Validating HTTP Requests in Azure Functions

HTTP Request from Client

```
POST /Schedule
{
    "flightNo": "RA01",
    "departing": "CYUL",
    "arriving": "KLAX",
    "codeshares": [{
        "partner": "Delta",
        "flightNo": "123"
    }],
    "tailnumber": "N74656",
    "scheduled": 1553559808,
    "revised": 1553559808
}
```

**Blob Input Binding
/schemas/scheduleSchema.json**

```
Sample JSON Schema
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-
07/schema#",
  "title": "Scheduled Flight Schema",
  "required": [
    "flightNo",
    "departing",
    "arriving",
    ...
  ],
  "properties": {
    "flightNo": {
      "$id": "#/properties/flightNo",
      "type": "string",
      "title": "The Flightno Schema",
      "default": "",
      "examples": [
        "456"
      ]
    }
    ...
  }
}
```

HTTP Trigger Function
Read HTTP Request Stream and parse to JObject
Read Schema Stream and Parse to JSchema
Validate Request against Schema using
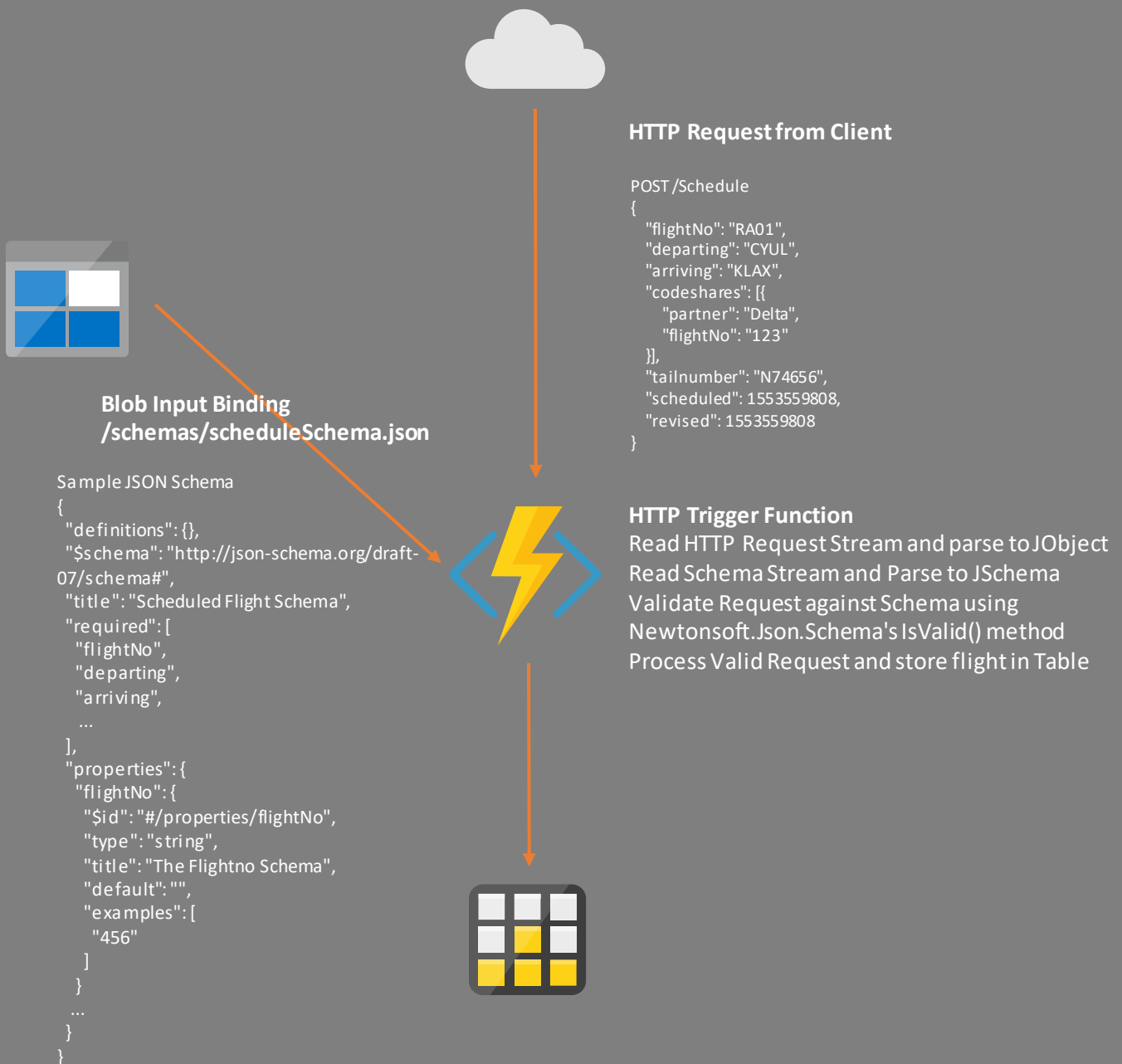Newtonsoft.Json.Schema's IsValid() method
Process Valid Request and store flight in Table

When To Use

- Non-trivial JSON validations
- JSON format is known ahead of time
- Low to medium traffic loads

Pros and Cons

- Handles complex JSON contracts
- Schema is hard to modify by hand
- Potential issues at large scale due to blob
stream retrieved on every call
- Can generate schema for existing contract

For details, code samples, and more, click here.

Written by Marc Roussy / MarcRoussy.com / @marcroussy