

Developers guide (Front-End)

Note : For Linux and Mac users only.

Overview :

This project was built for AssessMachines([this](#)). The project uses ace editor as the base editor. For more information have a look at [this](#). The editor will use it's intellisense for the Java, C++, C and C#.

1. Starting the virtual enviornment :

Type in `source bin/activate` in the console. There is no need to install Django specifically.

2. Intellisense data management:

2.1 Extracted Data :

We have extracted data from most of the important libraries and classes. All the `localhost:8000/data/lang_id/lib`.

Example : `localhost:8000/c_cpp/stdio.h` (Loads data for stdio.h) Example : `localhost:8000/c_cpp/stdio.h,stdlib.h` (Loads data for both stdio.h and stdlib.h)

The extracted data is then intellisensely shown into editor.

2.2 Snippet Data :

The extracted data is highly abstract and can go very complicated in a complex manner. So we also store some intellisense data in form of snippets which is a highly common way in which the user wants the intllisense data to be.

3. A guide through editor:

3.1 Embedding ace inside the code :

Project has it's ace code embedded in the file `/src/templates/index.html`. For details about embedding ace code, have a look at this [this](#).

3.2 Enabling ace's intellisense :

To enable ace's intellisense features just have a look through `/src/static/enableIntellisense.js`.

3.3 Language ID's :

Language ID	Language
<code>c_cpp</code>	C
<code>c_cpp</code>	C++

Language ID Language

java	Java
csharp	C#

3.4 Adding snippets :

Snippets are a easy way of storing intellisense features inside of ace. In snippets the data can be stored according to users need very easily. For adding more snippets to the editor , Go to `src/static/src-noconflict/snippets`. To select a particular languages snippet just choose the `lang_id.js` file from the current directory.

3.5 Showing Intellisense from Extracted data:

First add the function which will handle adding all the intellisense data in `/src/static/enableIntellisense.js`. All the intellisense data reterival is done in the function `functionExtractor` in the file `/src/templates/index.html`.

3.6 Adding more themes:

Currently the project has three editor themes. For adding more themes, look at `/src/src-noconflict/theme-id.js`.

4. A guide through functions:

All the functions below are in the file `/src/static/index.js`.

4.1 changLang:

The function detects the change in language and store it to the global variable `lang` which helps in fetching the intellisense data.

4.2 changTheme:

The function detects the change in theme and set the theme to the selected theme.