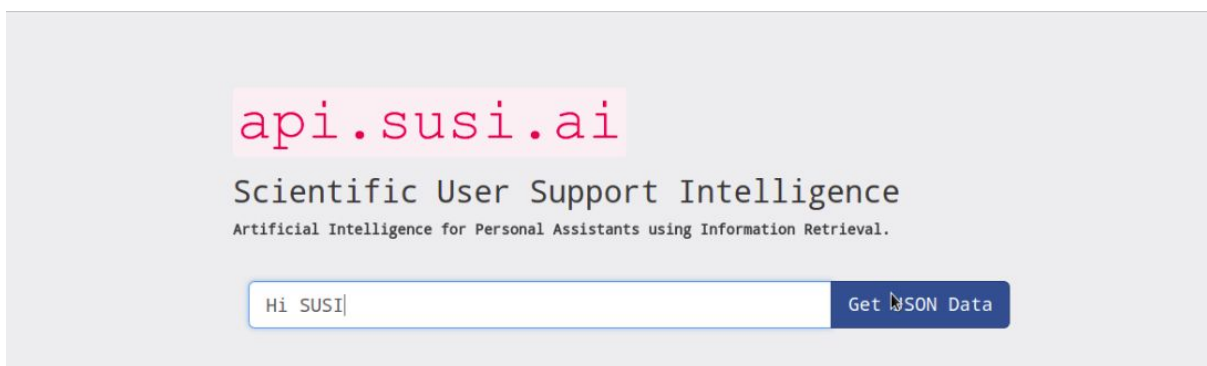


How api.susi.ai responds to a query and send response



The frontend to directly access the raw data for a query is api.susi.ai. It is an API of susi server which sends responses to a query by a user in form of a **JSON**(JavaScript Object Notation) object(more on JSON [here](#)). This JSON object is the raw form of any response to a query which is a bunch of key(attribute) value pair. This data is then send from the server to various APIs like chat.susi.ai, susi bots, android and ios apps of SUSI.AI.

Whenever a user in using an API, for example chat.susi.ai, the user send a query to the API. This query is then sent by the API as a request to the susi server. The server then process the request and sends the answer to the query in form of json data as a response to the request. This request is then used by the API to display the answer after applying stylings to the json data.



The json data looks like this(below is the response to hi susi by the server):

```
{
  "query": "hi susi",
  "count": 1,
  "client_id": "aG9zdF8xNjluMTU4LjE2NS4zMjF9iZTk0MGE4Yg==",
  "query_date": "2018-05-14T13:22:18.473Z",
  "answers": [{
    "data": [{
      "0": "hi susi",
      "1": "susi",
      "token_original": "hi",
      "token_canonical": "hi",
```

```

    "token_categorized": "hi",
    "timezoneOffset": "-330",
    "language": "en",
    "answer": "Hi! I'm SUSI",
    "skill_link":
    "https://github.com/fossasia/susi_server/blob/development/conf/susi/en_0700_ai_play.json",
    "query": "hi",
    "skill": "/susi_server/conf/susi/en_0700_ai_play.json"
  },
  "metadata": {"count": 1},
  "actions": [{
    "type": "answer",
    "language": "en",
    "expression": "Hi! I'm SUSI"
  }],
  "skills": ["/susi_server/conf/susi/en_0700_ai_play.json"],
  "persona": {}
},
"answer_date": "2018-05-14T13:22:18.479Z",
"answer_time": 6,
"language": "en",
"session": {"identity": {
  "type": "host",
  "name": "162.158.165.30_be940a8b",
  "anonymous": true
}}
}

```

Whenever a request to server is made the server process the data and looks for the skills file for the corresponding file. For our above example(hi susi) the required skill file is [susi_server/conf/susi/en_0700_ai_play.json](https://github.com/fossasia/susi_server/blob/development/conf/susi/en_0700_ai_play.json)

The format of file names for the skills is **<language>_<index>_<title_of_skill>.json**

A snippet of code of the [susi_server/conf/susi/en_0700_ai_play.json](https://github.com/fossasia/susi_server/blob/development/conf/susi/en_0700_ai_play.json) file is:

```

{"rules":[
  {
    "keys" :["help"],
    "phrases":[ {"type":"pattern", "expression":"I need some help with *"},
                  {"type":"prior", "expression":"I need some help"},
                  {"type":"pattern", "expression":"I'm in need of help *"},
                  {"type":"prior", "expression":"I need some help with this problem"}
    ],
    "actions":[ {"type":"answer", "select":"random", "phrases":[

```

```

    "Would love to help! In what way can I help you?"
  ]}
}, {
  "phrases": [ {"type": "pattern", "expression": "*sorry*"} ],
  "actions": [ {"type": "answer", "select": "random", "phrases": [
    "Please don't apologise.",
    "Apologies are not necessary.",
    "It did not bother me. Please continue."
  ]}
], {
  "phrases": [ {"type": "pattern", "expression": "hello*"},
    {"type": "pattern", "expression": "hey*"},
    {"type": "pattern", "expression": "hi*"}
  ],
  "actions": [ {"type": "answer", "select": "random", "phrases": [
    "Hi! I'm SUSI",
    "Hello!"
  ]}
],
},

```

In the rules attribute we can see a number of phrases attributes are there, each corresponding to a specific class of query like hello, sorry, help etc. In our example we have the query “hi susi” which corresponds to the hello type phrase. From this code block:

```

{
  "phrases": [ {"type": "pattern", "expression": "hello*"},
    {"type": "pattern", "expression": "hey*"},
    {"type": "pattern", "expression": "hi*"}
  ],
  "actions": [ {"type": "answer", "select": "random", "phrases": [
    "Hi! I'm SUSI",
    "Hello!"
  ]}
],
},

```

we see an array of length 3 with expressions with ‘hello’, ‘hi’, or ‘hey’ keywords. SUSI assumes all 3 types of queries same and performs same actions for these three types of expressions which are directed below in the actions attribute.

‘actions’ is an array with three columns:

- **type** : this defines the type of response(here answer).
- **select** : This defines the type of action like random or the value of a specific index of phrases[i] array. Here the value is random, which means that a random value from the phrases array will be the answer for this query with both “Hi! I’m SUSI” and “Hello!” have equal probability to appear as the answer.
- **phrases** : It’s an array which holds all the possible answers for a query.

The server process and create the response json data file with other attributes like the time of creation of an answer, language, the client Id, answer time, session information, link to the skills file and skill used etc. and sends the response to the client.

References:

- <https://en.wikipedia.org/wiki/JSON>
- https://github.com/fossasia/susi_server/blob/development/src/ai/susi/json/JsonFile.java
- <https://api.susi.ai/susi/chat.json?timezoneOffset=-330&q=hi+susi>
- <https://api.susi.ai/>