Manual de Usuario: uMakerAi.ToolFunctions

Introducción

La unidad ***uMakerAi.ToolFunctions*** proporciona un componente Delphi diseñado para gestionar y exponer funciones (o "tools") que pueden ser utilizadas por un modelo de Inteligencia Artificial (IA). Este componente permite definir, configurar y ejecutar funciones, permitiendo que el modelo de IA interactúe con el mundo exterior, acceda a información, o realice tareas específicas.

Componentes Principales

La unidad ***uMakerAi.ToolFunctions*** consta de los siguientes componentes principales:

- ***TAiFunctions***: Es el componente principal, que gestiona una colección de funciones.
- ***TFunctionActionItem***: Representa una función individual que puede ser llamada por el modelo de IA.
- ***TFunctionParamsItem***: Representa un parámetro individual de una función.
- ***TFunctionEvent***: Define el tipo de evento que se dispara cuando se ejecuta una función.

Tipos de Datos y Enumeraciones

- **TToolstype = (tt_function, ttNone);**:
    - Define los tipos de "tools" soportados. Actualmente solo tt_function está implementado.
- **TToolsParamType = (ptString, ptInteger, ptBoolean, ptFloat, ptDate, ptTime, ptDateTime, ptBase64);**:
    - Define los tipos de datos permitidos para los parámetros de las funciones.

Uso del Componente TAiFunctions

1. **Agregar el Componente al Formulario:**

- Arrastra y suelta el componente TAiFunctions desde la paleta de componentes en tu formulario Delphi. Esto creará una instancia del componente, por ejemplo, AiFunctions1.

2. **Definir Funciones:**
    - Haz doble clic en el componente AiFunctions1 en el formulario. Esto abrirá el editor de propiedades del componente.
    - Haz clic en los puntos suspensivos (...) al lado de la propiedad Functions. Esto abrirá el editor de colecciones de Delphi.
    - Haz clic en el botón ***Add*** para crear una nueva función (***TFunctionActionItem***).
    - Configura las propiedades de la función:
        - **FunctionName**: El nombre de la función (ej. getCurrentWeather). Este es el nombre que el modelo de IA usará para referirse a la función.
        - **Enabled**: Indica si la función está habilitada. Si está deshabilitada, no se expondrá al modelo de IA.
        - **Description**: Una descripción de la función (ej. "Obtiene el clima actual para una ciudad dada").
        - **Default**: Indica si esta función es la función por defecto que se ejecutará si el modelo de IA no especifica un nombre de función.
        - **Parameters**: Haz clic en los puntos suspensivos (...) al lado de la propiedad Parameters. Esto abrirá el editor de colecciones para los parámetros de la función.
            - Haz clic en el botón Add para crear un nuevo parámetro (TFunctionParamsItem).
            - Configura las propiedades del parámetro:
                - **Name**: El nombre del parámetro (ej. city).
                - **ParamType**: El tipo de datos del parámetro (ej. ptString).
                - **Required**: Indica si el parámetro es obligatorio.

- **Description**: Una descripción del parámetro (ej. "La ciudad para la cual se desea obtener el clima").
- **Enum**: Una lista de valores permitidos para el parámetro (opcional, separados por comas). Se utiliza para restringir los valores que puede tomar el parámetro. Por ejemplo, si ParamType es String, podrías restringirlo a "Londres,Nueva York,Paris".

- **OnAction**: Selecciona el evento OnAction y haz clic en los puntos suspensivos (...). Esto abrirá el editor de código para el evento. Aquí es donde debes escribir el código Delphi que se ejecutará cuando se llame a la función.

3. **Código para el Evento OnAction:**

   Dentro del evento OnAction, debes:

   - Obtener los valores de los parámetros de la función desde el objeto ToolCall (que es una instancia de TAiToolsFunction).
   - Realizar la lógica de la función.
   - Establecer la propiedad Handled a True si la función se ejecutó correctamente, o a False si ocurrió un error.
   - (Opcional) Asignar un resultado a la propiedad Result del objeto ToolCall para devolverlo al modelo de IA.

   Ejemplo:

```
procedure TForm1.GetCurrentWeatherAction(Sender: TObject; FunctionAction:
TFunctionActionItem; FunctionName: String; ToolCall: TAiToolsFunction; var
Handled: Boolean);
var
  City: string;
  WeatherInfo: string;
begin
  try
    City := ToolCall.Parameters.Values['city'];   // Obtener el valor del
parámetro "city"

    // Validar si existe el valor y no es vacío
    If Trim(City) = '' then
    Begin
      ShowMessage('Error: El parámetro "city" es obligatorio');
      Handled := False;
      Exit;
```

```
    End;

    // Código para obtener el clima actual para la ciudad (ej. llamando a
una API)
    WeatherInfo := GetWeatherFromAPI(City);

    // Asignar el resultado (si es necesario)
    ToolCall.Result := WeatherInfo;

    Handled := True;  // Indicar que la función se ejecutó correctamente
  except
    on E: Exception do
    begin
      ShowMessage('Error en GetCurrentWeatherAction: ' + E.Message);
      Handled := False;
    end;
  end;
end;
```

4. **Ejecutar una Función:**

   Cuando el modelo de IA decide utilizar una función, enviará una solicitud a tu aplicación Delphi, especificando el nombre de la función y los valores de los parámetros. Tu aplicación crea un objeto TAiToolsFunction con esta información y llama al método DoCallFunction del componente TAiFunctions para que la aplicación la ejecute y retorne el valor solicitado.

Métodos Principales del Componente TAiFunctions

- Constructor Create(AOwner: TComponent):
  - Crea una instancia del componente TAiFunctions.
- Destructor Destroy:
  - Destruye la instancia del componente.

- Function SetFunctionEnable(FunctionName : String; Enabled : Boolean) : Boolean:
  - Habilita o deshabilita una función por su nombre.
  - Parámetros:
    - **FunctionName**: El nombre de la función a habilitar o deshabilitar.
    - **Enabled**: True para habilitar la función, False para deshabilitarla.

- Valor de Retorno: True si la función se encontró y se pudo actualizar su estado, False si no se encontró la función.