**Adding a Language to Droplet Jeremiah Blanchard**

I recommend using ANTLR for any parsers you wish to add to droplet; this is the most effective general purpose branch for new language support.

If you’d like to, you could integrate a new and separate parser… but it probably isn’t necessary, and it’s probably more work than it is worth.

**What is ANTLR?**

ANTLR is a tool that can generate parsers and lexers, given a grammar, that can run in multiple languages. We’re using the JavaScript target in this case.

**How to Create a Parser/Lexer using ANTLR**

Put the grammar file for the new language in the antlr directory in the cloned repo and install antlr4 (Google for directions). You can generate the lexer and parser from the grammar (assuming the new language is called NewLanguage):

* antlr4 -Dlanguage=JavaScript NewLanguage.g4

This will generate several files that can then be used for Droplet. **NOTE: the grammar file must be modified / updated so that it can be handled properly by Droplet in some cases by duplicating grammar nodes and adding \_DropletFile in order to handle EOF elements.** This may be language-dependent; I’m still exploring whether any language do not need this. Your mileage may vary.

**Adding the Language to Droplet**

Droplet imports parsers in the src/antlr.coffee file. In this file, add the following to the map/dictionary **ANTLR\_PARSER\_COLLECTION**:

* 'NewLanguageLexer': require('../antlr/NewLanguageLexer'),  
  'NewLanguageParser': require('../antlr/NewLanguageParser'),

You’ll also need to define the syntax tree highlighting in src/languages/newlanguage.coffee.

Once this is complete, import the new language in src/modes.coffee add the new language to the map of **module.exports**:

* newlanguage = require './languages/newlanguage.coffee'  
  …  
  module.exports = {  
   …  
   ‘newlanguage’: newlanguage  
  }

**Language Definition File**

The language definition file (src/languages/newlanguage.coffee) identifies the coloring and placement of nodes in blocks modes. The easiest way to understand this is to…

Select another language

Generate a standalone parser for that language using the grammar file in Droplet

Parse a few sample source file

Identify how the nodes are classified in existing language files

The most complete language file is the C language definition; Java (not JVM!) is also a good place to look for ideas.