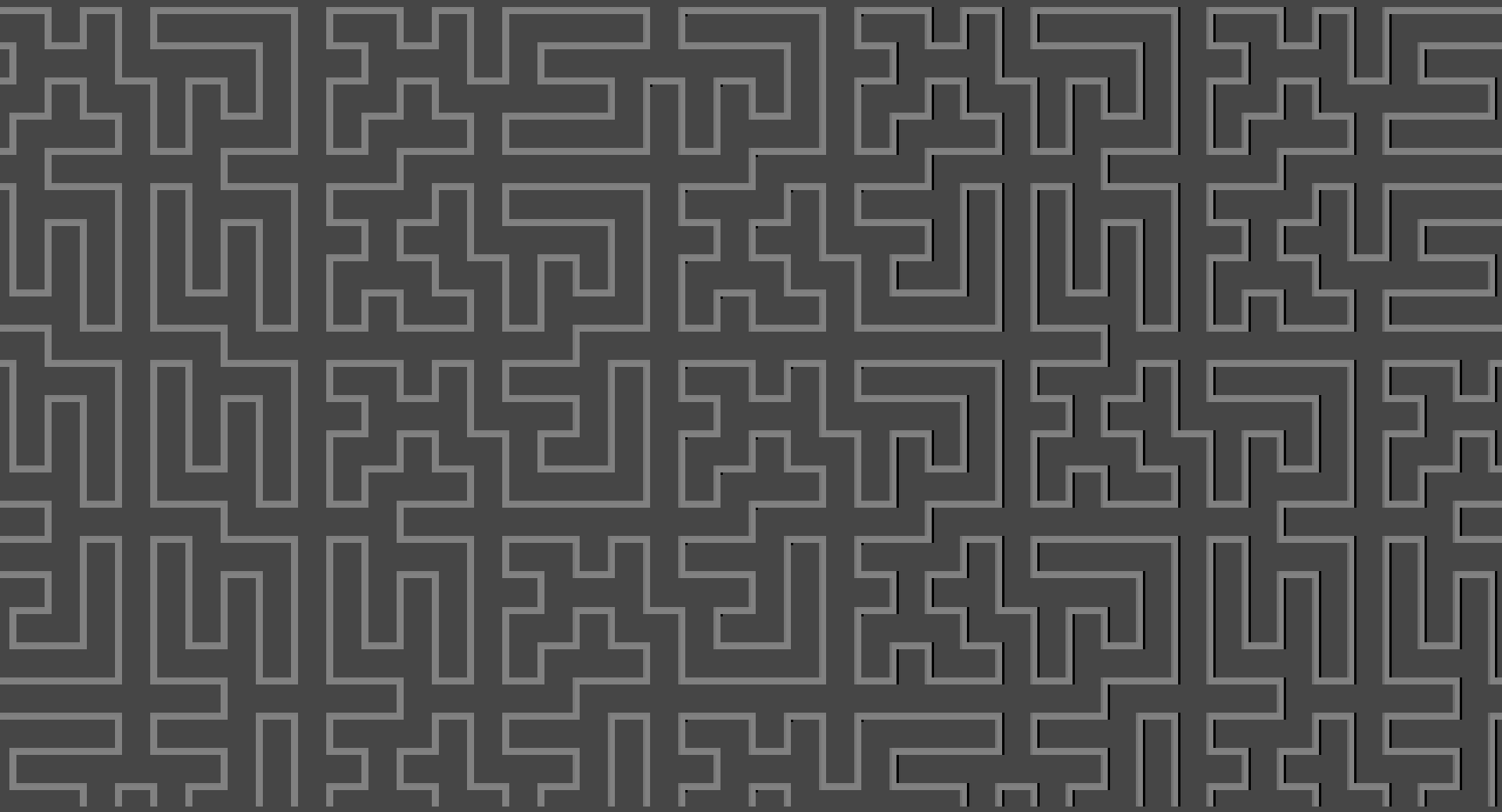


MOBILE DEVELOPMENT



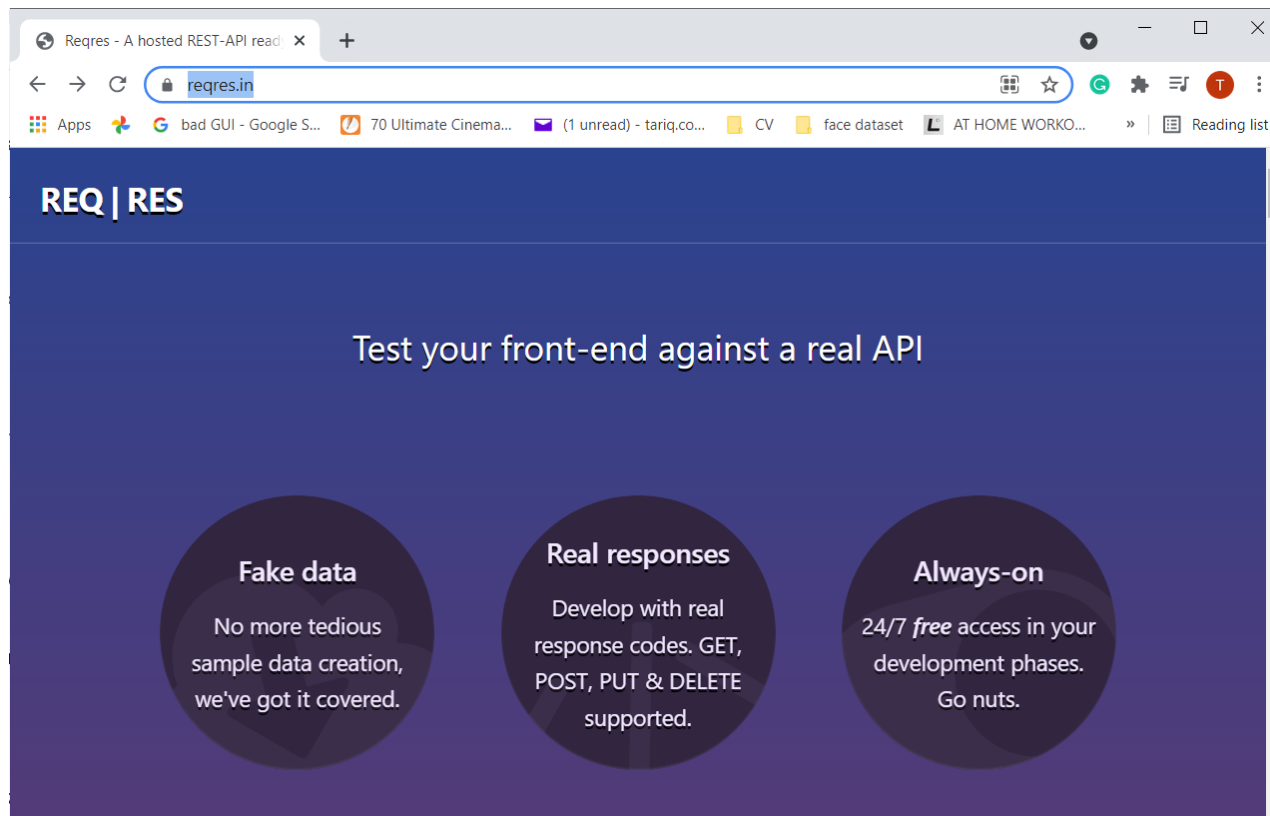
Web Services and JSON Using Volley

Web Services

- ❑ Google Volley library setup
- ❑ Open APIs – we will use apigeonet.org.nz
- ❑ Requests
- ❑ JSON Responses

Review- Test API

□ <https://reqres.in/>



HTTP GET

□ reqres.in/api/users?page=2

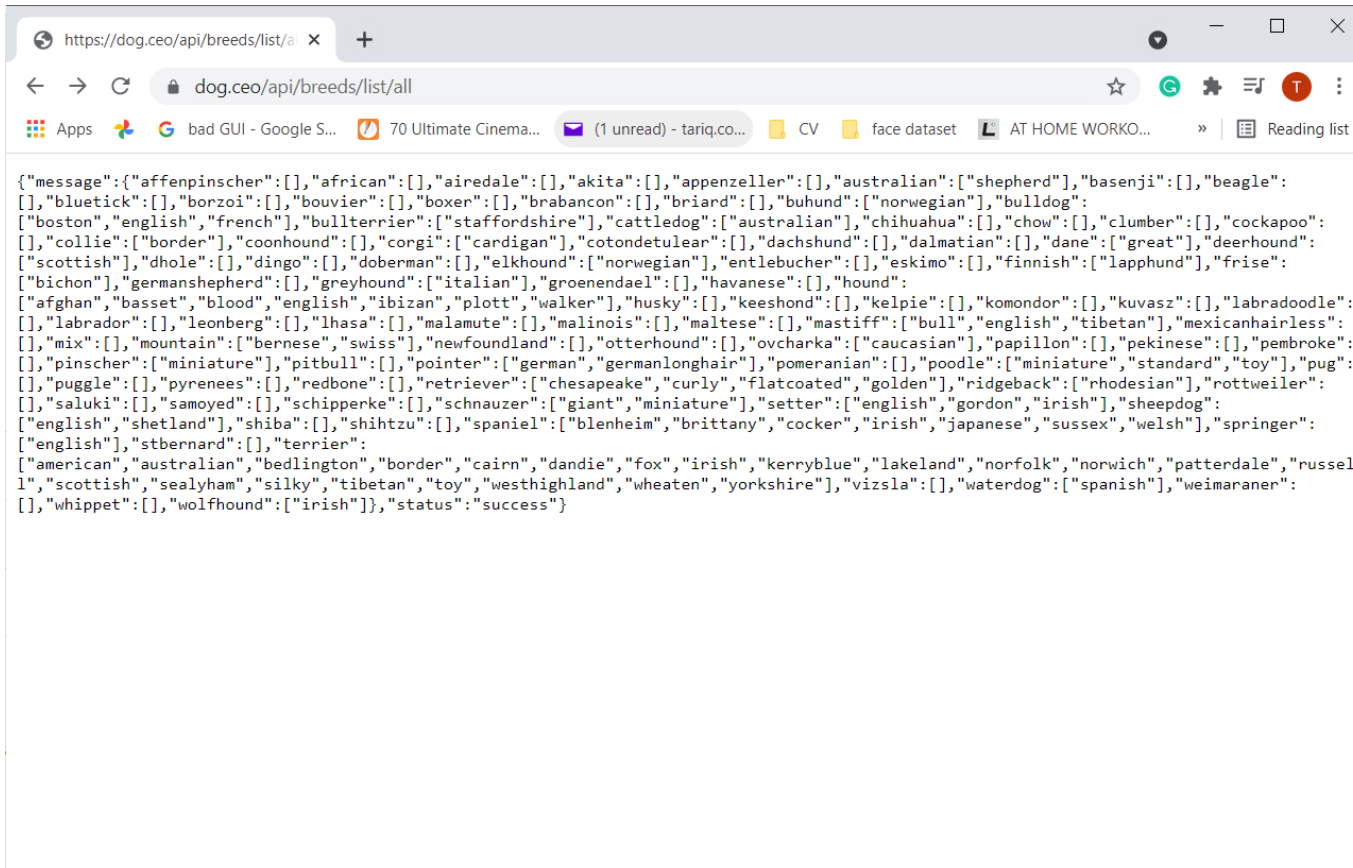
The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and buttons for '+ New', 'Import', 'Runner', and 'My Workspace'. The left sidebar shows a 'History' tab with a list of recent requests, including the current one: 'GET reqres.in/api/users?page=2'. The main panel displays the 'Untitled Request' for the GET method to 'reqres.in/api/users?page=2'. The 'Params' tab is active, showing a 'Query Params' table with a single entry: 'page' with value '2'. The 'Body' tab is also visible, showing the response in 'Pretty' format. The response is a JSON object with the following structure:

```
1 {
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",
10      "first_name": "Michael",
11      "last_name": "Lawson",
12      "avatar": "https://reqres.in/img/faces/7-image.jpg"
13    }
14  ]
15 }
```

The status bar at the bottom shows '200 OK', '38 ms', and '1.93 KB'. The bottom right corner of the image has a red box with the number '4'.

Dogs API

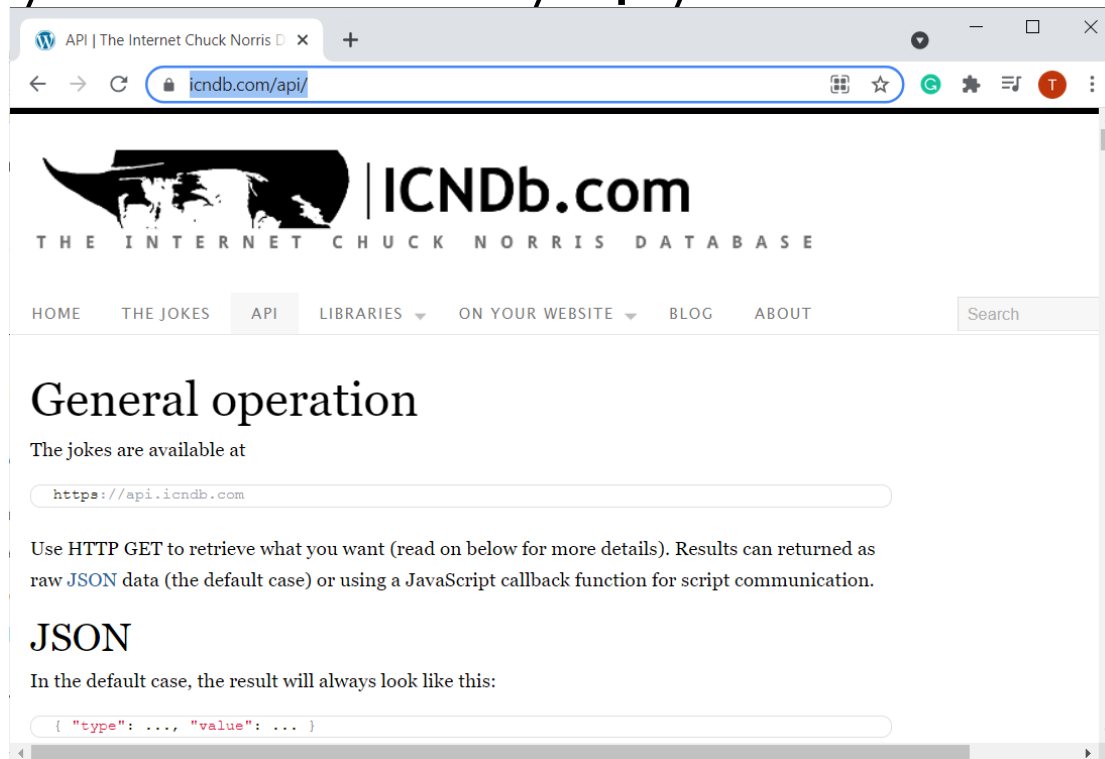
□ <https://dog.ceo/dog-api/>



```
{
  "message": {
    "affenpinscher": [],
    "african": [],
    "airedale": [],
    "akita": [],
    "appenzeller": [],
    "australian": [
      "shepherd"
    ],
    "basenji": [],
    "beagle": [
      "bluetick",
      "borzoi",
      "bouvier",
      "boxer",
      "brabancon",
      "briard",
      "buhund": [
        "norwegian"
      ],
      "bulldog": [
        "boston",
        "english",
        "french",
        "bullterrier": [
          "staffordshire"
        ],
        "cattledog": [
          "australian"
        ],
        "chihuahua": [],
        "chow": [],
        "clumber": [],
        "cockapoo": [
          "collie": [
            "border"
          ],
          "coonhound": [],
          "corgi": [
            "cardigan"
          ],
          "cotondetulear": [],
          "dachshund": [],
          "dalmatian": [],
          "dane": [
            "great"
          ],
          "deerhound": [
            "scottish"
          ],
          "dhole": [],
          "dingo": [],
          "doberman": [],
          "elkhound": [
            "norwegian"
          ],
          "entlebucher": [],
          "eskimo": [],
          "finnish": [
            "lapphund"
          ],
          "frise": [
            "bichon"
          ],
          "germanshepherd": [],
          "greyhound": [
            "italian"
          ],
          "groenendael": [],
          "havanese": [],
          "hound": [
            "afghan",
            "basset",
            "blood",
            "english",
            "ibizan",
            "plott",
            "walker",
            "husky": [],
            "keeshond": [],
            "kelpie": [],
            "komondor": [],
            "kuvasz": [],
            "labradoodle": [
              "labrador": [],
              "leonberg": [],
              "lhasa": [],
              "malamute": [],
              "malinois": [],
              "maltese": [],
              "mastiff": [
                "bull",
                "english",
                "tibetan"
              ],
              "mexicanhairless": [
                "mix": [
                  "mountain": [
                    "bernese",
                    "swiss"
                  ],
                  "newfoundland": [],
                  "otterhound": [],
                  "ovcharka": [
                    "caucasian"
                  ],
                  "papillon": [],
                  "pekinese": [],
                  "pembroke": [
                    "pinscher": [
                      "miniature"
                    ],
                    "pitbull": [],
                    "pointer": [
                      "german",
                      "germanlonghair"
                    ],
                    "pomeranian": [],
                    "poodle": [
                      "miniature",
                      "standard",
                      "toy"
                    ],
                    "pug": [
                      "puggle": [
                        "pyrenees": [
                          "redbone": [
                            "retriever": [
                              "chesapeake",
                              "curly",
                              "flatcoated",
                              "golden",
                              "ridgeback": [
                                "rhodesian"
                              ],
                                "rottweiler": [
                                  "saluki": [],
                                  "samoyed": [],
                                  "schipperke": [],
                                  "schnauzer": [
                                    "giant",
                                    "miniature"
                                  ],
                                    "setter": [
                                      "english",
                                      "gordon",
                                      "irish",
                                      "sheepdog": [
                                        "english",
                                        "shetland",
                                        "shiba": [],
                                        "shihtzu": [],
                                        "spaniel": [
                                          "blenheim",
                                          "brittany",
                                          "cocker",
                                          "irish",
                                          "japanese",
                                          "sussex",
                                          "welsh",
                                          "springer": [
                                            "english",
                                            "stbernard": [],
                                            "terrier": [
                                              "american",
                                              "australian",
                                              "bedlington",
                                              "border",
                                              "cairn",
                                              "dandie",
                                              "fox",
                                              "irish",
                                              "kerryblue",
                                              "lakeland",
                                              "norfolk",
                                              "norwich",
                                              "patterdale",
                                              "russell",
                                              "scottish",
                                              "sealyham",
                                              "silky",
                                              "tibetan",
                                              "toy",
                                              "westhighland",
                                              "wheaten",
                                              "yorkshire",
                                              "vizsla": [],
                                              "waterdog": [
                                                "spanish"
                                              ],
                                                "weimaraner": [
                                                  "whippet": [],
                                                  "wolfhound": [
                                                    "irish"
                                                  ]
                                                ],
                                                "status": "success"
                                              ]
                                            ]
                                          ]
                                        ]
                                      ]
                                    ]
                                  ]
                                ]
                              ]
                            ]
                          ]
                        ]
                      ]
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  }
}
```

Jokes

□ <https://www.icndb.com/api/>



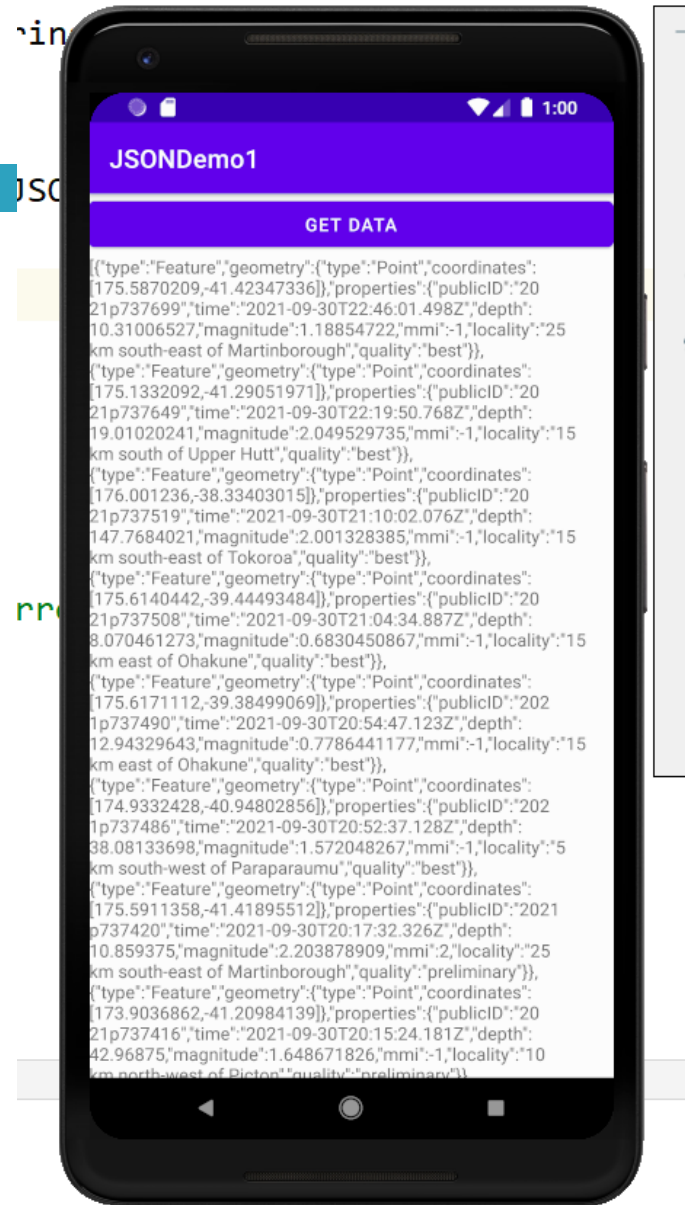
JSON Properties

- ❑ Define objects and arrays with key value pairs
- ❑ Square brackets define arrays
- ❑ Curly Brackets define objects
- ❑ Key-values define object properties

```
[  
  {  
    "color": "red",  
    "value": "#f00"  
  },  
  {  
    "color": "green",  
    "value": "#0f0"  
  },  
  {  
    "color": "blue",  
    "value": "#00f"  
  },  
  {  
    "color": "cyan",  
    "value": "#0ff"  
  }  
]
```

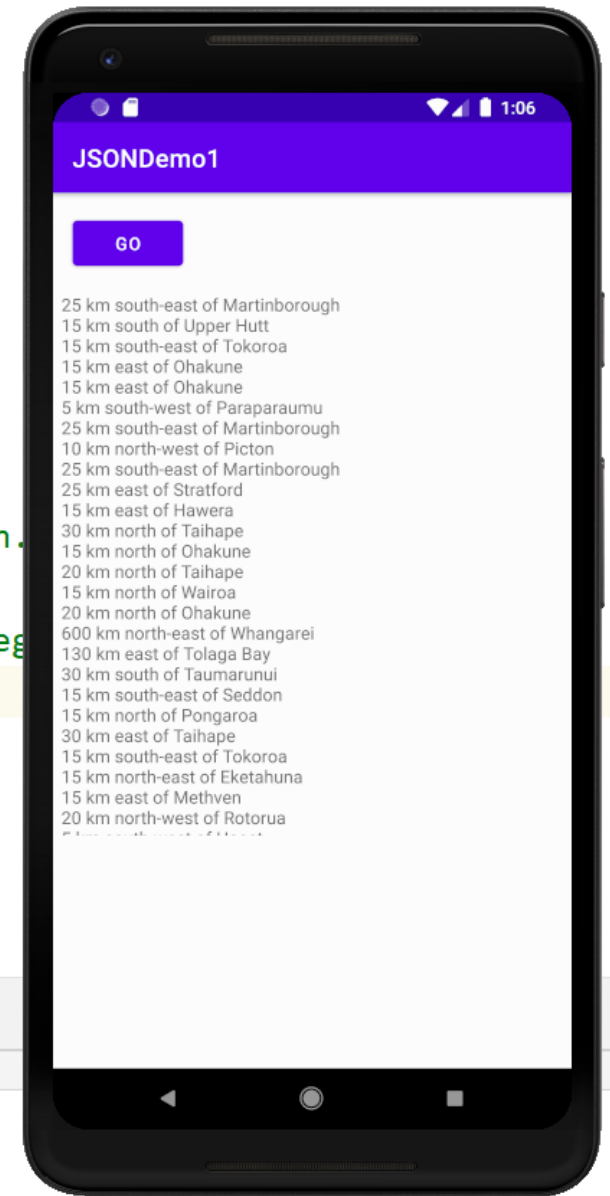
GeoNet App

- ❑ Just dumping the entire JSON response note the nesting of the of the server response



GeoNet Localities

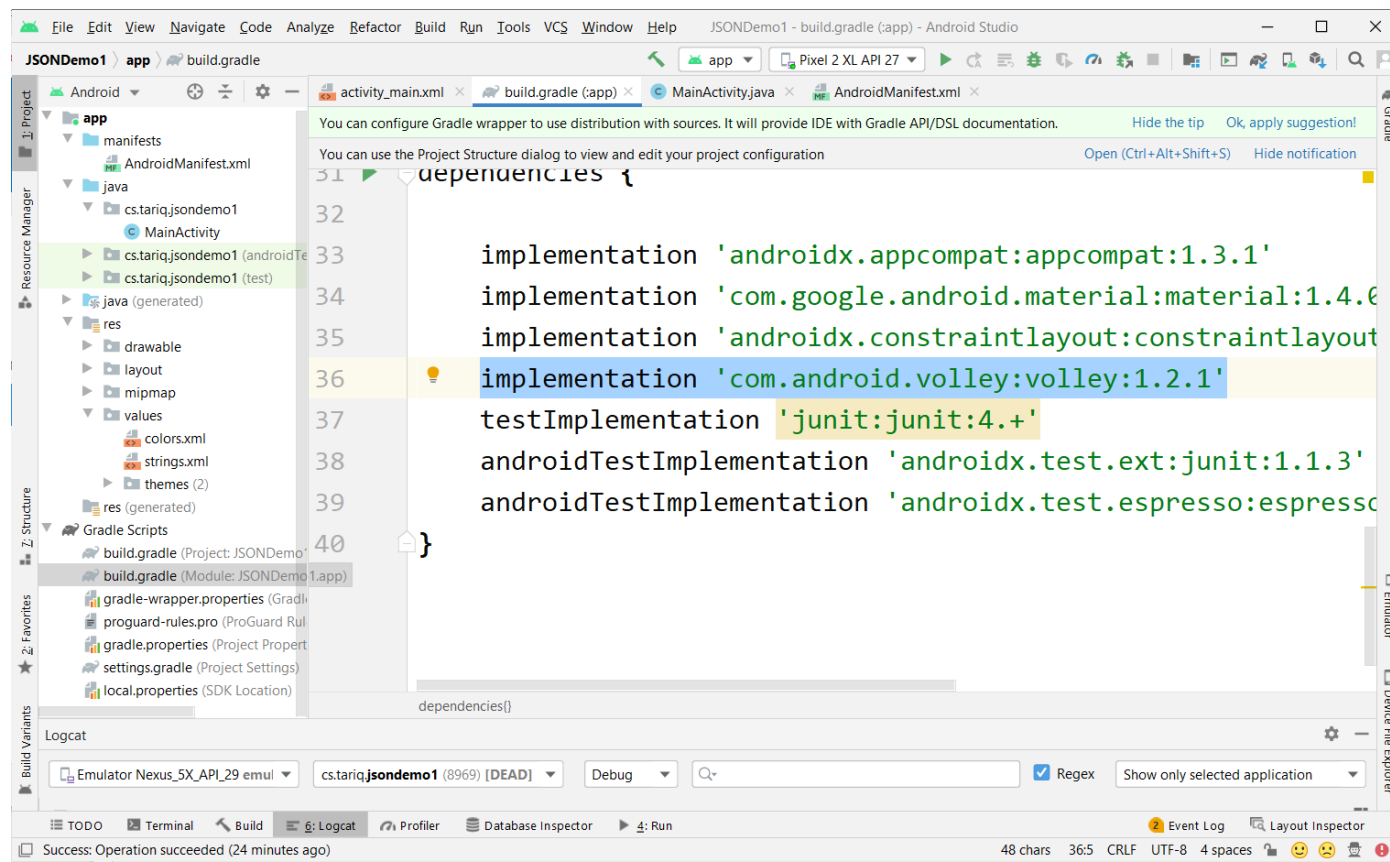
- ❑ Slice and dice the data to dig out the locality information from the array, and nested objects



Gradle

- Add new dependency to the bottom of your build.gradle file

implementation 'com.android.volley:volley:1.2.1'



Permissions

- uses-permission tags before application tags

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cs.tariq.jsondemo1">

    <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:supportsRtl="true"
        android:theme="@style/Theme.JSONDemo1">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Create A Request Queue

- ❑ Automates the entire connection process
- ❑ Some frameworks are a lot more complicated
- ❑ Just use the `Volley.newRequestQueue` factory method to define a queue
- ❑ Need to
 - ▣ `import com.android.volley.RequestQueue;`
- ❑ Does all of the hard work of reading / writing requests across the socket

```
RequestQueue queue = Volley.newRequestQueue(getApplicationContext());
```

Create JSON Request

- Define the request object and add it to the request queue

```
String url ="https://api.geonet.org.nz/quake?MMI=7";
```

```
JsonObjectRequest jsonObjRequest = new JsonObjectRequest
    (Request.Method.GET, url, null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    });
```

```
// Add the request to the RequestQueue.
queue.add(jsonObjRequest);
```

JSON Response Callback

- onResponse callback method does all of the work

```
new Response.Listener<JSONObject>() {  
  
    @Override  
    public void onResponse(JSONObject response) {  
  
        StringBuilder localities = new StringBuilder();  
  
        try {  
            JSONArray data = response.getJSONArray("features");  
  
            for(int index = 0; index < data.length(); index++)  
            {  
                JSONObject quake = data.getJSONObject(index);  
  
                JSONObject properties = quake.getJSONObject("properties");  
                localities.append(properties.getString("locality")+"\n");  
            }  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
        output.setText(localities.toString());  
    }  
}
```

Exercise

- Add some extra code and UI elements to extract the magnitude and point co-ordinates
- Read through:
 - ▣ <https://developer.android.com/training/volley>
 - ▣ <https://developer.android.com/training/volley/simple>
 - ▣ <https://developer.android.com/training/volley/requestqueue>
 - ▣ <https://developer.android.com/training/volley/request-custom>
- Work on your assignment task