

Background Report: TypeScript Webapp – Linkbait Article Generator

CPSC 311 Team T3

November 16, 2015

Name: Ashley Lee
Ugrad ID: k7y8
Student ID: 34959122
Email: alee238@hotmail.com

Name: Min Seok Ray Roh
Ugrad ID: l9x9a
Student ID: 33737123
Email: msroh94@gmail.com

Name: Michelle Wilson
Ugrad ID: a2d0b
Student ID: 60855087
Email: m888wilson@gmail.com

Name: Cecile Leung
Ugrad ID: o8r6
Student ID: 90600974
Email: cecilephl@gmail.com

Name: Norman Sue
Ugrad ID: h0e9
Student ID: 20396131
Email: normansue3@gmail.com

1 Project Overview

For the 100%-level milestone, we would complete the following:

1. Write and publish an `npm` module in TypeScript which generates a JSON representation of linkbait articles.
2. Write missing TypeScript type definition files for third-party `npm` modules that our `npm` module depends on.
3. Verify that all of our written TypeScript type definition files are bugfree with [tscheck](#).
4. Use our `npm` module to create a simple Node.js+Express API server that runs the web scraper code and sends the resulting JSON articles to the browser client.
5. Write a TypeScript single-page application client that utilizes the backend JSON and renders the article as a webpage.

An explanation of how what each of these components are and how they showcase specific TypeScript language features is given in the following subsections.

1.1 Write and publish an npm module in TypeScript which generates a JSON representation of linkbait articles

We've chosen to publish our project as an npm module because, according to modulecounts.com, it's the most popular JavaScript (and thus TypeScript) package management format. By using a widely-accepted format, we can ensure the core functionality of our project can be reused by other programmers.

TypeScript language features to be used by the module:

- TODO
- TODO
- TODO

1.2 Write missing TypeScript type definition files for third-party npm modules that our npm module depends on.

We plan on scraping the text content of existing websites in order to generate our own linkbait articles. This will require using the existing [pjschrape](https://github.com/pjschneider/pjschneider) JavaScript web scraping library.

TODO: Explain and mention that `tscheck` is based on a paper <https://cs.au.dk/~amoeller/papers/tscheck/paper.pdf>

1.3 Verify that all of our written TypeScript type definition files are bugfree with `tscheck`.

1.4 Use our npm module to create a simple Node.js+Express API server that runs the web scraper code and sends the resulting JSON articles to the browser client.

1.5 Write a TypeScript single-page application client that utilizes the backend JSON and renders the article as a webpage.

2 Project Value

Of the most popular compile-to-JS languages, TypeScript greatly lags behind in adoption and has significantly fewer exemplary repositories to draw inspiration from. There are currently 8,267 TypeScript repositories on GitHub, compared with the 1.7 million JavaScript and 54,605 CoffeeScript (the leading compile-to-JS language that competes with TypeScript) repositories.

Creating an additional TypeScript project provides future potential TypeScript programmers with an additional example of how they could utilize TypeScript's static typing features.

TODO: Add reference to research paper about how using a typed language increased productivity and make comment about how that adds value.

3 Project Importance

TODO: Write about the monetary value of link clicks, e.g. cost-per-click and web advertisements, referral links, SEO, etc.

4 Project Impact

Standardized support for writing TypeScript `npm` modules that compile to JavaScript by using `tsconfig.json` files to integrate with existing JavaScript and TypeScript `npm` modules has only been added 4 months ago in TypeScript 1.5.

By publishing a library using this relatively-new build process, we are contributing to the TypeScript community by providing an additional working example of how to use this new build feature of the language, since there are only 678 results when searching for GitHub code that use `tsconfig.json` files.

4.0.1 Reminders of Specific Requirements

show the respective language features

describe, at a high level, a project on this topic that could satisfy the 100% level goals.

Potential project—the one you’d do for the 100% level—is described in the context of the project topic.

Novel technical aspects of the project topic are summarized using appropriate programming languages terminology

The technical aspects you describe should include any needed for the project proof-of-concept and plan, even if you’re stopping at the 80% level.)

value in everyday programming practice

you must base that description on existing work. How much? That will differ from project to project, but we’ll try to quantify by saying “at least the equivalent of three CS conference research papers used suitably to support your description”

perhaps because of industrial or research impact relate your topic to its context (e.g., similar languages, analyses, semantic forms, or other projects and ideas). Project topic is well-situated in its context (i.e., similar languages, analyses, or language features, depending on your topic).

4.0.2 Unsorted refs

@100 <http://noeticforce.com/alternative-programming-languages-that-compile-to-javascript>

4.0.3 Possible refs

<https://users.soe.ucsc.edu/~abadi/Papers/FTS-submitted.pdf>

4.0.4 Ideas for pivoting main project idea to better suit requirements

- dynamic type discoverer
- metaprogramming something
- news article scraper with type checking

4.0.5 Misc Ideas

Ideas:

- find specific GitHub Issues of existing JS projects with **TypeError** or other runtime-related errors that could be prevented with TS
 - Q: What specific **Errors** does TS prevent?
- ES6