

CST-LOGGING: Capture & Query Applications for SAIT CST-Transmitters

TABLE OF CONTENTS

1. Introduction & Overview
2. Requirements
3. Installation
4. Overview of the Source Code
 1. Application.java Class
 2. CaptureApplication.java Class
 3. FileUtils.java Class
 4. LoggingUtils.java Class
 5. LogLineColumnNames.java Class
 6. LogLine.java Class
 7. LogReader.java Class
 8. MessageHandler.java Class
 9. QueryApplication.java Class
5. Building & Running the Application
 1. Capture Application
 2. Query Application

1. Introduction & Overview

The purpose of the cst-logging capture & query application is to read the real-time data from the SAIT CST300x transmitters, store that data into a local file and finally show that data in a simplified way and with a time filtering option to the user. With this application the user doesn't need to check the web application 24/7, in fact the user will be able to see past values from the transmitter.

2. Requirements

The cst-logging capture & query application is written in Java programming language. It is developed and tested on a computer which runs Microsoft Windows 10 Enterprise, version 1709 operating system. Also it is tested on a Apple MacBook which runs macOS High Sierra, version 10.13.3. The application is written with an IDE named Eclipse.

To run the application the user doesn't needs specifically use Eclipse. Users can use any Java IDE or even the command line, however it is written and tested in the Eclipse Environment. It is recommended to use Eclipse (Eclipse IDE for Java Developers, Version Photon 4.8.0 was used at the time).

You can find the download link here;

<https://www.eclipse.org/downloads/>

3. Installation

To install the application simply copy and paste the "cst-logging" directory somewhere in the computer. Launch Eclipse and follow these steps.

1. On the top left corner select **file**.
2. On the drop-down menu select **import**.
3. On the pop-up menu select **Existing Projects into Workspace** and click **Next**.

4. On the next pop-up menu mark **Select root directory** (this should be the default option).
5. Then click **Browse** and find the cst-logging directory.
6. Do not open the directory, just highlight it and click **select folder**.
7. It should appear in the field under **Projects**:
8. Make sure it is marked.
9. Click **finish**.
10. Wait for a few seconds and the application will be installed and be suitable for Eclipse.

4. Overview of the Source Code

The application consists of two parts, the capture part and the query part. The Java source code consist of nine classes. Comments are included in every class.

1. Application.java Class

This is the main class of the whole application. According to the first argument from the user it runs the **capture mode** or the **query mode**. If the user enters a totally different thing it prints necessary messages to the screen to help the user.

2. CaptureApplication.java Class

This class' main purpose is to create a file where the data will be stored using methods from the FileUtils.java class and also it connects with the websocket by sending the "init[]" command.

3. QueryApplication.java Class

This class is used during the query mode.

4. FileUtils.java Class

This class includes three methods to create a local file, write the data to that file and finally to read the data from that file.

5. LoggingUtils.java Class

This class includes the four message types that we will see through out the capture and query process. Its format is [TYPE] [DATE] [MESSAGE]. Type can be info, error, warn or debug depending on the situation. The date field shows the exact date and time of the message occurs. And the message part is the message. The debug mode is closed it can easily be opened when the user wants to debug the program.

6. LogLineColumnNames.java Class

This class just creates String values to be used in the LogLine.java class which are the column names from the query part of the application.

7. LogLine.java Class

This class and its methods are used by the query mode. This class creates a table for the query mode. The table's first column is the time stamp, showing when the data was initially stored. The other columns are the values from the slow monitoring(sm), hardware configuration(hc), current configuration(cc) and fast monitoring(fm). The table shows the values of the power amplifiers, power supplies and the DFS.

8. LogReader.java Class

This class includes two methods which to read the data from the file and another to add together the part that user wants to parse later on.

9. MessageHandler.java Class

This class handles with the websocket connection and the data which the program retrieves from the websocket.

5. Building & Running Application

If you are using Eclipse or any other IDE building is automatic however if you want to run the application in the command line, you'll need to rotate through your files using "cd" command find the application and to build it simply type "javac -filename.java".

1. Capture Application

To run the program in capture mode you must give two compulsory and one optional arguments.

[MODE] [IP ADDRESS] [DURATION]

Type “capture” for the capture mode as the first argument, then the ip address of the transmitter you want to connect to (i.e 1.1.1.1). And finally if you wish you can type a duration that the program will run. For example if you type 120 the application will run for 120 seconds.

2. Query Application

To run the program in query mode you must give two compulsory and two optional arguments.

[MODE] [TYPE] [START-TIME] [END-TIME]

Type “query” for the query mode as the first argument, then the type (i.e sm/fm/cc/ hc). If you only give these two arguments all of the data will be printed to the screen. To filter it you can give two more arguments. A start date and an end date. The format is; [YEAR]-[MONTH]-[DAY]T[HOUR]:[MINUTE]:[SECOND]Z (i.e 2018-08-29T12:30:00Z). Also the cst-logging directory includes two files that you can test the query mode in the cst-logs directory.

If you are using Eclipse to enter the arguments follow these steps;

1. On the top center click run.
2. On the drop-down menu click run configurations.
3. On the pop-up screen go to the arguments tab.
4. Enter your arguments as shown in to the program arguments field.

