

Using Web APIs for Digital Humanities Research

Sabbatical Leave Final Report

Nate Garrelts

Professor of English | Ferris State University

15 January 2018

Dear President Eisler and Members of the Sabbatical Committee,

Thank you for granting me a one semester sabbatical during the Fall 2017 term. Your generous support provided me with the uninterrupted time I needed to enhance my skills and pursue interdisciplinary research.

During my sabbatical, I researched Web APIs and learned new coding skills so that I could better work with these APIs. A significant amount of time was also spent developing my talents as I produced various sample web pages. I have included a detailed summary of my activities with this letter and attached other supporting evidence. Since I met all the project objectives as outlined in my sabbatical proposal, this project could be considered highly successful.

I am grateful for this sabbatical experience and will draw on my new knowledge as I produce interesting scholarship, interact with students, and serve the university community.

Sincerely,

A handwritten signature in black ink that reads "Nate Garrelts". The signature is written in a cursive style with a large initial "N" and "G".

Nate Garrelts
Professor of English

Using Web APIs for Digital Humanities Research

I. Description of Sabbatical Activities

Application Program Interfaces (APIs) allow programmers to easily access complex services and data from companies like Google and Facebook. Humanities projects also develop APIs to share research with the world. During my sabbatical semester, I familiarized myself with APIs that are useful for humanities research, enhanced my skills in several programming languages and web development frameworks, and practiced making API enhanced webpages. This sabbatical helped me gain the skills I need to work on other more ambitious digital humanities research projects.

When the sabbatical began, I had the following objectives:

1. Familiarize myself with APIs that may be useful for humanities research.
2. Become knowledgeable of XML and JSON formats used by REST based APIs.
3. Enhance my programming skills in languages useful for working with REST APIs (HTML, JavaScript, PHP, and Python).
4. Familiarize myself with popular web development frameworks for these languages (HTML/Bootstrap, JavaScript/AngularJS, and Python/Django).
5. Produce proof-of-concept API enhanced webpages for scholars working in the humanities.

During the sabbatical, I followed the work as it was initially proposed. Screenshots of my work have been attached in the appendix. This includes course completion certificates, sample code, and screenshots of webpages. All sabbatical activities took place at my home using my university issued computer and/or personal computer(s). I paid for the Code School registration fees with my own funds. IBM graciously allows faculty a free year of Bluemix, which I am currently using to host one of my project websites.

Week 1-2. At the start of the sabbatical, I collected information about APIs that may be useful for humanities research and used them in my web browser to give me a sense of different interesting mashups and the format of the data. I kept notes in a simple .TXT file. The most promising APIs included the New York Times, Marvel Comics, IBM Watson, Wikipedia, and NASA.

Week 3. During this time period, I researched and prepared my development environment, which changed as my needs changed. Initially, I considered both text editors and IDEs. Because it is cross platform compatible and light on resources, I chose Visual Studio Code from Microsoft as my primary code editor. At times, I also used a Cloud9 server that I hosted from my home along with an ElephantSQL database. For production, I later used IBM Bluemix with Compose for PostgreSQL.

Week 4-6. I began my studies by completing the HTML/CSS path on Code School, which included Bootstrap and other topics. This was informative, as I learned about changes in HTML5, mobile development, and other issues that impact web design for different browsers. While much of this training focused on CSS and the appearance of webpages at a depth that was not practical for my needs or interests, the Bootstrap modules were really useful. Using this framework is

much more practical than writing HTML and CSS from scratch.

I found throughout my early training on Code School that the basic modules often gave an overview of the language, the middle courses delved into the nuances, and the later modules gave practical tips that developers really use.

Week 7-9. I completed the JavaScript path on Code School, which covered JavaScript, jQuery, Angular, and other topics. Of the modules in Code School, the first JavaScript Road Trip module, JavaScript Best Practices, Try jQuery, and Node.js modules were the most enjoyable for me, which is something that I would not have predicted when I started this project. I was not as impressed with client side frameworks like Angular and AngularJS, which seemed overly complex to me (at least at the time). Upon revisiting Angular near the end of the project, I could understand the appeal but still preferred other methods for front end development.

Week 10-12. I completed the PHP/Laravel and Python/Django paths on Code School. Basic PHP was easy and useful, however Laravel was not as easy for me as I had hoped. And while the first two modules in Python were fun, I was not as impressed with using Django for web development—it just did not seem as graceful as other solutions. In fact, Python seemed to lose its appeal to me the more I worked with it. For example, I often struggled to format my code exactly as required by not indenting properly. This is not surprising in hindsight considering how much I enjoyed the forgiving and asynchronous nature of ECMAScript/JavaScript during my previous sprint.

Week 13-15. During this time, I practiced making API calls and webpages outside of CodeAcademy and focused my skillset. There was a lot of trial and error. In general modern web application coding is a chaotic mishmash of modules, languages, and approaches to make applications easier to code, run faster, and look better. While different approaches clearly work better in different situations, it made the most sense to develop a skillset that is efficient, effective, transferable, and not likely to disappear. So, I decided to code my pages in HTML, use Bootstrap to help with CSS formatting, make data calls to REST APIs using JQuery/Ajax, and to have Node.js on the backend interacting with my Postgres database, pulling data from the internet, and making APIs available to my web applications. The advantage of this model is that Javascript is used on both the front end and back end, Node.js is increasingly popular, and it helped me to quickly become a full stack developer.

I used the skills I gained to create increasingly complex sample webpages that call one or more APIs. Eventually, I began developing a project that used the Watson AI to study literature and devoted my time to that. I published this site on IBM Bluemix. In using Bluemix, I also learned more about GIT, Cloud Foundry, and the IBM Watson AI. Near the end of the project, after I understood more about full stack development, I also revisited some earlier topics. As one example, I made a book database web app using a MERN stack (MongoDB, Express, React, and Node). Users were authenticated using social media accounts and JSON Web Tokens (JWT).

Week 16. In the final week I collected project documentation and wrote a draft of this sabbatical report.

II. Publications/ Presentations (Scheduled)

A sample of my research is currently hosted live at aireads.com. I was also accepted to present my research at the International Association for Fantastic in the Arts conference in March 2018; however, recent health events will prevent my participation.

III. Graduate Courses and/or Seminars Attended

Although I did not complete any graduate courses, I did complete highly technical code training online and have attached a screenshot of my completion certificates/levels as evidence. Specifically, I completed the learning paths for these languages:

HTML/CSS- This learning path included 13 courses grouped according to these topics: Getting Started, Intermediate CSS, SVG, CSS Preprocessors, CSS Frameworks, Design, and Interactive Website.

JavaScript- This learning path included 17 courses grouped according to these topics: JavaScript Language, JQuery, Client-Side Frameworks, Server-Side Frameworks, Coffee Script, MEAN Stack, and AngularJS App.

PHP- This learning path included four courses on PHP and the Laravel framework.

Python- This learning path included four courses that introduced Python and Django.

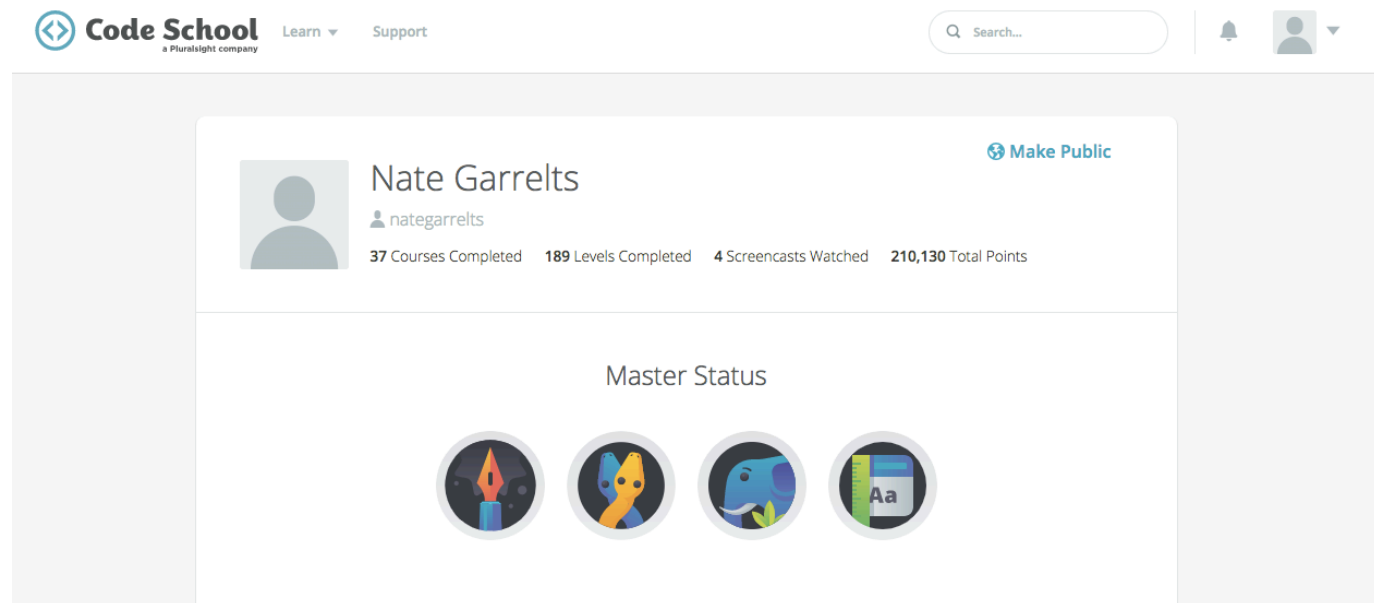
IV. Plans for Future Work

The main purpose of this research was to help me gain the skills I needed to work on other more ambitious research projects (like studying literature with AI). During the next phase of this research I will be developing other similar projects that utilize API data. I also plan to apply for a Fulbright research scholarship to continue my research. In pursuing this opportunity, I hope to extend my professional network and learn new skills.

V. Impact of the Leave on Professional Responsibilities

This research has enhanced my ability to support students in our TPC program who are interested in web design and social media. I am also prepared to serve as a resource for others on campus who may be working with this technology especially in the Accountancy, Finance, and Information Systems department. Since many topics discussed in information security are web related, I may be a valuable resource for students in ISIN and MS-ISI. Most of all, I am poised to produce some cool research that will excite students and bring positive attention to the university.

VI. Appendix



The screenshot shows the Code School user profile for Nate Garrelts. The profile includes a search bar, a notification bell, and a user menu icon. The user's name is Nate Garrelts, with the username nategarrelts. The profile statistics are: 37 Courses Completed, 189 Levels Completed, 4 Screencasts Watched, and 210,130 Total Points. A 'Make Public' button is visible in the top right corner of the profile card. Below the profile card, the 'Master Status' section displays four circular icons: a pen nib, a cat, a blue elephant, and a document with 'Aa' text.

Code School a Pluralsight company Learn ▾ Support

Q Search...

🔔 👤 ▾

Nate Garrelts [Make Public](#)

👤 nategarrelts

37 Courses Completed 189 Levels Completed 4 Screencasts Watched 210,130 Total Points

Master Status

🖋️ 🐱 🐘 📄 Aa

Figure 1. This shows the number of courses and levels I completed in Code School.

A/Reads.com

Where Machines Read

From genetic engineering in the kitchen using CRISPR to launching small computers into space, we live in an age where fantastic technology is both ubiquitous and accessible to the masses. As in the time of Mary Shelley, who wrote Frankenstein during the first industrial revolution, modern breakthroughs that alter the environment and displace workers are today welcomed by capitalists and often feared by the public. While governments and bioethicists have intervened in genetic engineering, effectively preventing any Frankenstein monsters, we have not been as proactive in thinking about and regulating artificial intelligence (AI). The admonitions of visionaries like Elon Musk and Steven Hawking are greeted with public derision, even while the media reports on studies about AI that guesses at sexuality (Kosinski & Wang, 2017) and criminal behavior (Xiaolin and Xi, 2016). But what if instead of using AI to police the world, labor in Martian colonies, or control our doomsday devices, we employed it in the humanities? This website documents my endeavor to use IBM Watson to better understand literature.

Select a text from the dropdown to see the report.

Abbott, Edwin Abbott - Flatland

Figure 2. This is a screenshot of the home page for my site that uses the Watson AI to analyze literature.

AIReads.com
Where Machines Read

[Home](#) [Login](#) [Logout](#) [Contact](#)

This form is for submitting data to AIReads database and IBM Watson. After processing, the data is stored on a PostgreSQL database that is later accessed to produce project webpages.

Author First Name

Author Last Name

Image of Author

Name of locally saved image file with extension.

Author Image Copyright Information

Paste in the copyright information or required attribution text.

Title of the Source

Year of Source

Original copyright year.

Figure 3. This is a portion of the web page that I use to enter data into my database.

Flatland

By Edwin Abbott Abbott



Image source: Public Domain/ Wikimedia Commons

Chapters 1- 10 of the complete text. Here is a short excerpt from that submission:

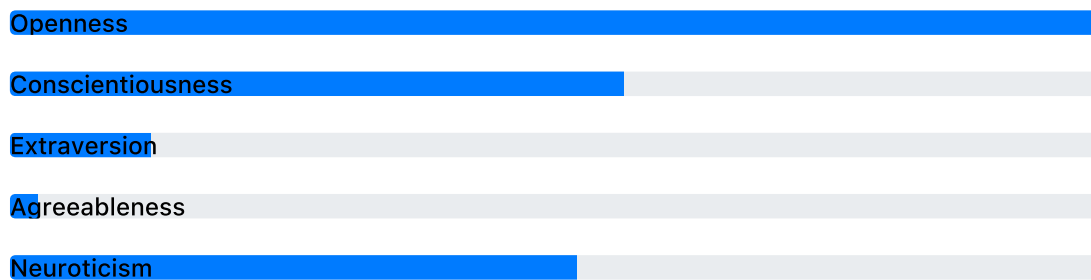
Flatland by Edwin A. Abbott 1884 To The Inhabitation of SPACE IN GENERAL
And H.C. IN PARTICULAR This Work is Dedicated By a Humble Native of
Flatland In the Hope that Even as he was Initiated into the Mysteries Of THREE
DIMENSIONS Having been previously conversant With ONLY TWO So the
Citizens of that Celestial Region May aspire yet higher and higher To the
Secrets of FOUR FIVE or EVEN SIX Dimensions Thereby contributing To the
Enlargment of THE IMAGINATION And the possible Development Of that most
and excellent Gift of MODESTY Among the Superior Races Of SOLID
HUMANITY *** FLATLAND PART 1 THIS WORLD SECTION 1 Of the Nature of

Figure 4. This is the first portion of the results that are returned to users. I used public domain texts and images.

Flatland I call our world Flatland, not because we call it so, but to make its nature clearer to you, my happy readers, who are privileged to live in Space. Imagine a vast sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remaining fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows--only hard with luminous edges--and you will then have a pretty correct notion of my country and countrymen. Alas, a few years ago, I should have said "my universe:" but now my mind has been opened to higher views of things. In such a country, you will perceive at once that it is impossible that there should be anything of what you call a "solid" kind; but I dare say you will suppose that we could at least distinguish by sight the Triangles, Squares, and other f

The full text can be found here: <https://www.gutenberg.org/ebooks/97>

Watson BIG 5 Personality Insights



Watson Needs Insights

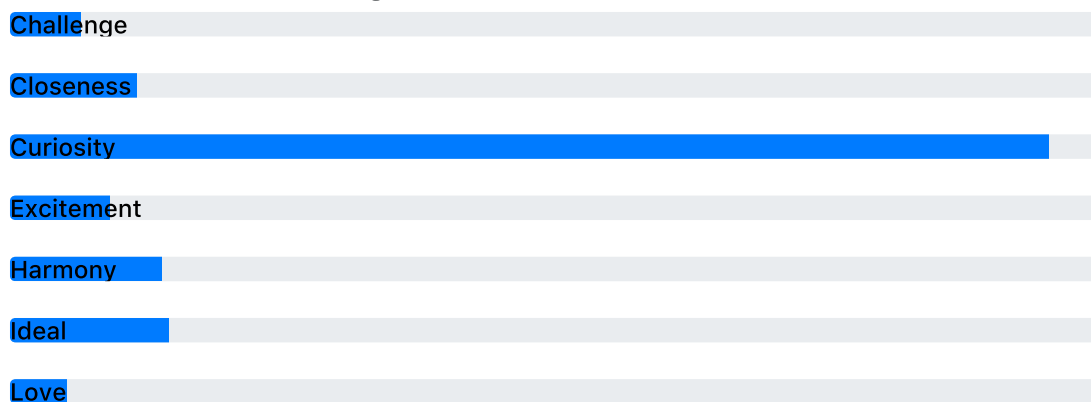


Figure 5. As users scroll down they see more information about the text and the Watson analysis. There are many other categories and this is just a very short sample.

Nate Garrelts specializes in games studies, critical informatics, and professional communication. He is currently a Professor of English at Ferris State University where he teaches upper-level writing courses in the English department and also works with undergraduate and graduate students in the Accountancy, Finance, and Information Systems department.

His first book, *Digital Gameplay* (2005), examines the intersections between games and gamers. Later books help readers understand a single game or franchise by presenting multiple academic perspectives. These books include *The Meaning and Culture of Grand Theft Auto* (2006), *Understanding Minecraft* (2014), and *Responding to Call of Duty* (2017). Dr. Garrelts is also a regular contributor to the website *Bad Subjects: Political Education for Everyday Life*, where he has written on topics ranging from information security to digital economies.

He is especially interested in using Artificial Intelligence to better understand the networks and rhythms of popular culture.

Contact

nategarrelts (at) ferris.edu.

Figure 6. The site also includes a contact page that gives information about the project.

```

<!DOCTYPE html>

<html lang="en">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <meta name="description" content="">
  <meta name="author" content="Nate Garrelts">
  <link rel="icon" href="">

  <title>AIReads.com</title>

  <!-- Bootstrap core CSS -->
  <link href="/bootstrap.min.css" rel="stylesheet">

  <!-- Custom styles for this template -->
  <link href="/narrow-jumbotron.css" rel="stylesheet">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

</head>

<body>

<script>
  //this is basically a handmade template that pulls information from the database
and puts into the page at the specified
  //id. all of the processing is done really by the javascript
  var searchParams = new URLSearchParams(window.location.search); //gets the
paraters from the URL
  var idsubmitted= searchParams.get("id")
  console.log(idsubmitted)

  $.ajax({
    type: "POST",
    url: "/samplerequest/"+idsubmitted, //notice when i submit the id do not put a
:
    dataType: "json",
    success: function(data) {
      console.log(data);
      var formattedResults = JSON.stringify(data, null);
      var obj = JSON.parse(formattedResults);
      $(document).ready(function() {
        $('#sourcetitle').html('<h1>' + obj[0].sourcetitle + '</h1>')

```

Figure 7. This is the first portion of the html code that renders webpages for each text that has been analyzed by Watson. The actual code is much longer and more complex.

```

//This is the api server for my IBM Watson Reader.
// These are the NPM packages that are required to make the server work they
need to be installed manually or as part of a package.json
// =====

var express    = require('express');          // this calls express which is a
webframework
var app        = express();                  // this defines app as using express
var bodyParser = require('body-parser');     // this requires bodyParser which is
middleware to process JSON for Express and expose the html body
var cookieParser = require('cookie-parser'); //this is cookie parsing
middleware needed for authenticating users
var session = require('express-session');    //this enables storig session data
and works with the cookie parser
var morgan = require('morgan');             //this is a debug logger to show what is
happening in the program
var myuser = require('./usercode');         //this is requiring an external js file
that actually just sets up the user login database
var path = require('path');                 // this is a core module and you don't
need to put in in the package.json it used to process paths
const PersonalityInsightsV3 = require('watson-developer-cloud/personality-
insights/v3'); //IMPORTANT the NPM for watson-developer-cloud needs to be
installed too notice the short name
const {Pool} = require('pg');              // this is the package to use a postgres
database the handlbars are an ECM6 thing
const pool = new Pool({                    // this is the connection iformation for the
pool to connect to the DB generallyly you would not put it here plainly
  host: 'sl-us-south-1-portal.10.dblayer.com', //this program uses a
postgreSQL database hosted by elephantsql
  user: 'admin',
  password: 'password deleted here',        //I have deleted mypassword for
security reasons
  database: 'compose',
  max: 10,
  port: 25113,
  idleTimeoutMillis: 30000,
  ssl: true
})

var port = (process.env.VCAP_APP_PORT || 3000); //set ibm blumix port or use
default
var host = (process.env.VCAP_APP_HOST || 'localhost'); //set ibm bluemix host or
use local host

// These functions further esablish our server envrionment
// =====
app.use(function(req, res, next) {          // this first section enables
CORS so that I can use the server accross domains
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-
Type, Accept");
  next();
});

```

Figure 8. This is a portion of the node.js code that saves data to my postgres database, sends data to Watson, and serves data to my webpage when requested. My code is heavily commented, which makes it difficult to reproduce in this format.