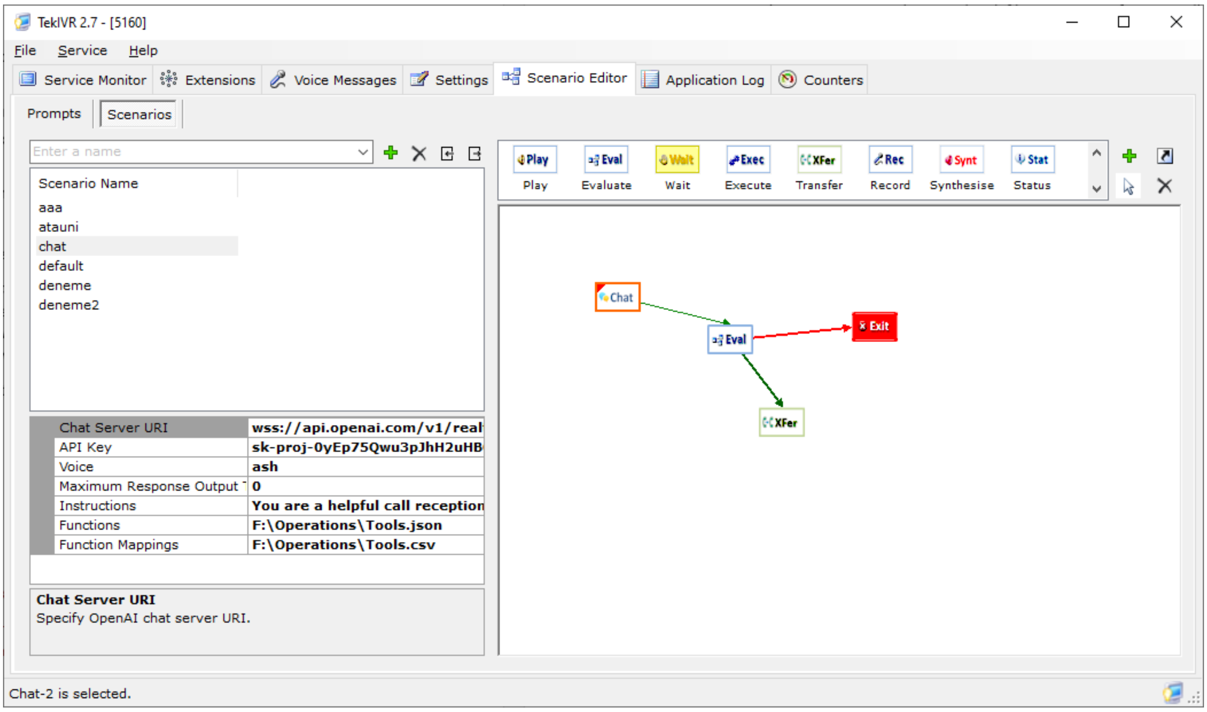
TekIVR Simple Auto Attendant with OpenAI

TekIVR Chat action in scenarios allows you access to OpenAI real time API. You can access OpenAI via real time API using voice commands. OpenAI real time API also allows you to execute user define functions and get / post data to enterprise databases and communicate with enterprise applications.

You can create a simple auto attendant with TekIVR using OpenAI real time API. TekIVR latest version introduces "Chat" action in scenarios. Chat action allows you to connect OpenAI real time API.



Scenario Name
aaa
atauni
chat
default
deneme
deneme2

Chat Server URI	wss://api.openai.com/v1/real
API Key	sk-proj-0yEp75Qwu3pJhH2uHB
Voice	ash
Maximum Response Output	0
Instructions	You are a helpful call reception
Functions	F:\Operations\Tools.json
Function Mappings	F:\Operations\Tools.csv

Chat Server URI
Specify OpenAI chat server URI.

Chat-2 is selected.

TekIVR Scenario Editor

Chat action has following parameters:

- **Chat Server URI.** This is a secure WebSocket URL (WSS) to connect (wss://api.openai.com/v1/realtime? model=gpt-4o-realtime-preview-2024-10-01)
- **API Key.** This is obtained from OpenAI dashboard Organications / Projects / API Keys section. You need to create a project for the IVR scenario.
- **Maximum Response Output.** The maximum number of tokens that can be generated in the chat completion.
- **Instructions.** The system instructions that the assistant uses.

  Instructions describe overall assistant behavior. You should restrict assistant operation only for transferring incoming calls to the desired department or extension. Example:

  You are a helpful call receptionist for company called KaplanSoft and you always speak in English. Greet the calling party with saying \"Welcome to KaplanSoft! How may I help you?\" without waiting for user input. Customer can ask his or her order status and customer can also request his or her call to be transferred to a company department.

- **Functions.** Full path of the JSON formatted text file which contains user defined functions which can be used by the assistant.

  Functions file specifies function to be called when needed. You need to have a function named "transfer_call" for this auto attendant scenario. Example:

```
[
  {
    "type": "function",
    "name": "transfer_call",
    "description": "Transfer call to an extension. Department extension numbers are accounting = 201,
                     sales = 202, marketing = 203 and technical support = 104. Say \"I'm transferring
                     your call. Please stay on the line.\" prior to transferring the calls. Do not ask
                     any further questions after transferring a call.",
    "parameters": {
      "type": "object",
      "properties": {
        "extension": {
          "type": "string",
          "description": "Extension number of the desired department"
        }
      },
      "additionalProperties": false,
      "required": [
        "extension"
      ]
    }
  }
]
```

  Descriptions can be written in your native language if it's supported by OpenAI. Try to be precise and use minimum words as much as possible.

  There are two built-in functions in TekIVR; transfer_call and terminate_call. You can override them by specifying your own functions. Default definitions for these built-in functions:

```
[
  {
    "type": "function",
    "name": "transfer_call",
    "description": "Transfer call to an extension",
    "parameters": {
      "type": "object",
      "properties": {
        "extension": {
          "type": "string",
          "description": "Extension number of the desired extension"
        }
      },
      "additionalProperties": false,
      "required": [
        "extension"
      ]
    }
  },
  {
    "type": "function",
    "name": "terminate_call",
    "description": "Terminate the current conversation. This function is called when the calling
                     party has no further requests or says bye."
  }
]
```

  transfer_call function sets TekIVR internal variable `%received_digits%` to received extension value. You need to evaluate `%received_digits%` variable value using Evaluate action and perform Transfer action based on Evaluate action. This typically done connecting Chat action to Evaluate action and connecting Evaluate action to Transfer actions.

- **Function Mappings.** Full path of the CSV file which contains function to local executable or web service URL mappings.

  This auto attendant scenario uses only “transfer_call” function and this is processed by TekIVR internally and this does not need a function mapping. Bu you can have a mapping to perform additional actions while transferring the call by specifying a mapping. You can perform a HTTP GET request to a web server URL or invoke a DOS batch file prior to transferring a call for instance. Example Tools.csv

  ```
  transfer_call,"F:\Operations\cdr.bat %calling_number% %called_number%"
  ```

  Contents of cdr.bat

  ```
  @echo off

  echo %1 %2 >> F:\Operations\cdr.csv

  echo Say \"Stay on the line, I'm transferring your call\" in the conversation language.
  ```

  Batch file output will be used as function description.

In our sample scenario, Chat action is linked to an Eval action which evaluates `%received_digits%` variable and decides next action to be performed in the scenario. Next action will be the default action if a numerical extension value is received otherwise the scenario will be terminated using Exit action.