

Payments LLM Prompt/Description of Tools

Journey Name Prediction:

journey_name: The journey name can be one of the following, payment_received, payment_cancel, payment_ambiguous, outofscope.

Always extract from the given choice, the journey name should not be any other from payment_received, payment_cancel, payment_ambiguous, outofscope

Use the journey descriptions from below to better understand which journey_name should be extracted.

Journey Descriptions:

1. payment_received: The user wants to know the details or want to check the status or confirm a receipt of a payment they had done on a card, this is different from transaction, payment only refers to the payment they have done to pay off their card. User might also want to check their payment history. The payment received journey only deals with payments that are currently processing or in the past. Check if the payment is posted, processing, applied to their card.
2. payment_cancel: The user wants to cancel a payment, or user want to make changes, edit or delete a payment they have done earlier on their card. Only deals with payments that are processing, scheduled in the future, cancelled or paid. The user wants to confirm that payment is cancelled or confirm that a payment was cancelled.
3. payment_bank: The user wants to know how to add, link, edit, manage, change bank accounts associated with their american express account to pay off their card.
4. payment_address: The user wants to know where and how they can mail a payment to american express to pay off their card.
5. payment_statement_balance: The user wants to know why their balances are calculated to a certain amount.
6. make_payment: The user wants to make a payment to pay off their card.
7. payment_due_date: The user has questions or asking about when is their payment due date.
8. change_payment_due_date: The user wants to change their payment due date.
9. payment_refund: The user wants a refund because they have overpaid their card or they have a credit balance that they want refunded.
10. payment_transfer: The user wants to transfer their credit balance or overpaid balance from one card to another.
11. payment_ambiguous: The user has asked a generic payment which does not refer to payment_received or payment_cancel then the journey_name should be extracted as payment_ambiguous.
12. outofscope: The user has asked a query which is not related to payment then the journey_name should be extracted as outofscope. The example queries include cancel a card, disputing a transaction, the user can also respond with something else or getting frustrated with the bot.
13. payment_methods: The user is seeking information about the valid and potential options or ways to pay off their card. (For example: Payments using credit cards, debit cards, gift cards, cash, wire transfers, money orders, checks)
14. autopay: The user wants to setup autopay, having issues with their autopay setup or may want to change their autopay details.
15. troubleshoot: The user is indicating that they are having trouble with a payment which could be a technical issue or other reasons.
16. hardship: The user is experiencing financial hardship and cannot pay towards their credit card balances. (For Example: Bankruptcy. Loss of Job, natural disasters).

Strictly adhere to the provided journey names listed above.

Focus on the current message and take it one step at a time.

Use the previous conversation history to decide the journey_name. Always extract the closest journey_name this field should never be None.

Card Type Prediction:

card_type: Type of credit card to extract from the user message (Example: Blue, Gold, Platinum, Blue Cash, Cash Magnet). Extract the card_type only from the current user message. The user can make mistakes or be ambiguous in mentioning the card like Bleu, Plat, or just mention Magnet card, cash card, do a fuzzy match and extract the best and closest possible card_type match from the following:

- Blue

- Gold
- Platinum
- Blue Cash
- Cash Magnet

Strictly adhere to the provided card types listed before. If the user didn't mention a card_type fill this slot with None. If the user has mentioned multiple card types extract a list of strings extracting multiple card types from the user message.

Card Number Prediction:

card_number: Card number to be extracted from the user message, the usual card number is consisting of five digits but user can also mention four or three digits to their card number. If the user is ambiguous about their card_type and card number and If the users card_list: {card_list} is available then try to fuzzy match the card number from the card_list and extract the card number from the card_list.

If the user has mentioned multiple card numbers extract a list of strings extracting multiple card numbers from the user message. If the user didn't mention a card_number fill this slot with None.

OpenAI Tool Input and Output:

tool_definition:

```
tool_n = {
  "type": 'function',
  "function": {
    "name": "Name of the tool",
    "description": " Description of the tool",
    "parameters": {
      "type": "object",
      "properties": {
        "arg1": {
          "type": 'object',
          "description": "Description of the argument",
          "items": {
            "type": 'Data type of the argument that is being extracted'
          }
        }
      }
    },
    "required": "List of required arguments"
  }
}

##Example

payment_amount_tool = {
  "type": 'function',
  "function": {
    "name": "extract_payment_details",
    "description": ""Extract payment amount from the current user message
                    payment_amount: The amount of payment to be extracted from the user message"",
    "parameters": {
      "type": "object",
      "properties": {
        "payment_amounts": {
          "type": 'array',
          "description": "payment amounts to extract from the user message",
          "items": {
            "type": 'string'
          }
        }
      }
    },
    "required": ["payment_amounts"]
  }
}
```

OpenAI payload:

```
payload = {
    "messages": prompt, #List of messages with the role and prompts
    "tools": tools, #List of tools
    "temperature": 0
}
```

OpenAI Example output:

```
{'choices': [{'content_filter_results': {},
               'finish_reason': 'tool_calls',
               'index': 0,
               'logprobs': None,
               'message': {'content': None,
                           'role': 'assistant',
                           'tool_calls': [{'function': {'arguments': '{\n'
                                                                ',\n'
                                                                '\n'
                                                                '    "journey_name": '\n'
                                                                '"payment_cancel"\n'
                                                                '}',
                                                'name': 'payment_cancel'}},
                                           {'id': 'call_5KzXIgqIR6GOQgKUQvxqUQbM',
                                            'type': 'function'}}]}],
 'created': 1709226249,
 'id': 'chatcmpl-8xe7Ffwxa9Zke0eBixD2mjLYHI4XM',
 'model': 'gpt-4',
 'object': 'chat.completion',
 'prompt_filter_results': [{'content_filter_results': {'hate': {'filtered': False,
                                                                'severity': 'safe'},
                                                         'self_harm': {'filtered': False,
                                                                      'severity': 'safe'},
                                                         'sexual': {'filtered': False,
                                                                    'severity': 'safe'},
                                                         'violence': {'filtered': False,
                                                                    'severity': 'safe'}}},
                           'prompt_index': 0}],
 'usage': {'completion_tokens': 18, 'prompt_tokens': 428, 'total_tokens': 446}}

# We are going to use the tool_calls field and extract the tool_name and arguments
```