

WebView

Na podstawie prezentacji udostępnionych przez Victor Matos, Cleveland State University
i tłumaczenia dr Tomasza Xięskiego

Trzy sposoby realizacji aplikacji na system Android

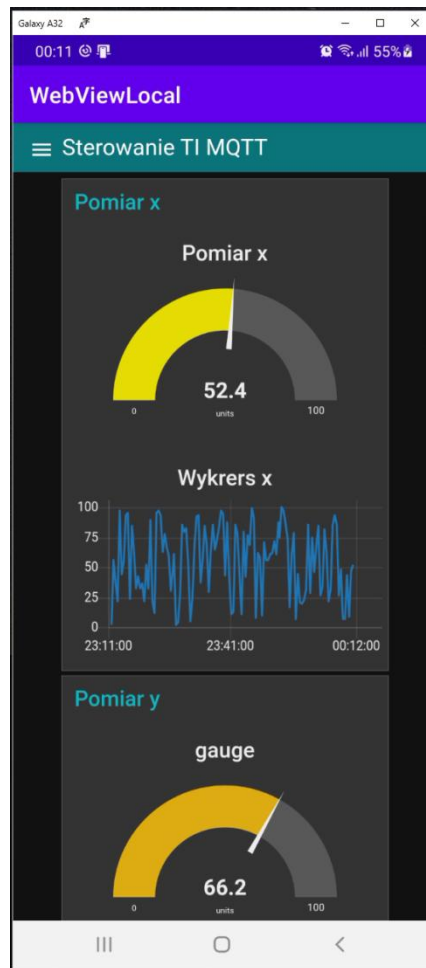
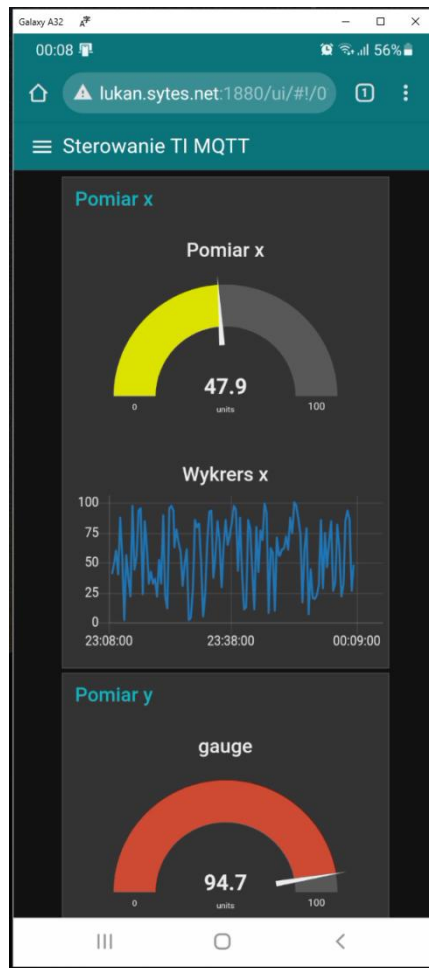
1. **Aplikacja natywna** Tworzona jest przy wykorzystaniu natywnego SDK i przechowywana jest na urządzeniu. Wykorzystuje wszelkie komponenty platformy Android jak widżety, usługi, aktywności, fragmenty, dostawcy treści, powiadomienia, itp.)
2. **Aplikacja internetowa** Zwykle realizowana jest jako aplikacja zdalna i umożliwia pobranie zasobów z serwera z wykorzystaniem zainstalowanej przeglądarki internetowej.
3. **Aplikacja hybrydowa** Realizowana jest jako aplikacja internetowa, jednakże oprócz stron html i wykorzystania zdalnych zasobów możliwa jest interakcja z lokalnymi zasobami i możliwościami urządzenia (np. dostęp do czujników)

Trzy sposoby realizacji aplikacji na system Android - charakterystyka

1. **Aplikacja natywna** pozwala na tworzenie bogatszego interfejsu i wykorzystuje w pełni dostępne zasoby sprzętowe, ale ograniczona jest do platformy Android. Często wykorzystanie innej platformy wymaga przepisania znacznej części bądź całości aplikacji.
2. **Aplikacja internetowa** pozwala uzyskać największy stopień przenośności. Każde urządzenie z dostępem do przeglądarki internetowej może otworzyć taką aplikację, jednakże brak jest bezpośredniego dostępu do zasobów sprzętowych urządzenia mobilnego.
3. **Aplikacja hybrydowa** stanowi kompromis między pozostałymi podejściami. Wykorzystanie standardu HTML (włącznie z elementami wersji piątej) oraz dostęp z poziomu języka skryptowego do fizycznych zasobów urządzenia zapewnia duże możliwości i zachowanie przyzwoitych możliwości przenośności.

Komponent WebView

1. Komponent WebView wykorzystuje otwarto-źródłowy silnik WebKit (znany z przeglądarek typu Chrome czy Opera).
2. Komponent WebView nie jest przeglądarką. Pomimo, że umożliwia wyświetlenie strony internetowej, ale nie posiada standardowych przycisków jak Wstecz, Dalej, Odśwież, itp.
3. Klasa WebView zawiera metody do:
 - Ładowania/Przeładowania strony, nawigacji wprzód lub wstecz poprzez historię przeglądarki,
 - Przybliżanie i oddalanie,
 - Wykorzystanie stylów CSS oraz skryptów JavaScript by zapewnić podobne GUI na różnych urządzeniach,
 - Wymiana danych z urządzeniem przez interfejs JavaScript,
 - Wyszukiwanie tekstu, pobierania zdjęć



Przeglądarka i WebView

Od API 27 (Android 8) połączenia internetowe domyślnie tylko HTTPS

Wpisów należy dokonać w pliku manifestu:

\$project_path\android\app\src\debug\AndroidManifest.xml

1. Pozwolenie na korzystanie z internetu:

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

2. Aby zezwolić na połączenia niebezpieczne HTTP należy w pliku manifestu należy dokonać wpisu w sekcji application:

```
<application  
    android:usesCleartextTraffic="true" />
```

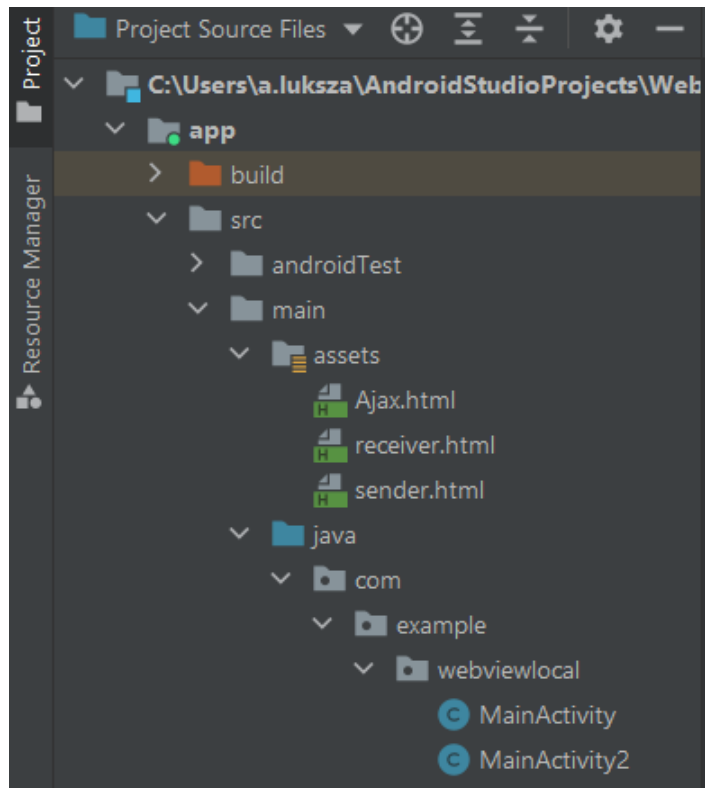
Zezwolenie na dostęp do internetu

```
?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.webviewlocal">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.WebViewLocal"
        android:usesCleartextTraffic="true">
```

Lokalizacja i adres url plików lokalnych



```
WebView webView;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    webView = (WebView) findViewById(R.id.webView);  
    webView.getSettings().setJavaScriptEnabled(true);  
    webView.getSettings().setDomStorageEnabled(true);  
    webView.getSettings().setAllowFileAccess(true);  
  
    webView.setWebViewClient(new WebViewClient());  
    //webView.loadUrl("file:///android_asset/sender.html");  
    webView.loadUrl("file:///android_asset/Ajax.html");  
}
```

Otwieranie WebView w nowej aktywności i przekazanie adresu URL

Aktywność 1

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

    findViewById(R.id.button1).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent myActivity = new Intent(MainActivity.this, MainActivity2.class);
            myActivity.putExtra("url", "file:///android_asset/sender.html");
            startActivity(myActivity);
        }
    });

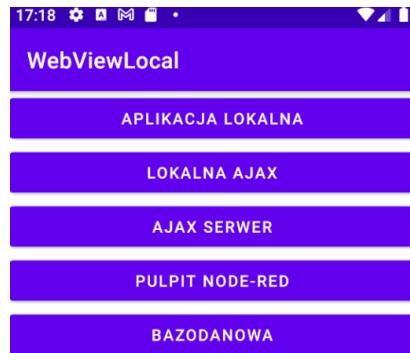
    findViewById(R.id.button2).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent myActivity = new Intent(MainActivity.this, MainActivity2.class);
            myActivity.putExtra("url", "file:///android_asset/Ajax.html");
            startActivity(myActivity);
        }
    });
}
```

Aktywność 2

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    WebView webView = (WebView) findViewById(R.id.webView);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.getSettings().setDomStorageEnabled(true);
    webView.getSettings().setAllowFileAccess(true);
    webView.setWebViewClient(new WebViewClient());

    webView.loadUrl(getIntent().getStringExtra("url").toString());
}
```



Aby otworzyć aktywność trzeba ją najpierw dodać do projektu z Menu: File/New/Activity/Empty Activity

Lokalne strony html – przesyłanie danych formularza

Sender.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
</head>
<body>
<FORM action="receiver.html" id=form1 method=GET name=form1>
  <h1>Form1: Sending Data</h1>
  First Name:
  <br>
  <INPUT id="txtFirstName" name="txtFirstName" value="Maria">
  <br>
  Last Name:
  <br>
  <INPUT id="txtLastName" name="txtLastName" value="Macarena">
  <p>
    <INPUT type="submit" value="Submit" id=submit1 name=submit1>
</FORM>
</body>
</html>
```



Reciver.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Android & Plain HTML Demo</title>
  <script language="javascript">
    function extracting(variable) {
      var query = window.location.search.substring(1);
      alert(query);
      var personName = getQueryVariable(variable)
      document.getElementById("myMsg").value = personName;
    }
    function getQueryVariable(variable) {
      var query = window.location.search.substring(1);
      var vars = query.split('&');
      for (var i=0;i<vars.length;i++) {
        var pair = vars[i].split('=');
        if ((pair[0] == variable) || (variable == 0)) {
          return pair[1];
        }
      }
      alert('Query Variable ' + variable + ' not found');
    }
  </script>
</head>
<body>
  <h1>Form2: Receiving Data</h1>
  <p><input TYPE="text" id="myMsg" ></p>
  <p><input TYPE="button" value="Display" onClick="extracting('txtFirstName');"></p>
  <p>Receiving data from the GET query-string
    <script>
      document.write("<p>Spying..." + new Date() );
      document.write("<p> You supplied the following data:");
      var querystring = window.location.search.substring(1);
      document.write("<p><b>" + querystring + "</b>");
      document.write("<p>txtLastName: " + getQueryVariable('txtLastName') );
    </script>
  </body>
</html>
```

Strona lokalna wykorzystująca AJAX

Ajax.html

```
<!DOCTYPE html>
<html>
<body>
  <h2>Odczyt wskazań czujnika BME280</h2>
  <p id="czas">Tu będą wyniki</p>
  <p id="demo">Tu będą wyniki</p>
  <button type="button" onclick="loadDoc()">Odczytaj dane</button>
</script>
var data = new Date();
document.getElementById("czas").innerHTML = "Czas wczytania strony: " + data.toLocaleTimeString();
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "http://lukan.sytes.net:1880/ajax/dane", true);
  xhttp.send();
}
</script>
</body>
</html>
```

Źródła danych

<http://lukan.sytes.net:1880/ajax/dane>

<http://lukan.sytes.net:1880/ajax/bme>

WebViewLocal

Odczyt wskazań czujnika
BME280

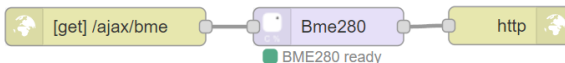
Czas wczytania strony: 16:49:22

Tu będą wyniki

Odczytaj dane



```
var data = new Date();
msg.payload = {
  "Czas": data.toLocaleTimeString(),
  "x": Math.random()
}
return msg;
```



Strona z serwera wykorzystująca AJAX

Czytanie pliku xml

```
<informatyka>
<IIrok id="IIrok">
<student nazwisko="Kowalski2" imię="Jan"/>
<student nazwisko="Nowak2" imię="Piotr"/>
<student nazwisko="Kowal2" imię="Stefan"/>
<student nazwisko="Widak2" imię="Igor"/>
</IIrok>
<IIIrok>
<student nazwisko="Kowalski3" imię="Jan"/>
<student nazwisko="Nowak3" imię="Piotr"/>
<student nazwisko="Kowal3" imię="Stefan"/>
<student nazwisko="Widak3" imię="Igor"/>
</IIIrok>
</informatyka>
```

WebViewLocal

The XMLHttpRequest Object

Tu będzie odpowiedź

WebViewLocal

The XMLHttpRequest Object

Wszyscy studenci

Kowalski2 Jan
Nowak2 Piotr
Kowal2 Stefan
Widak2 Igor
Kowalski3 Jan
Nowak3 Piotr
Kowal3 Stefan
Widak3 Igor

Studenci latami

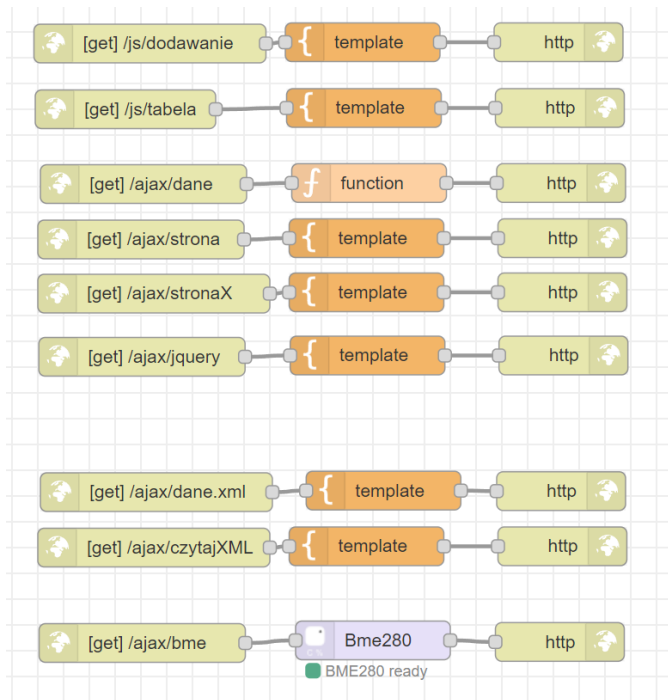
II rok

Kowalski2 Jan
Nowak2 Piotr
Kowal2 Stefan
Widak2 Igor

III rok

Kowalski3 Jan

Tworzenie usług serwerowych w Node-RED

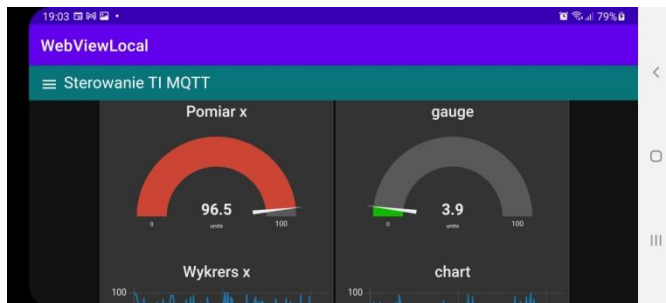
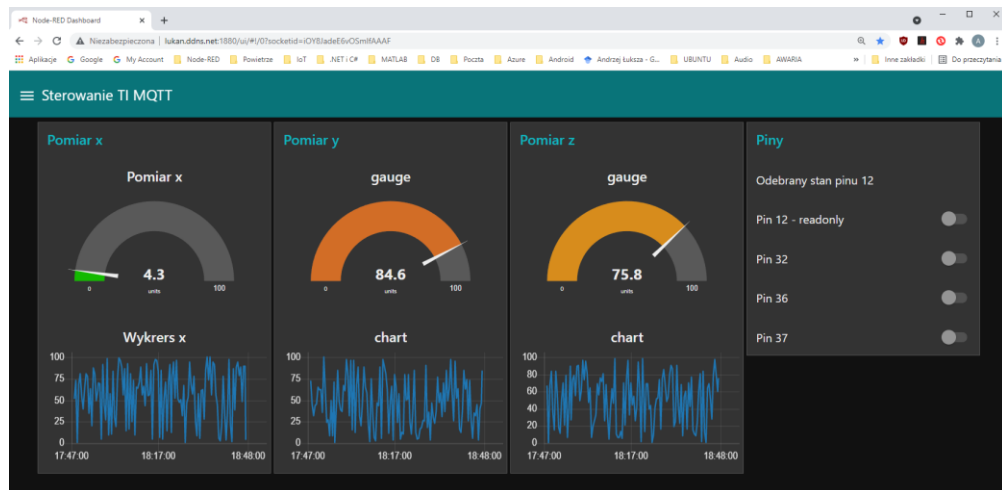


Przykłady

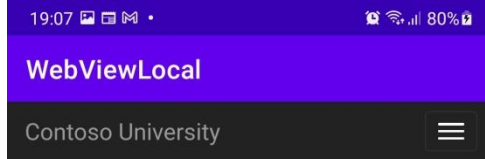
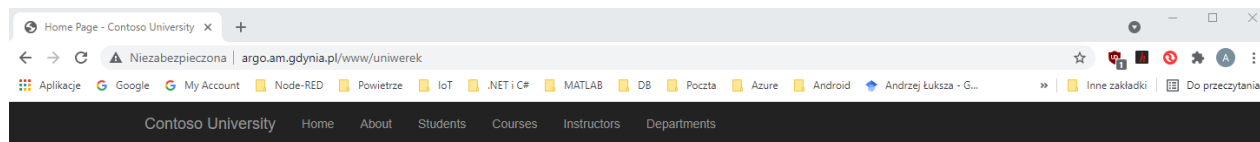
2. <https://www.google.com/maps/@54.4606871,18.4645007,14z>
3. <https://maps.google.com/maps?q=54.5,18.5&t=m&z=13>

Pulpit Node-RED na PC i smartfonie

Responsywność



Aplikacja ASP.NET na PC i urządzeniu przenośnym



Contoso University

Welcome to Contoso University

Contoso University is a sample application that demonstrates how to use Entity Framework 6 in an ASP.NET MVC 5 web application.

Build it from scratch

You can build the application by following the steps in the tutorial series on the ASP.NET site.

[See the tutorial »](#)

Download it

You can download the completed project from the Microsoft Code Gallery.

[Download »](#)



Dodatkowe materiały

- <https://developer.android.com/guide/webapps/webview.html>