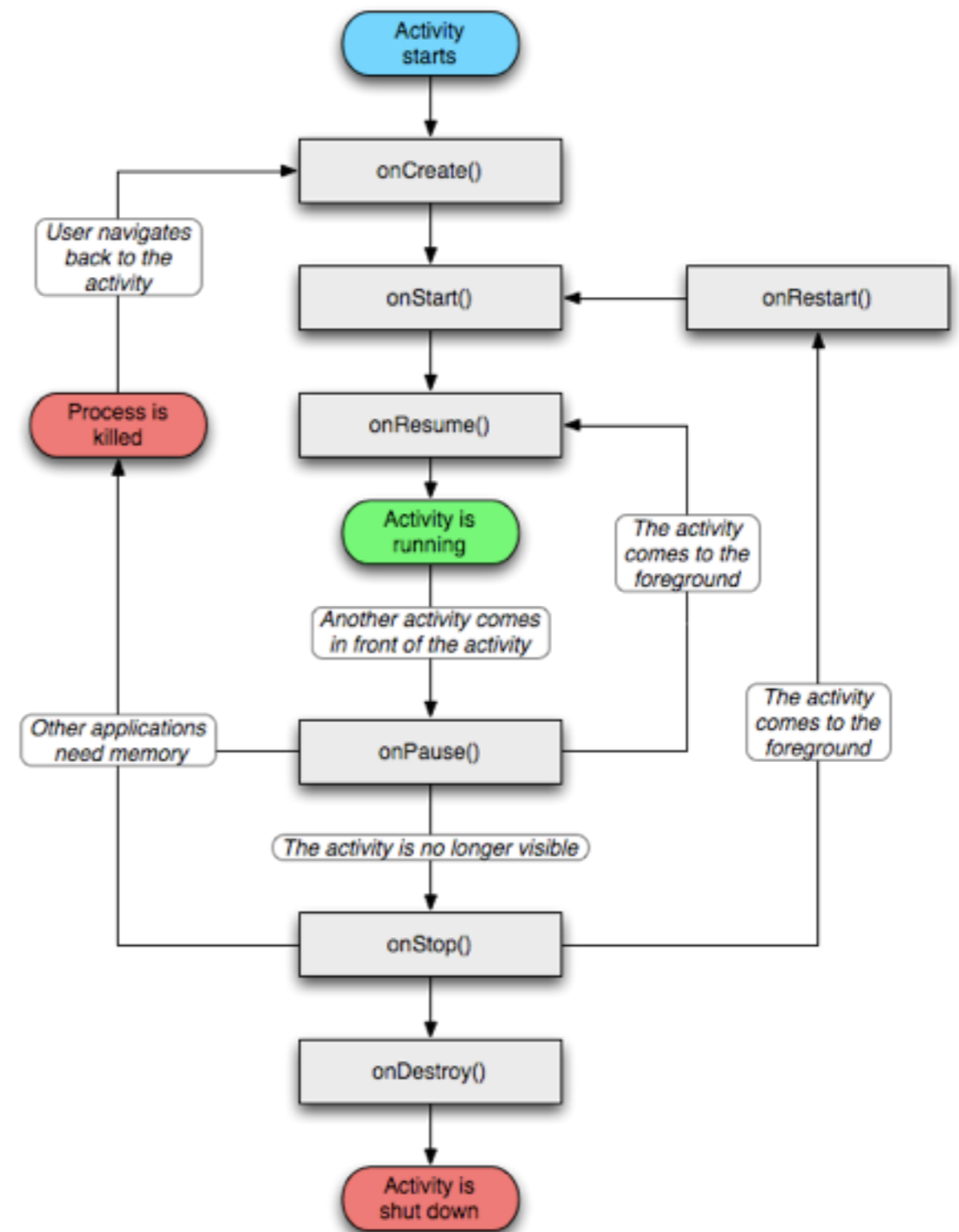
---

```
@Override
public void onClick(final View v) {
    new Thread(new Runnable() {
        public void run() {
            // ...
            URL url = new URL(REST_SERVICE);
            conn = (HttpURLConnection) url.openConnection();

            conn.setRequestMethod("DELETE");
            // ...
            v.post(new Runnable() {
                @Override
                public void run() {
                    TextView textView = (TextView) findViewById(R.id.editText1);
                    textView.setText(response);
                }
            });
        }
    }).start();
}
```

# Zmiany stanów Activity a wątki

- Zmiana orientacji ekranu
- Rozmowa przychodząca
- Brak pamięci
- Inne sytuacje wyjątkowe



---

# Pytania?



Jacek Laskowski

[jacek@japila.pl](mailto:jacek@japila.pl)

<http://www.JacekLaskowski.pl>