

# Polaris Message History Storage - Server Design

<b>wDate</b>	Nov 17, 2025
<b>Created By</b>	@Peer Fichman
<b>Epic</b>	<a href="#">PER-2121: Polaris - move chat history from local storage to our infra</a> <span>DONE</span>
<b>Ticket</b>	<a href="#">PER-2247: Polaris chat history - server design</a> <span>DONE</span>

## TOC

- [Overview](#)
- [Goals](#)
- [Flows](#)
- [Data](#)
  - MySQL
  - OpenSearch
- [API](#)
  - API Glossary
  - Send Chat Message
  - Get Chat Messages By Chat ID (User – Own Chats)
  - Upload Chats (Migration)

## Overview

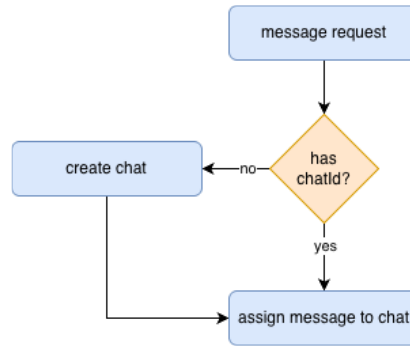
This document describes the process of migrating chat history storage from **client-side localStorage** to a **persistent external database** using **Snowflake**, exposing functionality around this data using the **brand-intelligence-console-server**.

## Goals

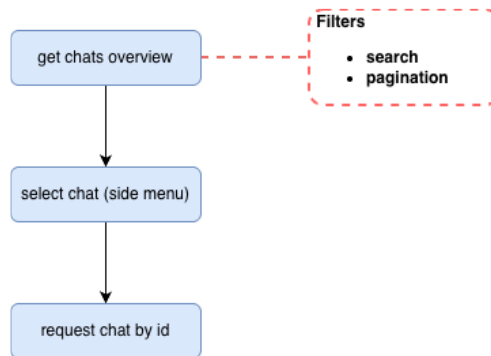
- Fully **match today's UX behavior** (click a chat → see the full conversation).
- **Safely migrates** existing localStorage chats so no data is lost.
- Enables future **admin oversight, chat forking/branching**.

## Flows

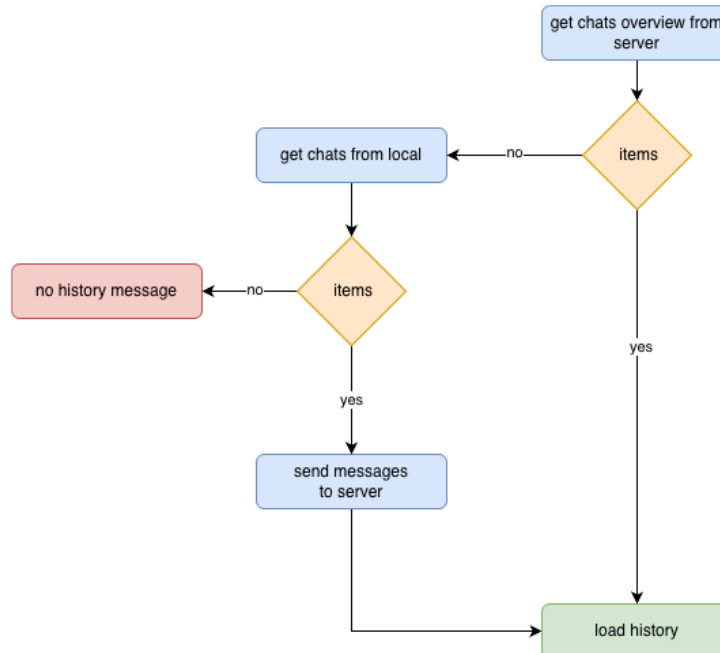
### Send chat message



### Get Chat



### Migration



## Data



- Chat history, and chat message must be associated with user.
- In order to support features as sharing chats across the account space and expose users with certain roles to other user's messages (within the same account), account and role must be registered.
- users and accounts have a many to many relation.

⚠ user given name will be added to custom claims in auth0

## MySQL

⚠ Required data currently stored in Auth0/MySQL.

- user id
- user role
- account key

### chat

Field Name	Data Type	Relation / Notes
chat_id	varchar (uuid)	<b>PK</b>
account_key	varchar	
user_id	varchar	
title	varchar	nullable
created_at	timestamp	server timestamp
last_modified	timestamp	client timestamp
parent_chat_id	varchar (uuid)	<b>FK → chat.chat_id</b> (nullable)

∨ ⓘ Invites (Phase one - optional)

#### collaboration\_invites

Field Name	Data Type	Relation / Notes
------------	-----------	------------------

user_owner_id	varchar	<b>PK</b>
user_invited_id	varchar	<b>PK</b>
chat_id	varchar(uuid)	<b>FK → chats.chat_id</b>
invite_message	varchar	
status	varchar	“pending“   “accepted“   “declined“   “cenceld“   “expired”

## OpenSearch

### Cluster Settings

<b>Shards per index</b>	3
<b>Replicas</b>	1 (1 replica = 2x storage)
<b>Refresh Interval</b>	1s
<b>Routing</b>	chatId (for clustering messages per chat)

message

```

1  {
2    "id": uuid,
3    "chatId": uuid,           // routing key
4    "brandKey": string | null,
5    "competitorsKeys": string[] | null,
6    "status": 'success' | 'pending' | 'failed' | null,
7    "polarisTimestamp": number | null,
8    "userPrompt": string,
9    "originalUserPrompt": string,
10   "response": string,
11   "userTimestamp": number,
12   "systemTimestamp": number | null,
13   "creativesURLs": string[],
14   "bbiRequestId": string,
15   "polarisLambdaRequestId": string | null,
16   "followUpQuestions": string[],
17   "insights": string[] | null,
18   "reasoning": string | null,
19   "grounding": {
20     "sqlQuery": string,
21     "dataSources": string[]
22   } | null,
23   "userPreferences": {
24     "userName": string,

```

```

25     "preferredBrands": string[],
26     "commonTopics": string[],
27     "preferredMetrics": string[],
28     "timePreferences": string,
29     "communicationStyle": string$
30 } | null,
31 "originalResponse": string,      //original response stringified
32 "graphs": [
33     {
34         "title": string,
35         "type": "bar_chart" | "line_chart" | "table" | "pie_chart",
36         "url": string,
37         "x": string[],
38         "series": {
39             "name": string,
40             "unit": string,
41             "y": number[]
42         }[],
43         "xAxis": string,
44         "yAxis": string,
45         "yIsPercentage": boolean
46     }
47 ] | null,
48 "widgetContext":{
49     "origin": string,
50     "name": string,
51     "metrics": string[],
52     "myBrand": string,
53     "myCompetitors": string[],
54     "dateRange": {
55         "from": string, // format: YYYY-MM-DD
56         "to": string, // format: YYYY-MM-DD
57     },
58     "countries": string[],
59     "device": string
60 } | null
61 }
62

```

## API

### API Glossary

#### Chat-Overview

```

1  {
2  | "chatId": uuid,
3  | "accountKey": string,
4  | "userId": string,
5  | "title": string,
6  | "parentChatId": uuid | null,
7  | "createdAt": "YYYY-MM-DDTHH:mm:ssZ"
8  }

```

#### Message

```

1  {
2  | "author": "user" | "assistant",
3  | "content": string
4  }

```

#### Wizard

```

1 {
2   "type": string,
3   "geos": string[],
4   "startDate": 'YYYY-MM-DD',
5   "startDate": 'YYYY-MM-DD',
6   "focusAreas": {
7     [key: string]: {
8       "subQuestions": string[]
9     }
10  }
11 }

```

## Graph

```

1 {
2   "title": string,
3   "type": 'bar_chart' | 'line_chart' | 'table' | 'pie_chart',
4   "url": string | null,
5   "x": string[],
6   "series": Series[],
7   "y_is_percentage": boolean,
8   "x_axis": string,
9   "y_axis": string
10 }

```

## Series

```

1 {
2   "name": string,
3   "y": number[],
4   "unit": string | null
5 }

```

## Widget-Context

```

1 {
2   "origin": string,
3   "name": string,
4   "metrics": string[],
5   "myBrand": string,
6   "myCompetitors": string[],
7   "dateRange": {
8     "from": string, // format: YYYY-MM-DD
9     "to": string, // format: YYYY-MM-DD,
10  }

```

## Message-Request

```

1 {
2   "brandKey"?: string,
3   "competitorsKeys"?: string,
4   "userPrompt": string,
5   "timestamp": 1700000000000,
6   "conversationHistory": Message[],
7   "debug": boolean, // defaults to false,
8   "wizard"?: Wizard,
9   "alert"?: Alert
10 }

```

## Message-Response

```

1 {
2   "id": uuid,

```

```

3   "status": string,
4   "userPrompt": string,
5   "response": string,
6   "timestamp": 1700000000000,
7   "polarisLambdaRequestId": string | null,
8   "insights": string[] | null,
9   "creativesURLs": string[],
10  "followUpQuestions": string[],
11  "graphs": Graph[],
12  "originalResponse": string // if debug = true
13 }

```

## Chat-Messages-Response

```

1  {
2    "chat": Chat-Overview,
3    "messages": Message-Response[],
4    "pagination": {
5      "limit": number,
6      "offset": number,
7      "total": number
8    }
9  }

```

## Message-Upload

```

1  {
2    "id": uuid,
3    "brandKey"?: string,
4    "competitorsKeys"?: string[],
5    "userPrompt": string,
6    "response": string,
7    "userTimestamp": number,
8    "author": 'user' | 'system',
9    "widgetContext"?: Widget-Context,
10   "insights": string[],
11   "graphs": Graph[],
12   "creativesURLs": string[],
13   "followUpQuestions": string[]
14 }
15 }

```

## Chat-Upload

```

1  {
2    "chatId": uuid,
3    "title": string,
4    "lastUpdated": timestamp,
5    "messages": Message-Upload[]
6  }

```

## Forbidden-Response

```

1  {
2    "statusCode": 403,
3    "error": "Forbidden"
4  }

```

## Send Chat Message

Send a chat message

### Permissions:

```
send:copilot-message
```

### Endpoint:

```
POST /polaris/v2/chat/message
```

### Body

```
1 | {  
2 |   "chatId": uuid,  
3 |   "createNewChat": boolean, // defaults to false  
4 |   "accountKey": string,  
5 |   "message": Message-Request  
6 | }
```

### Response (200):

```
1 | {  
2 |   "chat": Chat-Overview,  
3 |   "message": Message-Response  
4 | }
```

### Response (403):

- Account not assigned to user.

```
Forbidden-Response
```

### Response (409):

- Chat already exists with given id (createNewChat is true)

```
1 | {  
2 |   "message": "Chat already exists"  
3 | }
```

## Get Chats Overview

### Permissions:

```
read:copilot-chat *
```

### Endpoint:

```
GET /polaris/v2/accounts/:account_key/chats?limit=&offset=
```

### Path Params:

- account\_key

### Query Params:

- `limit` defaults to 20
- `offset` defaults to 0

### Response (200):

```
1 | {  
2 |   "items": Chat-Overview[],  
3 |   "total": number,  
4 |   "limit": number,  
5 |   "offset": number  
6 | }
```

### Response (403):

- Account key not assigned to user.

Forbidden-Response

### Get Chat Messages By Chat ID (User – Own Chats)

Retrieve chat by id for the authenticated user within an account. Supports searching by message content.

### Permissions:

`read:copilot-chat *`

### Endpoint:

`GET /polaris/v2/accounts/:account_key/chats/:id?all=&limit=&offset=`

### Path Params:

- `id` chat id
- `account_key`

### Query Params

- `all` : boolean - queries all chat messages
- `limit` : number - defaults to 20
- `offset` : number - defaults to 0

### Response (200):

Chat-Messages-Response

### Response (403):

- Account or chat id not assigned to user.

## Forbidden-Response

### Upload Chats (Migration)

Upload chats from local storage to server

#### Permissions:

```
send:copilot-message
```

#### Endpoint:

```
POST /polaris/v2/accounts/:account_key/chats
```

#### Path Params:

- account\_key

#### Body

```
1 {  
2   "chats": Message-Upload[]  
3 }
```

#### Response (200):

```
Chat-Messages-Response
```

#### Response (403):

- Account or chat id not assigned to user.

## Forbidden-Response

#### Response (500):

```
1 {  
2   "statusCode": 500,  
3   "message": "Chats upload failed - Failed to {create chat | put message in queue}:",  
4   "error.message": "",  
5   "error": "Internal Server Error"  
}
```

▼ ⓘ Admin oversight & Invites (Phase one - optional)

#### Get Chat History (Organization – Admin)

Retrieve chats across an entire account. Requires admin role in the account.

#### Permissions:

read:copilot-chat-admin \*

### Endpoint:

POST /polaris/admin/chats?search=&limit=&offset=

### Query Params:

- search (query, optional): Free-text search across message content.
- limit
- offset

### Body

```
1 {  
2   "users" : string[] // user_ids  
3 }
```

### Response (200):

```
1 {  
2   "items": Chat-Response[],  
3   "total": number,  
4   "page": number,  
5   "limit": number,  
6 }
```

### Fork Chat (Send Invite)

Fork an existing chat and invite another user in the same account to own or collaborate on the fork.

### Permissions:

fork:copilot-chat \*

### Endpoint:

POST /polaris/chats/invites

### Request Body:

```
1 {  
2   "chatId": uuid,  
3   "invitedUsers": string[], // user ids (not empty)  
4   "inviteMessage": string,  
5 }  
6
```

### Response (204):

```
1 {  
2   "message": "Invites sent."  
3 }
```

### Accept Fork Invite

Accept a fork invitation and gain access to the forked chat with its full message history.

#### Endpoint:

```
POST /api/v1/chats/invites/accept
```

#### Response (200):

```
1 | {  
2 |   Chat-Overview  
3 | }
```

### Decline Fork Invite

Decline a fork invitation.

#### Endpoint:

```
POST /api/v1/chats/invites/decline
```

#### Response (204):

```
1 | {  
2 |   "message": "Invitation declined."  
3 | }
```

\* To be added to auth0