Rozdział 1. Narzędzie systemowe dla AI

Narzędzie systemowe dla AI to komponent, który może zostać wykorzystany przez mechanizmy systemowe lub wtyczkowe związane z LLM do wykonania operacji na systemie PlusWorkflow.

Rozdział 2. Definiowanie narzędzia systemowego dla AI

Narzędzie systemowe dla AI tworzone jest w oparciu o definicję stworzoną przez programistę. Definicja taka musi zawierać następujące elementy:

1. Adnotację @AiToolComponent (jeżeli zadanie zaplanowane nie jest definiowane we wtyczce, musi ono pochodzić z pakietu `com.suncode`)
2. Implementować interfejs AiTool

Definicja może zawierać również adnotację @AiToolComponentScript. Adnotacja ta przekazuje ścieżkę do skryptu (z classpath) zawierającego definicję implementacji wywołania tool'a po stronie formularza zadania.

Przykładowa definicja przedstawiona jest poniżej:

```
@Slf4j
@AiToolComponent
public class ExampleAiTool
    implements AiTool
{

    private static final ObjectMapper OBJECT_MAPPER = new
ObjectMapper();

    private static final String INPUT_JSON_SCHEMA = """
        {
          "type": "object",
          "properties": {
            "exampleParam": {
              "type": "string",
              "description": "Example parameter"
            }
          },
          "required": ["exampleParam"]
        }
        """;


    @Override
    public void define( AiToolBuilder builder )
    {
        builder
            .id( "example-tool" )
            .name( "example-tool-name" )
            .userDescription(
"example-tool-description-shown-to-user" )
            .aiDescription(
```

```
"example-tool-description-shown-to-ai" )
            .inputJsonSchema( INPUT_JSON_SCHEMA )
            .scope( AiToolScope.SERVER );
    }

    @Override
    public String execute( String input )
    {
        try
        {
   String exampleParam = extractExampleParam( input );
            if ( StringUtils.isBlank( exampleParam ) )
            {
                return "Parameter exampleParam is required.";
            }
            return OBJECT_MAPPER.writeValueAsString(
exampleParam + " answered" );
        }
        catch ( Exception e )
        {
            return "Invalid input data";
        }
    }

    private String extractExampleParam( String input )
    {
        if ( StringUtils.isBlank( input ) )
        {
            return null;
        }
        try
        {
            JsonNode node = OBJECT_MAPPER.readTree( input );
            if ( node.isTextual() )
            {
                return node.asText();
            }
            return node.path( "exampleParam" ).asText( null );
        }
        catch ( Exception e )
        {
            throw new RuntimeException( "Could not extract
parameter", e );
        }
    }
```

```
}
```

Parametry:

- userDescription (wymagany)
  - opis narzędzia wyświetlany użytkownikowi
- aiDescription (wymagany)
  - opis narzędzia przekazywany do modelów LLM
- scope (wymagany)
  - enum określający możliwe strony wywołania narzędzia
  - możliwe wartości
    - SERVER - tylko po stronie serwera
    - CLIENT - tylko po stronie przeglądarki
    - ALL - możliwość wykonania w obu miejscach
- inputJsonSchema (opcjonalny)
  - string ze schematem JSON dla modelów LLM opisujący parametry wejściowe narzędzia przekazane do parametru String input

Rozdział 3. Implementacja metody execute

Metoda execute przyjmuje parametry wejściowe jako String od modelu LLM, który ma strukturę JSON o kształcie takim, jaki został opisany w parametrze narzędzia aiDescription oraz opcjonalnie inputJsonSchema.

Wartością zwracaną metody może być cokolwiek, co akceptuje model LLM - zwykle jest to JSON reprezentujący rezultat wykonania narzędzia.

Rozdział 4. Rejestracja narzędzia systemowego dla AI we wtyczce

Jeżeli narzędzie systemowe dla AI jest definiowane we wtyczce to należy dodatkowo zaznaczyć, że wtyczka ta udostępnia komponenty. W tym celu należy w pliku suncode-plugin.xml dodać wpis:

```
<!-- Udostpnianie komponentów (m.in. narzdzi systemowych dla
AI) -->
<workflow-components key="components" />
```

Rozdział 5. Wykonanie po stronie formularza zadania

Narzędzia systemowe dla AI mogą zostać wykonywane po stronie przeglądarki na formularzu zadania.

Rejestracja narzędzia po stronie formularza zadania

Wykonanie po stronie przeglądarki rejestrowane jest poprzez przekazanie ścieżki do skryptu js (z classpath wtyczki) zawierającego definicję obiektu z funkcją w adnotacji @AiToolComponentScript dodanej na poziomie klasy komponentu narzędzia systemowego dla AI.

Przykładowa rejestracja skryptu na klasie po stronie Javy:

Przekazanie ścieżki do skryptu w adnotacji @ScheduledTaskScript

```
@AiToolComponent
@AiToolComponentScript( "js/example-tool-definition.js" )
public class ExampleScheduledTask {
 @Override
 public void define( AiToolBuilder builder ) {
        builder
            .id( "example-tool" )  // identyfikator narzdzia
            ...  // pozostae parametry
    }

    ...
}
```

Przykładowa zawartość skryptu rejestrującego obiekt z funkcją po stronie Javascript:

Skrypt przekazany w adnotacji @ScheduledTaskScript

```
// kluczem w metodzie register jest identyfikator narzdzia
systemowego dla AI
PW.AiTools.register('example-tool', {
    execute: function (input) {
        var inputObj = JSON.parse(input);
        var result = inputObj.exampleParam + " answered";
        return JSON.stringify(result);
    }
})
```