```
# 🌐 Browser Monitors

> **Series:** SYNTH | **Notebook:** 2 of 6 | **Created:** December 2025

## Creating and Optimizing Browser-Based Synthetic Tests

This notebook covers browser monitors in Dynatrace, including single-URL
monitors, browser clickpaths, and performance analysis using the latest
Dynatrace platform capabilities.

---

## Table of Contents

1. Browser Monitor Types
2. Single-URL Monitors
3. Browser Clickpaths
4. Performance Metrics
5. Validation and Assertions
6. Analyzing Browser Results


## Prerequisites

- ✅ Access to a Dynatrace environment with Synthetic Monitoring
- ✅ Completed SYNTH-01 Fundamentals
- ✅ Web application URL to monitor



## 1. Browser Monitor Types

### Single-URL Browser Monitor

Loads a single page and captures performance metrics:

| Feature | Description |
|---------|-------------|
| **Execution** | Full Chrome browser render |
| **Metrics** | W3C Navigation Timing, resource timing |
| **Screenshots** | Automatic capture on completion/failure |
| **Validation** | Content validation, element checks |

**Best For:**
- Homepage availability
- Landing page performance
- Single page applications (SPA) initial load
```

```
### Browser Clickpath Monitor

Multi-step user journey simulation:

| Feature | Description |
|---------|-------------|
| **Steps** | Multiple pages/actions in sequence |
| **Interactions** | Click, type, select, wait |
| **State** | Cookies maintained within execution (session NOT maintained between executions) |
| **Content Validation** | Validate text/elements exist on page |

> **Note:** Unlike RUM session properties, browser clickpaths do not support extracting data into variables for use across executions. Content validation allows you to verify expected text/elements exist, but you cannot capture dynamic values for reuse.

**Best For:**
- Login flows
- Checkout processes
- Form submissions
- Multi-page workflows

### Configuration Path

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

## 2. Single-URL Monitors

### Creating a Single-URL Monitor

1. **URL Configuration**
   - Enter the full URL (https://...)
   - Set viewport size (desktop, tablet, mobile)
   - Configure user agent string

2. **Execution Settings**
   - Frequency: 5-60 minutes
   - Locations: Select public or private
   - Timeout: Maximum execution time

3. **Validation Rules**
   - HTTP status code validation
   - Content validation (text, regex)
   - Element presence checks
```

~~~
### Viewport Presets

| Preset | Dimensions | Use Case |
|--------|------------|----------|
| Desktop | 1920x1080 | Standard desktop |
| Laptop | 1366x768 | Common laptop |
| Tablet | 768x1024 | iPad portrait |
| Mobile | 375x667 | iPhone 8 |

```dql
// List all browser monitors
fetch dt.entity.synthetic_test
| filter isNotNull(browserMonitorSubtype)
| fields id, entity.name, isEnabled, browserMonitorSubtype
| sort entity.name asc
| limit 50
```

```dql
// Browser monitor execution results (last 24h)
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| fields timestamp,
         monitor = synthetic_test_id,
         location = synthetic_location_id,
         availability = execution_success,
         response_time_ms = toDouble(total_duration)
| sort timestamp desc
| limit 100
```

## 3. Browser Clickpaths

### Creating Clickpath Monitors

#### Option 1: Record with Browser Extension

1. Install Dynatrace Synthetic Recorder (Chrome extension)
2. Start recording session
3. Perform user journey in browser
4. Stop recording and export to Dynatrace

#### Option 2: Manual Script Creation

Define steps programmatically using the script editor.

### Common Actions
~~~

~~~
| Action | Description | Example |
|--------|-------------|----------|
| `navigate` | Go to URL | Navigate to login page |
| `click` | Click element | Click login button |
| `type` | Enter text | Type username |
| `selectOption` | Select dropdown | Select country |
| `wait` | Wait for condition | Wait for element visible |
| `javascript` | Execute JS | Custom validation |

### Element Selectors

| Selector Type | Example | Best Practice |
|--------------|---------|---------------|
| CSS | `#login-btn` | Preferred - stable |
| XPath | `//button[@id='login']` | Complex structures |
| Link Text | `Login` | Simple links |
| Data Attribute | `[data-testid='login']` | Test automation |

```dql
// Clickpath step performance analysis
fetch dt.synthetic.events, from: now() - 24h
| filter isNotNull(step_title)
| summarize {
    avg_duration_ms = avg(toDouble(step_duration)),
    max_duration_ms = max(toDouble(step_duration)),
    executions = count()
  }, by: {synthetic_test_id, step_title}
| sort avg_duration_ms desc
| limit 30
```

## 4. Performance Metrics

### W3C Navigation Timing

Browser monitors capture detailed timing metrics based on the W3C Navigation
Timing API:

![Navigation Timing]
~~~

(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA4MDAgMjUwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iZG5zR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgogICAgICA8c3RvcCBv
ZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZWY0NDQ0O3N0b3Atb3BhY2l0eToxIiAvPgogI
CAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkYzI2MjY7c3RvcC1vcG
FjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW50IGlkPSJ
jb25uZWN0R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgogICAgICA8c3Rv
cCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZjk3MzE2O3N0b3Atb3BhY2l0eToxIiAvP
gogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNlYTU4MGM7c3RvcC

```
1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW50IGl
kPSJzc2xHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAgIDxzdG9w
IG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNmNTllMGI7c3RvcC1vcGFjaXR5OjEiIC8+C
iAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Q5NzcwNjtzdG9wLW
9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQgaWQ
9InJlcXVlc3RHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAgIDxz
dG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4NGNjMTY7c3RvcC1vcGFjaXR5OjEiI
C8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzY1YTMwZDtzdG
9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQ
gaWQ9InJlc3BvbnNlR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgogICAg
ICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTBiOTgxO3N0b3Atb3BhY2l0e
ToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMwNTk2Nj
k7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWR
pZW50IGlkPSJkb21HcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAg
IDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMzYjgyZjY7c3RvcC1vcGFjaXR5O
jEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzI1NjNlYj
tzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGl
lbnQgaWQ9ImxvYWRHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAg
IDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4YjVjZjY7c3RvcC1vcGFjaXR5O
jEiIC8+CiAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzdjM2FlZD
tzdG9wLW9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8ZmlsdGVyIGlkPSJ
uYXZTaGFkb3ciPgogICAgICA8ZmVEcm9wU2hhZG93IGR4PSIxIiBkeT0iMSIgc3RkRGV2aWF0aW9u
PSIyIiBmbG9vZC1vcGFjaXR5PSIwLjE1Ii8+CiAgICA8L2ZpbHRlcj4KICA8L2RlZnM+CgogIDwhL
S0gQmFja2dyb3VuZCAtLT4KICA8cmVjdCB3aWR0aD0iODAwIiBoZWlnaHQ9IjI1MCIgZmlsbD0iI2
Y4ZjlmYSIgcng9IjEwIi8+CgogIDwhLS0gVGl0bGUgLS0+CiAgPHRleHQgeD0iNDAwIiB5PSIyOCI
gZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4IiBmb250LXdlaWdo
dD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+TmF2aWdhdGlvbiBUaW1pb
mcgQnJlYWtkb3duPC90ZXh0PgogIDx0ZXh0IHg9IjQwMCIgeT0iNDgiIGZvbnQtZmFtaWx5PSJBcm
lhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIzY2NiIgdGV4dC1hbmNob3I9Im1
pZGRsZSI+QnJvd3NlciBtb25pdG9yIHBlcmZvcm1hbmNlIHBoYXNlcyAoVzNDIE5hdmlnYXRpb24g
VGltaW5nIEFQSSk8L3RleHQ+CgogIDwhLS0gVGltZWxpbmUgYmFzZSAtLT4KICA8bGluZSB4MT0iN
TAiIHkxPSIxMzAiIHgyPSI3NTAiIHkyPSIxMzAiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZH
RoPSIyIi8+CgogIDwhLS0gVGltZWxpbmUgc2VnbWVudHMgLSBwcm9wb3J0aW9uYWwgd2lkdGhzIC0
tPgogIDwhLS0gRE5TOiA1MC0xMTAgKDYwcHgsIH4xNW1zKSAtLT4KICA8cmVjdCB4PSI1MCIgeT0i
OTAiIHdpZHRoPSI2MCIgaGVpZ2h0PSI0MCIgcng9IjQiIGZpbGw9InVybCgjZG5zR3JhZCkiIGZpb
HRlcj0idXJsKCNuYXZTaGFkb3cpIi8+CiAgPHRleHQgeD0iODAiIHk9IjEwNSIgZm9udC1mYW1pbH
k9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZml
sbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkROUzwvdGV4dD4KICA8dGV4dCB4PSI4MCIg
eT0iMTIwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpb
Gw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTVtczwvdGV4dD
4KCiAgPCEtLSBDb25uZWN0OiAxMTAtMTgwICg3MHB4LCB+MjVtcykgLS0+CiAgPHJlY3QgeD0iMTE
yIiB5PSI5MCIgd2lkdGg9IjcwIiBoZWlnaHQ9IjQwIiByeD0iNCIgZmlsbD0idXJsKCNjb25uZWN0
R3JhZCkiIGZpbHRlcj0idXJsKCNuYXZTaGFkb3cpIi8+CiAgPHRleHQgeD0iMTQ3IiB5PSIxMDUiI
GZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaH
Q9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Db25uZWN0PC90ZXh0Pgo
gIDx0ZXh0IHg9IjE0NyIgeT0iMTIwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZv
bnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZ
GRsZSI+MjVtczwvdGV4dD4KCiAgPCEtLSBTU0w6IDE4MC0yNzAgKDkwcHgsIH40NW1zKSAtLT4KIC
```

```
A8cmVjdCB4PSIxODQiIHk9IjkwIiB3aWR0aD0iOTAiIGhlaWdodD0iNDAiIHJ4PSI0IiBmaWxsPSJ
1cmwoI3NzbEdyYWQpIiBmaWx0ZXI9InVybCgjbmF2U2hhZG93KSIvPgogIDx0ZXh0IHg9IjIyOSIg
eT0iMTA1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvb
nQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U1NML1RMUz
wvdGV4dD4KICA8dGV4dCB4PSIyMjkiIHk9IjEyMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXN
lcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5j
aG9yPSJtaWRkbGUiPjQ1bXM8L3RleHQ+CgogIDwhLS0gUmVxdWVzdDogMjcwLTM0MCAoNzBweCwgf
jMwbXMpIC0tPgogIDxyZWN0IHg9IjI3NiIgeT0iOTAiIHdpZHRoPSI3MCIgaGVpZ2h0PSI0MCIgcn
g9IjQiIGZpbGw9InVybCgjcmVxdWVzdEdyYWQpIiBmaWx0ZXI9InVybCgjbmF2U2hhZG93KSIvPgo
gIDx0ZXh0IHg9IjMxMSIgeT0iMTA1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZv
bnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9I
m1pZGRsZSI+UmVxdWVzdDwvdGV4dD4KICA8dGV4dCB4PSIzMTEiIHk9IjEyMCIgZm9udC1mYW1pbH
k9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU
1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjMwbXM8L3RleHQ+CgogIDwhLS0gUmVzcG9uc2U6
IDM0MC00NTAgKDExMHB4LCB+ODVtcykgLS0+CiAgPHJlY3QgeD0iMzQ4IiB5PSI5MCIgd2lkdGg9I
jExMCIgaGVpZ2h0PSI0MCIgcng9IjQiIGZpbGw9InVybCgjcmVzcG9uc2VHcmFkKSIgZmlsdGVyPS
J1cmwoI25hdlNoYWRvdykiLz4KICA8dGV4dCB4PSI0MDMiIHk9IjEwNSIgZm9udC1mYW1pbHk9IkF
yaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0i
d2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlJlc3BvbnNlPC90ZXh0PgogIDx0ZXh0IHg9IjQwM
yIgeT0iMTIwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIG
ZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+ODVtczwvdGV
4dD4KCiAgPCEtLSBET006IDQ1MC02MDAgKDE1MHB4LCB+MTUwbXMpIC0tPgogIDxyZWN0IHg9IjQ2
MCIgeT0iOTAiIHdpZHRoPSIxNTAiIGhlaWdodD0iNDAiIHJ4PSI0IiBmaWxsPSJ1cmwoI2RvbUdyY
WQpIiBmaWx0ZXI9InVybCgjbmF2U2hhZG93KSIvPgogIDx0ZXh0IHg9IjUzNSIgeT0iMTA1IiBmb2
50LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJ
ib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RE9NIFByb2Nlc3Npbmc8L3Rl
eHQ+CiAgPHRleHQgeD0iNTM1IiB5PSIxMjAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZ
iIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj
0ibWlkZGxlIj4xNTBtczwvdGV4dD4KCiAgPCEtLSBMb2FkOiA2MDAtNzQwICgxNDBweCwgfjEyMG1
zKSAtLT4KICA8cmVjdCB4PSI2MTIiIHk9IjkwIiB3aWR0aD0iMTMwIiBoZWlnaHQ9IjQwIiByeD0i
NCIgZmlsbD0idXJsKCNsb2FkR3JhZCkiIGZpbHRlcj0idXJsKCNuYXZTaGFkb3cpIi8+CiAgPHRle
HQgeD0iNjc3IiB5PSIxMDUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaX
plPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGx
lIj5Mb2FkIEV2ZW50PC90ZXh0PgogIDx0ZXh0IHg9IjY3NyIgeT0iMTIwIiBmb250LWZhbWlseT0i
QXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsM
C45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+MTIwbXM8L3RleHQ+CgogIDwhLS0gVGltZSBtYXJrZX
JzIC0tPgogIDx0ZXh0IHg9IjUwIiB5PSIxNTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJ
pZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4YiI+MG1zPC90ZXh0PgogIDx0ZXh0IHg9IjQw
MCIgeT0iMTU1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiI
GZpbGw9IiM2NDc0OGIiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjI1MG1zPC90ZXh0PgogIDx0ZXh0IH
g9Ijc0MiIgeT0iMTU1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0
iMTAiIGZpbGw9IiM2NDc0OGIiIHRleHQtYW5jaG9yPSJlbmQiPjQ3MG1zPC90ZXh0PgoKICA8IS0t
IExlZ2VuZCAtLT4KICA8cmVjdCB4PSI1MCIgeT0iMTc1IiB3aWR0aD0iNzAwIiBoZWlnaHQ9IjYwI
iByeD0iNiIgZmlsbD0iI2ZmZiIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4KIC
A8dGV4dCB4PSI0MDAiIHk9IjE5NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb25
0LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNob3I9Im1p
ZGRsZSI+S2V5IERRTCBGaWVsZHMgZm9yIEFuYWx5c2lzPC90ZXh0PgogIDx0ZXh0IHg9IjEyMCIge
T0iMjIwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3ND
```

hiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5kbnNfdGltZTwvdGV4dD4KICA8dGV4dCB4PSIyMzAiIHk
9IjIyMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4
YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Y29ubmVjdF90aW1lPC90ZXh0PgogIDx0ZXh0IHg9IjM0M
CIgeT0iMjIwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNj
Q3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5zc2xfdGltZTwvdGV4dD4KICA8dGV4dCB4PSI0NTA
iIHk9IjIyMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0
NzQ4YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+dHRmYjwvdGV4dD4KICA8dGV4dCB4PSI1NjAiIHk9I
jIyMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4Yi
IgdGV4dC1hbmNob3I9Im1pZGRsZSI+ZG9tX2ludGVyYWN0aXZlPC90ZXh0PgogIDx0ZXh0IHg9IjY
4MCIgeT0iMjIwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIj
NjQ3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5sb2FkX2V2ZW50X2VuZDwvdGV4dD4KPC9zdmc+C
g==)

~~~
### Key Performance Metrics

| Metric | Description | Good | Needs Work |
|--------|-------------|------|------------|
| **First Contentful Paint** | First content rendered | < 1.8s | > 3.0s |
| **Largest Contentful Paint** | Largest element rendered | < 2.5s | > 4.0s |
| **Time to Interactive** | Page fully interactive | < 3.8s | > 7.3s |
| **Total Blocking Time** | Main thread blocked | < 200ms | > 600ms |
| **Cumulative Layout Shift** | Visual stability | < 0.1 | > 0.25 |

> **Note:** Targets based on Google's Core Web Vitals. Actual acceptable
thresholds vary by application type and user expectations. Set baselines
based on your own monitoring data.

```dql
// Browser monitor performance breakdown
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| summarize {
    avg_dns_ms = avg(toDouble(dns_lookup_time)),
    avg_connect_ms = avg(toDouble(tcp_connect_time)),
    avg_ssl_ms = avg(toDouble(ssl_handshake_time)),
    avg_ttfb_ms = avg(toDouble(time_to_first_byte)),
    avg_dom_interactive_ms = avg(toDouble(dom_interactive_time)),
    avg_load_ms = avg(toDouble(total_duration)),
    executions = count()
  }, by: {synthetic_test_id}
| sort avg_load_ms desc
| limit 20
```

```dql
// Performance trend over time
fetch dt.synthetic.events, from: now() - 7d
~~~

~~~
| filter matchesValue(event_type, "*browser*")
| makeTimeseries {
    avg_response_time = avg(toDouble(total_duration)),
    p95_response_time = percentile(toDouble(total_duration), 95)
  }, interval: 1h
```

## 5. Validation and Assertions

### Content Validation

| Validation Type | Description | Example |
|-----------------|-------------|----------|
| **Text Present** | Page contains text | "Welcome" |
| **Text Absent** | Page doesn't contain | "Error" |
| **Regex Match** | Pattern matching | `Order #\d{6}` |
| **Element Exists** | DOM element present | `#success-message` |
| **Element Content** | Element has text | Button says "Submit" |

### HTTP Validation

| Check | Default Behavior | Customization |
|-------|------------------|---------------|
| Status Code | Fails on 4xx/5xx (400-599) | Can configure to ignore specific codes |
| Response Body | Content validation | Text/regex matching |
| Screenshots | Captured on success/failure | Automatic |

> **Note:** Response size validation and header validation are not available as built-in options. Use content validation or JavaScript steps for advanced checks.

### Visual Validation

- Automatic screenshots on success/failure
- Visual comparison (pixel diff)
- Layout validation

```dql
// Failed browser monitor executions
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| filter execution_success == false
| fields timestamp,
         monitor = synthetic_test_id,
         location = synthetic_location_id,
         error = error_message
| sort timestamp desc
~~~

~~~
| limit 50
```

## 6. Analyzing Browser Results

```dql
// Browser monitor availability by location
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| summarize {
    total = count(),
    successful = countIf(execution_success == true),
    failed = countIf(execution_success == false)
  }, by: {synthetic_test_id, synthetic_location_id}
| fieldsAdd availability_pct = round((successful * 100.0) / total, decimals: 2)
| sort availability_pct asc
| limit 30
```

```dql
// Response time distribution by location
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| filter execution_success == true
| summarize {
    min_ms = min(toDouble(total_duration)),
    avg_ms = avg(toDouble(total_duration)),
    p50_ms = percentile(toDouble(total_duration), 50),
    p95_ms = percentile(toDouble(total_duration), 95),
    max_ms = max(toDouble(total_duration)),
    executions = count()
  }, by: {synthetic_location_id}
| sort avg_ms desc
| limit 20
```

```dql
// Slowest page loads (outliers)
fetch dt.synthetic.events, from: now() - 24h
| filter matchesValue(event_type, "*browser*")
| filter execution_success == true
| filter toDouble(total_duration) > 5000  // > 5 seconds
| fields timestamp,
         monitor = synthetic_test_id,
         location = synthetic_location_id,
         response_time_ms = toDouble(total_duration)
| sort response_time_ms desc
~~~

~~~~
| limit 20
```

---

## Summary

In this notebook, you learned:

✅ **Browser monitor types** - Single-URL vs clickpath monitors
✅ **Creating monitors** - URL configuration, viewports, locations
✅ **Clickpath automation** - Recording, actions, selectors
✅ **Performance metrics** - W3C timing, Core Web Vitals
✅ **Validation** - Content, HTTP, and visual checks
✅ **Analysis queries** - Availability, response times, failures

---

## Next Steps

Continue to **SYNTH-03: HTTP Monitors** to learn about lightweight API
monitoring.

---

## References

- [Browser Monitors](https://docs.dynatrace.com/docs/platform-
modules/digital-experience/synthetic-monitoring/browser-monitors)
- [Browser Clickpaths](https://docs.dynatrace.com/docs/platform-
modules/digital-experience/synthetic-monitoring/browser-monitors/browser-
clickpaths)
- [Synthetic Recorder](https://docs.dynatrace.com/docs/platform-
modules/digital-experience/synthetic-monitoring/browser-monitors/record-
browser-clickpath)
- [Performance Metrics](https://docs.dynatrace.com/docs/platform-
modules/digital-experience/synthetic-monitoring/analysis-and-
alerting/analyze-synthetic-monitors)
~~~~