

# My Friend JSON.

Explained for a 6 year old.

A Kate Catlin Production.

# Game Plan

- We're going to talk about what JSON is.
- We're going to talk about how a JSON call is written.
- We're going to walk through an example (with puppets).

# Who is Kate Catlin?

- Android Developer at Detroit Labs
- Recent Apprentice Graduate
- Founder of [www.WomenRising.co](http://www.WomenRising.co)



A photograph of two women laughing together. One woman is in the foreground, smiling broadly, wearing a red cardigan over a white shirt. The other woman is partially visible behind her, also smiling. They are in a bright room with large windows in the background.

**Studies show that mentors and friends determine your career trajectory. Find that support group here.**

Email

Join Now

@Kate\_Catlin || @detroitlabs

# What's a JSON?

“JSON (JavaScript Object Notation) is a lightweight data-interchange format.”

“JSON is a way to store information in an organized, easy-to-access manner. It gives us a human-readable collection of data that we can access in a really logical manner.”

“JSON is a syntax for storing and exchanging data. It’s an easier to use alternative to XML. It defines objects and arrays of information about those objects.”

# What's a “Jay-son?” (TRANSLATION)

1. Makes a very specific request for info to a specific website (which is an API, not just any website) and gathers very specific info from that source.
2. Puts the info I asked for into one, loooooooooong “string” so dummy computers can read it, but you can still kind of understand what it says as a human.
3. Lets you read that big long string to your computer.
4. Later - You can sort through the sentence real quick like for the info you need.

# JSON String:

```
{"count":50100, "results":[{"listing_id":121804322,"state":"active","user_id":26346108,"category_id":68888142,"title":"Moss Agate Cabochon, Natural Stone, Rusty Yellow, Red, Brown, Translucent, Free Form (186)","description":"This is a free-form Moss Agate cabochons that is translucent with rusty yellow, red, and brown patterns that are suspended within the stone.. It is approximately 30mm x 25mm , and weighs 9g. \r\n\r\nEtsy186","creation_tsz":1414766342,"ending_tsz":1425137942,"original_creation_tsz":1359302332,"last_modified_tsz":1414766342,"price":"19.00","currency_code":"USD","quantity":"1","tags":["Agate","cabochon","supplies","wire wrapping","bezel setting","semiprecious stone","natural cabochon","stone cabochon","free form","jewelry supply","lapidary","loose stone","moss agate"],"category_path":["Supplies","Cabochon"],"category_path_ids":[69150433,68888142],"materials":["Stone","agate","moss agate"],"shop_section_id":12488034,"featured_rank":null,"state_tsz":1414586427,"url":"https://www.etsy.com/listing/121804322/moss-agate-cabochon-natural-stone-rusty?utm_source=jsonpractice&utm_medium=api&utm_campaign=api","views":15,"num_favorers":0,"shipping_template_id":null,"processing_min":1,"processing_max":2,"who_made":"i_did","is_supply":true,"when_made":"2010_2014","is_private":false,"recipient":null,"occasion":null,"style":null,"non_taxable":false,"is_customizable":false,"is_digital":false,"file_data":null,"language":"en-US","has_variations":false,"used_manufacturer":false},{"listing_id":121923378,"state":"active","user_id":26346108,"category_id":68888142,"title":"Picasso Stone Cabochon, Natural Stone, Black, Rust, Gray, Spiderweb, Free Form (192)","description":"A free-form Picasso Stone Cabochon with interesting spiderweb patterns of black, gray and rusty tan colors. It is approximately 30mm x 10mm and weighs 5g.\r\n\r\nEtsy192","creation_tsz":1414766341,"ending_tsz":1425137941,"original_creation_tsz":1359394411,"last_modified_tsz":1414766341,"price":"14.00","currency_code":"USD","quantity":"1","tags":["cabochon","supplies","wire wrapping","bezel setting","semiprecious stone","natural cabochon","stone cabochon","jewelry supply","lapidary","loose stone","Picasso Stone","free form","Marble"],"category_path":["Supplies","Cabochon"],"category_path_ids":[69150433,68888142],"materials":["Stone","Marble","Picasso Stone"]}]}
```

# What kind of gossip can we get from your friend JSON?

- A string of weather forecasts for the next week from a forecast API.
- The title, author and genre of the books on the NYT best sellers list.
- CIA info on each member of your family (this is a joke)
- ...all in a long “string” format!

# JSON String:

```
{"count":50100, "results":[{"listing_id":121804322,"state":"active","user_id":26346108,"category_id":68888142,"title":"Moss Agate Cabochon, Natural Stone, Rusty Yellow, Red, Brown, Translucent, Free Form (186)","description":"This is a free-form Moss Agate cabochons that is translucent with rusty yellow, red, and brown patterns that are suspended within the stone.. It is approximately 30mm x 25mm , and weighs 9g. \r\n\r\nEtsy186","creation_tsz":1414766342,"ending_tsz":1425137942,"original_creation_tsz":1359302332,"last_modified_tsz":1414766342,"price":"19.00","currency_code":"USD","quantity":"1","tags":["Agate","cabochon","supplies","wire wrapping","bezel setting","semiprecious stone","natural cabochon","stone cabochon","free form","jewelry supply","lapidary","loose stone","moss agate"],"category_path":["Supplies","Cabochon"],"category_path_ids":[69150433,68888142],"materials":["Stone","agate","moss agate"],"shop_section_id":12488034,"featured_rank":null,"state_tsz":1414586427,"url":"https://www.etsy.com/listing/121804322/moss-agate-cabochon-natural-stone-rusty?utm_source=jsonpractice&utm_medium=api&utm_campaign=api","views":15,"num_favorers":0,"shipping_template_id":null,"processing_min":1,"processing_max":2,"who_made":"i_did","is_supply":true,"when_made":"2010_2014","is_private":false,"recipient":null,"occasion":null,"style":null,"non_taxable":false,"is_customizable":false,"is_digital":false,"file_data":null,"language":"en-US","has_variations":false,"used_manufacturer":false},{"listing_id":121923378,"state":"active","user_id":26346108,"category_id":68888142,"title":"Picasso Stone Cabochon, Natural Stone, Black, Rust, Gray, Spiderweb, Free Form (192)","description":"A free-form Picasso Stone Cabochon with interesting spiderweb patterns of black, gray and rusty tan colors. It is approximately 30mm x 10mm and weighs 5g.\r\n\r\nEtsy192","creation_tsz":1414766341,"ending_tsz":1425137941,"original_creation_tsz":1359394411,"last_modified_tsz":1414766341,"price":"14.00","currency_code":"USD","quantity":"1","tags":["cabochon","supplies","wire wrapping","bezel setting","semiprecious stone","natural cabochon","stone cabochon","jewelry supply","lapidary","loose stone","Picasso Stone","free form","Marble"],"category_path":["Supplies","Cabochon"],"category_path_ids":[69150433,68888142],"materials":["Stone","Marble","Picasso Stone"]}]}
```

# What does JSON look like? (key value pairing)

Start of JSON object



The colon separates the “name” and “value” of a property of the object

```
{  
    "textProns": [],  
    "sourceDictionary": "ahd-legacy",  
    "exampleUses": [],  
    "relatedWords": [],  
    "labels": [],  
    "citations": [],  
    "word": "devotion",  
    "text": "Ardent, often selfless affection and dedication, as to a person or principle. See Synonyms at love.",  
    "sequence": "0",  
    "score": 10.132973,  
    "partOfSpeech": "noun",  
    "attributionText": "from The American Heritage® Dictionary of the English Language, 4th Edition"  
},
```

Here's another “key value” pairing.

Aaaaand here's another key value pairing.

Also at the end of a JSON object

# What does JSON look like? (multiple objects in one array)

Starts entire JSON

```
{  
    "totalResults": 1523,  
    "results": [  
        {  
            "textProns": [],  
            "sourceDictionary": "ahd-legacy", ← First object  
            "exampleUses": [],  
            "relatedWords": [],  
            "labels": [],  
            "citations": [],  
            "word": "devotion",  
            "text": "Ardent, often selfless affection and dedication, as to a person or principle. See Synonyms at love.",  
            "sequence": "0",  
            "score": 10.132973,  
            "partOfSpeech": "noun",  
            "attributionText": "from The American Heritage® Dictionary of the English Language, 4th Edition"  
        },  
        {  
            "textProns": [],  
            "sourceDictionary": "ahd-legacy", ← Second object  
            "exampleUses": [],  
            "relatedWords": []  
        }  
    ]  
}
```

# What does JSON look like? (nesting)

Here we are starting our entire "JSON object"...

```
{  
  "id": 517335,  
  "word": "festination",  
  "examples": [  
    {  
      "url": "http://api.wordnik.com/v4/mid/515a10e4ba035ad6230414483429",  
      "text": "There's also, in some of his works, a sense of festination  
      \"...my music is also seductive, even spiritual\"",  
      "id": 337411048  
    },  
    {  
      "url": "http://api.wordnik.com/v4/mid/d46150115f3752a3f476364a7dbc",  
      "text": "Strive not to run, like Hercules, a furlong in a breath:  
      \"Library of the World's Best Literature, Ancient and Modern\",  
      \"id\": 1174539620  
    }  
,  
  "definitions": [  
    {  
      "text": "Specifically in <em>medicine</em>, involuntary hurrying in  
      \"partOfSpeech\": \"noun\",  
      \"source\": \"century\"  
    },  
    {  
      "text": "A state of being hurriedly or compulsively active.  
      \"partOfSpeech\": \"noun\",  
      \"source\": \"century\"  
    }  
,  
  ]
```

But look! Now that "[" starts an array!

Oh and here's another array of info!

# Ok let's get our hands dirty!

Let us take a leap down the rabbit hole with a real example.

Follow the code along at:

[https://github.com/KateCatlin/  
JSONForDummies](https://github.com/KateCatlin/JSONForDummies)

# This is Etsy:

Etsy

Home Favorites You Cart

Your Feed   You have new activity! Click to refresh.

Trending Items

Handmade Wooden Chess Pieces \$16.44 | MadelInTheMancave  
One of our most favorite items

Femkit ★★★★★ Unique dresses and small series...  
One of our most popular shops

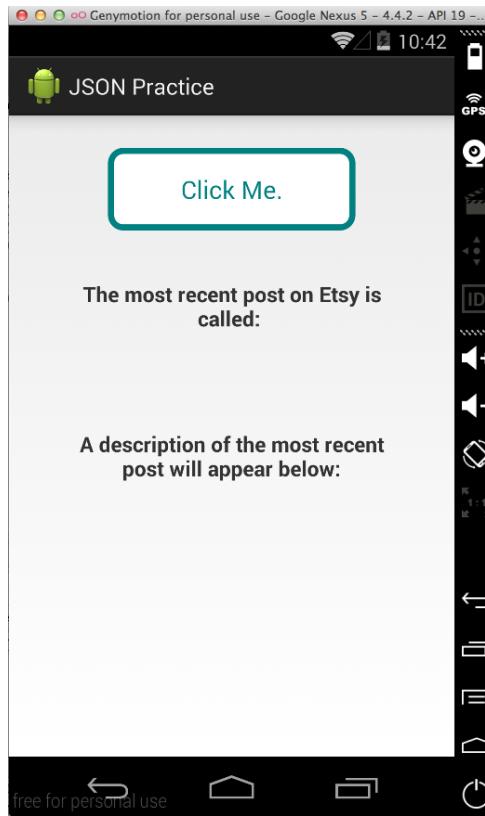
Browse

Featured Shop  CeeCeeandBee   
  
  
40 items >

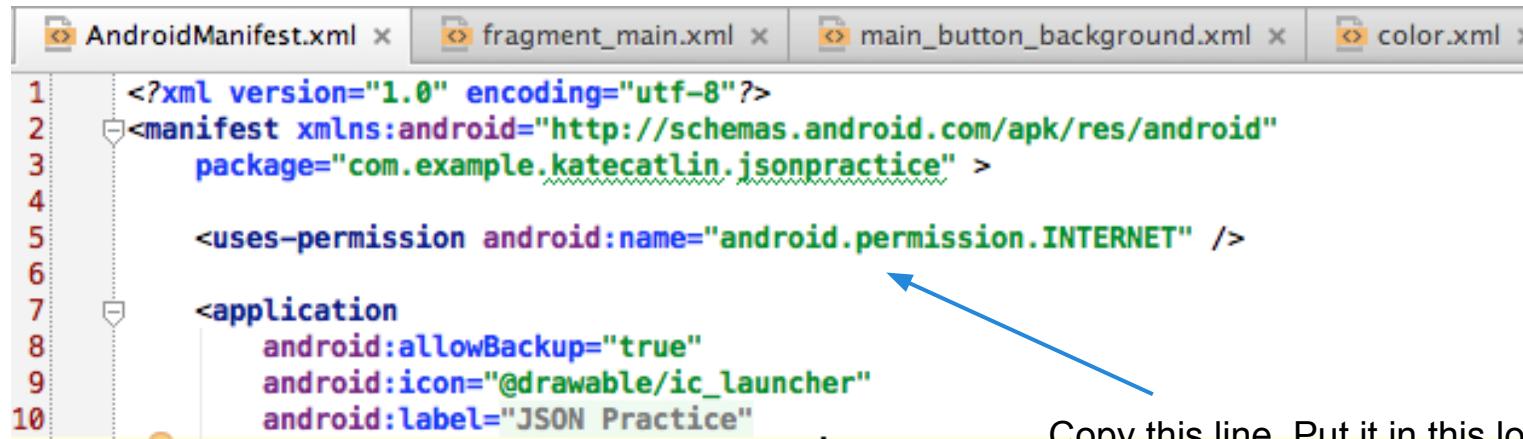
Recent Blog Posts  Inspiring Workspaces: Dellamorte & Co.  
  
Read the next

@Kate\_Catlin || @detroitlabs

# Today we'll make the Etsy App.



# 1.) Internet Permissions (duh):



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.katecatlin.jsonpractice" >
4
5      <uses-permission android:name="android.permission.INTERNET" />
6
7      <application
8          android:allowBackup="true"
9          android:icon="@drawable/ic_launcher"
10         android:label="JSON Practice" .
```

Copy this line. Put it in this location.

Follow along at: <https://github.com/KateCatlin/JSONForDummies>

## 2. Call the Async Task

This is you creating an AsyncTask :)

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        GetJSONInfo jsonCaller = new GetJSONInfo();
        jsonCaller.execute(); ←
    }
});
```

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_main, container, false);

    textView = (TextView) rootView.findViewById(R.id.text_title);
    Button button = (Button) rootView.findViewById(R.id.main_button);
    button.setOnClickListener((v) -> {
        GetJSONInfo jsonCaller = new GetJSONInfo();
        jsonCaller.execute();
        textView.setText("");
    });
    return rootView;
}
```

This is you creating an instance of an AsyncTask class and calling it somewhere else.

What goes in  
(nothing).

Progress  
Units

What comes out

Follow along at: <https://github.com/KateCatlin/JSONForDummies>

# Ok number two is--- WAIT? WHY DO WE NEED AN ASYNC???

Because your resources are constrained and you don't want the app to freeze up while you're calling the internet.

Follow along at: <https://github.com/KateCatlin/JSONForDummies>

# Async is like having a friend help you get ready for a cookie party.



# Async == two big methods

```
protected String[ ] doInBackground(Void... params) {  
}
```

Here's where we will get the JSON object (a big long computer-y string)

```
private String[ ] parsePostingFromJsonString(String etsyJsonString) throws  
JSONException {  
}
```

Here's where we parse out the info we want from the JSON object.

Follow along at: <https://github.com/KateCatlin/JSONForDummies>

# 3. Make Your URL

```
final String URL_BASE = "https://openapi.etsy.com/v2/listings/active?api_key=";
final String API_KEY = "8yuclbhlmoijgd6jld92l6t4";  
  
try {  
    String urlString = URL_BASE + API_KEY;  
    URL urlToUse = new URL(urlString);
```

Figure out your HTML call...

Put it together down here...

Follow along at: <https://github.com/KateCatlin/JSONForDummies>

# WAIT A MO'. How'd you get the URL?

<http://developer.wordnik.com/docs.html#!/words>

See Detroit?

See Noun?

Parameters		
Parameter	Value	Description
query	Detroit	Search term
findSenseForWord		Restricts words an closest sense
includeSourceDictionaries		Only include these delimited source d
excludeSourceDictionaries		Exclude these con delimited source d
includePartOfSpeech	Noun	Only include these delimited parts of s
excludePartOfSpeech		Exclude these con

# WAIT A MO'. How'd you get the URL?

<http://developer.wordnik.com/docs.html#!/words>

See Detroit!!!

The screenshot shows a developer documentation interface. At the top left, it says "400". Below that are "Try it out!" and "Hide Response" buttons. Under "Request URL", there is a code block containing the URL: "http://api.wordnik.com:80/v4/words.json/reverseDictionary?query=Detroit&includePartOfSpeech=Noun". Two blue arrows point from the text "See Detroit!!!" and "See Noun (mostly)!!!!" to this URL. Below the URL, under "Response Body", is a JSON snippet:

```
{  
  "totalResults": 52,  
  "results": [  
    {  
      "textProns": [],  
      "sourceDictionary": "ahd-legacy",  
      "exampleUses": []  
    }  
  ]  
}
```

See Noun  
(mostly)!!!!

# WAIT A MO'. How'd you get the URL?

[https://www.etsy.com/developers/documentation/getting\\_started/api\\_basics](https://www.etsy.com/developers/documentation/getting_started/api_basics)

## Accessing the API

The Etsy API accessed under the `openapi` subdomain. A sample unauthenticated request might look like:

```
GET https://openapi.etsy.com/v2/listings/active?api_key={YOUR_API_KEY}
```

To retrieve the most recent active listings on Etsy.

# WAIT A MO'. How'd you get the URL?

[https://www.etsy.com/developers/documentation/getting\\_started/api\\_basics](https://www.etsy.com/developers/documentation/getting_started/api_basics)

## Parameter Types [edit]

Many API methods take one or more parameters, either as query parameters of the URL itself, or as POST parameters. The documentation for each method references these standard types:

Param Type	Meaning
string	Any string (up to 255 characters).
int	A whole number value.
epoch	A whole number value representing <a href="#">UNIX epoch time</a> , or any string accepted by PHP's <code>strtotime()</code> function.
float	A number with or without a decimal point. Represented in output as a string, to avoid precision errors.
boolean	A logical true or false value. May be passed to API requests as the strings "true" or "false" or "1" and "0". In JSON output, symbolic constants are used.
user_id_or_name	Either a user's numeric ID or login name.
shop_id_or_name	Either a shop's numeric ID or login name. (Note: shop IDs are not interchangeable with user IDs.)

# (PAUSE FOR VOLUNTEERS)

Don't waste time! Introduce yourself to three people near you and out:

- Where they are from.
- If they had a rock star or DJ alter ego what their name would be (Ex: Hannah Montana).
- What their favorite pair of shoes is and what it says about them.

## 4. Translate website -> String

```
HttpURLConnection urlConnection = (HttpURLConnection)  
urlToUse.openConnection();
```

```
urlConnection.setRequestMethod("GET");  
urlConnection.connect();
```

*(Translation: Open the website and connect)*

# URLConnection is like turning on the oven:



# JSON “GET” request is like a restaurant server bringing cookies.



JSON “POST” request is like a boat man between 2 cookie colonies.



# What our JSON website looks like:

The screenshot shows a web browser window with the URL [https://openapi.etsy.com/v2/listings/active?api\\_key=8yuclbhlmoijgd6jld92l6t4](https://openapi.etsy.com/v2/listings/active?api_key=8yuclbhlmoijgd6jld92l6t4). The browser interface includes a toolbar with icons for back, forward, search, and refresh, and a tab bar showing other open tabs like 'TheBusyCode...gle Drive' and 'JSON Calls For Dummies - Google Slides'. The main content area displays a large JSON object representing a collection of items from the Etsy API. The JSON structure includes fields such as 'count', 'results', and various item details like 'title', 'description', 'category\_id', 'price', and 'tags'. The text is heavily redacted for privacy.

```
{"count":50100,"results":[{"listing_id":209150026,"state":"active","user_id":32630678,"category_id":69169413,"title":"Vintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt","description":"LABEL\r\nUnknown\r\n\r\nDESCRIPTION\r\nVintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt\r\n\r\nFLAT MEASUREMENTS\r\nLength: 28#39;#39;\r\nShoulder to Shoulder: 23quot;\r\nSleeve Length: 8quot;\r\nUnder Arm to Under Arm: 24quot;\r\nHem: 24quot;\r\n\r\nGARMENT SIZE\r\nLarge\r\n\r\nMODEL SIZE\r\nMedium\r\n\r\nFABRICATION\r\n50% Cotton / 50% Polyester\r\n\r\nCONDITION\r\nExcellent\r\n\r\nThis is a carefully selected vintage piece. Expect elements of wear in the item. All flaws are noted in the condition section. Vintage sizes are sometimes different from today's standard sizes so we have made the appropriate suggestion in the size section. We have also included the item's measurements for your convenience.\r\n\r\nIF YOU HAVE ANY QUESTIONS, PLEASE DO NOT HESITATE TO CONTACT US!","creation_tsz":1414705857,"ending_tsz":1425163857,"original_creation_tsz":1414705857,"last_modified_tsz":1414705857,"price":"21.00","currency_code":"USD","quantity":1,"tags":["Tee","90s Tshirt","Christmas Tshirt","Ugly Christmas","Summer Christmas","Seapunk Tshirt","Ho Ho Ho Tshirt","Xmas Stockings","1990s Tshirt","90s Christmas","Blue Tshirt","Christmas Tree"],"category_path":["Vintage","Clothing","Tshirt"],"category_path_ids":[69150437,69154945,69169413],"materials":["Cotton","Polyester"],"shop_section_id":13474636,"featured_rank":null,"state_tsz":1414705857,"url":"https://www.etsy.com/listing/209150026/vintage-90s-stuff-it-ho-ho-ho-ugly?utm_source=jsonpractice&utm_medium=api&utm_campaign=api","views":0,"num_favorers":0,"shipping_template_id":null,"processing_min":1,"processing_max":1,"who_made":"someone_else","is_supply":false,"when_made":"1990_1994","is_private":false,"recipient":"unisex_adults","occasion":"christmas","style":["Kitsch","Retro"],"non_taxable":false,"is_customizable":false,"is_digital":false,"file_data":"","language":"en-US","has_variations":false,"used_manufacturer":false},{"listing_id":194344338,"state":"active","user_id":14418755,"category_id":69189681,"title":"Just plain creepy","description":"The expression, the extra head, the red paint\u2026 it just all came together.", "creation_tsz":1414705856,"ending_tsz":1425163856,"original_creation_tsz":1403767303,"last_modified_tsz":1414705856,"price":"30.00","currency_code":"USD","quantity":1,"tags":["goth","weird","creepy","angel","black","punk","halloween","dead"],"category_path":["Art","Mixed Media","3d"],"category_path_ids":[68887312,68890894,69189681],"materials":["ceramic"],"shop_section_id":null,"featured_rank":null,"state_tsz":1414395021,"url":"https://www.etsy.com/listing/194344338/just-plain-creepy?utm_source=jsonpractice&utm_medium=api&utm_campaign=api","views":51,"num_favorers":3,"shipping_template_id":null,"processing_min":3,"processing_max":5,"who_made":"i_did","is_supply":false,"when_made":"2010_2014","is_private":false,"recipient":null,"occasion":null,"style":null,"non_taxable":false,"is_customizable":false,"is_digital":false,"file_data":"","language":"en-US","has_variations":false,"used_manufacturer":false},
```

# 4. Translate website -> String

```
inputStream = urlConnection.getInputStream();
```

*(Translation: Get the bytes from the internet)*

**InputStream is like the container bowl  
for the ingredients (takes in bytes):**



## 4. Translate website -> String

```
BufferedInputStream bufferedInputStream =  
    new BufferedInputStream(inputStream);
```

*(Translation: Gather up all the bytes and prepare to process  
them in a huge batch rather than one by one)*

# BufferedInputStream is like cooking in batches (lotsa bytes -> chars at once):



## 4. Translate website -> String

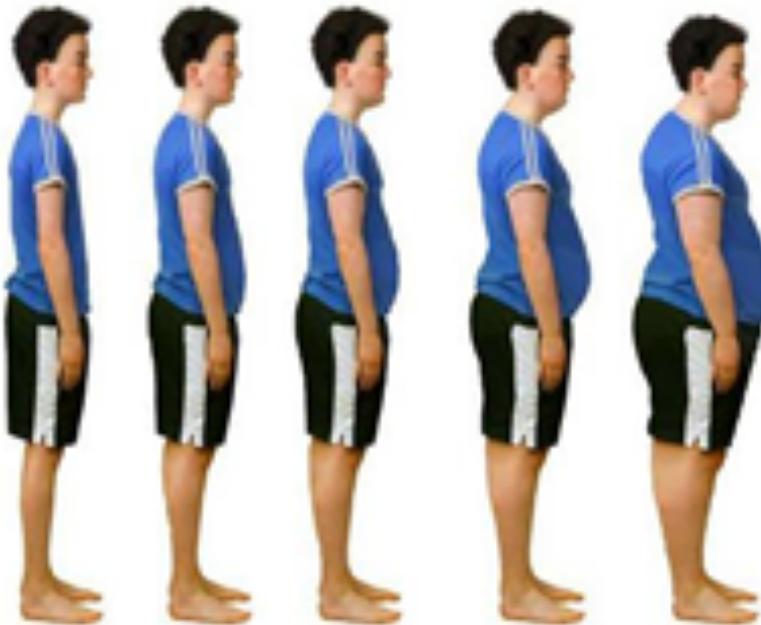
```
StringBuilder stringBuilder = new StringBuilder();  
  
while ((bytesRead = bufferedInputStream.read()) != -1) {  
    stringBuilder.append((char)bytesRead);  
}  
etsyJsonString = stringBuilder.toString();
```

*(Translation: Translate all the bytes into strings)*

# BufferedInputStream.read is like the oven (turns bytes -> chars):



**StringBuilder is like an expandable belly (holds more and more chars):**



## 4. Translate website -> String

Look, we're done with doInBackground! Now we use that JSON string we generated and make actual cool information out of it through "parsing" in another class!



```
try {  
    String[] theInfoYouWanted =  
    parsePostingFromJsonString(etsyJsonString);  
  
    return theInfoYouWanted;  
}
```

# What our JSON looks like:

```
{"count":50100,"results":[{"listing_id":209150026,"state":"active","user_id":32630678,"category_id":69169413,"title":"Vintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt", "description":"LABEL\r\nUnknown\r\n\r\nDESCRIPTION\r\n\r\nVintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt..."}
```

```
{"listing_id":194344338,"state":"active","user_id":14418755,"category_id":69189681,"title":"Just plain creepy","description":"The expression, the extra head, the red paint\u2026it just all came together."...}
```

# Where we at again?

```
protected String[ ] doInBackground(Void... params) {  
    ...String[ ] theInfoYouWanted =  
    parsePostingFromJsonString(etsyJsonString);  
    return theInfoYouWanted;}  
  
private String[ ] parsePostingFromJsonString(String  
    etsyJsonString) throws JSONException {  
    ...  
}
```

The diagram illustrates the state of the code being developed. It shows two main sections of Java code. The first section is a protected method `doInBackground` that includes code to parse a JSON string into an array of strings. An annotation with an arrow points to the line `parsePostingFromJsonString`, stating "We WERE working on this part." The second section is a private method `parsePostingFromJsonString` that takes a JSON string and throws a `JSONException`. An annotation with an arrow points to the opening brace of this method, stating "Now we're working on this part!" This visual cue helps track the progress and current focus of the developer.

# 5. Parse the JSON string.

```
JSONObject postingJSONObject = new JSONObject  
(etsyJsonString);
```

*(Translation: We gotta make a `JSONObject` (an imported class) to understand all the rules and syntax and stuff that comes with the big long JSON string. )*

# What our JSON looks like:

```
{"count":50100,"results":[{"listing_id":209150026,"state":"active","user_id":32630678,"category_id":69169413,"title":"Vintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt", "description":"LABEL\r\nUnknown\r\n\r\nDESCRIPTION\r\n\r\nVintage 90s Stuff It Ho Ho Ho Ugly Christmas Stockings Oversized Large Blue Tshirt..."}
```

```
{"listing_id":194344338,"state":"active","user_id":14418755,"category_id":69189681,"title":"Just plain creepy","description":"The expression, the extra head, the red paint\u2026it just all came together."...}
```

## 5. Parse the JSON string.

```
private final String RESULTS = "results";
```

```
JSONArray arrayOfPostings = postingJSONObject.  
getJSONArray(RESULTS);
```

*(Translation: Get that big long array called “results” within our JSON object. )*

# 5. Parse the JSON string.

```
JSONObject mostRecentPosting = arrayOfPostings.  
getJSONObject(0);
```

*(Translation: Make a new JSONObject JUST for the MOST recent of the most recent postings, which would be at array position zero. )*

## 5. Parse the JSON string.

```
private final String NAME_OF_GOOD = "title";
```

```
String title = mostRecentPosting.getString  
(NAME_OF_GOOD);
```

*(Translation: Search within that JSON object to find a string named “title”. Then copy them to our newly created “title” string. )*

# 5. Parse the JSON string.

```
private final String DESCRIPTION = "description";
```

```
String description = mostRecentPosting.getString  
(DESCRIPTION);
```

*(Translation: Search within that JSON object to find a string named “description”. Then copy them to our newly created “description” string. )*

## 5. Parse the JSON string.

```
String[ ] arrayOfTitleAndDescription = new String[2];  
arrayOfTitleAndDescription[0] = title;  
arrayOfTitleAndDescription[1] = description;  
  
return arrayOfTitleAndDescription;
```

*(Translation: Save our newly created “title” and “description” string in a new array of strings, then return it to the doInBackground class.*

# Where we at now?

```
protected String[ ] doInBackground(Void... params) {  
    ...//insert lots of stuff  
    String[ ] theInfoYouWanted = parsePostingFromJsonString(etsyJsonString);  
    return theInfoYouWanted;  
}
```

to here.

Now we're returning the whole monty! But where does it go?

```
private String[ ] parsePostingFromJsonString(String etsyJsonString) throws  
JSONException {  
    ...//insert lots of stuff  
    return arrayOfTitleAndDescription;  
}
```

We just returned this string array...

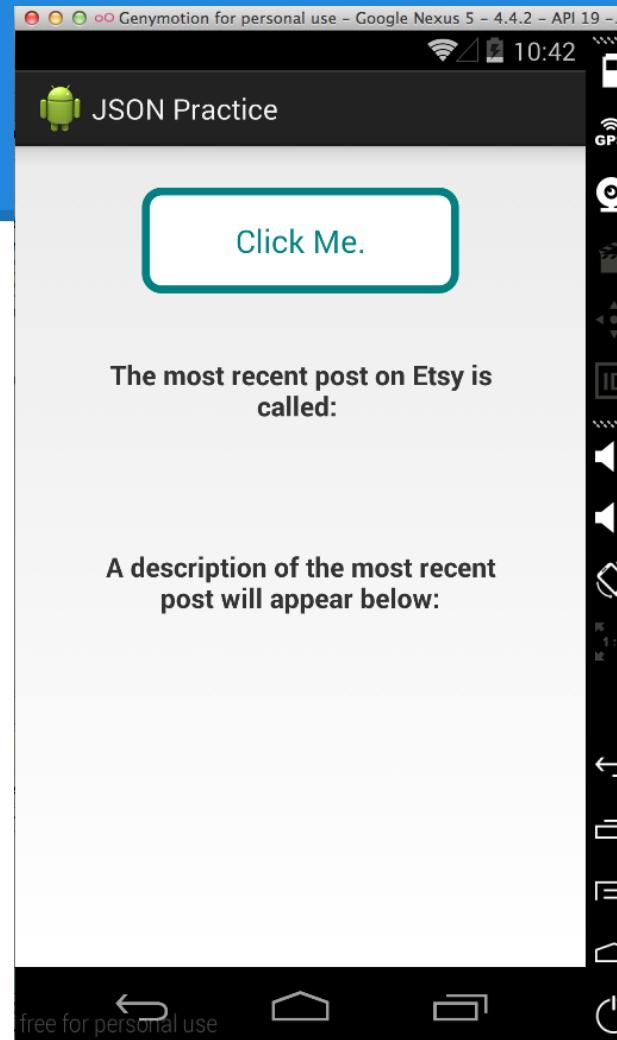
# 6. Make that info do watcha want!

```
protected void onPostExecute(String[] result) {  
    if (result != null) {  
        textView.setText(result[0]);  
        descriptionView.setText(result[1]);  
    }  
}
```

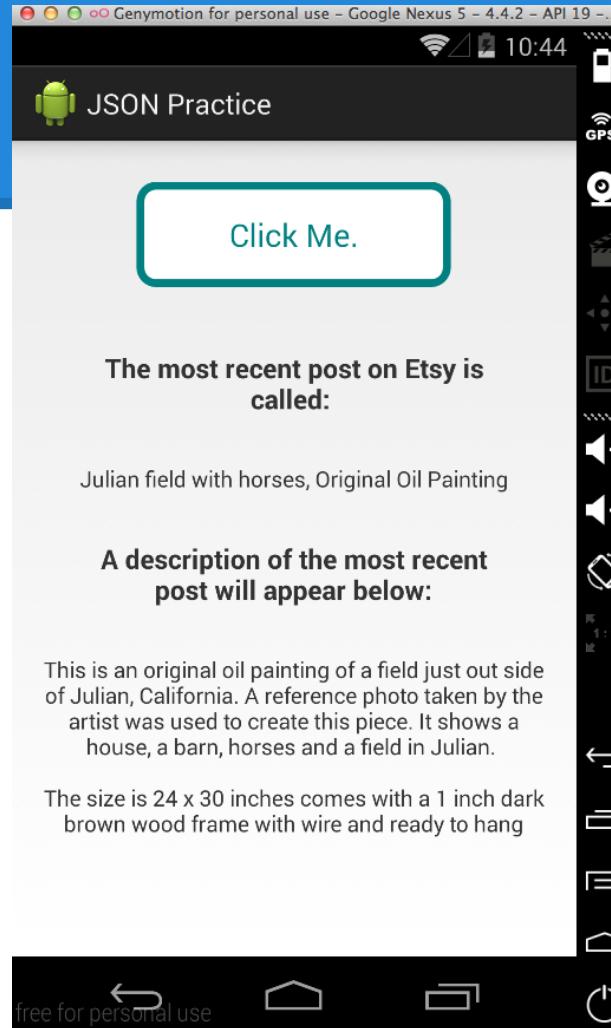
Change the text in our app to the title of the post.

Change the other text in our app to the description from our post.

# Before...



# AFTER!!!



# The End.

Have fun? Tweet at @Kate\_Catlin or  
@detroitlabs!