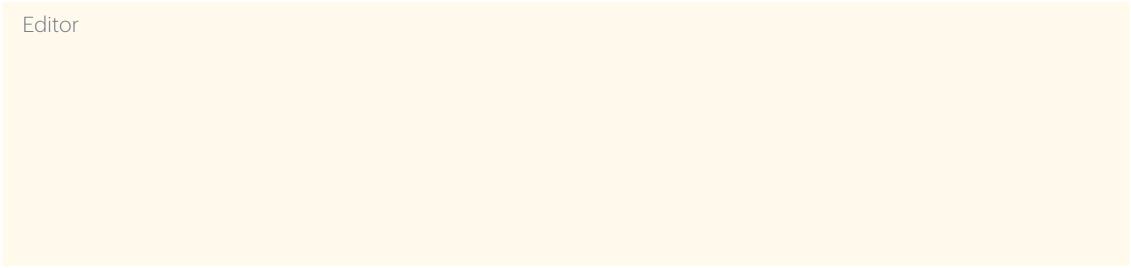


BroadAI MAS Framework

to develop applications enhanced by AI capabilities



```
import BroadAI from 'broadai';
import rs from 'broadai-agents/Researcher';
const ai = new BroadAI([rs.agent], {/**/});
rs.register(ai.config, {
    appid: "dswZsbf",
    personality: { name: "Jack",
      role: "assistant", pov: "first-person" },
    llm: {
      provider: "openai",
      model: "gpt-3.5-turbo-0125",
      apikey: env['OPENAI_KEY']
    history: {
      enabled: false, max_exchanges: 5
```















































































































Finally, simply call **go()** method to invoke Generative Al

capability on behalf of your application

go() works like this:

- Based on the question, BroadAI generates an objective and a plan to achieve that objective
- If the 2nd parameter is **true**, BroadAI **audits** the **plan** to ensure it can accurately and optimally achieve the objective.
- The updated **plan** is a series of steps comprising of identified agents and skills that can provide incremental results to the overall objective.
- This **plan** is executed by BroadAI and results are collected..
 - Collected results are used to ground the last step which also takes any past **conversations** into account to generate the final **response** to user's original **question**.

```
ai.go(question, true, [ ]).then((resp)=>{
 console.log(resp);
});
```

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

```
Editor
import BroadAI from 'broadai';
import rs from 'broadai-agents/Researcher';
const ai = new BroadAI([rs.agent], {/**/});
rs.register(ai.config, {
    appid: "dswZsbf",
    personality: { name: "Jack",
      role: "assistant", pov: "first-person" },
    llm: {
      provider: "openai",
      model: "gpt-3.5-turbo-0125",
      apikey: env['OPENAI_KEY']
    history: {
      enabled: false, max_exchanges: 5
  });
```

Finally, simply call **go()** method to invoke Generative AI capability on behalf of your application

```
ai.go(question, true, []).then((resp)=>{
  console.log(resp);
});
```

go() works like this:

- Based on the question, BroadAI generates an objective and a plan to achieve that objective
- If the 2nd parameter is **true**, BroadAl **audits** the **plan** to ensure it can accurately and optimally achieve the objective.
- The updated **plan** is a series of steps comprising of identified agents and skills that can provide incremental results to the overall objective.
- This **plan** is executed by BroadAI and results are collected..
- Collected results are used to ground the last step which also takes any past conversations into account to generate the final response to user's original question.

BroadAI MAS Framework

to develop applications enhanced by AI capabilities

Question: Find fast food joints around Columbus airport