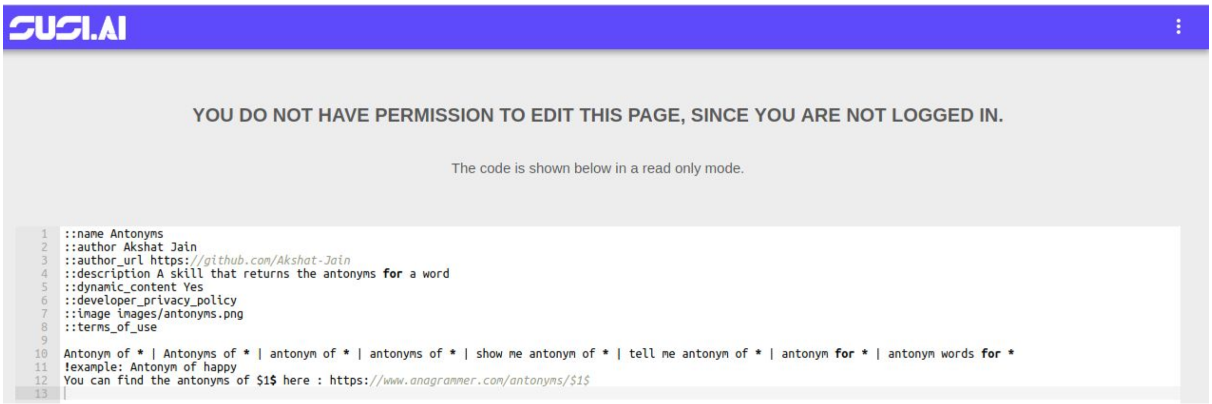


YOU DO NOT HAVE PERMISSION TO EDIT THIS PAGE, SINCE YOU ARE NOT LOGGED IN.

The code is shown below in a read only mode.

```
::name Antonyms
::author Akshat Jain
::author_url https://github.com/Akshat-Jain
::description A skill that returns the antonyms for a word
::dynamic_content Yes
::developer_privacy_policy
::image images/antonyms.png
::terms_of_use

Antonym of * | Antonyms of * | antonym of * | antonyms of * | show me antonym of * | tell me antonym of * | antonym for * | antonym words for *
!example: Antonym of happy
You can find the antonyms of $1$ here : https://www.anagrammer.com/antonyms/$1$
```

Skill Development in SUSI.AI

There are a lot of personal assistants around like Google Assistant, Apple’s Siri, Windows’ Cortana, Amazon’s Alexa, etc. What is then special about SUSI which makes it stand apart from all the different assistants in the world ?

Well, the thing that makes SUSI different is that it gives the Users the ability for them to create their own skills. You don’t need to be a developer to be able to enhance SUSI.

SUSI is an Open Source personal assistant which can do a lot of incredible stuff for you, made by you.

How to make your own skill ?

So, let’s say you want to create your own Skill and add it to the existing SUSI Skills. So, these are the steps you need to follow regarding the same -

1. The current SUSI Skill Development Environment is based on an Etherpad. An Etherpad is a web-based

collaborative real-time editor. https://dream.susi.ai/ is one such Etherpad. Open https://dream.susi.ai/ and name your dream (in lowercase letters).

2. Define your skill in the Etherpad. The general skill format is

```
::name <Skill_name>
::author <author_name>
::author_url <author_url>
::description <description>
::dynamic_content <Yes/No>
::developer_privacy_policy <link>
::image <image_url>
::term_of_use <link>

#Intent
User query1|query2|query3....
Answer answer1|answer2|answer3...
```

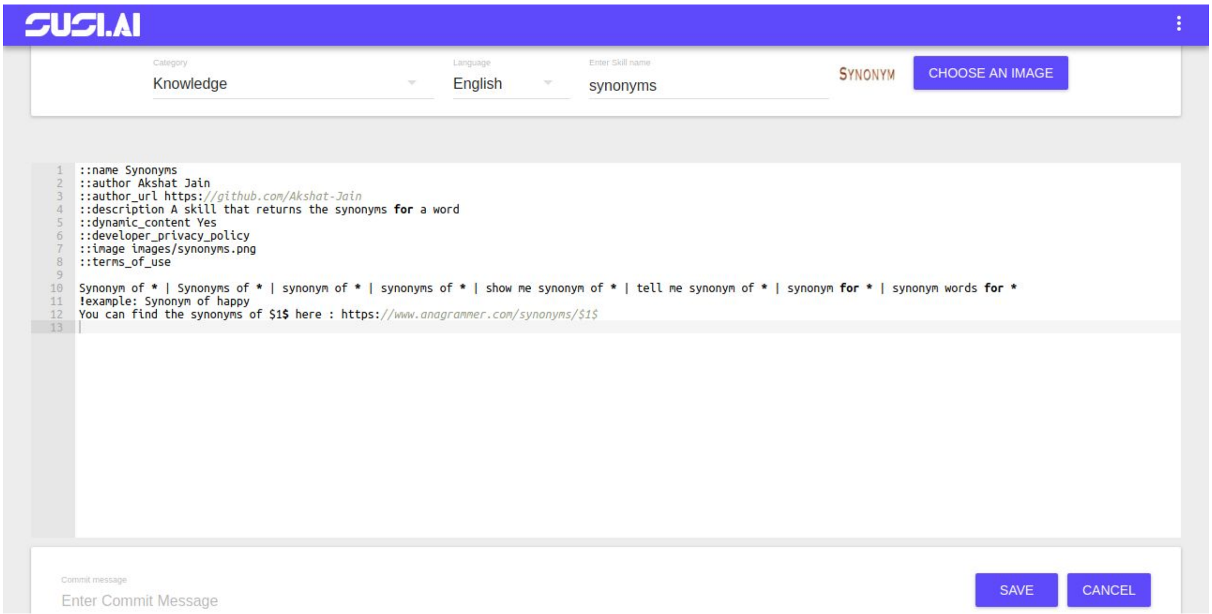
   Patterns in query can be learned easily via this tutorial.

3. Open any SUSI Client and then write dream <your dream name> so that dreaming is enabled for SUSI. Once dreaming is enabled, you can now test any skills which you’ve made in your Etherpad.
4. Once you’ve tested your skill, write ‘stop dreaming’ to disable dreaming for SUSI.
5. If the testing was successful and you want your skill to be added to SUSI Skills, send a Pull Request to susi_skill_data repository providing your dream name.

How do you modify an existing skill ?

SUSI Skill CMS is a web interface where you can modify the skills you’ve made. All the skills of SUSI are directly in sync with the susi_skill_data.
To edit any skill, you need to follow these steps -

1. Login to SUSI Skill CMS website using your email and password (or Sign Up to the website if you haven’t already).
2. Click on the skill which you want to edit and then click on the “pencil” icon.
3. You can edit all aspects of the skill in the next screen. The screen looks like the below image.



Category: Knowledge
Language: English
Enter Skill name: synonyms

```
::name Synonyms
::author Akshat Jain
::author_url https://github.com/Akshat-Jain
::description A skill that returns the synonyms for a word
::dynamic_content Yes
::developer_privacy_policy
::image images/synonyms.png
::terms_of_use

Synonym of * | Synonyms of * | synonym of * | synonyms of * | show me synonym of * | tell me synonym of * | synonym for * | synonym words for *
!example: Synonym of happy
You can find the synonyms of $1$ here : https://www.anagrammer.com/synonyms/$1$

```

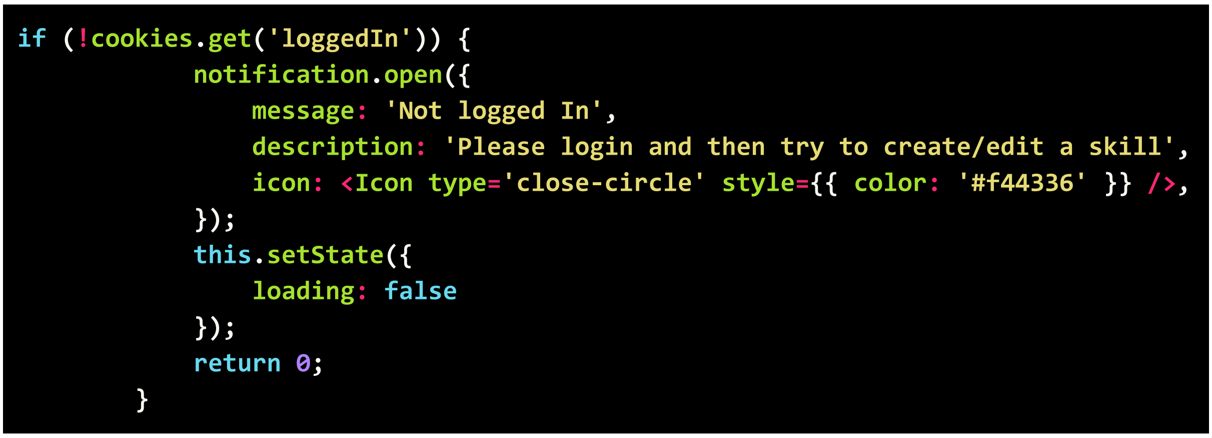
Commit message: Enter Commit Message

4. Make the changes and then click on “SAVE” button to save the skill.

What’s happening Behind The Scenes in the EDIT process ?

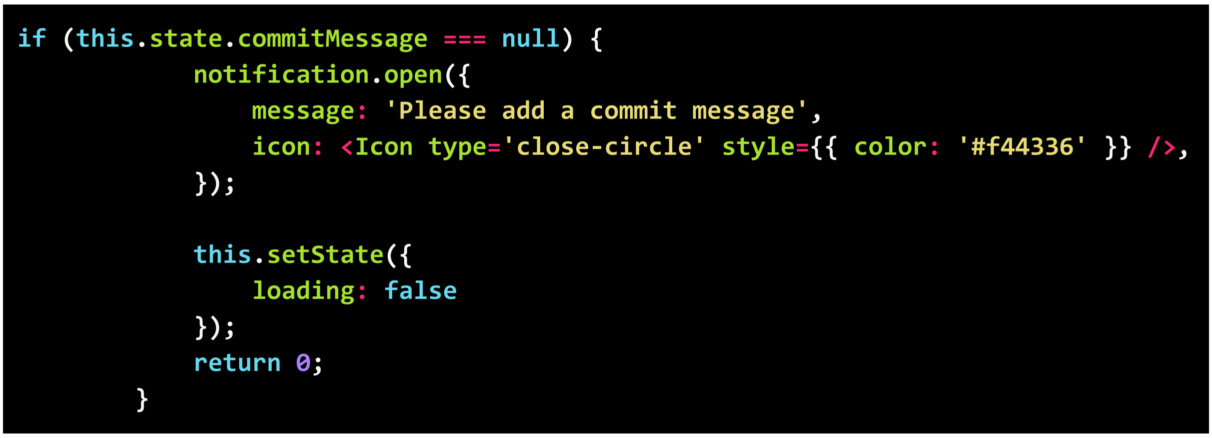
➢ SkillEditor.js is the file which is responsible for keeping a check over various validations in the Skill Editing process. There are certain validations that need to be made in the process. Those are as follows -

- **Check whether User has logged in or not**



```
if (!cookies.get('loggedIn')) {
            notification.open({
                message: 'Not logged In',
                description: 'Please login and then try to create/edit a skill',
                icon: <Icon type='close-circle' style={{ color: '#f44336' }} />,
            });
            this.setState({
                loading: false
            });
            return 0;
        }
```

- **Check whether Commit Message has been entered by User or not**



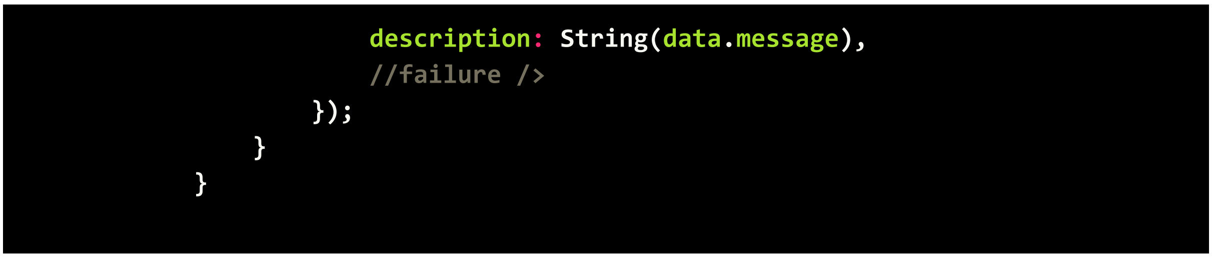
```
if (this.state.commitMessage === null) {
            notification.open({
                message: 'Please add a commit message',
                icon: <Icon type='close-circle' style={{ color: '#f44336' }} />,
            });

            this.setState({
                loading: false
            });
            return 0;
        }
```

- **Check to ensure that request is sent only if there are some differences in old values and new values**

```
if (this.state.oldGroupValue === this.state.groupValue &&
          this.state.oldExpertValue === this.state.expertValue &&
          this.state.oldLanguageValue === this.state.languageValue &&
          !this.state.codeChanged && !this.state.image_name_changed) {
            notification.open({
                message: 'Please make some changes to save the Skill',
                icon: <Icon type='close-circle' style={{ color: '#f44336' }} />,
            });
            self.setState({
                loading: false
            });
            return 0;
        }
```

➢ After doing the above validations, a request is sent to the Server and the User is shown a notification accordingly, whether the Skill has been uploaded to the Server or there has been some error.

```
$.ajax(settings)
            .done(function (response) {
                this.setState({
                    loading: false
                });
                let data = JSON.parse(response);
                if (data.accepted === true) {
                    notification.open({
                        message: 'Accepted',
                              description: 'Your Skill has been uploaded to the
server',
                        //success/>
                    });
                }
                else {
                    this.setState({
                        loading: false
                    });
                    notification.open({
                        message: 'Error Processing your Request',
```



```
                        description: String(data.message),
                        //failure />
                    });
                }
            }
```

- ➢ If the User is notified with a Success notification, then to verify whether the Skill has been added or not, the User can go to [susi_skill_data]() repo and see if he has a recent commit regarding the same or not.

Resources

- Notifications - https://ant.design/components/notification/
- Material UI - https://www.material-ui.com/#/
- Tutorials for Skill Development - https://github.com/fossasia/susi_skill_cms/blob/master/docs/Skill_Tutorial.md
- Relevant Blogs -
    - https://blog.fossasia.org/skills-for-susi/
    - https://blog.fossasia.org/how-to-teach-susi-ai-skills-using-external-apis/