

Field descriptions for DeltaV

The importance of describing your fields in a service

At registration of an Agent Function on the [Agentverse 1 \(opens in a new tab\)](#), providing comprehensive information, especially in the description section, is paramount. A detailed Function description not only elucidates the functionality of the Agent Function but also aids users in grasping its essence. This clarity fosters effective interactions, empowering users to navigate the system with confidence, knowing precisely what to anticipate.

Similarly, the field description of the Agent Function needs to be detailed enough so to provide a clear picture of each dataModel the Function requires for it to be executable for Objective execution. Hence, these descriptions serve as a guiding beacon for users, developers, and LLM alike. Each field explanation elucidates the purpose, usage, and expected inputs required by eachModel class, streamlining interactions and facilitating execution. Enhanced understanding of field descriptions enables the AI Engine to interpret user requests accurately, thereby ensuring accurate Agent Functions execution.

For clarity, reinforcing key concepts could be done through repetition of keywords within the description. Such an approach minimizes ambiguity and mitigates the risk of misunderstandings. Whether for users or LLM, a clear and descriptive field description lays the foundation for smooth interactions and reliable Agent Function delivery and execution.

Consider the following Agent Function:

Providing a detailed field description for the `num_rolls` dataModel is important to correctly execute the Function. A well written field description enhances the LLM understanding of the type of Function the user objective requires to be executed. This would help in the accurate interpretation and execution of users' requests.

It is possible to call additional Secondary Functions from Primary or Secondary ones by specifying this within the field description itself. For Primary and Secondary Functions network, please refer to the below examples.

Example 1: News Reading System

Primary Function Field Description

This Function helps user to read news of a specific type. This Function calls a Secondary function to generate news.

i Remember to always provide a comprehensive description for triggering the Secondary Function in the field description so to ensure that the Secondary Function is always initiated.

The field description for the `news` data model describes the news that will be presented to the user. It should be mentioned that it should be always provided by triggering of Secondary Function. In our case, a good field description would be: Describes the news which will be generate from the "Generate News" secondary function. Always go for "Generate News" secondary function only never ask this field from user. All the news articles generated are presented as strings.

Secondary Function Field Description

This Function helps task to generate news and send it to task or another Secondary Function. Below, there is an example where the users ask for news category they want to read and provide news to Primary Function. For a better understanding, you can find the overall guide for this example [here 1](#).

The field description for the `category` data model describes the category for which the user wants to read the news. It should be mentioned that it should be always provided by user. In our case, a good field description would be: Describes the category provided by user about which he want to get news for. This should always be provided by user in all cases. This primary function responds to "Generate News" secondary function. This should be from options business, entertainment, general, health, science, sports, technology.

i Remember to always include the secondary function trigger in the field description. This is very important to ensure that the secondary function is being called. When selecting one of multiple secondary functions, you need to use different names for each one of the,. This way, we avoid confusion for the LLM.

Example 2: Hugging Face Text Classification Models

This Function helps user to choose a text classification model and make a query to that model. This task hugging face system goes for Secondary Functions to get make request to hugging face API which in turn trigger model list Secondary Functions to get list of model related to a search keyword. User can make query to their selected model.

Primary Function Field Description

The field description for the response describes the response to be provided to the user. The Primary Function always triggers secondary function Hugging Face Request. The ideal field description in this case will be like Describes the response to the user query. Always go for Hugging face request secondary function to get this field. Never ask this from user.

Secondary Function Field Description

Hugging Face Request

Hugging face request : This function has two fields `model_id` (to which query is to be made) and `query` (what request you want to make to the model). The `model_id` field again triggers Model List secondary function which asks user for search keyword to get most downloaded models related to that keyword.

Ideal field description for this example will be like:

`model_id`: Always go for model list secondary function. Never ask this field to user. `query`: Describes the query user wants to ask to the model. Always ask this to user after `model_id` is given by model list secondary function.

i Remember to always provide a comprehensive description in `query` field that first we should fetch `model_id` using secondary function before asking `query` to user.

Model List

Model List : This secondary function has field `Search` which searches for top 10 downloaded models related to that keyword from hugging face API and lets user select from provided model id's.

Ideal field description for this example will be like:

`Search`: This is the type of model user wants to make query to. Always ask this to user. This always gives list of options to the user. Make user select one from these options.

Example 3: Local Business Finder

This Function helps user to find local businesses in any locality and specified category. This Agent Function returns list of 10 businesses to user and gives information about selected business by the user.

Primary Function Field Description

There are 3 fields in this primary function `category` (Type of business which user wants to search), `city` (City in which user wants to look around for business) and `name` (Name of the business which user wants to search).

Ideal field descriptions for this example will be like:

`category`: This describes business category provided for which user wants to search business. Ask this from user. `city`: always go for city finder secondary function to get user's city. Ask user if they want to use this city in yes/no question. ['yes': 'use this city', 'no': 'ask for city they want to search for']. `name`: Always go for secondary function business finder service's response. Never ask this from user.

Secondary Function Field Description

1. City Finder
2. : This secondary function takes user location from DeltaV and looks for their current location. Below given is the ideal description for location.
3. location:
4. This describes the coordinates of the city given in `str(lat,long)` format.
5. Business Finder
6. : This secondary function takes `category` and `city` from the primary function and looks around for 10 business name in that area. Asks user to select one and then sends back to task as `name` field.

`category`: This is the category provided by the user for which they want to get businesses in primary function business details service, Use this category for task as well. `city`: This is the city responded by city finder or one given by user.

Agent Function registration examples

For additional information and guidelines on how to register an Agent Function on the Agentverse and examples of descriptions, head over to our dedicated resources:

- [Agent Functions ↗](#)
- [Register Agentverse Functions ↗](#)
- .
- [Register a coin toss agent as a Function ↗](#)
- .

Was this page helpful?

[Agentverse Functions: coin toss agent Agentverse Command Line Interface \(AVCTL\)](#)