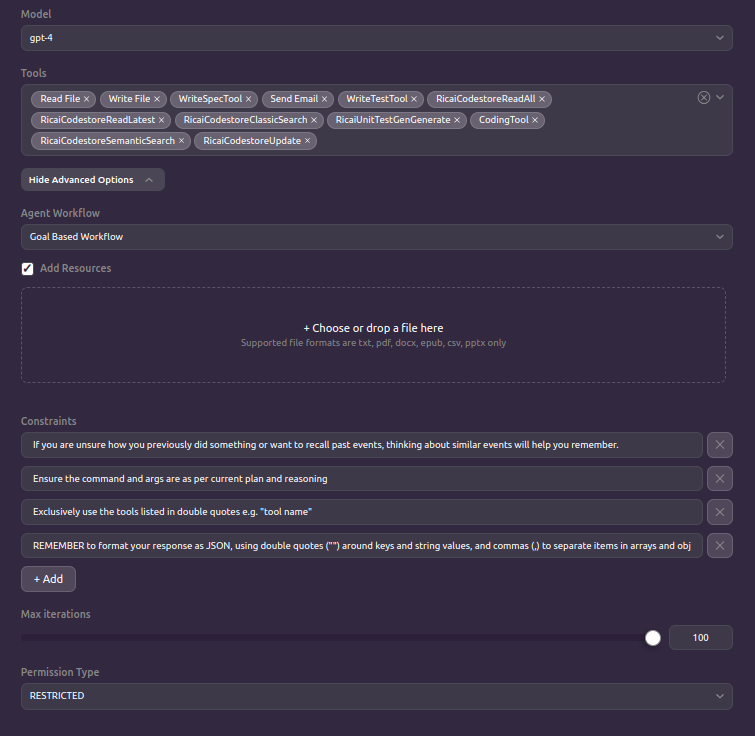
**The Agent configuration**

Goals (in order):

* Use the tool named "RicaiCodestoreClassicSearch" in order to get the the specific code from the github repository. For the "repository" parameter of the tool, supply "CV\_generator" and for the "location" parameter of the tool, supply "app.py". It returns a json string.
* Using the Resource tool, inside the resources find the file "requirement\_doc.txt" and read its contents.
* Now, use the tool named "RicaiUnitTestGenGenerate" in order to generate unit tests for given code with provided requirements. For tool parameter "code\_files" supply the json string with code obtained from the first step using "RicaiCodestoreReadClassiSearch" and for the parameter "requirements" supply the plain text extraced in second step from the "requirement\_doc.txt".
* After the "RicaiUnitTestGenGenerate" tool has completed its work, demonstrate its output on the screen and write its output to a file named "test\_app.py".
* Using the Resource tool, inside the resources find the file "report.txt" and read its contents.
* Now imagine that your task is to read the contets of "report.txt" document from previous step, which contains software test reports and you have to describe each test case in the document in human understandable language. Feel free to read all the document, add yur comments and suggestions.
* After you finished writing your document in the previous step, save all your work into a file named "agent\_report.txt".
* Using the email tool, send the file "agent\_report.txt" to the email address "<your address>".

Other configuration:



(most of that is taken from the SuperCoder template)