

```
store_policy_kb:
  type: kb agent
  description: I can answer questions related to the
store's policy.
  sources:
    - https://www.bookstores.com/policies/shipping
    - https://www.bookstores.com/about
  faq:
    - q: Thanks, bye
      a: Looking forward to serving you next time.
```

```
book_recommendation:
  type: llm agent
  description: I can recommend books to customers.
  args:
    - genre
    - book_name
    - book_info
    - book_wanted
  prompt: |
    1. Ask the user if they have a preferred book
genre.
    2. If the user has a favorite book, call the
"query_book_genre" function based on their favorite
book to obtain the "genre".
    3. Using the genre, call the "find_best sellers"
function to recommend relevant books to the user.
Then ask user if they need to order this one.
    4. If the user agree, append this book to
argument "book_wanted", which should be an array,
and then complete this agent.
  uses:
    - query_book_genre
    - find_best_sellers
```

```
order:
  type: flow agent
  description: I can place an order.
  args:
    - books
    - order_status
  fallback: Sorry, I didn't understand that. Could
you rephrase it?
  steps:
    - bot: "I'll place the order for you."
    - label: confirm_books
    - bot: "You have selected these books so far:
${books}. Would you like to add anything else to
your order?"
    - user
    - if: the user claims "Yeah", "Do you have other
types of books?"
      then:
        - call: book_recommendation
        - next: confirm_books
    - else if: the user claims "No", "I don't have
anything else I want to buy."
      then:
        - next: start_ordering_operation
      else:
        - next: confirm_books

start_ordering_operation:
  - call: place_order
  args:
    ordered_book: books
    date: triage.date
  - if: place_order.status == True
    then:
      - return: success, Order placed
successfully.
    else:
      - return: error, Order failed.
```

```
guardrail:
  type: llm agent
  description: I can check if the user's input is
relevant to the bookstore.
  args: ["is_relevant"]
  prompt: |
    Determine whether the user's message is highly
unrelated to a typical bookstore customer. It is
acceptable for the customer to send messages such as
"Hi," "OK," or other conversational responses.
However, if the message is non-conversational, it
must still be at least somewhat related to books.
Return is_relevant = True if it is, otherwise return
False.
```

```
triage:
  type: ensemble agent
  description: Select an agent to respond to users.
  args: ["book"]
  contains:
    - guardrail
    - store_policy_kb
    - book_recommendation:
      args:
        book_wanted: ref book
    - order:
      args:
        books: ref book
  steps:
    - bot: "Hi, I'm your bookstore assistant. How
can I help you?"
    policy: Call the guardrail first each time, and
decide whether to proceed with the conversation
based on its output.
    exit: default
```

```
main:
  type: flow agent
  steps:
    - call: triage
```