

# Rich Text and Hyperlinks in responses in DeltaV

## Introduction

This guide will help you understand how to utilize rich text and hyperlinks within responses in your [DeltaV](#) application based on a practical example. This is particularly useful for developers who need to generate interactive and informative content.

❗ Check out the [AI Engine package \(opens in a new tab\)](#) to download it and start integrating your Agents with this tool!

Current version of the AI Engine package is 0.6.0

Remember to check out the following [guide](#) to make your agents AI Engine and DeltaV compatible!

## Understanding Rich Text and Hyperlinks

### Rich Text and Hyperlinks

Rich Text is a text format that supports a variety of text styles and formatting options beyond plain text.

These options include:

- Bold, Italic, and Underline styles
- Headings and Subheadings
- Lists (bullet points and numbered)
- Hyperlinks

Rich text makes your content more visually appealing and easier to read, improving the user experience.

Hyperlinks are clickable texts or elements that redirect users to another resource, such as a web page or document. Hyperlinks are often used to connect related content or provide additional information.

In DeltaV, you can include hyperlinks in your responses, allowing users to easily navigate to relevant resources.

## Why Rich Text and Hyperlinks?

1. Enhanced communication
2. : Rich text allows for better formatting and structure, making your messages clearer and more effective.
3. Interactive content
4. : Hyperlinks enable users to access additional information or related content directly from your responses.
5. Improved user experience
6. : These features make content more engaging and accessible, leading to higher user satisfaction.

## Implementing Rich Text and Hyperlinks in DeltaV responses

In DeltaV, you can integrate rich text and hyperlinks to enhance the responses generated by your agents or automation workflows. Let's look at how you can implement these features. Let's consider a scenario where you want your DeltaV agent to respond with a list of news articles related to a company, with each article presented as a clickable hyperlink.

The script below shows how to integrate external data (e.g., stock symbols or news articles) into a DeltaV agent's response, and how to format that response using rich text, specifically by creating hyperlinks. This approach enhances the communication capabilities of the DeltaV agent, making its responses more dynamic, interactive, and useful for the end-user.

Here's the example of how to do this in:

Self hosted example.py import requests from ai\_engine import UAgentResponse , UAgentResponseType

## Define a news content structure using the Model class to represent a news content

```
class
```

```
NewsRequest ( Model ): company_name :
```

```
str
```

# Create a protocol for the Hugging Face finbert agent

## news\_protocol

```
Protocol ( "Institute News" )
```

```
async
```

```
def
```

```
fetch_symbol ( company_name ): url =
```

```
f "https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords= { company_name } &apikey= { ALPHA_VANTAGE_API_KEY } " response = requests . get (url) if response . status_code ==
```

```
200 : data = response . json ()
```

## Typically, the best match will be the first item in the bestMatches list

```
if data . get ( 'bestMatches' )
```

```
and
```

```
len (data[ 'bestMatches' ])
```

```
0 :
```

## Return the symbol of the best match

### symbol

```
data [ 'bestMatches' ] [ 0 ][ '1. symbol' ] return symbol return
```

```
'No Symbol found'
```

```
async
```

```
def
```

```
analyze_news ( company_name ): """Analyze news sentiment for the company using GNews.""" url =
```

```
f "https://gnews.io/api/v4/search?q= { company_name } &token= { GNEWS_API_KEY } &lang=en" response = requests . get (url) articles = response . json (). get ( 'articles' , [])
```

## Extract URLs and convert them to hyperlinks with titles

### button\_list

```
[] for article in articles : title = article . get ( "title" , "No Title" ) url = article . get ( "url" , "No Url" ) if url !=
```

```
"No Url" :
```

## Create a clickable button with the title as text

```
button_list . append ( f '{ title }' )
```

## Combine buttons into a single string with line breaks

# buttons\_combined

```
"\n" . join (button_list)

return buttons_combined

@news_protocol . on_message (model = NewsRequest, replies = UAgentResponse) async
def
on_institute_news_request ( ctx : Context ,
sender :
str ,
msg : NewsRequest): ctx . logger . info (msg.company_name) symbol =
await
fetch_symbol (msg.company_name) ctx . logger . info (symbol) news_content =
await
analyze_news (symbol) ctx . logger . info (news_content) await ctx . send (sender, UAgentResponse (message =
news_content, type = UAgentResponseType.FINAL))
```

agent . include (news\_protocol) <sup>△</sup> Remember that you need to get a `ALPHA_VANTAGE_API_KEY` Key from [GNews](#) ([opens in a new tab](#)) to correctly run this example. The script integrates two key concepts: fetching data from APIs and formatting the response with rich text (specifically, HTML hyperlinks). We first import the needed libraries and then define the `NewsRequest` Data Model. This takes in a `company_name` as input, which will be used to fetch relevant data from GNews. We proceed and define a communication [protocol](#) within the DeltaV agent named `Institute News` . Protocols are used to handle specific types of messages or requests, in this case, news-related requests about a company. We go on and define 3 different functions:

1. `fetch_symbol()`
2. : Fetches the stock symbol of a company by querying the Alpha Vantage API
3. with the company name. The function constructs the API request URL, sends the request, and parses the JSON response to extract the first match from the `bestMatches`
4. list. If no match is found, it returns `No Symbol found`
5. .
6. `analyze_news()`
7. : This function fetches news articles related to the company using the GNews API
8. . The function processes the list of articles, extracts their URLs, and formats each URL as a clickable hyperlink using the HTML
9. tag. The hyperlinks are then combined into a single string, with each link on a new line, and returned.
10. `on_news_request()`
11. : This is the main function that handles incoming requests based on the `NewsRequest`
12. model. The flow is:
  - \* Logs the company name received in the request.
- 13.
14.
  - Calls `fetch_symbol()`
15.
  - to retrieve the company's stock symbol.
16.
  - Calls `analyze_news()`
17.
  - to fetch and format the news articles into clickable links.
18.
  - Sends the final response containing the news content back to the sender, using the `UAgentResponse`
19.
  - class with a `FINAL`
  - message type.

**i** Use HTML tags to format text as rich text. For example, use **for bold**, *for italics*, and for underline. **Embed hyperlinks using the tag.**

**Example:**

[Click here to visit Example.com](#) When users click on [Click here to visit Example](#) , they will be taken

to <https://example.com> .

## **Expected output**

**This guide details how to enhance DeltaV responses using rich text and hyperlinks. Rich text allows for enhanced formatting of responses, while hyperlinks make content interactive by providing clickable links to additional resources. The example code demonstrates how to fetch data from APIs, format it with HTML links, and handle requests within a DeltaV agent. Proper use of these features improves readability, engagement, and the overall effectiveness of your responses.**

**Last updated on October 17, 2024**

**Was this page helpful?**

**You can also leave detailed feedback [on Github](#)**

**[Field descriptions for DeltaV Agentverse: Teams](#)**

### **On This Page**

- [Introduction](#)
- [Understanding Rich Text and Hyperlinks](#)
- [Rich Text and Hyperlinks](#)
- [Why Rich Text and Hyperlinks?](#)
- [Implementing Rich Text and Hyperlinks in DeltaV responses](#)
- [Expected output](#)
- [Edit this page on github\(opens in a new tab\)](#)