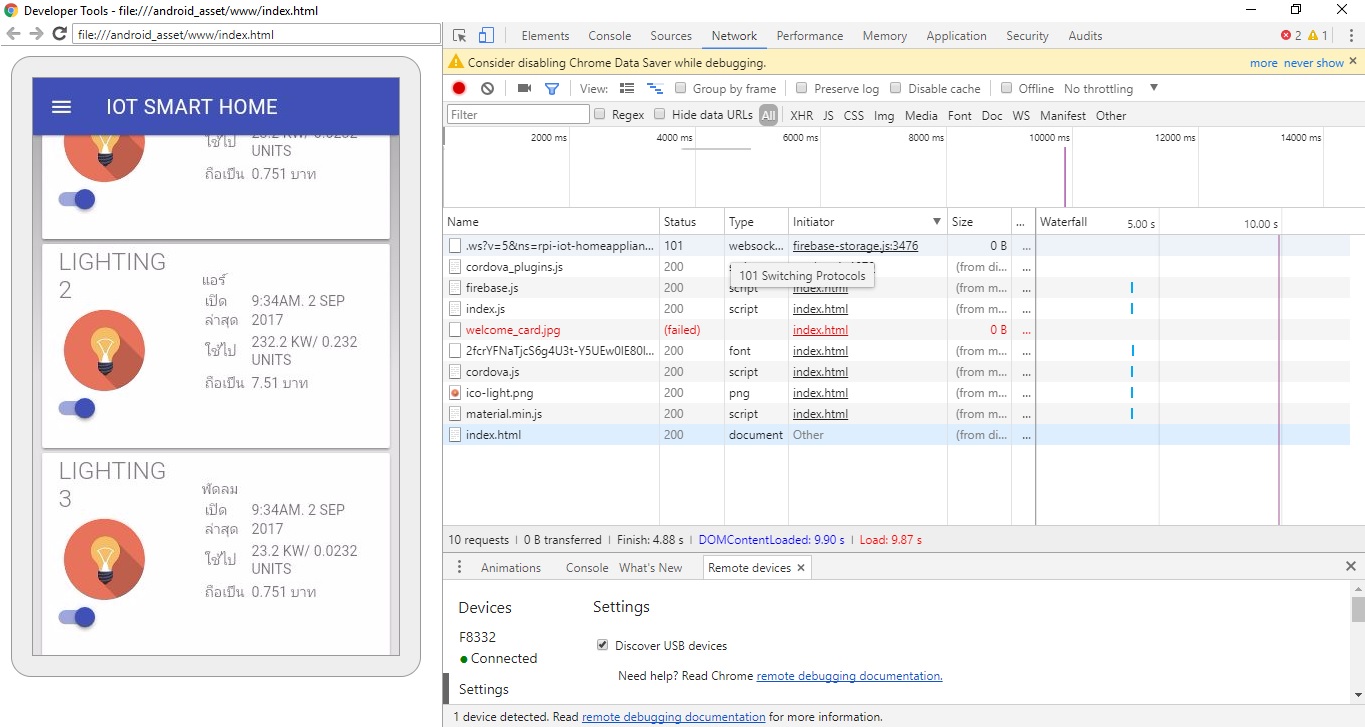
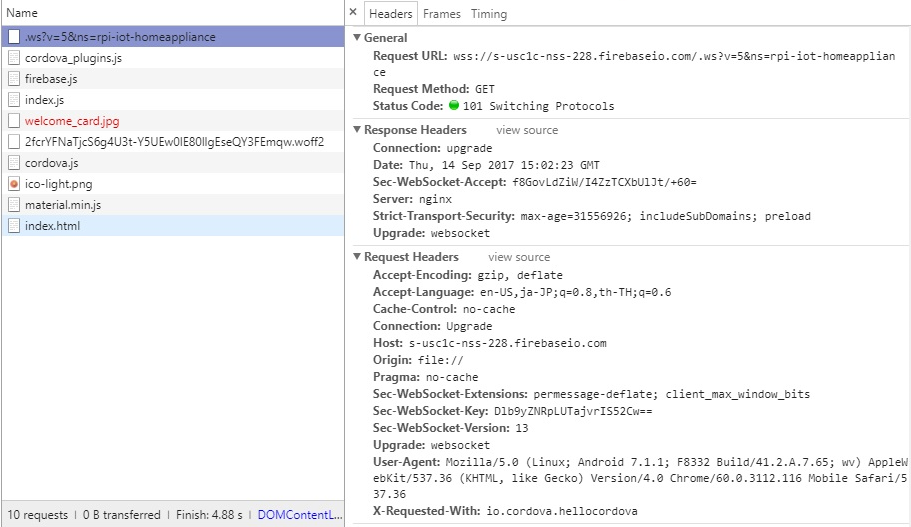
**IOT SMART HOME**

รุป 1

จากรูป 1 Application มีการทำงานโดยมีการติดต่อสื่อสารระหว่าง ฐานข้อมูล firebase โดยมี process wss://s-usc1c-nss-228.firebaseio.com/.ws?v=5&ns=rpi-iot-homeappliance ซึ่งมี status code เป็น 101 แปลว่า เมื่อเครื่องServer ได้รับ request แล้วก็จะ return นี้คืนมาให้พร้อมกับเปลี่ยน protocol ให้เป็นไปตามที่ request ในไฟล์ firebase-storage.js port 3476



รูป 2

จากรูป 2 ในส่วน Header

โดยเริ่ม request ขอ URL ไปที่ server ของ wss://s-usc1c-nss-228.firebaseio.com/.ws?v=5&ns=rpi-iot-homeappliance มีการใช้ Request Method GET

ซึ่งมี status code เป็น 101 Switching Protocols และยังคงเป็น stay alive จาก Function ที่ชื่อว่า Present ใน code js ของ Client ทำให้ยังไม่มีการปิด connetion

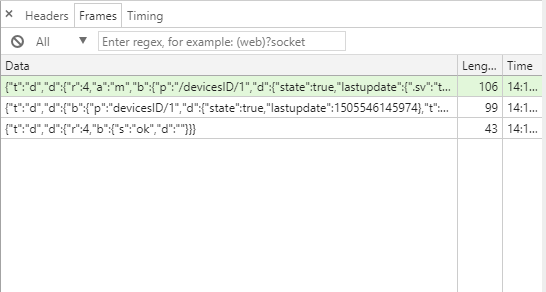
Server firebase ตอบรับ Response Header status conection เป็น upgrade Websocket ยอมรับ RCZR4CMpk44+KLRiXCJ/+LpEhiQ=

ร้องขอ Request Header รูปแบบการเข้ารหัส บีบอัดข้อมุล compression data ประเภท gzip, deflate

รองรับ ภาษา english japanese thai ไม่ระบุ catche



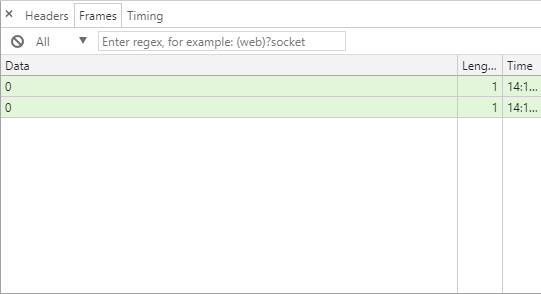
**รูป 3.1**



**รูป 3.2**

จากรูป 3.1 และ 3.2 ในส่วนนี้แสดง ส่วน FRAME โดยแสดงให้เห็นถึงข้อมูลใน frame ที่มีการส่งระหว่าง Client- Server (Application - Server Firebase) ซึ่งภายใน frame จะมีการแกะออกและผสมกับ code command คำสั่งของ firebase เพื่อทำการจัดการกับข้อมูล เช่น จากรูป “deviceID”, “q..’,”ok”,”d”,”deviceID/1”...,”lastupdate”:sv…,timestamp…. โดยแต่ละ frame จะแสดงขนาดหรือความยาว size,length ของData และ time

ซึงตอนนี้จากรูป แสดงการส่งข้อมูล อยู่ 3 frame โดยมี สถานะเปิดการใช้งานอุปกรณ์ที่ 1 (ไฟในห้องนอน) โดยแถบ frame ที่เน้นสีเหลือง({"t":"d","d":{"r":4,"a":"m","b":{"p":"/devicesID/1","d":{"state":true,"lastupdate":{".sv":"timestamp"}}}}}) แสดงให้เห็นถึงการ reqest จาก application จากนั้น อีก 2 frameเป็นการตอบกลับเพื่ออัพเดทสถานะของอุปกรณมาที่ application ให้มาแสดงในส่วน user interface เพื่อให้ user ทราบถึงสถานะการทำงานของอุปกรณ์

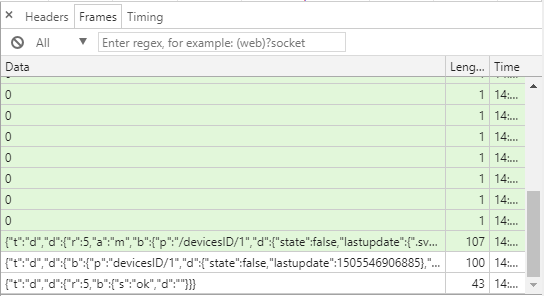


**รูป 4**

ต่อมาจากรูป แสดงเมื่อไม่มีการสั่งงานใดๆอยู่ในสถานะ stanby application จะส่ง frame ซึ่งมีข้อมูล เป็น 0 แต่จะยังมีไม่มีการปิด connection (stay live) ให้มีการพร้อมใช้งานเสมอ



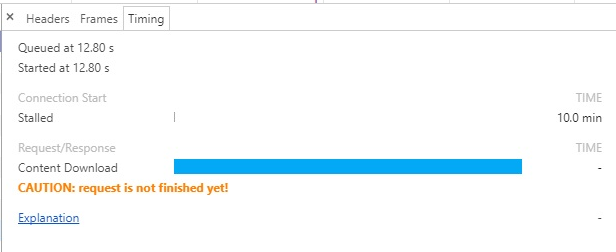
**รูป 5.1**



**รูป 5.2**

จากรูป 5.1 และ 5.2 เมื่อเราทำการปิด อุปกรณ์ จะมีการส่ง frame ซึ่งมี data {"t":"d","d":{"r":5,"a":"m","b":{"p":"/devicesID/1","d":{"state":false,"lastupdate":{".sv":"timestamp"}}}}}

ไปยัง server และอัพเดท database ของ firebase หลังจากนั้น return ค่า response กลับมา 2 frame ซึ่งมี data {"t":"d","d":{"r":5,"a":"m","b":{"p":"/devicesID/1","d":{"state":false,"lastupdate":{".sv":"timestamp"}}}}} และ {"t":"d","d":{"r":5,"b":{"s":"ok","d":""}}} เพื่ออัพเดทสถานะของอุปกรณมาที่ application ให้มาแสดงในส่วน user interface เพื่อให้ user ทราบถึงสถานะการทำงานของอุปกรณ์



**รูป 6**

จากรูป 6 ในส่วนสุดท้าย Timimg แสดงระยะเวลาทั้งหมด การเชื่อมต่อ เริ่มต้นและสิ้นสุด