

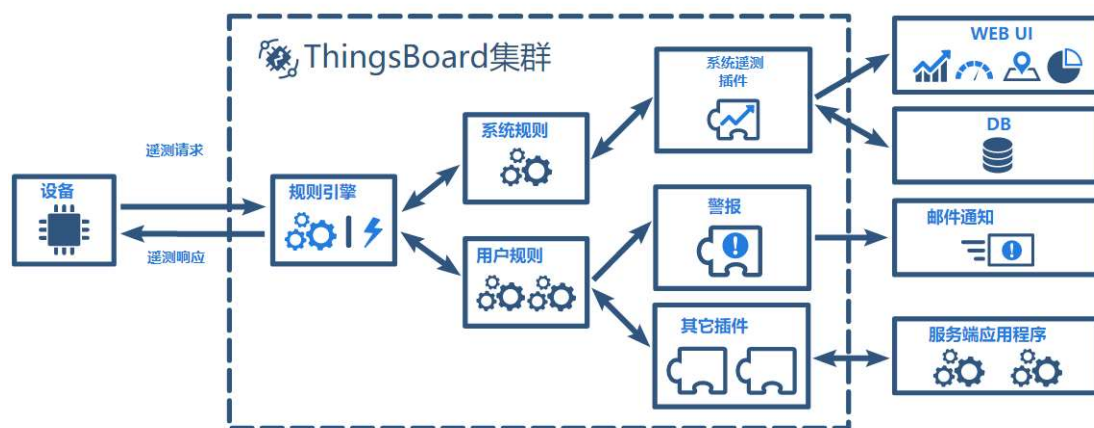
遥测数据应用

1 描述

ThingsBoard 提供与遥测数据操作相关的 API:

- 集成 MQTT, CoAP 或 HTTP 协议从设备采集数据
- 集成 Cassandra 数据库存储时间序数据
- 通过时间序列和时间间隔查询最新数据
- 集成 Websocket 订阅数据更新服务 (用于可视化或实时分析)
- 通过配置实现可以定制的 widgets and dashboards
- 通过灵活的规则引擎过滤和分析数据
- 通过采集数据实现异常报警
- 通过规则引擎节点实现与外部数据交互 (例如 Kafka 或 RabbitMQ 规则节点)

遥测数据处理



2 上传遥测数据 API

ThingsBoard 提供了一个用于上传时间序列键值数据的 API。

键值格式的灵活性和简单性允许与市场上几乎所有 IoT 设备轻松无缝地集成。

遥测上传 API 基于特定的网络协议。您可以在相应的参考页面中查看 API 和示例：

- [MQTT API 参考](#)
- [CoAP API 参考](#)
- [HTTP API 参考](#)

3 遥测服务

遥测服务负责将时间序列数据持久保存至数据库中；提供服务器端 API 来查询和订阅数据更新服务。

2.1 数据存储

ThingsBoard 使用 Cassandra NoSQL 数据库或 SQL 数据库来存储所有数据。


虽然您可以直接查询数据库，但是 ThingsBoard 提供了一组 RESTful 和 Websocket

API，可简化调用过程并应用某些安全策略：

- Tenant 管理员能够管理所拥有实体属性。
- Customer 用户只能管理 Tenant 分配的实体属性。

2.2 数据查询

遥测服务提供以下 REST API 来获取实体数据

 **swagger**

thingsboard (v2/api-docs?group=thingsboard) Authorize Explore

Thingsboard REST API

For instructions how to authorize requests please visit [REST API documentation page](#).

Created by Thingsboard team
 See more at <http://thingsboard.io>
[Contact the developer](#)
[Apache License Version 2.0](#)

admin-controller : Admin Controller	Show/Hide	List Operations	Expand Operations
alarm-controller : Alarm Controller	Show/Hide	List Operations	Expand Operations
asset-controller : Asset Controller	Show/Hide	List Operations	Expand Operations
audit-log-controller : Audit Log Controller	Show/Hide	List Operations	Expand Operations
auth-controller : Auth Controller	Show/Hide	List Operations	Expand Operations
blob-entity-controller : Blob Entity Controller	Show/Hide	List Operations	Expand Operations
component-descriptor-controller : Component Descriptor Controller	Show/Hide	List Operations	Expand Operations
converter-controller : Converter Controller	Show/Hide	List Operations	Expand Operations
custom-menu-controller : Custom Menu Controller	Show/Hide	List Operations	Expand Operations

注意: 上图中的 API 可通过 Swagger UI 使用, 如获取更多详细信息请查看 [REST API](#)。

支持如下实体类型: TENANT, CUSTOMER, USER, RULE, DASHBOARD, ASSET, DEVICE, ALARM

4 Timeseries data keys API

您可以使用下面的 GET 请求地址获取指定 entity 类型和 entity id 的所有属性 key 列

表

`http(s)://host:port/api/plugins/telemetry/{entityType}/{entityId}/keys/timeseries`

➤ 请求

```
curl -v -X GET http://localhost:8080/api/plugins/telemetry/DEVICE/ac8e6020-ae99-11e6-b9bd-2b15845ada4e/keys/timeseries \
--header "Content-Type:application/json" \
--header "X-Authorization: $JWT_TOKEN"
```

➤ 返回

```
["gas", "temperature"]
```

5 Timeseries data values API

您可以使用下面的 GET 请求地址获取指定 entity 类型和 entity id 的所有属性 value

列表

`http(s)://host:port/api/plugins/telemetry/{entityType}/{entityId}/values/timeseries?keys=key1,key2,key3`

➤ 请求

```
curl -v -X GET http://localhost:8080/api/plugins/telemetry/DEVICE/ac8e6020-ae99-11e6-b9bd-2b15845ada4e/values/timeseries?keys=gas,temperature \
--header "Content-Type:application/json" \
--header "X-Authorization: $JWT_TOKEN"
```

➤ 返回

```
{
  "gas": [
    {
      "ts": 1479735870786,
      "value": "1"
    }
  ],
  "temperature": [
    {
      "ts": 1479735870786,
      "value": "3"
    }
  ]
}
```

支持的实体类型为: TENANT, CUSTOMER, USER, DASHBOARD, ASSET, DEVICE,

ALARM, ENTITY_VIEW

您可以使用下面的 GET 请求地址获取指定 entity 类型和 entity id 的所有属性历史

value 列表

`http(s)://host:port/api/plugins/telemetry/{entityType}/{entityId}/values/timeseries?keys=key1,key2,key3&startTs=1479735870785&endTs=1479735871858&interval=60000&limit=100&agg=AVG`

参数说明：

1. keys-以逗号分隔的要获取的遥测键列表。
2. startTs -Unix 时间戳，标识间隔的开始（以毫秒为单位）。
3. endTs -Unix 时间戳，标识间隔的结束时间（以毫秒为单位）。
4. interval-聚合间隔，以毫秒为单位。
5. agg-聚合函数。MIN, MAX, AVG, SUM, COUNT, NONE 之一。
6. limit-要返回的最大数据点数或要处理的间隔。

ThingsBoard 将使用 startTs, endTs 和 interval 来标识聚合分区或子查询，并对利用内置聚合功能的数据库执行异步查询。

支持的实体类型为: TENANT, CUSTOMER, USER, DASHBOARD, ASSET, DEVICE,

ALARM, ENTITY_VIEW

➤ 请求

```
curl -v -X GET "http://localhost:8080/api/plugins/telemetry/DEVICE/ac8e6020-ae99-11e6-b9bd-2b15845ada4e/values/timeseries?keys=gas,temperature&startTs=1479735870785&endTs=1479735871858&interval=60000&limit=100&agg=AVG" \
--header "Content-Type:application/json" \
--header "X-Authorization: $JWT_TOKEN"
```

➤ 返回

```
{
  "gas": [
    {
      "ts": 1479735870786,
      "value": "1"
    },
    {
      "ts": 1479735871857,
      "value": "2"
    }
  ],
  "temperature": [
    {
```

```

    "ts": 1479735870786,
    "value": "3"
  },
  {
    "ts": 1479735871857,
    "value": "4"
  }
]
}

```

6 Websocket API

Thingsboard Web UI 积极使用 Websocket。Websocket API 复制了 REST API 功能，并提供了订阅设备数据更改的功能。您可以使用以下 URL 打开与遥测服务的 Websocket 连接 [ws\(s\)://host:port/api/ws/plugins/telemetry?token=\\$JWT_TOKEN](ws(s)://host:port/api/ws/plugins/telemetry?token=$JWT_TOKEN) 发送订阅相关的命令。

- cmdId-命令唯一标识（在相应的 Websocket 连接中）
- entityType-唯一实体类型标识。
- EntityId-唯一实体标识符
- keys-逗号分隔的 keys 参数列表
- timeWindow-订阅的时间间隔，以毫秒为单位。
- startTs-获取历史数据查询的间隔的开始时间（以毫秒为单位）。
- endTs-获取历史数据查询的间隔的结束时间（以毫秒为单位）。

实例

token – JWT 令牌

entityId – 设备标识

```

<!DOCTYPE HTML>
<html>
<head>

```

```
<script type="text/javascript">
    function WebSocketAPIExample() {
        var token = "YOUR_JWT_TOKEN";
        var entityId = "YOUR_DEVICE_ID";
        var websocket = new
WebSocket("ws(s)://host:port/api/ws/plugins/telemetry?token=" + token);

        if (entityId === "YOUR_DEVICE_ID") {
            alert("Invalid device id!");
            websocket.close();
        }

        if (token === "YOUR_JWT_TOKEN") {
            alert("Invalid JWT token!");
            websocket.close();
        }

        websocket.onopen = function () {
            var object = {
                tsSubCmds: [
                    {
                        entityType: "DEVICE",
                        entityId: entityId,
                        scope: "LATEST_TELEMETRY",
                        cmdId: 10
                    }
                ],
                historyCmds: [],
                attrSubCmds: []
            };
            var data = JSON.stringify(object);
            websocket.send(data);
            alert("Message is sent: " + data);
        };

        websocket.onmessage = function (event) {
            var received_msg = event.data;
            alert("Message is received: " + received_msg);
        };

        websocket.onclose = function (event) {
            alert("Connection is closed!");
        };
    }

```

```
    }  
  </script>  
  
</head>  
<body>  
  
<div id="sse">  
  <a href="javascript:WebSocketAPIExample()">Run WebSocket</a>  
</div>  
  
</body>  
</html>
```

7 数据可视化

ThingsBoard 提供了配置和自定义仪表板以进行数据可视化的功能。

[数据可视化指南](#)

8 规则引擎

ThingsBoard 提供了配置数据处理规则的功能。设备属性可在规则过滤器内使用。这允许基于某些设备属性应用规则。

[规则引擎指南](#)

原文地址: <https://thingsboard.io/docs/user-guide/telemetry/>