

Challenge6 Step By Step

Alex Fletcher edited this page now · 1 revision

Edit

New page

Challenge 6 - Step-by-Step

Introduction

We are going to use Semantic Kernel to create the recommendation engine. Lets get started!

Prerequisites

I created a pull request that you need to accept. Simply accept it and it will give you the starting point of the solution. After accepting the pull request I created, you will find the frontend code is already done. It calls into our catalog API where there is a new controller called `EventRecommendations`. This controller already has a constructor that creates the semantic kernel instance that you can use.

Getting a list of events available next month

To get a list of available events, you can call into the `_eventRepository` and request it to give you all available events with the method `GetEvents()`.

Next, you can filter the list of events by using a simple LINQ query on the returned list. This is done as follows:

```
var availableEvents = events.Where(e => e.Date > now && e.Date < now.AddMonths(1)).ToList();
```

We can use this list as the input to the Kernel, asking it to provide the recommendations.

Getting the Recommendation

For this, we create a prompt that instructs the LLM to generate us an output of a comma-separated list that contains the recommended events.

We start by defining the prompt.

```
const string prompt = "Given the artist ({input_artist}), recommend similar artists from the following events: " +
    "ONLY return the event ids of the recommended events as a comma-separated list, nothing else. ";
```

You see we have two inputs here: the artist the customer likes and the list of events we just got back from our own service.

Next, we need to define these arguments in the way Semantic Kernel can handle them. This is done as follows:

```
var arguments = new KernelArguments
{
    ("input_artist", artist),
    ("event_list", string.Join(Environment.NewLine, availableEvents.Select(e => JsonSerializer.Serialize(e))))
};
```

Now we have everything ready to call the Kernel and use the LLM to get a recommendation.

This is done as follows:

```
var result = await _kernel.InvokePromptAsync(prompt, arguments);
```

And now we need to parse the results back to a list of events that we previously retrieved from the datastore. This is done as follows:

```
var chosenEvents = result.ToString()
    .Split(",")
    .Select(eventId => eventId.Trim())
    .Select(Guid.Parse);
```

And finally, we return the results to the caller.

```
availableEvents.Where(e => chosenEvents.Contains(e.EventId)).ToList();
```

Summary

You selected the events used in the answer from the database. You created a prompt to use with an LLM to get a recommendation. You asked it explicitly to get you a comma-separated list back. You parsed that result and returned the list of events to the caller.

And done, you have now created the logic behind a pretty sophisticated recommendation engine, and it now helps your customers buy concert tickets that are similar to the artist the customer loves.

+ Add a custom footer

Pages 11

Find a page...

Home

Challenge1 Braindump

Challenge2 Braindump

Challenge3 Braindump

Challenge3 Step By Step

Challenge4 Braindump

Challenge4 Step By Step

Challenge5 Braindump

Challenge6 Braindump

Challenge6 Step By Step

Challenge6 Step-by-Step

Introduction

Prerequisites

Getting a list of events available next month

Getting the Recommendation

Summary

Enable CopilotVoucher

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/globaldevopsexper>