- Introduction to JSON or JavaScript Object Notation

# TOPICS

- Introduction
- JSON
- JSON Array
- Retrieving data  as JSON Object
- Examples

# EXAMPLE

- - Data in database on server side →( e.g. MySQL )
- If we want to send these data to client side (e.g. Mobile App) to be displayed on screen

( how to send the data?

- Ans: using **JSON** or **XML** )

| id | name | phone | email |
|---|---|---|---|
| 1 | Tim | 555-5555 | tim@yahoo.ca |
| 2 | Jane | 444-555 | jane@yahoo.ca |
| 3 | joe | 333-3333 | joe@gamil.com |
| 4 | ….. | ……. | …… |

# JSON

- stand for **JavaScript Object Notation**
- popular data interchange format
- text-based
- Lightweight
- Human /machine-readable
- format for data exchange between clients and servers
- closely resemble JavaScript objects
- Supported by many languages e.g. C#, PHP, Java, C++, Python, and Ruby.

# JSON VS XML

- Prior to JSON, XML was considered to be the chosen data interchange format
- XML often tend to be
  - heavy
  - Verbose
    - Every element in the tree has a name, and the element must be enclosed in a matching pair of tags.
  - take up a lot of bandwidth

- The JSON specification states that data can be structured in either of the two following compositions:
  1. A collection of name/value pairs (JSON Object)
  2. An ordered list of value ( JSON Array)

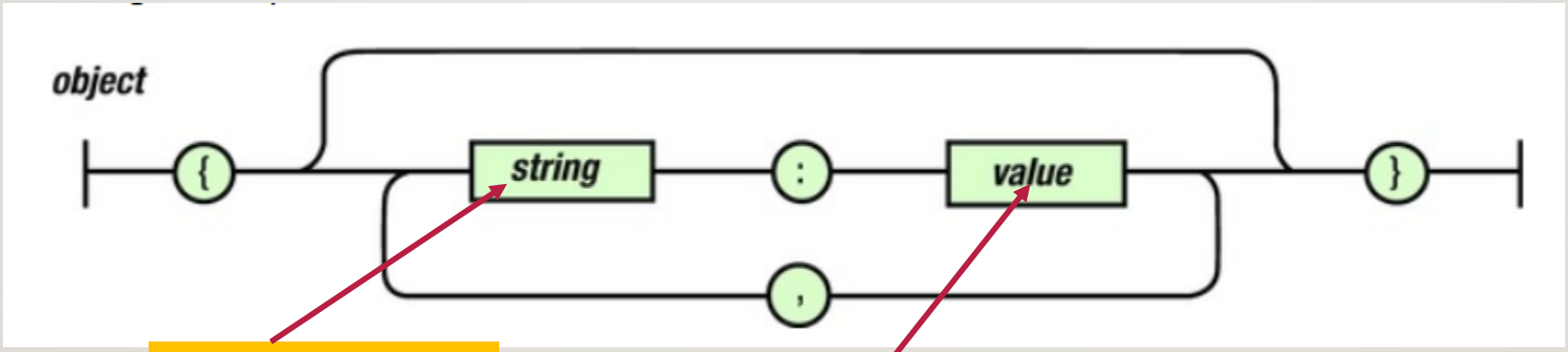
# JSON OBJECT*

- is an unordered collection of **name/value** pairs.
- with **colons** between the names and values
- **put** methods for adding or replacing values by name
- Values can be : String, Boolean, Number, JSONObject or JSONArray
- **get** and **opt** methods for accessing the values by name
- JSONObject  json = new JSONObject().put("JSON", "Hello, World!")
- {"JSON": "Hello, World"}

*http://www.json.org/javadoc/org/json/JSONObject.html

# JSON OBJECT

- name- value pair
-




Example >>
jsonObj = { “color”:“red” ,“size” : 40 }

- value can hold
  - String → { “name” : “joe”}
  - Number→ {“age” : 23 }
  - Boolean → { “validNumber” : true}
  - JSONObject → {“user” :{“name”:”Jane”, “age”:20}}
  - Array → {“array”:[ 3, 5 , 4] }
  - null

- Examples { “name”:”joe”, “age”:22}

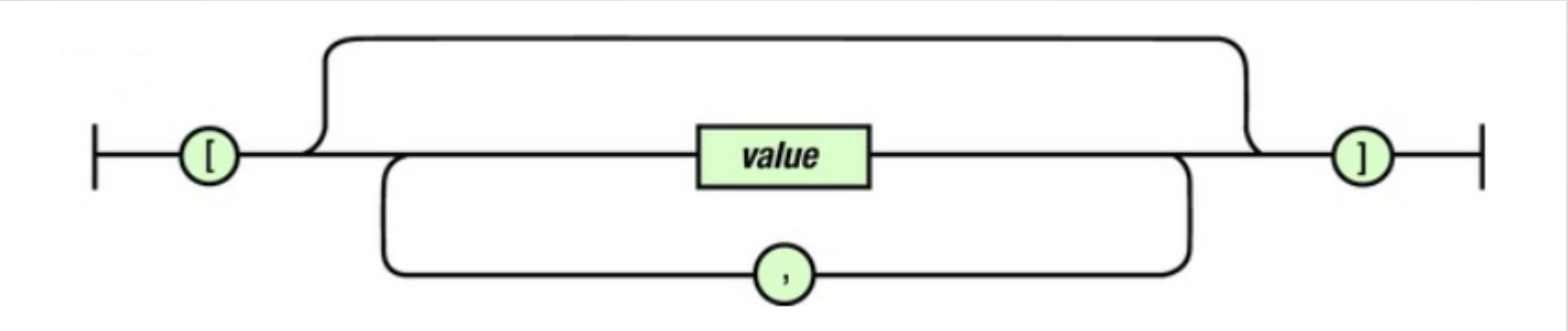
# EXERCISE 1

- Create JSON Object using the following info.

- name = Tim
- phone  = 3456
- email  = user@emailcom
- age =  22
-

# JSON ARRAY

Json Array start with [
Values separated by comma




jsonArray1 = [ “Math” ,“ Java”,“C#”]

jsonArray = [{ “test” :80} , {“test2” : 70 }, {“test3” : 65}]

# EXAMPLE

- Create JSON Array that represent the users shown in the table below.

| id | name | phone | email |
|---|---|---|---|
| 1 | Tim | 555-5555 | tim@yahoo.ca |
| 2 | Jane | 444-555 | jane@yahoo.ca |
| 3 | joe | 333-3333 | joe@gamil.com |
| 4 | ….. | ……. | …… |

# EXAMPLES

- myObj = {“name”:”Tim”, “age”: 20, “gender”, “male”}
- String name = myObj.getString(”name”);

- JSONObject ob = new JSONObject();
- ob.put(“user”,myObj);
- Ob = {“user”:{“name”:”Tim”, “age”: 20, “gender”, “male”}};

# EXAMPLE

```
JSONObject user1 = new JSONObject();
JSONObject user2 = new JSONObject();
JSONArray jsonArray = new JSONArray();
try {
    user1.put("name" , "Tim");
    user1.put("age" , 20);

    user2.put("name" , "Jane");
    user2.put("age" , 21);

     jsonArray.put(user1);
    jsonArray.put(user2);
```

```
String name = user1.getString("name");
    Log.d(TAG, "name: "+name);
    Log.d(TAG, "user1: "+user1.toString());
    Log.d(TAG, "user2 : "+user2.toString());
    Log.d(TAG, "jsonArray: "+jsonArray.toString());

} catch (JSONException e) {
    e.printStackTrace();
```

```
}
```

# JSON ARRAY

- jsonArray =

```
[
  {”name":“Tim", ”age":23},
   {”name":“Jane", ”age":33},
    {”name":“Joe", ”age":21}
]
```

- ```
  for (int i = 0; i < jsonArray.length(); i++) {
      JSONObject c = (JSONObject) jsonArray.get(i);

     String userName = c.getString("name");
      int userAge = c.getInt("age");
     Log.d(TAG, "userName: "+userName);
     Log.d(TAG, "userAge: "+userAge);

  }
  ```
  ```

# STRING TO JSON OBJECT

```
// well formed JSON String
String val = "{\"name\":\"Tim\",\"age\":20}";
try {
    JSONObject obj =  new JSONObject(val);
    String name = obj.getString("name");
    Log.d(TAG, "name: "+name);
} catch (JSONException e) {
    e.printStackTrace();
    Log.d(TAG, "Error  ");
}
```

# EXAMPLE

- Retrieving data as JSON. From a server and displaying it
- url
- https://jsonplaceholder.typicode.com/todos

# REFERENCES


- http://developer.android.com/index.html
- https://developer.android.com/reference/org/json/JSONObject
- https://developer.android.com/reference/org/json/JSONArray