

Parsing JSON

CS 371M Mobile Computing

JSON (JavaScript Object Notation) Example

```
{  
  "Title": "The Cuckoo's Calling",  
  "Author": "Robert Galbraith",  
  "Detail": { "Pages": 494 },  
  "Price": [  
    { "type": "Hardcover",  
      "price": 16.65 },  
    { "type": "Kidle Edition",  
      "price": 7.03 }  
  ]  
}
```

<http://www.w3resource.com/JSON/introduction.php>

Elements of JSON

- Four basic and built-in data types in JSON.
 - Strings, numbers, Booleans (i.e true and false) and null.
 - These can be used as values
- Two structured data types
- Objects are wrapped within '{' and '}'.
 - Objects are list of label-value pairs.
- Arrays are enclosed by '[' and ']'.
 - Arrays are list of values.
- Both objects and arrays can be nested.

JSONObject from string

```
JSONObject = new JSONObject(String);
```

Where does my JSON come from?

- https://www.reddit.com/r/aww/search.json?raw_json=1&q=cat&sort=hot

```
{"kind": "Listing", "data": {"facets": {}, "modhash": "", "children": [{"kind": "t3", "data": {"domain": "imgur.com", "banned_by": null, "media_embed": {}, "subreddit": "AdviceAnimals", "selftext_html": null, "selftext": "", "likes": null, "suggested_sort": null, "user_reports": [], "secure_media": null, "link_flair_text": null, "id": "3hhn1g", "from_kind": null, "gilded": 1, "archived": false, "clicked": false, "report_reasons": null, "author": "the-cat-killa", "media": null, "score": 6443, "approved_by": null, "over_18": false, "hidden": false, "preview": {"images": [{"source": {"url": "https://i.redditmmedia.com/4DMMaAtt0YMHNTYHybK5AQyMo1I1TBhO3NJIqeDH3ss.jpg?s=ea8e48e0a0a3343a1fba2c720e74a885", "width": 1137, "height": 1137}, "resolutions": [{"url": "https://i.redditmmedia.com/4DMMaAtt0YMHNTYHybK5AQyMo1I1TBhO3NJIqeDH3ss.jpg?fit=crop\u0026crop=faces%2Centropy\u0026arh=2\u0026w=108\u0026s=2cc6e293cedab69234ae614a990bba37", "width": 108, "height": 108}, {"url": .....}]
```

Help from JSON Pretty Printer

- <https://jsonformatter.curiousconcept.com/>

Formatted JSON Data

```
{
  "kind": "Listing",
  "data": {
    "facets": {

    },
    "modhash": "",
    "children": [
      {
        "kind": "t3",
        "data": {
          "domain": "imgur.com",
          "banned_by": null,
          "media_embed": {

          },
          "subreddit": "AdviceAnimals",
          "selftext_html": null,
          "selftext": "",
          "likes": null,
          "suggested_sort": null,
          "user_reports": [

          ]
        }
      }
    ]
  }
}
```

Use Actual Documentation

- <https://github.com/reddit/reddit/wiki/API>
 - Make no more than thirty requests per minute.
 - NEVER lie about your user-agent.

https://www.reddit.com/dev/api#GET_search

search

GET `[/r/subreddit]/search`

read

rss support

[view code](#) <#>

Search links page.

This endpoint is [a listing](#).

<code>after</code>	<code>fullname</code> of a thing
<code>before</code>	<code>fullname</code> of a thing
<code>count</code>	a positive integer (default: 0)
<code>include_facets</code>	boolean value
<code>limit</code>	the maximum number of items desired (default: 25, maximum: 100)
<code>q</code>	a string no longer than 512 characters
<code>restrict_sr</code>	boolean value
<code>show</code>	(optional) the string <code>all</code>
<code>sort</code>	one of (relevance, hot, top, new, comments)
<code>sr_detail</code>	(optional) expand subreddits
<code>syntax</code>	one of (cloudsearch, lucene, plain)
<code>t</code>	one of (hour, day, week, month, year, all)
<code>type</code>	(optional) comma-delimited list of result types (sr, link)

```

JSONObject j0 = .....;
try {
    if( j0.isNull("data") ) {
        return;
    }
    j0 = j0.getJSONObject("data");
    JSONArray jA = j0.getJSONArray("children");
    while( jsonIndex < jA.length() ) {
        j0 = jA.getJSONObject(jsonIndex);
        if( j0.isNull("data") ){
            jsonIndex++;continue;
        }
        j0 = j0.getJSONObject("data");
        // Make sure there is a thumbnail and an image url
        // Most thumbnails I see are jpg or png, but some urls are to imagur, which
        // contain html
        if( j0.isNull("thumbnail")
            || (!j0.getString("thumbnail").endsWith("jpg")
                && !j0.getString("thumbnail").endsWith("png"))
            || j0.isNull("url")
            || (!j0.getString("url").endsWith("jpg") && !j0.getString("url").endsWith("png"))
            || j0.isNull("title")) {
            jsonIndex++;continue;
        }
    }
}

```

Formatted JSON Data

```

{
  "kind": "Listing",
  "data": {
  }
}

```

Formatted JSON Data

```

{
  "kind": "Listing",
  "data": {
    "facets": {
    },
    "modhash": "",
    "children": [
    ],
    "after": "t3_3g0ra4",
    "before": null
  }
}

```

Finish traversal

```
String title = j0.getString("title");
String thumbnailURL = j0.getString("thumbnail");
String imageURL = j0.setString("url");
} catch( Exception e ) {
    e.printStackTrace();
}
// Update loop index
jsonIndex++;
}
} catch( Exception e) {
    e.printStackTrace();
}
```

Formatted JSON Data

```
{
  "kind": "Listing",
  "data": {
    "facets": {

    },
    "modhash": "",
    "children": [
      {
        "kind": "t3",
        "data": {

        },
        { },
        { }
      ]
    }
  }
}
```