Version 6 / Examples

JsonGeneratorExample.ino

Description

This example shows how to generate a JSON document with ArduinoJson.

An online demo of this example is available on wandbox.org.

Source code

```
#include <ArduinoJson.h>

void setup() {
  // Initialize Serial port
  Serial.begin(9600);
  while (!Serial) continue;

  // Allocate the JSON document
  //
  // Inside the brackets, 200 is the RAM allocated to this document.
  // Don't forget to change this value to match your requirement.
  // Use arduinojson.org/v6/assistant to compute the capacity.
  StaticJsonDocument<200> doc;

  // StaticJsonObject allocates memory on the stack, it can be
  // replaced by DynamicJsonDocument which allocates in the heap.
  //
  // DynamicJsonDocument  doc(200);

  // Add values in the document
  //
  doc["sensor"] = "gps";
  doc["time"] = 1351824120;

  // Add an array.
  //
  JsonArray data = doc.createNestedArray("data");
  data.add(48.756080);
  data.add(2.302038);

  // Generate the minified JSON and send it to the Serial port.
  //
  serializeJson(doc, Serial);
  // The above line prints:
  // {"sensor":"gps","time":1351824120,"data":[48.756080,2.302038]}

  // Start a new line
  Serial.println();

  // Generate the prettified JSON and send it to the Serial port.
  //
  serializeJsonPretty(doc, Serial);
  // The above line prints:
  // {
  //   "sensor": "gps",
  //   "time": 1351824120,
  //   "data": [
  //     48.756080,
  //     2.302038
  //   ]
  // }
}

void loop() {
  // not used in this example
}
```

Things used in this example

Classes

- JsonDocument
  - StaticJsonDocument
  - operator[]
  - createNestedArray()
- JsonArray

  - `add()`

Functions

- `serializeJson()`
- `serializeJsonPretty()`

Libraries

- Core
  - `Serial`

See also

- Serialization tutorial
- JsonParserExample.ino

Home / Version 6 / Examples / JsonGeneratorExample.ino

ArduinoJson

A JSON library for embedded C++.
Simple, efficient, and versatile.

Newsletter

Your email Subscribe

Stay informed of the major changes.

GitHub

About Contact Privacy