

# Mobile Application Development

Higher Diploma in Science in Computer Science

---

Produced  
by

Eamonn de Leastar (edelestar@wit.ie)

Department of Computing, Maths & Physics  
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



# File IO in MyRent

---



MyRent



52.253456, -7.187162

Registered: Sep 6, 2014 11:48:29 AM



52.253456, -7.187162

Registered: Sep 6, 2014 11:48:31 AM



12,12

Registered: Sep 14, 2014 12:00:00 AM



- Json Representation

```
[
  {
    "date"      : 1410000509060,
    "geolocation": "52.253456, -7.187162",
    "id"        : "6a8d44df-7534-43c5-8130-444678fcc187",
    "rented"    : false
  },
  {
    "date"      : 1410000511943,
    "geolocation": "52.253456, -7.187162",
    "id"        : "483eb7da-0dd2-41d3-b53e-ba3d51f9c55d",
    "rented"    : false
  },
  {
    "date"      : 1410649200000,
    "geolocation": "12,12",
    "id"        : "9cf116a4-2c75-40a8-b28b-f3bc9a3939a6",
    "rented"    : true
  }
]
```

# Residence

---

- Introduce JSON Support libraries

```
import org.json.JSONException;  
import org.json.JSONObject;
```

- Define string IDs for each field

```
private static final String JSON_ID           = "id"           ;  
private static final String JSON_GEOLOCATION  = "geolocation"  ;  
private static final String JSON_DATE       = "date"         ;  
private static final String JSON_RENTED     = "rented"       ;
```

# Residence

---

- Write a json Residence Object

```
public JSONObject toJSON() throws JSONException
{
    JSONObject json = new JSONObject();
    json.put(JSON_ID, id.toString());
    json.put(JSON_GEOLOCATION, geolocation);
    json.put(JSON_DATE, date.getTime());
    json.put(JSON_RENTED, rented);
    return json;
}
```

- Read a json Residence Object

```
public Residence(JSONObject json) throws JSONException
{
    id = UUID.fromString(json.getString(JSON_ID));
    geolocation = json.getString(JSON_GEOLOCATION);
    date = new Date(json.getLong(JSON_DATE));
    rented = json.getBoolean(JSON_RENTED);
}
```

# Residence

- Complete json support

```
public class Residence
{
    public UUID id;

    public String geolocation;
    public Date date;
    public boolean rented;

    private static final String JSON_ID = "id" ;
    private static final String JSON_GEOLOCATION = "geolocation" ;
    private static final String JSON_DATE = "date" ;
    private static final String JSON_RENTED = "rented" ;

    public Residence()
    {
        this.id = UUID.randomUUID();
        this.date = new Date();
        this.geolocation = "";
    }

    public Residence(JSONObject json) throws JSONException
    {
        id = UUID.fromString(json.getString(JSON_ID));
        geolocation = json.getString(JSON_GEOLOCATION);
        date = new Date(json.getLong(JSON_DATE));
        rented = json.getBoolean(JSON_RENTED);
    }

    public JSONObject toJSON() throws JSONException
    {
        JSONObject json = new JSONObject();
        json.put(JSON_ID, id.toString());
        json.put(JSON_GEOLOCATION, geolocation);
        json.put(JSON_DATE, date.getTime());
        json.put(JSON_RENTED, rented);
        return json;
    }

    public String getDateString()
    {
        return "Registered: " + DateFormat.getDateTimeInstance().format(date);
    }
}
```

# PortfolioSerializer

---

- A class to orchestrate the Serialization of a collection of Residence Objects to/from Disk
- Constructor + 2 Methods
- Fully Encapsulates serialisation mechanisms

```
public class PortfolioSerializer
{
    private Context mContext;
    private String mFilename;

    public PortfolioSerializer(Context c, String f)
    {
        mContext = c;
        mFilename = f;
    }

    public void saveResidences(ArrayList<Residence> residences) throws ...
    {
        //...
    }

    public ArrayList<Residence> loadResidences() throws ...
    {
        //...
    }
}
```

# Portfolio

- Uses the serialiser to read and write the residence list
- Read - in constructor when Portfolio create
- Write - in 'saveResidences()' method, called when data may have changed

```
public class Portfolio
{
    public ArrayList<Residence> residences;
    private PortfolioSerializer serializer;

    public Portfolio(PortfolioSerializer serializer)
    {
        this.serializer = serializer;
        try
        {
            residences = serializer.loadResidences();
        }
        catch (Exception e)
        {
            info(this, "Error loading residences: " + e.getMessage());
            residences = new ArrayList<Residence>();
        }
    }

    public boolean saveResidences()
    {
        try
        {
            serializer.saveResidences(residences);
            info(this, "Residences saved to file");
            return true;
        }
        catch (Exception e)
        {
            info(this, "Error saving residences: " + e.getMessage());
            return false;
        }
    }

    //...
}
```



# MyRentApp

---

- Creates the serializer, giving it a file name to use
- Passes the serialiser to the Portfolio object, which will be responsible for loading / saving to the file using the serializer

```
public class MyRentApp extends Application
{
    private static final String FILENAME = "portfolio.json";

    public Portfolio portfolio;

    @Override
    public void onCreate()
    {
        super.onCreate();
        PortfolioSerializer serializer = new PortfolioSerializer(this, FILENAME);
        portfolio = new Portfolio(serializer);

        info(this, "RentControl app launched");
    }
}
```

# ResidenceActivity

---

- Call to save triggered by 'onPause' event detected by ResidenceActivity
- This will occur when the user leaves the activity

```
public class ResidenceActivity extends Activity implements ...
{
    //...
    private Portfolio portfolio;

    public void onPause()
    {
        super.onPause();
        portfolio.saveResidences();
    }
    //...
}
```

# The Serialization Mechanism

```
public class Residence
{
    public UUID id;

    public String geolocation;
    public Date date;
    public boolean rented;

    private static final String JSON_ID = "id" ;
    private static final String JSON_GEOLOCATION = "geolocation" ;
    private static final String JSON_DATE = "date" ;
    private static final String JSON_RENTED = "rented" ;
}
```

```
public class PortfolioSerializer
{
    private Context mContext;
    private String mFilename;

    public PortfolioSerializer(Context c, String f)
    {
        mContext = c;
        mFilename = f;
    }

    public void saveResidences(ArrayList<Residence> r)
    {
        //...
    }

    public ArrayList<Residence> loadResidences() thro
    {
        //...
    }
}
```

```
[
  {
    "date" : 1410000509060,
    "geolocation": "52.253456, -7.187162",
    "id" : "6a8d44df-7534-43c5-8130-444678fcc187",
    "rented" : false
  },
  {
    "date" : 1410000511943,
    "geolocation": "52.253456, -7.187162",
    "id" : "483eb7da-0dd2-41d3-b53e-ba3d51f9c55d",
    "rented" : false
  },
  {
    "date" : 1410649200000,
    "geolocation": "12,12",
    "id" : "9cf116a4-2c75-40a8-b28b-f3bc9a3939a6",
    "rented" : true
  }
]
```

```
return "Registered: " + DateFormat.getDateInstance().format(date);
}
```

# saveResidences

---

- Create a JSONArray object
- Place each residence in turn into the object
- Write the object to the file
- Close the file
- If any exceptions occur, 'propagate' to the caller (whoever that is)

```
public void saveResidences(ArrayList<Residence> residences)
    throws JSONException, IOException
{
    // build an array in JSON
    JSONArray array = new JSONArray();
    for (Residence c : residences)
        array.put(c.toJSON());

    // write the file to disk
    Writer writer = null;
    try
    {
        OutputStream out = mContext.openFileOutput(mFilename, Context.MODE_PRIVATE);
        writer = new OutputStreamWriter(out);
        writer.write(array.toString());
    }
    finally
    {
        if (writer != null)
            writer.close();
    }
}
```

# LoadResidences

- Create Residence Array
- Attach a reader to the file (via a buffered reader)
- Read the file into a string
- Tokenize the string into individual json objects
- Extract each object in turn and create a new Residence Object from it
- Add to our Residence list

```
public ArrayList<Residence> loadResidences() throws IOException, JSONException
{
    ArrayList<Residence> residences = new ArrayList<Residence>();
    BufferedReader reader = null;
    try
    {
        // open and read the file into a StringBuilder
        InputStream in = mContext.openFileInput(mFilename);
        reader = new BufferedReader(new InputStreamReader(in));
        StringBuilder jsonString = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null)
        {
            // line breaks are omitted and irrelevant
            jsonString.append(line);
        }
        // parse the JSON using JSNTokener
        JSONArray array = (JSONArray) new JSNTokener(jsonString.toString()).nextValue();
        // build the array of residences from JSONObjects
        for (int i = 0; i < array.length(); i++)
        {
            residences.add(new Residence(array.getJSONObject(i)));
        }
    }
    catch (FileNotFoundException e)
    {
        // we will ignore this one, since it happens when we start fresh
    }
    finally
    {
        if (reader != null)
            reader.close();
    }
    return residences;
}
```

# Open Android Device Monitor (ADM) from Studio

The screenshot shows the Android Studio interface with the Android Device Monitor (ADM) window open. The ADM window is divided into several tabs: Devices, Threads, Heap, Allocation Tracker, Network Statistics, File Explorer, Emulator Control, and System Information. The File Explorer tab is active, showing a file tree for the application. The 'org.wit.myrent' directory is expanded, showing sub-directories like 'cache' and 'files'. The 'files' directory is selected, and the file 'portfolio.json' is highlighted. The LogCat window is visible at the bottom, showing a list of log messages.

Name	Size	Date	Time	Permissions	Info
com.example.android.apis		2015-08-05	11:00	drwxr-x--x	
com.example.android.live...		2015-08-05	11:00	drwxr-x--x	
com.example.android.soft...		2015-08-05	11:00	drwxr-x--x	
com.example.donation		2015-09-17	18:31	drwxr-x--x	
com.example.edeleastar...		2015-08-05	11:03	drwxr-x--x	
com.google.android.apps...		2015-08-05	11:01	drwxr-x--x	
com.google.android.gms		2015-10-12	13:44	drwxr-x--x	
com.google.android.gsf		2015-08-05	11:01	drwxr-x--x	
com.google.android.gsf.l...		2015-08-05	11:00	drwxr-x--x	
com.google.android.street		2015-08-05	11:00	drwxr-x--x	
com.svox.pico		2015-08-05	11:03	drwxr-x--x	
jp.co.omronsoft.openwnn		2015-08-05	11:00	drwxr-x--x	
org.wit.myrent		2015-10-12	13:44	drwxr-x--x	
cache		2015-09-29	12:59	drwxrwx--x	
files		2015-10-12	13:47	drwxrwx--x	
portfolio.json	292	2015-10-12	13:47	-rw-rw----	
lib		2015-09-29	12:59	lrwxrwxrwx	-> /data/a...
dontpanic		2015-08-05	11:00	drwxr-x--x	
drm		2015-08-05	11:00	drwxrwx--x	
local		2015-08-05	11:00	drwxr-x--x	
lost+found		1970-01-01	01:00	drwxrwx---	
media		2015-08-05	11:00	drwxrwx---	
mediadrms		2015-08-05	11:00	drwxrwx---	
misc		2015-08-05	11:00	drwxrwx--t	
nativebenchmark		2015-02-14	16:45	drwxrwx--x	
nativetest		2015-02-14	16:45	drwxrwx--x	
property		2015-10-14	06:43	drwx-----	
resource-cache		2015-08-05	11:00	drwxrwx--x	
security		2015-08-05	11:00	drwx--x--x	

Level	Time	PID	TID	Application	Tag	Text
I	10-14 06:45:57.964	1782	1789	com.google.android...	art	Debugger is no longer active
I	10-14 06:45:57.964	1849	1856	android.process.m...	art	Debugger is no longer active
I	10-14 06:45:57.964	1863	1872	com.android.calen...	art	Debugger is no longer active
I	10-14 06:45:57.965	1929	1936	com.android.provi...	art	Debugger is no longer active
I	10-14 06:45:57.965	1956	1964	com.android.dialer	art	Debugger is no longer active
I	10-14 06:45:57.965	1972	1974	com.android.keych...	art	Debugger is no longer active
I	10-14 06:45:57.966	1998	2004	com.android.manag...	art	Debugger is no longer active
I	10-14 06:45:57.966	2025	2032	com.android.mms	art	Debugger is no longer active
I	10-14 06:45:57.966	2124	2132	com.android.setti...	art	Debugger is no longer active
I	10-14 06:45:57.967	2166	2173	com.android.email	art	Debugger is no longer active
I	10-14 06:45:57.967	2190	2197	com.android.excha...	art	Debugger is no longer active
I	10-14 06:45:57.967	2288	2290	com.svox.pico	art	Debugger is no longer active
I	10-14 06:45:57.968	1405	1433	com.android.phone	art	Debugger is no longer active
I	10-14 06:45:57.968	1649	1656	com.google.proces...	art	Debugger is no longer active
I	10-14 06:45:57.968	1900	1902	org.wit.myrent	art	Debugger is no longer active
I	10-14 06:46:23.093	1678	3730	com.google.proces...	VacuumService	Vacuum at: now=1444801583093 tag=VacuumService

• Select File Explorer

• Browse / data/data/.. application id



Android Device Monitor

Quick Access DDMS Emulator Control System Information

Devices Threads Heap Allocation Tracker Network Statistics File Explorer Emulator Control System Information

Name	Online	Process ID	Package Name
Nexus_5...	Online		Nexus_5...
com.googl...	1728	8600	
com.andro...	1411	8601	
android.pr...	1539	8602	
system_pr...	1220	8603	
com.andro...	1863	8604	

Name	Size	Date	Time	Permissions	Info
com.example.android.apis		2015-08-05	11:00	drwxr-x--x	
com.example.android.live...		2015-08-05	11:00	drwxr-x--x	
com.example.android.soft...		2015-08-05	11:00	drwxr-x--x	
com.example.donation		2015-09-17	18:31	drwxr-x--x	
com.example.edeleastar...		2015-08-05	11:03	drwxr-x--x	
com.google.android.apps...		2015-08-05	11:01	drwxr-x--x	

File Explorer View:

- jp.co.omronsoft.openwnn (2015-08-05)
- org.wit.myrent
  - cache
  - files
    - portfolio.json (292)
    - lib
  - dontpanic

LogCat View:

Search for messages. Accepts Java regexes. Prefix with #

Level	Time	PID	TID	Application	Tag
I	10-14 06:45:57.964	1782	1789	com.google.android...	art
I	10-14 06:45:57.964	1849	1856	android.process.m...	art
I	10-14 06:45:57.964	1863	1872	com.android.calen...	art
I	10-14 06:45:57.965	1929	1936	com.android.provi...	art
I	10-14 06:45:57.965	1956	1964	com.android.dialer	art
I	10-14 06:45:57.965	1972	1974	com.android.keych...	art
I	10-14 06:45:57.966	1998	2004	com.android.manag...	art
I	10-14 06:45:57.966	2025	2032	com.android.mms	art
I	10-14 06:45:57.966	2124	2132	com.android.setti...	art
I	10-14 06:45:57.967	2166	2173	com.android.email	art
I	10-14 06:45:57.967	2190	2197	com.android.excha...	art
I	10-14 06:45:57.967	2288	2290	com.svox.pico	art
I	10-14 06:45:57.968	1405	1433	com.android.phone	art
I	10-14 06:45:57.968	1649	1656	com.google.proces...	art
I	10-14 06:45:57.968	1900	1902	org.wit.myrent	art
I	10-14 06:46:23.093	1678	3730	com.google.proces...	Vacuum

Save file to host

```
[
  {
    "id": "6c4bae0b-9f59-44bf-9592-5c60c9e7f6e8",
    "geolocation": "52.253456,-7.187162",
    "date": 1444654019019,
    "rented": false,
    "tenant": ": none presently"
  },
  {
    "id": "93042416-e5b5-460e-b9a3-00a629a0f2ea",
    "geolocation": "52.253456,-7.187162",
    "date": 1444654029951,
    "rented": true,
    "tenant": ": none presently"
  }
]
```

# Alternatively ... log in to emulator

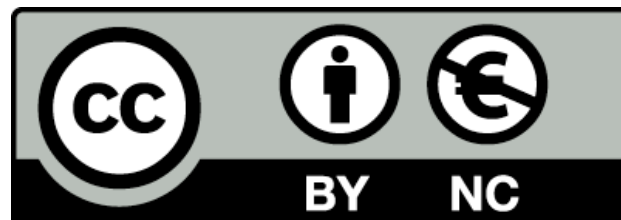
- “abd shell” + navigate using shell commands to folder

```
2. adb
Last login: Tue Oct 14 15:50:12 on ttys001
localhost:~ edeleastar$ adb shell
root@vbox86p:/ # cd data/data
root@vbox86p:/data/data # cd org.wit.myrent
root@vbox86p:/data/data/org.wit.myrent # cd files
root@vbox86p:/data/data/org.wit.myrent/files # cat portfolio.json
[{"date":1410000509060,"geolocation":"52.253456, -7.187162","id":"6a8d44df-7534-43c5-8130-444678fcc187","rented":false},{"date":1410000511943,"geolocation":"52.253456, -7.187162","id":"483eb7da-0dd2-41d3-b53e-ba3d51f9c55d","rented":false},{"date":1410649200000,"geolocation":"12,12","id":"9cf116a4-2c75-40a8-b28b-f3bc9a3939a6","rented":true}]root@vbox86p:/data/data/org.wit.myrent/files #
```

```
[
  {
    "date"      : 1410000509060,
    "geolocation": "52.253456, -7.187162",
    "id"        : "6a8d44df-7534-43c5-8130-444678fcc187",
    "rented"    : false
  },
  {
    "date"      : 1410000511943,
    "geolocation": "52.253456, -7.187162",
    "id"        : "483eb7da-0dd2-41d3-b53e-ba3d51f9c55d",
    "rented"    : false
  },
  {
    "date"      : 1410649200000,
    "geolocation": "12,12",
    "id"        : "9cf116a4-2c75-40a8-b28b-f3bc9a3939a6",
    "rented"    : true
  }
]
```

```
adb shell
root@vbox86p:/ # cd data/data
root@vbox86p:/data/data # cd org.wit.myrent
root@vbox86p:/data/data/org.wit.myrent # cd files
root@vbox86p:/data/data/org.wit.myrent/files # cat portfolio.json
[{"date":1410000509060,"geolocation":"52.253456,
-7.187162","id":"6a8d44df-7534-43c5-8130-444678fcc187","rented":fal
se}, {"date":1410000511943,"geolocation":"52.253456,
-7.187162","id":"483eb7da-0dd2-41d3-b53e-
ba3d51f9c55d","rented":false}, {"date":
1410649200000,"geolocation":"12,12","id":"9cf116a4-2c75-40a8-b28b-
f3bc9a3939a6","rented":true}]root@vbox86p:/data/data/
org.wit.myrent/files #
```





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

