```
# 🔗 HTTP Monitors

> **Series:** SYNTH | **Notebook:** 3 of 6 | **Created:** December 2025

## Lightweight API and Endpoint Monitoring

This notebook covers HTTP monitors for API health checks, endpoint
validation, and multi-step API workflows using the latest Dynatrace platform.

---

## Table of Contents

1. HTTP Monitor Overview
2. Single Request Monitors
3. Multi-Step HTTP Monitors
4. Authentication
5. Response Validation
6. SSL Certificate Monitoring
7. Analyzing HTTP Results


## Prerequisites

- ✅ Access to a Dynatrace environment with Synthetic Monitoring
- ✅ Completed SYNTH-01 Fundamentals
- ✅ API endpoint(s) to monitor



## 1. HTTP Monitor Overview

HTTP monitors execute lightweight HTTP requests without browser overhead:

### HTTP vs Browser Monitors

| Aspect | HTTP Monitor | Browser Monitor |
|--------|-------------|-----------------|
| Execution | Direct HTTP call | Full browser |
| Speed | Fast (< 1s typical) | Slower (3-30s) |
| Resources | Minimal | Chrome instance |
| JavaScript | Not executed | Fully executed |
| Frequency | 1-60 minutes | 5-60 minutes |
| Cost | Lower | Higher |

### Use Cases
```

~~~
| Scenario | HTTP Monitor Type |
|----------|-------------------|
| API health check | Single request |
| REST API endpoint | Single request |
| Auth token + API call | Multi-step |
| GraphQL queries | Single/Multi-step |
| Webhook testing | Single request |
| Certificate expiry | Single request + SSL check |

### Configuration Path

**Dynatrace menu → Synthetic → Create synthetic monitor → Create HTTP
monitor**

## 2. Single Request Monitors

### Request Configuration

| Setting | Description | Example |
|---------|-------------|----------|
| **URL** | Full endpoint URL | `https://api.example.com/health` |
| **Method** | HTTP verb | GET, POST, PUT, DELETE, PATCH |
| **Headers** | Custom headers | `Authorization: Bearer ...` |
| **Body** | Request payload | JSON, form data, raw |
| **Timeout** | Max wait time | 30 seconds (default) |

### Common HTTP Methods

| Method | Use Case | Body |
|--------|----------|------|
| `GET` | Retrieve data | None |
| `POST` | Create resource | Required |
| `PUT` | Update resource | Required |
| `PATCH` | Partial update | Required |
| `DELETE` | Remove resource | Optional |
| `HEAD` | Check existence | None |
| `OPTIONS` | CORS preflight | None |

### Request Headers

```
Content-Type: application/json
Accept: application/json
Authorization: Bearer
X-API-Key:
User-Agent: Dynatrace Synthetic
```
~~~

~~~
```dql
// List all HTTP monitors
fetch dt.entity.http_check
| fields id, entity.name
| sort entity.name asc
| limit 50
```

```dql
// HTTP monitor execution results (last 24h)
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| fields timestamp,
         monitor = dt.entity.synthetic_test,
         location = dt.entity.synthetic_location,
         availability = synthetic.availability,
         response_time_ms = toDouble(synthetic.response_time),
         status_code = synthetic.http_status_code
| sort timestamp desc
| limit 100
```

## 3. Multi-Step HTTP Monitors

Chain multiple HTTP requests with data passing between steps:

### Multi-Step Workflow Example

![HTTP Monitor Workflow]
~~~

```
(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA4MDAgMzIwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
ic3RlcDFHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAgPHN0
b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmNjtzdG9wLW9wYWNpdHk6MSIgL
z4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMjU2M2ViO3N0b3
Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCB
pZD0ic3RlcDJHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzEwYjk4MTtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMDU5NjY5O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0ic3RlcDNHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzhiNWNmNjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojN2MzYW
VkO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0iZXh0cmFjdEdyYWRIVFRQIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIx
MDAlIj4KICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y1OWUwYjtzd
G9wLW9wYWNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2
xvcjojZDk3NzA2O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDx
```

```
maWx0ZXIgaWQ9Imh0dHBTaGFkb3ciPgogICAgICA8ZmVEcm9wU2hhZG93IGR4PSIxIiBkeT0iMSIg
c3RkRGV2aWF0aW9uPSIyIiBmbG9vZC1vcGFjaXR5PSIwLjE1Ii8+CiAgICA8L2ZpbHRlcj4KICAgI
DxtYXJrZXIgaWQ9Imh0dHBBcnJvdyIgbWFya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIH
JlZlg9IjkiIHJlZlk9IjMuNSIgb3JpZW50PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbnRzPSI
wIDAsIDEwIDMuNSwgMCA3IiBmaWxsPSIjNjQ3NDhiIi8+CiAgICA8L21hcmtlcj4KICA8L2RlZnM+
CgogIDwhLS0gQmFja2dyb3VuZCAtLT4KICA8cmVjdCB3aWR0aD0iODAwIiBoZWlnaHQ9IjMyMCIgZ
mlsbD0iI2Y4ZjlmYSIgcng9IjEwIi8+CgogIDwhLS0gVGl0bGUgLS0+CiAgPHRleHQgeD0iNDAwIi
B5PSIyOCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4IiBmb25
0LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+TXVsdGktU3Rl
cCBIVFRQIE1vbml0b3IgV29ya2Zsb3c8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSI0OCIgZm9ud
C1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjNjY2IiB0ZX
h0LWFuY2hvcj0ibWlkZGxlIj5DaGFpbmVkIEFQSSByZXF1ZXN0cyB3aXRoIHZhcmlhYmxlIGV4dHJ
hY3Rpb248L3RleHQ+CgogIDwhLS0gU3RlcCAxOiBBdXRoZW50aWNhdGUgLS0+CiAgPHJlY3QgeD0i
MzAiIHk9Ijc1IiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjEwMCIgcng9IjEwIiBmaWxsPSJ1cmwoI3N0Z
XAxR3JhZCkiIGZpbHRlcj0idXJsKCNodHRwU2hhZG93KSIvPgogIDx0ZXh0IHg9IjEzMCIgeT0iMT
AwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2V
pZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U3RlcCAxOiBBdXRo
ZW50aWNhdGU8L3RleHQ+CiAgPHRleHQgeD0iMTMwIiB5PSIxMjAiIGZvbnQtZmFtaWx5PSJtb25vc
3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbm
Nob3I9Im1pZGRsZSI+UE9TVCAvYXBpL2xvZ2luPC90ZXh0PgogIDx0ZXh0IHg9IjEzMCIgeT0iMTQ
wIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJn
YmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+eyJ1c2VyIjogInRlc3Qif
TwvdGV4dD4KICA8dGV4dCB4PSIxMzAiIHk9IjE2MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLX
NlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5
jaG9yPSJtaWRkbGUiPkV4cGVjdDogMjAwIE9LPC90ZXh0PgoKICA8IS0tIEV4dHJhY3QgVG9rZW4g
LS0+CiAgPHJlY3QgeD0iODAiIHk9IjE4MCIgd2lkdGg9IjEwMCIgaGVpZ2h0PSIzMCIgcng9IjYiI
GZpbGw9InVybCgjZXh0cmFjdEdyYWRIVFRQKSIgZmlsdGVyPSJ1cmwoI2h0dHBTaGFkb3cpIi8+Ci
AgPHRleHQgeD0iMTMwIiB5PSIyMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9
udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0i
bWlkZGxlIj5FeHRyYWN0OiB0b2tlbjwvdGV4dD4KCiAgPCEtLSBBcnJvdyB0byBTdGVwIDIgLS0+C
iAgPHBhdGggZD0iTTIzMCwxMjUgTDI4MCwxMjUiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZH
RoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZW5kPSJ1cmwoI2h0dHBBcnJvdykiLz4KICA8cGF0aCB
kPSJNMTgwLDE5NSBMMTgwLDIzMCBMMzEwLDIzMCBMMzEwLDE4MCIgc3Ryb2tlPSIjZjU5ZTBiIiBz
dHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIHN0cm9rZS1kYXNoYXJyYXk9IjQsMiIgbWFya2VyL
WVuZD0idXJsKCNodHRwQXJyb3cpIi8+CiAgPHRleHQgeD0iMjQ1IiB5PSIyNDUiIGZvbnQtZmFtaW
x5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzkyNDAwZSI+VXNlIHR
va2VuIGluIGhlYWRlcjwvdGV4dD4KCiAgPCEtLSBTdGVwIDI6IEdldCBVc2VyIC0tPgogIDxyZWN0
IHg9IjI5MCIgeT0iNzUiIHdpZHRoPSIyMDAiIGhlaWdodD0iMTAwIiByeD0iMTAiIGZpbGw9InVyb
Cgjc3RlcDJHcmFkKSIgZmlsdGVyPSJ1cmwoI2h0dHBTaGFkb3cpIi8+CiAgPHRleHQgeD0iMzkwIi
B5PSIxMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9
udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5TdGVwIDI6
IEdldCBVc2VyPC90ZXh0PgogIDx0ZXh0IHg9IjM5MCIgeT0iMTIwIiBmb250LWZhbWlseT0ibW9ub
3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW
5jaG9yPSJtaWRkbGUiPkdFVCAvYXBpL3VzZXJzL21lPC90ZXh0PgogIDx0ZXh0IHg9IjM5MCIgeT0
iMTQwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9
InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+QXV0aDogQmVhcmVyI
Ht0b2tlbn08L3RleHQ+CiAgPHRleHQgeD0iMzkwIiB5PSIxNjAiIGZvbnQtZmFtaWx5PSJBcmlhbC
wgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB
```

```
0ZXh0LWFuY2hvcj0ibWlkZGxlIj5FeHBlY3Q6IDIwMCBPSzwvdGV4dD4KCiAgPCEtLSBFeHRyYWN0
IHVzZXJJZCAtLT4KICA8cmVjdCB4PSIzNDAiIHk9IjE4MCIgd2lkdGg9IjEwMCIgaGVpZ2h0PSIzM
CIgcng9IjYiIGZpbGw9InVybCgjZXh0cmFjdEdyYWRIVFRQKSIgZmlsdGVyPSJ1cmwoI2h0dHBTaG
Fkb3cpIi8+CiAgPHRleHQgeD0iMzkwIiB5PSIyMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1
zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0
LWFuY2hvcj0ibWlkZGxlIj5FeHRyYWN0OiB1c2VySWQ8L3RleHQ+CgogIDwhLS0gQXJyb3cgdG8gU
3RlcCAzIC0tPgogIDxwYXRoIGQ9Ik00OTAsMTI1IEw1NDAsMTI1IiBzdHJva2U9IiM2NDc0OGIiIH
N0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNodHRwQXJyb3cpIi8
+CiAgPHBhdGggZD0iTTQ0MCwxOTUgTDQ0MCwyNjAgTDU3MCwyNjAgTDU3MCwxODAiIHN0cm9rZT0i
I2Y1OWUwYiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBzdHJva2UtZGFzaGFycmF5PSI0L
DIiIG1hcmtlci1lbmQ9InVybCgjaHR0cEFycm93KSIvPgogIDx0ZXh0IHg9IjUwNSIgeT0iMjc1Ii
Bmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM5MjQ
wMGUiPlVzZSB1c2VySWQgaW4gcGF0aDwvdGV4dD4KCiAgPCEtLSBTdGVwIDM6IEdldCBPcmRlcnMg
LS0+CiAgPHJlY3QgeD0iNTUwIiB5PSI3NSIgd2lkdGg9IjIyMCIgaGVpZ2h0PSIxMDAiIHJ4PSIxM
CIgZmlsbD0idXJsKCNzdGVwM0dyYWQpIiBmaWx0ZXI9InVybCgjaHR0cFNoYWRvdykiLz4KICA8dG
V4dCB4PSI2NjAiIHk9IjEwMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXN
pemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRk
bGUiPlN0ZXAgMzogR2V0IE9yZGVyczwvdGV4dD4KICA8dGV4dCB4PSI2NjAiIHk9IjEyMCIgZm9ud
C1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NS
wwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5HRVQgL2FwaS91c2Vycy97dXNlcklkfS9vcmRlcnM
8L3RleHQ+CiAgPHRleHQgeD0iNjYwIiB5PSIxNDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1z
ZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY
2hvcj0ibWlkZGxlIj5BdXRoOiBCZWFyZXIge3Rva2VufTwvdGV4dD4KICA8dGV4dCB4PSI2NjAiIH
k9IjE2MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWx
sPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkV4cGVjdDogMjAw
LCBvcmRlcnMubGVuZ3RoICZndDsgMDwvdGV4dD4KCiAgPCEtLSBGaW5hbCB2YWxpZGF0aW9uIC0tP
gogIDxyZWN0IHg9IjU4MCIgeT0iMTgwIiB3aWR0aD0iMTYwIiBoZWlnaHQ9IjMwIiByeD0iNiIgZm
lsbD0iI2QxZmFlNSIgc3Ryb2tlPSIjMTBiOTgxIiBzdHJva2Utd2lkdGg9IjEiLz4KICA8dGV4dCB
4PSI2NjAiIHk9IjIwMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9
IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzA0Nzg1NyIgdGV4dC1hbmNob3I9Im1pZGRsZ
SI+VmFsaWRhdGUgUmVzcG9uc2U8L3RleHQ+CgogIDwhLS0gTGVnZW5kIC0tPgogIDxyZWN0IHg9Ij
MwIiB5PSIyOTUiIHdpZHRoPSI3NDAiIGhlaWdodD0iMTgiIHJ4PSI0IiBmaWxsPSIjZTBmMmZlIi8
+CiAgPHRleHQgeD0iNDAwIiB5PSIzMDgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIg
Zm9udC1zaXplPSIxMCIgZmlsbD0iIzAzNjlhMSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+VmFyaWFib
GVzIGV4dHJhY3RlZCBmcm9tIG9uZSBzdGVwIGNhbiBiZSB1c2VkIGluIHN1YnNlcXVlbnQgc3RlcH
MgdmlhICR7dmFyaWFibGVfbmFtZX0gc3ludGF4PC90ZXh0Pgo8L3N2Zz4K)


### Variable Extraction

| Source | Syntax | Example |
|--------|--------|----------|
| JSON path | `$.data.token` | Extract from JSON body |
| Response header | `header:X-Request-Id` | Extract from headers |
| Regex | `token":"([^"]+)` | Pattern matching |

### Using Variables
```

~~~~
Reference extracted variables in subsequent steps:
- URL: `https://api.example.com/users/${userId}`
- Header: `Authorization: Bearer ${token}`
- Body: `{"userId": "${userId}"}`

```dql
// Multi-step HTTP monitor step performance
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| filter isNotNull(synthetic.step_name)
| summarize {
    avg_duration_ms = avg(toDouble(synthetic.step_duration)),
    success_rate = countIf(synthetic.step_success == true) * 100.0 / count(),
    executions = count()
  }, by: {dt.entity.synthetic_test, synthetic.step_name}
| sort avg_duration_ms desc
| limit 30
```

## 4. Authentication

### Supported Authentication Methods

| Method | Configuration | Use Case |
|--------|---------------|----------|
| **Basic Auth** | Username/password encoded | Legacy APIs |
| **Bearer Token** | Header: `Authorization: Bearer ` | OAuth2, JWT |
| **API Key** | Header: `X-API-Key: ` | Third-party APIs |
| **OAuth2** | Token endpoint + credentials | Modern APIs |
| **Client Certificate** | mTLS | High-security APIs |

### Credential Vault

Store sensitive credentials securely:

1. **Settings → Integration → Credential vault**
2. Add credential (username/password, token, certificate)
3. Reference in monitor: `${credentials.vault.myCredential}`

### OAuth2 Flow Example

```
Step 1: Get Token
    POST https://auth.example.com/oauth/token
    Body: grant_type=client_credentials
          &client_id=${vault.clientId}
          &client_secret=${vault.clientSecret}
~~~~

~~~
    Extract: access_token

Step 2: API Call
    GET https://api.example.com/data
    Header: Authorization: Bearer ${access_token}
```

## 5. Response Validation

### HTTP Status Validation

| Status Range | Meaning | Default Behavior |
|--------------|---------|------------------|
| 2xx | Success | Pass |
| 3xx | Redirect | Follow/Pass |
| 4xx | Client Error | Fail |
| 5xx | Server Error | Fail |

### Content Validation

| Type | Description | Example |
|------|-------------|----------|
| **Contains** | Text present | `"status": "ok"` |
| **Not Contains** | Text absent | `"error"` |
| **Regex** | Pattern match | `"id":\s*\d+` |
| **JSON Path** | Value at path | `$.status == "success"` |

### JSON Path Assertions

```json
// Response:
{
  "status": "success",
  "data": {
    "users": [
      {"id": 1, "name": "John"},
      {"id": 2, "name": "Jane"}
    ]
  }
}

// Assertions:
$.status == "success"           // Check status
$.data.users.length > 0         // Array not empty
$.data.users[0].name == "John"  // First user name
```

```dql
~~~

~~~
// HTTP status code distribution
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| summarize {
    count = count()
  }, by: {dt.entity.synthetic_test, synthetic.http_status_code}
| sort count desc
| limit 30
```

```dql
// Failed HTTP requests with error details
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| filter synthetic.availability == false
| fields timestamp,
         monitor = dt.entity.synthetic_test,
         location = dt.entity.synthetic_location,
         status_code = synthetic.http_status_code,
         error = synthetic.error_message
| sort timestamp desc
| limit 50
```

## 6. SSL Certificate Monitoring

HTTP monitors automatically check SSL certificates:

### Certificate Checks

| Check | Description | Alert Threshold |
|-------|-------------|-----------------|
| **Validity** | Certificate not expired | Configurable days |
| **Chain** | Valid certificate chain | Any break |
| **Hostname** | Matches request domain | Mismatch |
| **Trust** | Issued by trusted CA | Untrusted |

### Expiration Alerts

Configure alerts for certificates expiring within:
- 30 days (warning)
- 14 days (critical)
- 7 days (emergency)

```dql
// SSL certificate expiration status
~~~

~~~
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter isNotNull(synthetic.ssl_certificate_expiry)
| summarize {
    latest_check = max(timestamp),
    certificate_expiry = takeFirst(synthetic.ssl_certificate_expiry),
    days_until_expiry = takeFirst(toDouble(synthetic.ssl_days_until_expiry))
  }, by: {dt.entity.synthetic_test}
| filter days_until_expiry < 30
| sort days_until_expiry asc
| limit 20
```

## 7. Analyzing HTTP Results

```dql
// HTTP monitor availability summary
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| summarize {
    total = count(),
    successful = countIf(synthetic.availability == true),
    failed = countIf(synthetic.availability == false)
  }, by: {dt.entity.synthetic_test}
| fieldsAdd availability_pct = round((successful * 100.0) / total, decimals: 2)
| sort availability_pct asc
| limit 30
```

```dql
// Response time percentiles by monitor
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| filter synthetic.availability == true
| summarize {
    p50_ms = percentile(toDouble(synthetic.response_time), 50),
    p95_ms = percentile(toDouble(synthetic.response_time), 95),
    p99_ms = percentile(toDouble(synthetic.response_time), 99),
    max_ms = max(toDouble(synthetic.response_time)),
    executions = count()
  }, by: {dt.entity.synthetic_test}
| sort p95_ms desc
| limit 20
```
~~~

~~~
```dql
// HTTP timing breakdown (DNS, connect, SSL, TTFB)
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| filter synthetic.availability == true
| summarize {
    avg_dns_ms = avg(toDouble(synthetic.dns_time)),
    avg_connect_ms = avg(toDouble(synthetic.connect_time)),
    avg_ssl_ms = avg(toDouble(synthetic.ssl_time)),
    avg_ttfb_ms = avg(toDouble(synthetic.time_to_first_byte)),
    avg_total_ms = avg(toDouble(synthetic.response_time)),
    executions = count()
  }, by: {dt.entity.synthetic_test}
| sort avg_total_ms desc
| limit 20
```

```dql
// Response time trend over time
fetch bizevents, from: now() - 7d
| filter event.provider == "dynatrace.synthetic"
| filter matchesValue(event.type, "*http*")
| filter synthetic.availability == true
| makeTimeseries {
    avg_response_ms = avg(toDouble(synthetic.response_time)),
    p95_response_ms = percentile(toDouble(synthetic.response_time), 95)
  }, interval: 1h
```

---

## Summary

In this notebook, you learned:

✅ **HTTP monitor types** - Single request vs multi-step
✅ **Request configuration** - Methods, headers, body
✅ **Multi-step workflows** - Variable extraction and chaining
✅ **Authentication** - Basic, Bearer, OAuth2, API keys
✅ **Response validation** - Status codes, content, JSON path
✅ **SSL monitoring** - Certificate expiration alerts
✅ **Analysis queries** - Availability, timing breakdown

---

## Next Steps
~~~

```
Continue to **SYNTH-04: Scripted Monitors** to learn about advanced scripting capabilities.

---

## References

- [HTTP Monitors](https://docs.dynatrace.com/docs/platform-modules/digital-experience/synthetic-monitoring/http-monitors)
- [Multi-step HTTP Monitors](https://docs.dynatrace.com/docs/platform-modules/digital-experience/synthetic-monitoring/http-monitors/create-http-monitor)
- [Credential Vault](https://docs.dynatrace.com/docs/manage/identity-access-management/credential-vault)
- [SSL Certificate Monitoring](https://docs.dynatrace.com/docs/platform-modules/digital-experience/synthetic-monitoring/analysis-and-alerting/synthetic-ssl-certificates)
```