

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1\ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform news, updates & events. Need technical help? @TwitterDev",
    "place": { },
    "entities": {
      "urls": [
        {
          "url": "https://cards.twitter.com/cards/18c5wgo/3xolc",
          "expanded_url": "https://twitter.com/TwitterDev/status/850006245121695744",
          "display_url": "Building the Future of the Twitter API Platform"
        }
      ],
      "user_mentions": [ ]
    }
  },
  "retweet_count": 0,
  "retweeted_by": [ ],
  "retweeted_with_retweet": false,
  "retweeted_status": {
    "created_at": "Thu Apr 06 15:24:15 +0000 2017",
    "id_str": "850006245121695744",
    "text": "1\ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
    "user": {
      "id": 2244994945,
      "name": "Twitter Dev",
      "screen_name": "TwitterDev",
      "location": "Internet",
      "url": "https://dev.twitter.com/",
      "description": "Your official source for Twitter Platform news, updates & events. Need technical help? @TwitterDev",
      "place": { },
      "entities": {
        "urls": [
          {
            "url": "https://t.co/XweGngmx1P",
            "expanded_url": "https://twitter.com/TwitterDev/status/850006245121695744",
            "display_url": "Building the Future of the Twitter API Platform"
          }
        ],
        "user_mentions": [ ]
      }
    },
    "retweet_count": 0,
    "retweeted_by": [ ],
    "retweeted_with_retweet": false,
    "retweeted_status": {
      "created_at": "Thu Apr 06 15:24:15 +0000 2017",
      "id_str": "850006245121695744",
      "text": "1\ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nhttps://t.co/XweGngmx1P",
      "user": {
        "id": 2244994945,
        "name": "Twitter Dev",
        "screen_name": "TwitterDev",
        "location": "Internet",
        "url": "https://dev.twitter.com/",

```

{Introduction to Tweet JSON}

Introduction to Tweet JSON

Breaking down Tweet JSON

The 1 October Twitter Data Tutorial Series will ask you to download and later analyze a collection of Twitter data. Before you interact with the dataset, it is important to understand the structure of the data format you will be working with. This tutorial will walk you through the basics of JavaScript Object Notation, or JSON, a human-readable data format. It will then highlight some of the name-value pairs found in Tweet JSON.

Difficulty level: Beginner

Tutorial Key

- JSON and Tweet JSON will be displayed in this format
- 🎉 means you have come to the end of the tutorial

Lesson objectives

- Understand the basic structure of JSON and Tweet JSON
- Get familiar with some of the fundamental attributes of the Tweet Object

Key Terms

- JSON - JavaScript Object Notation
 - A minimal, human-readable format for structuring data. Twitter data is in JSON format.

Table of Contents

Lesson objectives	1
Key Terms	1
Table of Contents	1
Introduction to JSON	2

Tweets	3
Tweet JSON	5

Introduction to JSON

JSON, or JavaScript Object Notation, is a syntax for storing and exchanging data that is easy for humans to read and write. In the examples below, we will explore the two structures JSON is built on, objects and arrays.

Example of an object

An object is an unordered set of name-value pairs. Here's how you can spot them:

1. Objects begin with a left curly brace and end with a right curly brace.
2. Each name is followed by a colon
3. Name-value pairs are separated by a comma

```
{  
  "gender": "female",  
  "age": 17  
  "grade": 12  
}
```

In the example above, the **object** starts with a left curly brace and ends with a right curly brace. The first name-value pair `"gender": "female"`, begins with the name, `"gender"` is separated by a colon, has a value of `"female"`, and is separated from the following name-value pair by a comma.

Example of an array

An array is an ordered collection of values. Here's how you can spot them:

1. Arrays begin with a left bracket and end with a right bracket.
2. Each value is separated by a comma

```
{  
  "gender": "female",  
  "age": 17  
  "grade": 12  
  "classes": ["science", "technology", "engineering", "art", "math"]  
}
```

```
}
```

In the example above, the **array** starts with a left bracket and ends with a right bracket. Each value is separated by a comma.

Example of a Name

In the name-value pairs highlighted in the examples above, names are simply the attributes for the values they are paired with. Names must be text wrapped in double quotation marks.

```
{  
  "gender": "female",  
  "age": 17  
  "grade": 12  
  "classes": ["science", "technology", "engineering", "art", "math"]  
}
```

In this example "name", "age", "grade", and "classes" are all names.

Example of a value

In the name-value pairs highlighted in the examples above, values are paired with names. Values can come in the following formats.

1. Quoted text: "female"
2. The following unquoted words: true, false, or null
3. An unquoted number: 17
4. An array wrapped in brackets: ["science", "technology", "engineering", "art", "math"]
5. Another JSON object wrapped in curly braces

```
{  
  "gender": "female",  
  "age": 17  
  "grade": 12  
  "classes": ["science", "technology", "engineering", "art", "math"]  
}
```

Tweets

Before we break down tweet JSON, let's identify information a Twitter user can easily obtain when interacting with a tweet. Consider the [example below](#) from the Twitter Developer account.



Without looking at tweet JSON, we can still identify basic elements of [this tweet](#)

User handle: *Unique handle of the user who sent the tweet*

User name: *User's Twitter name*

Verified status: *A blue check mark indicates this account has been verified*

Message: *Message of the tweet*

Link: *Link to a source*

Link description: *Description of linked source*

Time: *Time published*

Date: *Date published*

Retweets: *Number of times the tweet has been retweeted*

Likes: *Number of times the tweet has been liked*

Even without seeing the JSON objects that make up this tweet, we can get an idea of what kind of name-value pairs we will see by listing out the basic elements of a tweet visible to all users.

Tweet JSON

We learned that JSON is based on name-value pairs. Each name is an attribute that is paired with an associated value. Attributes are used to describe objects, which, in JSON, are wrapped in curly braces. We're going to break down some of the most common objects seen in Tweet JSON. A Tweet can have over 150 attributes associated with it. We will be using [a sample](#) of Tweet JSON published by Twitter. If you're interested in learning more you can visit Twitter's [developer site](#).

Tweet Object

It comes as no surprise that Tweets are the main building block of the Twitter platform. There are a number of fundamental attributes like the time the tweet was created at ("**created_at**"), the unique identifier of the tweet ("**id_str**"), and the actual message ("**text**") of the tweet.

The Tweet object is considered the 'parent object' to several child objects like the sender of the tweet ("**user**"), and the additional metadata that comes with the contextual content in tweets like links, hashtags, and attached media ("**entities**" and "**extended_entities**").

The example below shows some of the name-value pairs you will see in the Tweet object.

```
{
  "created_at": "Thu May 10 15:24:15 +0000 2018",
  "id_str": "850006245121695744",
  "text": "Here is the Tweet message.",
  "user": {
  },
  "place": {
  },
  "entities": {
  },
  "extended_entities": {
  }
}
```

Let's break down each of the above names and their corresponding values:

"created_at": Time the tweet was created. All tweets are recorded in UTC.

"id_str": Every tweet has a unique identifier. **"id_str"** is the string representation of the unique identifier.

"text": The actual text of the tweet.

"**user**": The user who posted the tweet.

"**place**": Place is not always present. It is important to note the distinction between "**place**" and "**coordinates**". Place indicates that a tweet is associated with a place, but does not necessarily originate from a place. Coordinates represent the geographic location reported by the user or application. To learn more, check out the [ninth tutorial](#) in this series.

"**entities**": Additional metadata that provides contextual information about a Tweet like links, hashtags, and attached media

"**extended_entities**": Tweets with photos, videos, and animated GIFs will include an "**extended_entities**" JSON object

As you begin to incorporate Twitter data into your research, it is important to consider the full list of Tweet JSON attributes and the significance of their values. A [Tweet Data Dictionary](#) provided by Twitter is available and is updated as the platform changes.

🎉 Nice work! You've reached the end of the first tutorial. 🎉