The ATUAV Eye Tracker Service

# Starting the Service

1. define processors in a text file in the following format:

each line begins with a unique Tobii eye tracker product ID

separated by a tab comes a comma separated list of IDs to name each EmdatProcessor listening to that eye tracker.

if “-c” follows a processor ID, that processor will collect data cumulatively

e.g. the below will create 2 EmdatProcessors for TT120-204080900268. “observer” will be interval based and “experiment-c” will be cumulative

TT120-204-80900268 observer,experiment-c

2. run with arguments:

-b = base address for the service (e.g. localhost:8080)

-p = filepath/to/processors.txt

# Calling the Service

1. start data collection by calling start with a runID and AOI definitions

localhost:8080/start?runId={runId}&aois={aois}

- sets AOIs

- starts data collection

2a. periodically poll for whether a condition has been met by the data collected by a processor

localhost:8080/condition?processorId={processorId}&condition={condition}&callback={callback}

- returns true if condition was met, false otherwise

2b. alternatively poll for all features and do the analysis client side

localhost:8080/features?processorId={processorId}&callback={callback}

- returns a JSON representation of features and values

3. stop data collection

localhost:8080/stop

- stops data collection

- clears data

# Defining Conditions

Conditions are hardcoded in the ATUAV\_RT project. To create a new condition just create a new class that implements the **Condition** interface. That’s it, no further configuration is needed since Conditions are loaded by reflection and automatically attached to every EmdatProcessor.

# Tobii Documentation

* [Tobii Studio](http://www.tobii.com/Global/Analysis/Downloads/User_Manuals_and_Guides/Tobii_Studio1.X_UserManual.pdf)
* [Tobii SDK 3.0](http://www.tobii.com/Global/Analysis/Downloads/User_Manuals_and_Guides/Tobii%20SDK%203.0%20Release%20Candidate%201%20Developers%20Guide.pdf)