



BOBA MANATEE SERVER BACKEND (2024 EDITION)

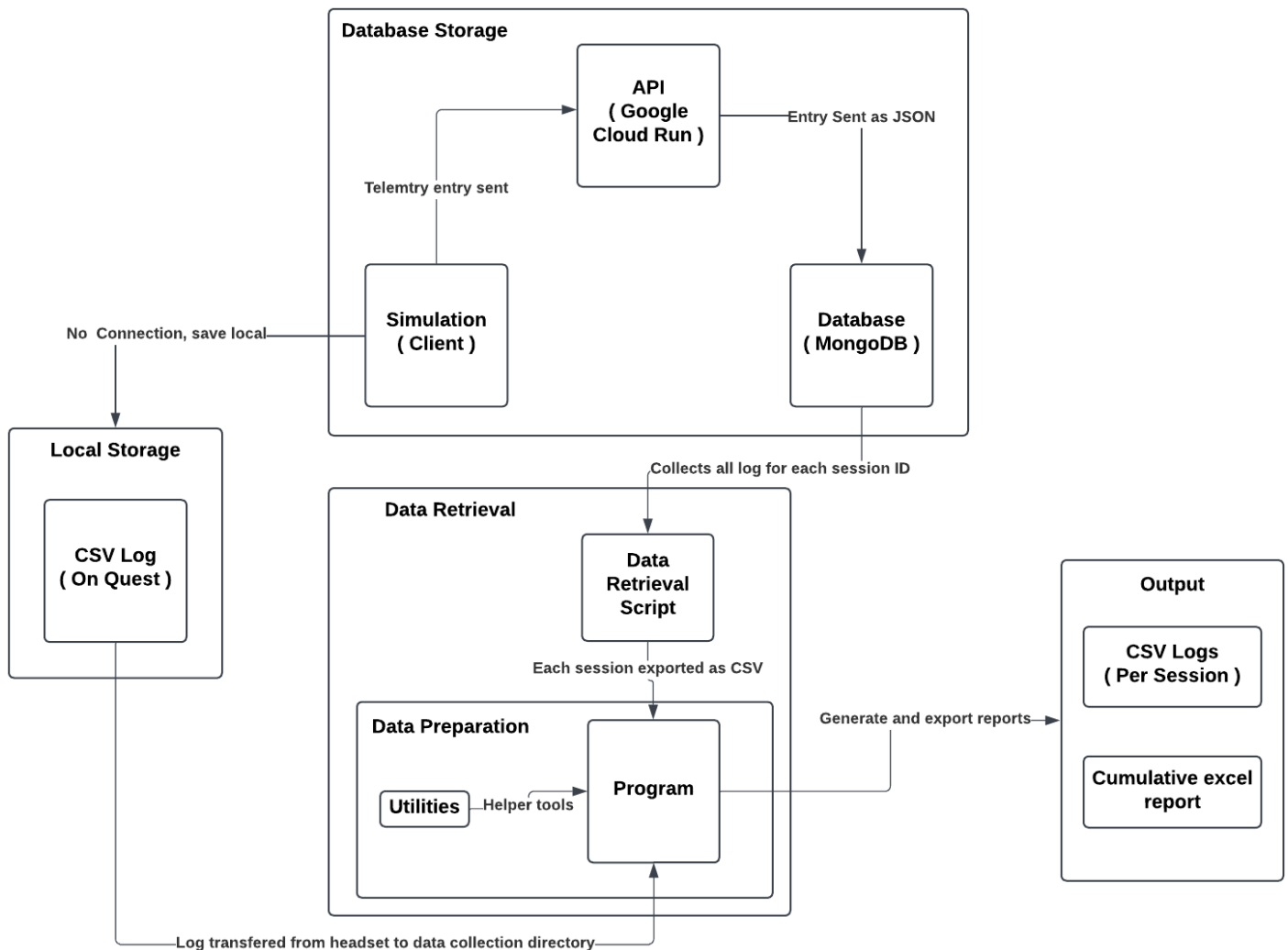
Instructions for use

A guide on collecting telemetry data from the Boba Manatee
Simulation.

Seamus Jackson
Seamus.jackson24@ncf.edu

Introduction

All core systems both within the simulation and externally are based on the original systems build by Alex Wills and Ender Fluegge in years prior. I have merely rebuilt the old tools with less config in mind. The system continues to support local and online data storage. With online data being the primary focus of the study, and local data storage being included as more of a precaution.



(System Architecture of data systems)

1.) Setting up

- **Ensure Python 3.11+ is installed**
- **Clone this repository**
 - With git installed do ``git clone https://github.com/Schweem/BobaManateeAnalysis.git``
 - If you're using GitHub desktop, paste the link in there and follow instructions
- **Setup your .env file. These means defining "ENDPOINT_URL" in your .env file**
 - **(.env file goes in the '/tools' directory alongside the python scripts)**
 - <https://www.geeksforgeeks.org/how-to-create-and-use-env-files-in-python/>
 - ``touch .env`, `micro .env`, `vi .env`, `nano .env`, etc...` This creates the file
 - **Once created, open it and in the first line write 'ENDPOINT_URL = APIURLHERE'**
 - you don't need any quotes in here. If you want to put them around the API URL that is OK but not needed.
- Install requirements '**pip install -r requirements.txt**' in the root directory (These are for both server and analysis)
 - Pip is a python package manager that should be installed alongside python. Running this command will attempt to install of the packages listed inside of the requirements file.

Starting a Session

A main goal of the 2024 simulation was to streamline the backend, both storing data on it and retrieving data from it. To accomplish this *Google Cloud Run* has been leveraged to deploy an API we can always access. When running a study, it is no longer required to setup or configure a new server, the simulation is built with a direct link to the API.

That said, to retrieve data and build reports there are still a list of prerequisite steps that must be taken.

Requirements:

- Python 3.11+ (To ensure compatibility with all libraries)
- While deployment is already established, the code and dockerfile for hosting a new deployment are included within the **'/tools/bobaserver'** directory.
 - Requirements for running the server are also included within the **'requirements.txt'** file in the main directory.

To begin collecting data, start the simulation after ensuring your Quest headset is connected to the internet. Once the simulation has started, the participant should turn around and face the door of the boat. Above it there is an indicator light and accompanying text:

- **Red** indicates a failed database connection.
 - The error causing the failure will be displayed and the user will be given the option to switch to *local data storage*.
 - *A local storage button will appear prompting the user to enable local data storage. At this point the light will turn green, and the text will indicate that data is being stored locally.*
- **Green** indicates a successful database connection
 - When displaying green and the connection message, you can start playing and data will be sent up to the server.

Once connected either to local storage or the database, the study can be run.

Retrieving a Session

Once a study has been run and all the individual sessions are complete data can be collected.

It is worth noting that data can be retrieved at any time, however it is most efficient and minimizes redundancy to collect all logs as once.

To retrieve logs from the database:

- Clone boba backend repository:
 - Git clone <https://github.com/Schweem/BobaManateeAnalysis/tree/main>
- Install requirements from 'requirements.txt'
 - `Pip install -r requirements.txt`, the file is located in the **/tools** directory
 - This includes all dependencies for all the scripts included in the repository
 - You must do this step in the same directory as the requirements file, or this will not work.
- With requirements installed, navigate into '**/tools**' and run '**dataRetrieval.py**'
 - ensure dependencies are created
 - '**pip install -r requirements.txt**'
 - This will create a 'telemetry_reports' directory in the same directory as the data retrieval script.

To retrieve logs from local storage:

- Plug Quest headset into your computer:
 - On windows you should be able to explore the file system immediately
 - On mac you will need to install an android file viewing utility
 - This is not a flaw with our design but rather a limitation of the interaction of the hardware.
 - Meta doesn't love Apple.
- Navigate to the '**Android/data/com.TR.Boba {sim version} /files/SessionDataLogs/[sessionID].csv**'
 - *Sim version will correspond to the title of the version of the simulation you ran. (I.e. Boba, BobaTaro, etc...)*
- Copy CSV reports from directory on headset to your PC:
 - create a '**telemetry_reports**' directory in the same directory as the data retrieval script ('**/tools**') just as the script would have done.

Creating Reports

Generating CSV and Excel reports:

- Inside of the **‘/tools’** directory you are in you will find **‘Program.py’** and **‘Utilities.py’**.
- Run **‘Program.py’**. || ``python program.py`` or ``python3 program.py``
 - For this to work you must have requirements installed and a **telemetry_reports** directory
- ``python program.py`` OR if that doesn't work for you ``python3 program.py``
 - This will scan the **‘/telemetry_reports’** directory and create both individual summary reports as well as an excel document with all reports aggregated onto one sheet in the **‘/output’** directory.

Clearing Data after Reports

The simplest is to delete documents from mongo atlas. It is the web UI they have designed to make interacting with your data more accessible and hands on.

- Navigate to MongoDB, link to collection is directly available on next step
- Browse collections
<https://cloud.mongodb.com/v2/672cd5a6723a0941430ed6b9#/metrics/replICASet/672cd649d90fae3f8bf81e3b/explorer>
- Delete the collection titled **‘TelemetryCollection’**
- Once deleted, click the plus icon that appears when you hover over the database name **‘TelemetryDB’**
- Create a new collection called **TelemetryCollection** with the default settings, the logs are now cleared.
 - This can be repeated following the same steps for the other **‘sessions’** table, however given the smaller number of sessions it is also practical to just delete the individual sessions, or if they are left and no corresponding logs are made, they will be observed but skipped during report generation.

Resources

- **Main repo link (Boba Server)**
 - <https://github.com/Schweem/BobaManateeAnalysis>
- **Old repo link (Dumpling Server)**
 - <https://github.com/AlexWills37/Dumpling-Backend-Server/tree/main>
- **Simulation (Current) – Private repo**
 - <https://github.com/Kinjen0/Boba-Manatee-Simulation>
- **Deployments – Requires access**
 - <https://cloud.mongodb.com/v2/672cd5a6723a0941430ed6b9#/metrics/repllicaSet/672cd649d90fae3f8bf81e3b/explorer/TelemetryDB/TelemetryCollection/find> - **Mongo**
 - <https://console.cloud.google.com/run/detail/us-central1/bobabackend/metrics?authuser=0&project=light-sunup-441618-s6&inv=1&inv=AbjWXA> – **Google Cloud Run**