

Homework #1: Anatomy of an Ad

HIST 318, Dr. McDaniel
Due Wednesday, January 22, 3 p.m.

Introduction

What information from a runaway slave ad would need to be captured in digital form in order for the information to be useful to a historian? How can that information be stored in a way that makes it easier for a computer to understand it?

In this assignment you will use three runaway slave advertisements from a nineteenth-century Texas newspaper to come up with a simple *schema* for digitizing the data in the ads. After deciding on the elements in your schema and their data types, you will put the information into valid **JSON**, a simple data formatting language that is easy for both humans and machines to read.

Note: This instruction sheet is long and detailed in order to make the assignment as clear as possible, but don't be daunted by the length! If you work through this step-by-step, you will be able to complete the assignment for full credit, and help will be available!

Objectives

1. To reflect on the decisions a digital historian must make about which information from a primary source to capture in a digital representation of the source.
2. To understand the care and labor required to represent data in a form that makes it easily digestible by a computer program.
3. To gain a basic introduction to one of the standard data formatting languages on the open web, JavaScript Object Notation (JSON).

Before You Begin

For this assignment, you will need to create a [free, personal GitHub account](#), which you will be using primarily for [creating Gists](#), and a [free Twitter account](#).

If you already have an account on either service that you want to use for this class, that's fine (though be aware that Dr. McDaniel and other classmates would then be able to see the other things you post). You may also choose to create "disposable" accounts dedicated to this class that can then be taken down after the semester.

Try to choose the same, short username for both services; shorter is better. You may choose to use a pseudonymous alias to protect your privacy rights (as described in the online participation consent form distributed in class). Spend some time looking at the account settings options [for GitHub](#) and [for Twitter](#) to make sure that the security and privacy settings match your preferences.

If you wish, you may choose to protect your Tweets so that they are only visible to those you approve, so long as you agree to allow Dr. McDaniel and other members of the class see your tweets. Be aware that in this case, tweets asking for help will only be seen by members of the class and you won't be able to solicit technical help on Twitter from experts outside of our class. Conversely, if you choose to make your accounts public, be sure that your username and the things that you publish are as professional and polite as you would be in person.

After creating your accounts, send your usernames for both to Dr. McDaniel. You should also receive from Dr. McDaniel an email inviting you to join a Google Drive spreadsheet containing runaway slave advertisements from a Texas newspaper, together with links to the full-text images of the ads. Before proceeding, make sure that you can get into this spreadsheet, and email Dr. McDaniel if you have any problems.

Requirements

Non-Technical

A runaway slave advertisement contains a great deal of information that can be useful to an historian, such as the name of the runaway or the date of the advertisement's publication. You should identify the pieces of information in your three runaway slave ads that are most interesting to you and make a list of these elements. Think of these as database "fields," or as the labels that you would put in a header row if you were arranging information from these ads in a spreadsheet like the one on Google Docs.

Your schema will probably not completely represent all of the information in these ads; it's up to you to decide which pieces of information are most important for now. But your schema should contain *at least 12 name/value pairs* that can be collected from ads like these. If you wish, you may reuse the labels in the header row in the Google Docs spreadsheet to help get you started, or you can create totally new labels or "fields."

Technical

There are multiple ways that you could represent individual ads using your data schema; a spreadsheet like the one on Google Docs is only one. For this assignment, you will format your three ads in a single, valid JSON document.

Here's what such a document might look like if you were working with two tweets instead of three runaway slave ads:

```
{"tweet1":
{
  "user": {"username": "wcaleb", "followers": 205},
  "date_sent": "January 14, 2014",
  "text": "I heart Cheerios.",
  "hashtags": [],
```

```

        "coordinates": null,
        "has_image": false
    },
    "tweet2":
    {
        "user": {"username": "wcaleb", "followers": 205},
        "date_sent": "September 1, 2013",
        "text": "Does anyone have any good #json links?",
        "hashtags": ["#json"],
        "coordinates": [-75.14310264, 40.05701649]
    }
}

```

This is only one example of many potential, technically valid ways of representing the same information. The technical requirements of this assignment are as follows:

- Data types should remain consistent. For example, if a string called “hash-tags” is paired with a Boolean value at some point in your object, it should not be paired with a string in the others.
- Your JSON should contain *at least one example* of each of the following types of values: `null`, Boolean (`true` or `false`), and array.
- Each ad should be its own object in your JSON document, and each object corresponding to an ad should have the same set of name/value pairs (even though the values will be different for each ad). The example above violates this requirement because “tweet1” includes the “has_image” pair and “tweet2” does not.
- Your document should clearly represent when one ad ends and another begins. I recommend that you choose one of the following two approaches to this: you could make each member of the main object a pair whose value is another object corresponding to a single ad (that’s the approach in the sample above), or you could make the element of your main object a single pair whose value is an array of objects, each of which corresponds to an ad (the approach taken in [the first example here](#)).
- The object for each ad should contain a pair whose value is the permalink URL to the runaway slave ad, which you will have found in the Google spreadsheet.
- When you copy and paste your JSON into the [JSONLint](#) text box and click the “Validate” button, you should see a green message that reads “Valid JSON” under the results.
- You should draft your JSON document using [a public or secret Gist](#), created with your Github account. (Note that [a secret gist is still public](#) to anyone to whom you send the URL, but it is not searchable or discoverable from the Gist directory.) Once you have a valid JSON document, tweet the URL of the finished Gist using your Twitter account and include the course hashtag `#ricedh` in the tweet.

Before turning in your assignment, be sure that it meets each of the above requirements.

Step-By-Step Instructions

- Start by selecting three ads from the Google spreadsheet that interest you. They can be any three ads from any year. They can even be reprints of the same ad!
- Look closely at your three ads and brainstorm a list of types of information contained in them, much like we did in class on Friday.
- Without paying much attention to formatting yet, sketch out a simple list on paper of what you hope to capture from each ad. Next to your list of types of information, jot down whether you think data of this type would best be represented with a number, a Boolean value (**true** or **false**), a string (words in quotation marks), or a collection of several of these things (for example, a list of strings or numbers; or an object containing several more name/value pairs).
- Go to <http://gist.github.com> and log in with your Github account name and password if you haven't already.
- Type in a brief Gist description such as "Homework #1 for HIST 318." (See Figure 1.)
- In the "name this file..." box, give your gist a short name with no spaces, followed by the filename extension ".json", which should cause Gist to recognize the language as JSON. (See Figure 2.)
- In the text editing box, begin entering a draft for your JSON document. Once you have a few lines written, click on "Create a Public Gist" or "Create a Secret Gist," which will cause your draft to be saved. You can click on "Edit" to go back to editing your draft, and click on "Update Gist" to save your changes. All changes will also be visible if you click on "Revisions" after updating your Gist. (See Figure 3.)
- Use the resources at <http://www.json.org> and <http://en.wikipedia.org/wiki/Json> to conform your Gist to JSON's formatting rules. Once you think you have a valid JSON document, save your changes (i.e., update the Gist) and then copy the text to your clipboard.
- Paste your JSON text into the editing box at <http://jsonlint.com> and try to "Validate" it.
- If you get a red error message, return to your Gist and edit it to see if you can correct the problem that was making your JSON invalid. Update your Gist and try validating again until you get a green success message. (See Figure 4.)

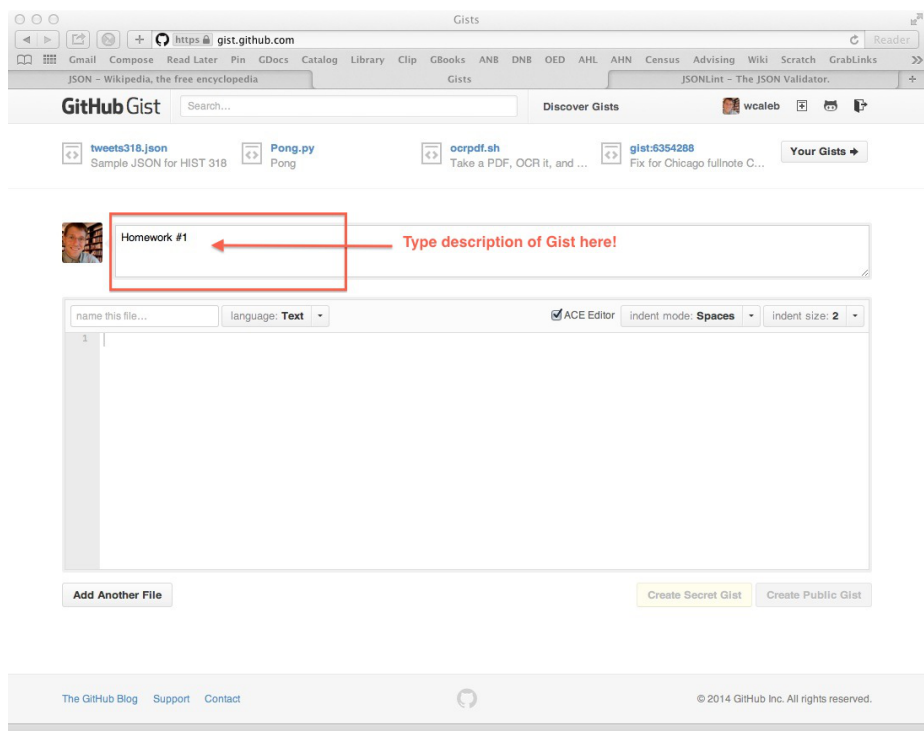


Figure 1: Describe your Gist

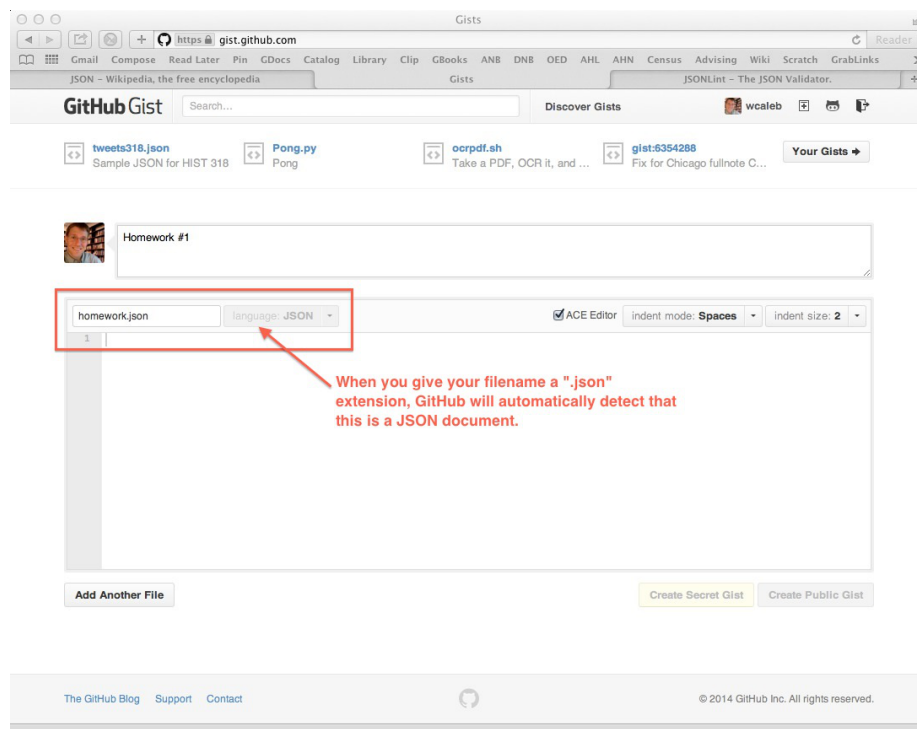


Figure 2: Name your Gist

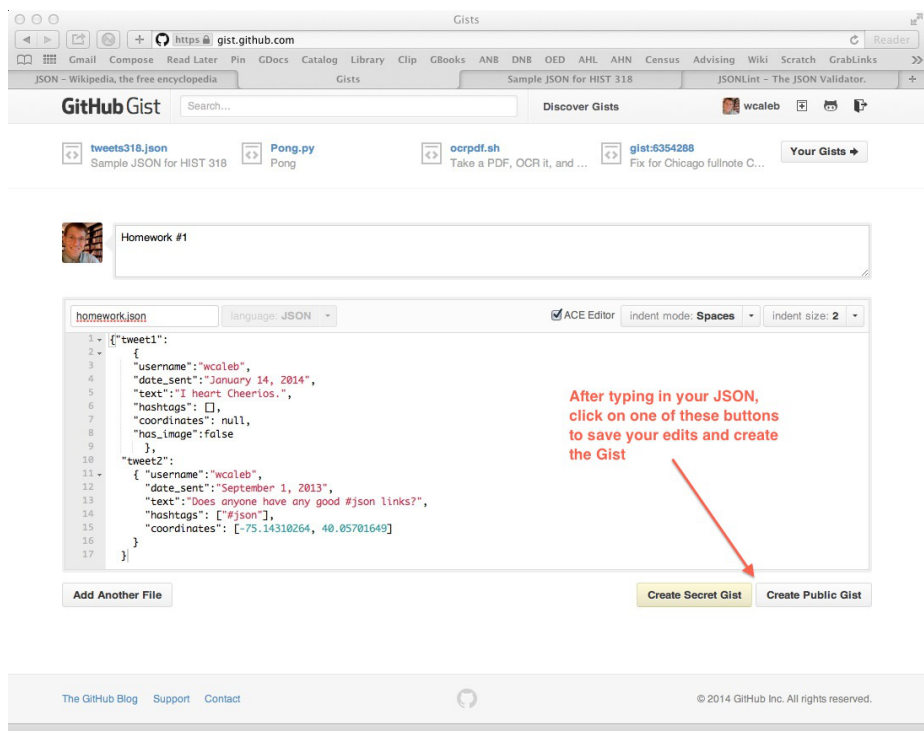


Figure 3: Type your JSON

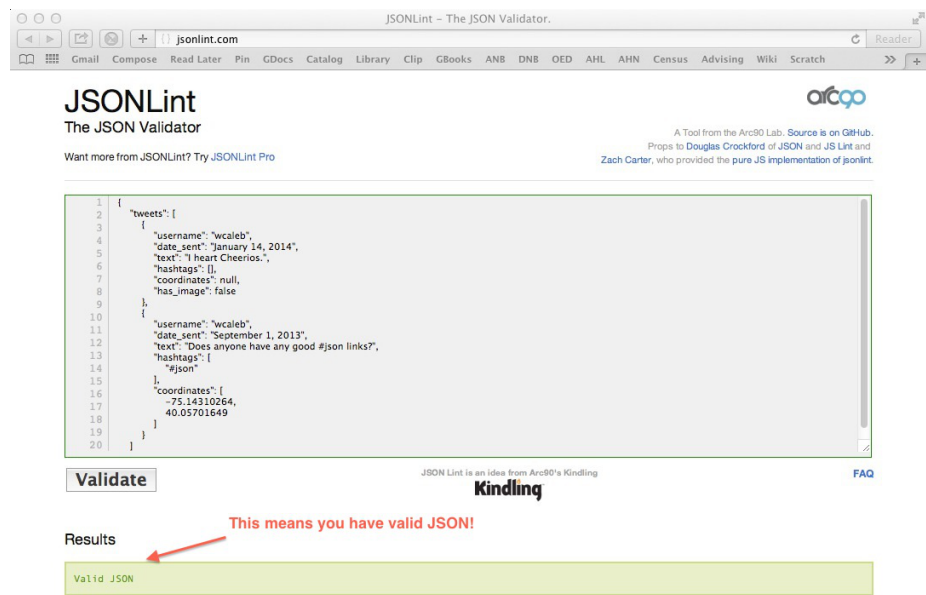


Figure 4: Success!

- Once you have a green success message from JSONLint, return to the technical and non-technical requirements above and be sure that your document satisfies all the other conditions specified. If you change anything about your JSON, be sure to make sure that it is still valid with JSONLint.
- When you're finished, go back to your Gist and copy the URL from your browser bar to your clipboard. Then paste it into a tweet with the class hashtag and announce your success! (See Figure 5.)



Figure 5: A sample tweet

Help! I'm Stuck!

Since this is the first homework of the semester, there are lots of moving parts here, and you most likely will have trouble along the way. That's normal, so don't despair!

There are multiple ways you can get help, especially if you are having trouble getting JSONLint to validate your JSON:

- Click on the "FAQ" link on JSONLint to see if the error you are getting is addressed there.
- If you have general questions about what the requirements above mean, or why something isn't working, tweet your question with the #ricedh hashtag so that another classmate who may have figured out your problem can answer. You can do this from the very beginning of the assignment—just be sure that your question is as clear and specific as possible.
- If you keep getting a weird error message, try Googling some of the error message or some words about what you are trying to do and see if someone else online has solved the basic problem.

- If you're really stuck, you can tweet the link to your Gist using the course hashtag, so that I or other students can look at your text and offer tips. However, you should not send out a URL to your Gist until you have attempted at least five separate "revisions" on your own (we will be able to see how many revisions you have made by looking at the Gist).
- If all else fails, we will be spending time troubleshooting any problems in class on January 22; the assignment requirements must be satisfied by the *end* of that class, but there will be time in class to smooth out any wrinkles.

Helping other students out is a form of [Team Participation](#) and is encouraged in this course, as is seeking out help on Twitter if you get stuck. However, be sure to abide by the course guidelines for academic integrity published on the course blog.