



main

Looker\_Agent / README.md



wrenchchat Update README.md

19c96bb · 2 days ago



199 lines (193 loc) · 10.3 KB

Preview

Code

Blame

Raw



# Looker Agent

## Created with Vertex AI Agent Builder, using Data Stores, OpenAPI, powered by Gemini 1.5 Flash

This repository contains the configuration and instructions for setting up the Looker Agent in Vertex AI. This agent assists users with the Looker ecosystem, LookML, and SQL related to BigQuery.

### Table of Contents

#### 1. Repository Setup

#### 2. Required Roles, Permissions, and APIs

- Get a Gemini API key
- Roles and Permissions
- APIs to Enable
- Instructions

#### 3. Vertex AI Agent Configuration

- Syncing the Agent with this Repository
- Agent Handoff Instructions
- Using the Looker and BigQuery Assistants

## 4. Contributing

- Required Files Checklist
- Roadmap
- Contact Information

---

## Repository Setup

---

To clone this repository and set it up locally, run:

```
git clone git@github.com:doit/Looker_Agent.git
```



Ensure you have access to this repo. If not, request access from the repo owner.

## Required Roles, Permissions, and APIs

---

Use or create a billing-enabled project in Google Cloud Platform, add the following roles to your service account, and enable the following APIs.

### Get a Gemini API key

To use the Gemini API, you need an API key. You can create a key with a few clicks in Google AI Studio.

[GET GEMINI API KEY](#)

## Roles and Permissions

### 1. Roles Required:

- BigQuery Admin: Manages all BigQuery resources.
- Logging Admin: Manages Cloud Logging.
- Dialogflow Admin: Full access to Dialogflow agents and configurations.
- Dataform Admin: Manages Dataform repositories and workflows.
- Dataplex Admin: Manages Dataplex data lakes.
- Monitoring Viewer: Read-only access to monitoring resources.
- Project Viewer / Browser: Basic view permissions.

## 2. Special Permission:

- analysisResult.accessControlLists.accesses.permission
- This may require a custom role if not included in predefined roles.

## APIs to Enable

- Analytics Hub API: [analyticshub.googleapis.com](https://analyticshub.googleapis.com)
- BigQuery API: [bigquery.googleapis.com](https://bigquery.googleapis.com)
- BigQuery Connection API: [bigqueryconnection.googleapis.com](https://bigqueryconnection.googleapis.com)
- BigQuery Data Policy API: [bigquerydatapolicy.googleapis.com](https://bigquerydatapolicy.googleapis.com)
- BigQuery Migration API: [bigquerymigration.googleapis.com](https://bigquerymigration.googleapis.com)
- BigQuery Reservation API: [bigqueryreservation.googleapis.com](https://bigqueryreservation.googleapis.com)
- BigQuery Storage API: [bigquerystorage.googleapis.com](https://bigquerystorage.googleapis.com)
- Cloud Logging API: [logging.googleapis.com](https://logging.googleapis.com)
- Cloud Monitoring API: [monitoring.googleapis.com](https://monitoring.googleapis.com)
- Cloud Resource Manager API: [cloudresourcemanager.googleapis.com](https://cloudresourcemanager.googleapis.com)
- Dialogflow API: [dialogflow.googleapis.com](https://dialogflow.googleapis.com)
- Dataform API: [dataform.googleapis.com](https://dataform.googleapis.com)
- Dataplex API: [dataplex.googleapis.com](https://dataplex.googleapis.com)
- Discovery Engine API: [discoveryengine.googleapis.com](https://discoveryengine.googleapis.com)
- Enterprise Search API: [enterprisesearch.googleapis.com](https://enterprisesearch.googleapis.com)
- Contact Center AI Platform API: [contactcenterai.googleapis.com](https://contactcenterai.googleapis.com)
- Contact Center AI Insights API: [contactcenteraiinsights.googleapis.com](https://contactcenteraiinsights.googleapis.com)
- Vertex AI API: [aiplatform.googleapis.com](https://aiplatform.googleapis.com)

## Instructions

### 1. Enabling APIs via Google Cloud Console:

- Go to the Google Cloud Console API Library.
- Search for and enable each required API by clicking Enable.

### 2. Enabling APIs via gcloud command-line:

```
gcloud services enable aiplatform.googleapis.com
gcloud services enable analyticshub.googleapis.com
gcloud services enable bigquery.googleapis.com
gcloud services enable bigqueryconnection.googleapis.com
```



```
gcloud services enable bigquerydatapolicy.googleapis.com
gcloud services enable bigquerymigration.googleapis.com
gcloud services enable bigqueryreservation.googleapis.com
gcloud services enable bigquerystorage.googleapis.com
gcloud services enable contactcenterai.googleapis.com
gcloud services enable contactcenteraiinsights.googleapis.com
gcloud services enable cloudresourcemanager.googleapis.com
gcloud services enable dataform.googleapis.com
gcloud services enable dataplex.googleapis.com
gcloud services enable dialogflow.googleapis.com
gcloud services enable discoveryengine.googleapis.com
gcloud services enable enterprisesearch.googleapis.com
gcloud services enable logging.googleapis.com
gcloud services enable monitoring.googleapis.com
```

### 3. Assigning Roles via Google Cloud Console:

- Go to the IAM & Admin page.
- Select the user or service account, then click Edit Principal.
- Add the required roles from the dropdown menu and click Save.

### 4. Assigning Roles via gcloud:

```
# replace customRoleName with your desired custom role name
# replace YOUR_PROJECT_ID with your Google Cloud project ID
gcloud projects add-iam-policy-binding <PROJECT_ID> \
  --member="user:<USER_EMAIL>" \
  --role="roles/bigquery.admin"
gcloud projects add-iam-policy-binding <PROJECT_ID> \
  --member="user:<USER_EMAIL>" \
  --role="roles/logging.admin"
gcloud projects add-iam-policy-binding <PROJECT_ID> \
  --member="user:<USER_EMAIL>" \
  --role="roles/dialogflow.admin"
# Repeat for other roles as needed.
```



### 5. Create a Custom Role with the Required Permission:

```
gcloud iam roles create customRoleName \
  --project=YOUR_PROJECT_ID \
  --title="Custom Role with Specific Permission" \
  --
permissions="analysisResult.accessControlLists.accesses.permission" \
  --stage="GA"
# add role to user or service account:
gcloud projects add-iam-policy-binding YOUR_PROJECT_ID \
```



```
--member="user:USER_EMAIL" \  
--role="projects/YOUR_PROJECT_ID/roles/customRoleName"
```

## Vertex AI Agent Configuration

---

### 1. Export the Agent (optional):

- If you need to export your existing Vertex AI agent, follow these steps:
  - Navigate to your Vertex AI Console > Agents.
  - Export the agent configuration as a ZIP file for backup or transfer.

### 2. Import the Agent:

- To use this repository with a new Vertex AI agent:
  - Navigate to your Vertex AI Console > Agents.
  - Click Create New Agent or select an existing agent.
  - Under Settings, select GitHub as the repository source and link it to this repo: ([git@github.com:doit/Looker\\_Agent.git](https://github.com:doit/Looker_Agent.git)).

### 3. Configure the Agent:

- Once the repo is linked, configure the agent's behavior:
  - Set up intents, tools, and handoffs using the pre-defined instructions in the codebase.
  - Ensure the Looker Data Store and BigQuery Data Store tools are linked correctly.

## Syncing the Agent with this Repository

### 1. Sync the Repo with Vertex AI:

- Go to the Vertex AI Console > Agent Settings.
- Under the Repository section, ensure the repo `git@github.com:doit/Looker_Agent.git` is synced.
- Click Sync to pull the latest changes from the repository into Vertex AI.

### 2. Handling Updates:

- After syncing the repo with Vertex AI, any changes made to the repo (e.g., LookML instructions, tool configurations) will be reflected in the agent.

### 3. Collaborating:

- Multiple collaborators can work on the agent by committing changes to the GitHub repository. Pull requests and branching strategies are recommended for collaboration.

## Agent Handoff Instructions

This agent utilizes the following tools and handoffs:

### 1. Looker Assistant:

- Handles LookML syntax validation and dashboard troubleshooting using the Looker Data Store.

### 2. BigQuery Assistant:

- Handles SQL troubleshooting and query optimization using the BigQuery Data Store.

### 3. Default Generative Agent:

- Handles general queries and tasks using the OpenAPI and code-interpreter tools.
- Handoffs to either Looker or BigQuery Assistant based on user queries.

### 4. Example Handoff:

- If a query involves LookML, the agent will detect the keyword and trigger a handoff to the Looker Assistant.
- Example: "It sounds like you're asking about LookML. I'll pass this to the Looker Assistant for help."

## Using the Looker and BigQuery Assistants

- Looker Assistant:
  - The Looker Assistant is designed to handle queries related to LookML and dashboards.
  - It uses the Looker Data Store to process requests and validate LookML models.
- BigQuery Assistant:
  - The BigQuery Assistant assists with SQL performance and troubleshooting.
  - It uses the BigQuery Data Store to handle optimization tasks.

# Contributing

---

We welcome contributions to improve the Looker Agent. To contribute:

1. Fork the repository.
2. Create a new branch for your feature or bugfix.
3. Submit a pull request with a detailed description of your changes.

## Required Files Checklist

- Agent Configuration: agent.json ✓
- Flows: Default Start Flow is present ✓
- Generative Settings: The generative settings file is present ✓
- Guides: Blueprint for defining, testing agent and JSON for DialogFlow ✓
- Intents: Default Negative and Welcome Intents are present with training phrases ✓
- Playbooks: Both Default Generative Agent and LookML Syntax Assistant are present ✓
- README.md: Documentation is included ✓

## Roadmap

### 1. Add Support for **Looker Studio Pro**

- Expand Toolset: Introduce a new tool, `${TOOL: Looker_Studio_Pro_Data_Store}`, for handling Looker Studio Pro queries.
- Create Intents: Add Looker Studio Pro-specific intents, such as issues with report generation, dashboard syncing, and visualizations.
- Documentation: Store relevant Looker Studio Pro documentation and Zendesk support tickets in the Looker Studio Pro Data Store.
- API Integration: Integrate the Looker Studio API to provide real-time assistance for report issues.

### 2. Create a Datastore of **Zendesk Support Tickets**

- Zendesk API Access: Set up an ETL pipeline to extract, transform, and load Zendesk support tickets related to Looker, Looker Studio, and BigQuery into a BigQuery datastore.
- Train the AI: Use the datastore of past support tickets to train the agents for automated responses to common queries.

- Searchable Datastore: Allow support staff to search for tickets by **Issue, Resolution, or CSAT**.

### 3. Integrate with DoiT's AI Chat, **Ava**

- Handoff Mechanism: Implement a system where the Vertex AI agents can escalate complex issues to Ava if they can't resolve the queries.
- Cross-Agent Collaboration: Ensure seamless handoffs between the Vertex AI agent and Ava for deeper issue resolution.

### 4. Improve User Experience for **DoiT CREs**

- Automated Diagnostics: Enable the AI agents to pre-emptively diagnose issues based on Zendesk ticket history and suggest common fixes.
- Custom Recommendations: Train the AI on DoiT's best practices for Looker, Looker Studio, and BigQuery to provide personalized troubleshooting suggestions.

---

## Contact Information



For issues, reach out to the repo owner at [dion@doit.com](mailto:dion@doit.com)