# Test-Project for Applicants (Android, iOS, WindowsPhone)

## Network

The application queries a http-address with a GET-request to receive a file.
Network-Address: http://foo.bragi.net/numbers.json

```
{
  "numbers": [
    4, 150, 12, 21, 136, 16, 3
  ]
}
```

## Parsing

The file received in JSON-format contains a number-array which content needs to be parsed.

The numbers must be converted into bitfields which will be split afterwards into three different numbers:

- [0, 1] (two least significant bits): section number
- [2, 6]: item number
- [7]: checkmark set or not

The section number must be matched against the section-enum which looks like:

- Section1: 0
- Section2: 1
- Section3: 2
- Section4: 3

The item number must me matched against the item-enum which looks like:

- Item1: 0
- Item2: 1
- Item3: 2
- Item4: 3
- Item5: 4
- Item6: 5

The checkmark bit sets the checkbox

- unchecked: 0
- checked: 1

Example:

- Value: 140
- as bitfield: 10001100
- Section bits: 00 = 0
- Item bits: 00011 = 3
- Checkmark bit: 1 = 1
- Result: Section1, Item4, checked

## Layout

According to the information read from the bitfield, a new tablerow gets inserted into a table (occupying the whole screen of the device). For every new section, a header row gets inserted which headlines the according items that are referred to it. Furthermore, every non-header-row holds a checkbox that is either checked or not checked dependent on the checkmark-bit.

The example from above would look like:

```
----------------------
SECTION 1
----------------------
Item4            ✅
----------------------
```

After reading other values, the table could look like (note that sections and items are ordered by their numeric value):

```
----------------------
SECTION1
----------------------
Item4            ✅
----------------------
Item5            ❌
----------------------
SECTION2
----------------------
Item2            ✅
```

# Requirements

- The application shall work on mobile phones running either Android 4.3+, iOS 8.x or Windows Phone 8.1+
- No external libraries or frameworks
- Clean, maintainable code
- Basic documentation
- Robust error handling
- Effective, performant processes
- Clear seperation between data and ui
- Full project files (Android: Android Studio or Eclipse, iOS: XCode, Windows Phone: Visual Studio)