


```
index.js x App.js x Questions.js x Dispatchers.js x package.json x Constants.js x utils.js x stationsDb.js x
78
79 return fetch('http://aqmthai.com/includes/getMultiManReport.php', {
80   method: 'POST', body, header: body.getHeaders()
81 }).then(res => res.text())
82 .then(body => {
83   const xml = parse(body)
84   const rows = xml.root.children[2].children
85   const trHeader = rows.shift().children
86   sensorTitleMap = trHeader.map(val => {
87     const [field1, field2] = val.content.split('_')
88     return field2 || '_time'
89   })
90   rows.splice(-5) // remove average fields 5 last fields
91   return rows
92 })
93 .then(rows => {
94   return rows.map(v => {
95     let c = v.children.shift().content
96     let [yyyy, m, d, hh, mm, ss] = c.split(',')
97     let time = moment.tz([yyyy, m - 1, d, hh, mm, ss], 'Asia/Bangkok').toDate()
98     let values = v.children.map(v => parseFloat(v.content) || -1)
99     let data = Object.entries([time, ...values]).reduce((prev, [idx, val]) => {
100       (val !== -1) && (prev[sensorTitleMap[idx]] = val)
101       return prev
102     }, {})
103     delete data._time
104     return Object.assign({}, {
105       data,
106       extra: {
107         time: time,
108         stationId: stationId,
109         jobId,
110         stationName: stations[stationId]
111       }
112     })
113   })
114 })
115 .catch(err => {
116
```

get()







index.js x App.js x Questions.js x Dispatchers.js x package.json x Constants.js x utils.js x stationsDb.js x

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

```
return fetch('http://aqmthai.com/includes/getMultiManReport.php', {
  method: 'POST', body, header: body.getHeaders()
}).then(res => res.text())
  .then(body => {
    const xml = parse(body)
    const rows = xml.root.children[2].children
    const trHeader = rows.shift().children
    sensorTitleMap = trHeader.map(val => {
      const [field1, field2] = val.content.split('_')
      return field2 || '_time'
    })
    rows.splice(-5) // remove average fields 5 last fields
    return rows
  })
  .then(rows => {
    return rows.map(v => {
      let c = v.children.shift().content
      let [yyyy, m, d, hh, mm, ss] = c.split(',')
      let time = moment.tz([yyyy, m - 1, d, hh, mm, ss], 'Asia/Bangkok').toDate()
      let values = v.children.map(v => parseFloat(v.content) || -1)
      let data = Object.entries([time, ...values]).reduce((prev, [idx, val]) => {
        (val !== -1) && (prev[sensorTitleMap[idx]] = val)
        return prev
      }, {})
      delete data._time
      return Object.assign({}, {
        data,
        extra: {
          time: time,
          stationId: stationId,
          jobId,
          stationName: stations[stationId]
        }
      })
    })
  })
  .catch(err => {
```

get()

bl Terminal

is ready to update. (5 minutes ago)

79:10 LF UTF-8 2 spaces Git: master



influxdb