# 1 VAS-AI

Il servizio si occupa della gestione dell’interazione utente-sistema combinando l’interazione statica, rappresentata dalla struttura intrinseca del template, con quella dinamica, rappresentata da un motore di ricerca basato su contesti.

Al suo interno sono presenti metodi che consentono di popolare tutte le risorse necessarie a garantire un’interazione completa.

La modalità con cui si verifica l’interazione varia al variare del templateType.

## 1.1 DatasetController

Il datasetController contiene tutti i metodi necessario alla creazione-eliminazione dei documenti interni alla collection mongo “Dataset”. Ogni documento della collection contiene un elemento del dataset, inserito dal data scientist in fase di creazione del template. Tali documenti vengono utilizzati nello scenario di interazione “generale” per la corretta identificazione del template di interesse.

### 1.1.1 POST/dataset

Il metodo consente di popolare la collection “Dataset” sulla base del dataset allegato al template in fase di creazione. A partire dal base64 del file, ricostruisce il documento e lo scorpora secondo dinamiche definite internamente. Ogni elemento così identificato viene allocato all’interno di un documento della collection.

L’output indica il numero di documenti inseriti all’interno della collection.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Name** | **URI** | **Input** | **Output** |
| POST | /dataset/{templateId} | http://localhost:8043/vas-ai/nlp/dataset/{templateId} | Parameters:   * templateId (String) | String |

### 1.1.2 DELETE/dataset

Il metodo, che riceve in ingresso l’id del template d’interesse, consente di eliminare i documenti della collection “Dataset” relativi ad un determinato template.

L’output indica il numero di documenti rimossi dalla collection.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Name** | **URI** | **Input** | **Output** |
| DELETE | /dataset/{templateId} | http://localhost:8043/vas-ai/nlp/dataset/{templateId} | Parameters:   * templateId (String) | String |

## 1.2 NLP controller

### 1.2.1 POST findNextQuestionExp

Il metodo consente di identificare la domanda successiva da porre all’utente durante l’interazione con uno specifico template. Il metodo riceve in ingresso l’id del template di riferimento, la frase digitata o pronunciata dall’utente e (opzionale) l’id della domanda che il sistema presenta all’utente.

Qualora il parametro questionId non venga specificato, il metodo identifica la domanda-nodo iniziale del template, essendo essa l’unica a non avere link entranti ma solamente uscenti.

L’output presenta una struttura unica , indipendentemente dal tipo di template a cui fa riferimento, che varia nella valorizzazione dei suoi campi.

Nello specifico, la risposta presenta i seguenti campi:

* nextQuestionId : l’id della domanda successiva da presentare all’utente
* foglia : indica se il nodo successivo da analizzare è un nodo foglia o meno
* ambiguity : consente di identificare i casi di ambiguità, adattando di conseguenza la risposta del sistema
* fieldFlag : nel caso di template di tipo anagrafica, consente di identificare il campo a cui fa riferimento l’informazione inserita dall’utente

l’identificazione della domanda successiva viene effettuata tramite tre layers di confronto frase-keyword del template, rispettivamente identità, inclusione, identità di token.

Il metodo implementa un sistema di disambiguazione che consente di gestire casi di ambiguità in cui più keyword, relative a differenti link, vengano identificate come corrispondenti alla frase inserita dall’utente.

Qualora la ricerca statica, basata sul flusso descritto dalla struttura del template, non dovesse condurre ad un esito positivo. Il metodo sfrutta un motore di ricerca sviluppato nel modulo IR e basato sul contesto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Name** | **URI** | **Input** | **Output** |
| DELETE | /findNextQuestionExp?sentence= &templateId=&questionId= | http://localhost:8043/vas-ai/nlp/findNextQuestionExp?sentence= &templateId=&questionId | Parameters:   * sentence (String) * templateId (String) * questionId (String, optional) | VasResponse |

### 1.2.2 POST findTemplate

il metodo consente di identificare il template in grado di soddisfare le richieste dell’utente sulla base della frase digitata-pronunciata e dei documenti della collection relativi ai template associati al gruppo di cui l’utente che effettua l’interazione è associato.

Il metodo riceve in ingresso la frase digitata dall’utente, un token che consente di autenticare l’utente, il nome del gruppo a cui appartiene l’utente.

Il metodo identifica il template di pertinenza effettuando tre layers di verifica (identità, inclusione, identità di token) tra la frase digitata e i documenti della collection dataset relativi a tutti i template che sono associati al gruppo di appartenenza dell’utente.

Il metodo implementa un sistema di gestione dell’ambiguità.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Name** | **URI** | **Input** | **Output** |
| POST | findTemplate?groupName=&sentence= | <http://localhost:8043/v->  ai/nlp/findTemplate  ?groupName=  sentence= | Parameters:   * sentence (String) * accessToken (String) * groupName (String) | VasResponse |

## 1.3 Strutture dati vas-AI

Di seguito vengono riportate le strutture dati utilizzate durante la gestione dell’interazione con il VVAS.

### 1.3.1 NLPController

VasResponse

{

Foglia : boolean

nextQuestionId : String

fieldFlag : String

ambiguity : String

}

### 1.3.2 DatasetController

Dataset

{

\_id : String

templateId : String

sentence : String

}