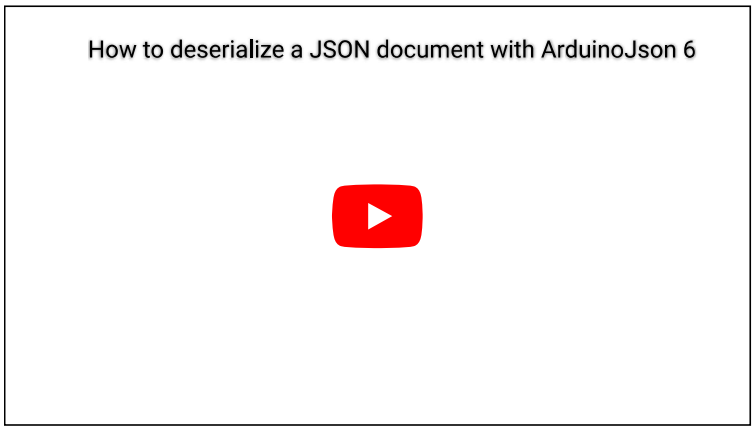
JsonParserExample.ino



Description

This example shows how to deserialize 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 capacity of the memory pool in bytes.
  // Don't forget to change this value to match your JSON document.
  // Use arduinojson.org/v6/assistant to compute the capacity.
  StaticJsonDocument<200> doc;

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

  // JSON input string.
  //
  // Using a char[], as shown here, enables the "zero-copy" mode. This mode uses
  // the minimal amount of memory because the JsonDocument stores pointers to
  // the input buffer.
  // If you use another type of input, ArduinoJson must copy the strings from
  // the input to the JsonDocument, so you need to increase the capacity of the
  // JsonDocument.
  char json[] =
      "{\"sensor\":\"gps\",\"time\":1351824120,\"data\":[48.756080,2.302038]}";

  // Deserialize the JSON document
  DeserializationError error = deserializeJson(doc, json);

  // Test if parsing succeeds.
  if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
  }

  // Fetch values.
  //
  // Most of the time, you can rely on the implicit casts.
  // In other case, you can do doc["time"].as<long>();
  const char* sensor = doc["sensor"];
  long time = doc["time"];
  double latitude = doc["data"][0];
  double longitude = doc["data"][1];

  // Print values.
  Serial.println(sensor);
  Serial.println(time);
  Serial.println(latitude, 6);
  Serial.println(longitude, 6);
}

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

Things used in this example

Classes

- JsonDocument
  - StaticJsonDocument
  - operator[]

- `DeserializationError`

Functions

- `deserializeJson()`

Libraries

- Core
  - `Serial`

See also

- Deserialization tutorial
- JsonHttpClient.ino