When we want to create json from **single string which is in json format already**  and  
**select elements** from that JSON

public String buildResult(String response) {  
 String errorCode = "";  
 String redirectUrl = "";  
 try {  
 org.json.simple.parser.JSONParser parser = new JSONParser();  
 Object obj = parser.parse(response);  
 JSONObject objson = (JSONObject) obj;  
 if (response.contains("code")) {  
 errorCode = objson.get("code").toString();  
 }  
 if (response.contains("token")) {  
 String token = objson.get("token").toString();  
 redirectUrl = this.preUrlRedirect + token;  
 }  
 } catch (Exception ex) {  
 this.logger.error(ex.getMessage(), ex);  
 }  
 return errorCode + "|" + redirectUrl;  
}

when we want to make **a json object from many plain texts :**

<dependency>  
 <groupId>com.google.code.gson</groupId>  
 <artifactId>gson</artifactId>  
 <type>jar</type>  
</dependency>  
  
<dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
</dependency>  
  
<dependency>  
 <groupId>com.fasterxml.jackson.dataformat</groupId>  
 <artifactId>jackson-dataformat-xml</artifactId>  
</dependency>

Result is a string in Json format

Map<String, Object> json = new HashMap<>();  
json.put("uploadPath", uploadPath);  
json.put("pdfPath", pdfPath);  
  
String data = "";  
try {  
 ObjectMapper mapper = new ObjectMapper();  
 data = mapper.writeValueAsString(json);  
} catch (Exception e) {  
 log.info("", e);  
}  
return ResponseEntity.*ok*().body(data);

ex

@GetMapping(value = "/demo", produces = MediaType.*APPLICATION\_JSON\_VALUE*)  
 public ResponseEntity<Object> demo() {  
 Map<String, Object> jsonMap = new HashMap<>(1);  
 jsonMap.put("leg", new User(1l, "Anynomous"));  
 jsonMap.put("leg1", 500);  
 ObjectMapper objectMapper = new ObjectMapper();  
 String result = "";  
 try {  
 result = objectMapper.writeValueAsString(jsonMap);  
// org.json.JSONArray jsonArray= new JSONArray(result);  
// System.out.println(jsonArray.get(1));  
 org.json.JSONObject jsonObject = new JSONObject(result);  
 System.*out*.println(jsonObject.accumulate("Name", "Quan"));  
 System.*out*.println(jsonObject.getJSONObject("leg"));  
 System.*out*.println(jsonObject.getInt("leg1"));  
 jsonObject.put("leg1", new User(1l, "quanzip"));  
 } catch (JsonProcessingException | JSONException e) {  
 e.printStackTrace();  
 }  
 return ResponseEntity.*ok*().body(result);  
 }

{"leg1":500,"leg":{"name":"Anynomous","id":1},"Name":"Quan"}

{"name":"Anynomous","id":1}

500