Technical Report

Attempting to use the Lingo Client

We followed the ‘Getting started’ guide on http://docs.codelingo.io/ to set use the on premise instance of CodeLingo. To start with we attempted to add a folder to the %PATH% as it specified in the guide, however we was unfamiliar with the concept and wasn’t sure we had succeeded so we ended up placing the Lingo binary into one of the directories already defined in %PATH% (fallcj2\AppData\Local\Microsoft\WindowsApps)

From there we opened the git bash and navigated to the folder where we ran ‘./lingo.exe setup’ and were asked for a CodeLingo username. We then went to codelingo.io and created an account and input the username, then we were prompted for a User-Token, we attempted to get one from https://codelingo.io/lingo-token but the button to generate a token didn’t seem to work, at first we believed that we were missing something, we checked to make sure the token wasn’t on our clipboard and even used the inspect element tool to try discover the secrets of the button. When that didn’t work we tried changing browsers, initially we were using Internet Explorer, and we tried Google Chrome and Firefox but with no luck.

Next session we met with Carlin and found that we were getting the token from the wrong place, we needed a token from the local instance of CodeLingo which we could access at 10.118.26.6:30303. From there we set the local variable ‘onprem’ to 10.118.26.6 and created and initialized a Git repo and created a test php file as outlined by the guide and then ran the ‘lingo new’ command as specified in the guide, however it didn’t work and just brought up a help menu. After this we spent the remainder of the session experimenting with the other possible commands such as ‘list-facts‘ and ‘describe-fact’

The next session we had Carlin come in to help us, apparently it wasn’t working because we were trying to run the lingo.exe from inside the Git repo, once we removed it worked and we spent the rest of the session using the Lingo client to look through PHP code for functions.

The C# lexicon

We Received a C# lexicon from CodeLingo and soon began to try figure out how it functions,

To convert the C# code to Java there were several steps, first on all the C# code would be commented out so it could be referred to, then we would examine the line of code and figure out what it does, usually the msdn documentation was helpful for this, and then replicate the code in Java, often stackoverflow was very helpful with

this.

```
//Object relativeUri = fromUri.MakeRelativeUri(toUri);
URI relativeUri = fromUri.relativize(toUri);
//Object relativePath = URI.UnescapeDataString(relativeUri.toString());
String temp = relativeUri.toString();
URL tempURL = new URL(temp);
String relativePath = tempURL.toURI().toString();
//https://stackoverflow.com/questions/8317505/c-sharp-uri-escapedatastring-equivalent-for-java
```

The Docker File

We looked at the docker files, and decided to leave those until we had managed to replicate the lexicon. Though as we never succeeded in completely recreating the lexicon we also didn't get back to the Docker File.