



# Webサイト上の ユーザー体験観測の基本

NRU 300



# ウェビナー 各種ご連絡

1. ご質問がある場合は、“Q&A”からご入力ください。



① 画面下  
「Q&A」をクリック！

こちらにご質問をご記入し、  
「送信」をクリックしてください！



②

1. 本日の資料はこの後“チャット”でURLを共有します。アクセスできない場合は、“Q&A”よりお名前とメールアドレスをご連絡ください。

# Safe Harbor

This presentation and the information herein (including any information that may be incorporated by reference) is provided for informational purposes only and should not be construed as an offer, commitment, promise or obligation on behalf of New Relic, Inc. ("New Relic") to sell securities or deliver any product, material, code, functionality, or other feature. Any information provided hereby is proprietary to New Relic and may not be replicated or disclosed without New Relic's express written permission.

Such information may contain forward-looking statements within the meaning of federal securities laws. Any statement that is not a historical fact or refers to expectations, projections, future plans, objectives, estimates, goals, or other characterizations of future events is a forward-looking statement. These forward-looking statements can often be identified as such because the context of the statement will include words such as "believes," "anticipates," "expects" or words of similar import.

Actual results may differ materially from those expressed in these forward-looking statements, which speak only as of the date hereof, and are subject to change at any time without notice. Existing and prospective investors, customers and other third parties transacting business with New Relic are cautioned not to place undue reliance on this forward-looking information. The achievement or success of the matters covered by such forward-looking statements are based on New Relic's current assumptions, expectations, and beliefs and are subject to substantial risks, uncertainties, assumptions, and changes in circumstances that may cause the actual results, performance, or achievements to differ materially from those expressed or implied in any forward-looking statement. Further information on factors that could affect such forward-looking statements is included in the filings New Relic makes with the SEC from time to time. Copies of these documents may be obtained by visiting New Relic's Investor Relations website at [ir.newrelic.com](http://ir.newrelic.com) or the SEC's website at [www.sec.gov](http://www.sec.gov).

New Relic assumes no obligation and does not intend to update these forward-looking statements, except as required by law. New Relic makes no warranties, expressed or implied, in this presentation or otherwise, with respect to the information provided.

# このウェビナーについて

## ● ゴール

- ユーザー視点の監視の重要性について理解する
- New Relic でどのようなユーザー体験観測ができるかを理解する
- New Relic でユーザー体験観測を設定できるようになる

## ● 対象者

- New Relic を知ってはいるがまだ触ったことがないエンジニアの方
- New RelicのAPMやInfrastructureは触ったことがあるが、BrowserやSyntheticsは使っていないエンジニアの方
- New Relicのフロントエンド監視、分析機能にご興味のある方

# アジェンダ

- 15:00-15:10** 座学(1)インフラ監視からユーザー視点の監視
- 15:10-15:20** 座学(2)ユーザー視点の観測の種類とできること
- 15:20-15:40** ハンズオン(0)環境を確認する  
ハンズオン(1)サービス死活監視を設定してみる
- 15:40-15:50** 座学(3)サービス死活監視から外形監視(Synthetics)へ
- 15:50-16:15** ハンズオン(2)Syntheticsで死活監視以外を設定する
- 16:15-16:20** 座学(4)リアルユーザーのWebパフォーマンスの分析
- 16:20-16:55** ハンズオン(3)リアルユーザーのWebパフォーマンスの分析
- 16:55-17:00** まとめ、アンケートご回答



New Relic  
技術統括コンサルティング部  
ソリューションコンサルタント

## 高木 憲弥

兵庫県尼崎市出身 広島在住

経歴：

- ・ 日系Sler

フルスタックエンジニア

金融顧客にて開発・運用を担当

システム障害発生時には火消し隊長

- ・ 得意領域

JVMのトラブルシューティング

クエリチューニング



# 座学(1) インフラ監視 からユーザー視点の 監視

# ユーザー視点の監視、どれくらいされていますか？

- 何もしていない
- Webサイトのping監視
- それ以外

# 監視のアンチパターンとデザインパターン

## アンチパターン：OSのメトリクスのアラート

” MySQLが継続的にCPU全部を使っていたとしても、レスポンスタイムが許容範囲に収まっていれば何も問題ありません。”

“OSのメトリクスは診断やパフォーマンス分析にとっては重要です。しかし99%の場合、これらのメトリクスは誰かを叩き起こすには値しません。”

出典: 入門監視 (Oreilly, 2019)



# 監視のアンチパターンとデザインパターン

## デザインパターン：ユーザー視点の監視

“ユーザーが気にするのは、アプリケーションが動いているかどうかです。”

“ユーザー視点優先の監視によって、個別のノードを気にすることから解放されます。”

出典: 入門監視 (Oreilly, 2019)

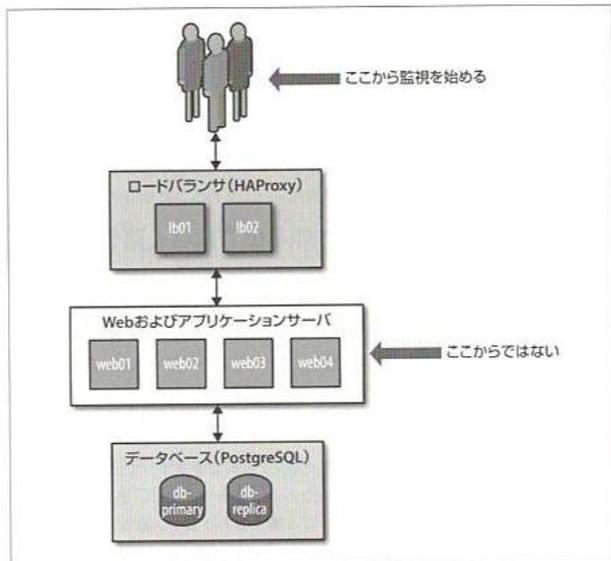


図2-1 できるだけユーザーに近いところから監視を始める

# Webアプリケーションにおけるユーザー体験の重要性

57%

ページ表示に3秒以上  
かかった場合にサイトから  
離脱する割合

引用: [How Site speed affects your Conversion rate](#)

80%

悪い体験をしたユーザーが  
他のサイトを利用する割合

引用: [How Site speed affects your Conversion rate](#)

+82%

サイト品質に問題があること  
によるサポート問い合わせ  
件数の増加率

引用: 2022 State of o11y

サイト品質の維持がユーザー体験に大きく影響を及ぼし、  
利用者の定着・維持の重要な要因になり得る  
→ユーザー体験を知り、改善をすることはビジネスにとっても重要！

# ユーザー視点の監視の難しさ

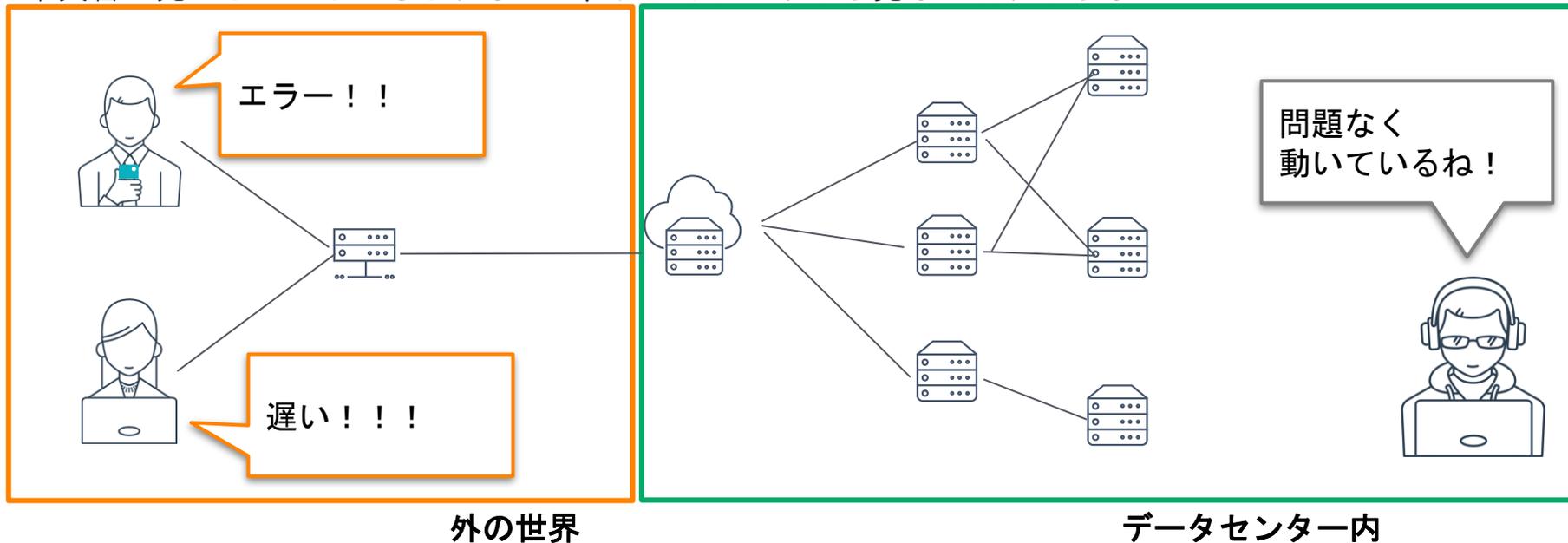
以下の2つの理由から、ユーザー側からデータを収集しないと把握ができない

- ① データセンターの外での変動要素
- ② 近年のWebアプリケーション構成の特性

# ① データセンターの外での変動要素

ユーザーにとってはリクエストを投げて画面に描画されるまでが応答時間

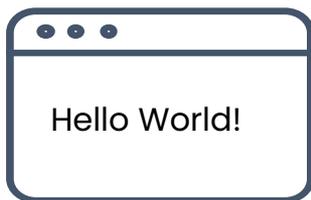
データセンター内で問題がなくても、外の世界で発生するエラーや遅延によってクライアント側では不具合が発生しているかもしれないが、クライアントから見ないとわからない



## ②近年のWebアプリケーション構成の特性

クライアントの情報をみないと、ユーザーの動きがわからない構成へと変遷している

静的画面



```
<html>
  <head />
  <body>
    Hello World!
  </body>
</html>
```

HTML  Backend

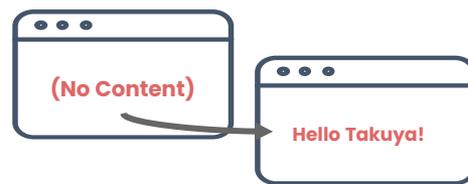
動的画面(サーバーサイド)



```
<html>
  <head />
  <body>
    Hello ${name}!
  </body>
</html>
```

HTML  Backend

動的画面(クライアントサイド)

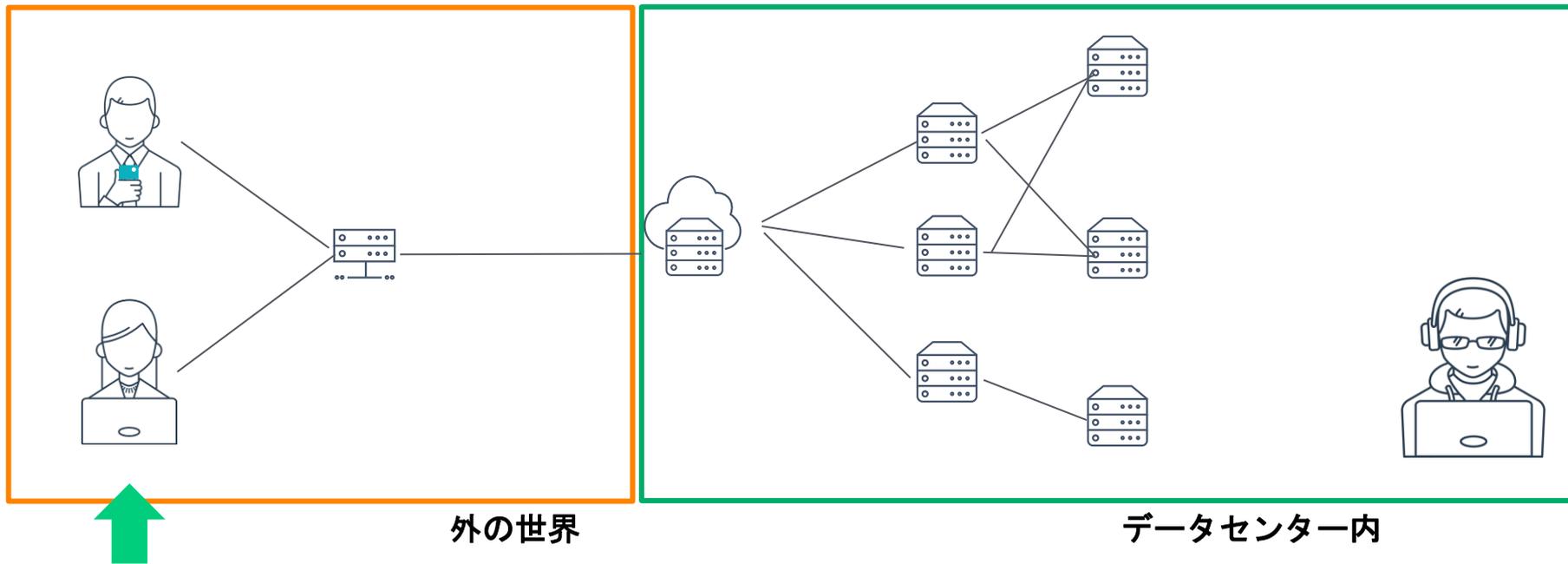


```
<html>
  <head />
  <body>
    <div id=content />
    <script>
      $('#content')=***
    </script>
  </body>
</html>
```

HTML  Backend

AJAX  Backend

# New Relicはユーザー側からデータを収集可能



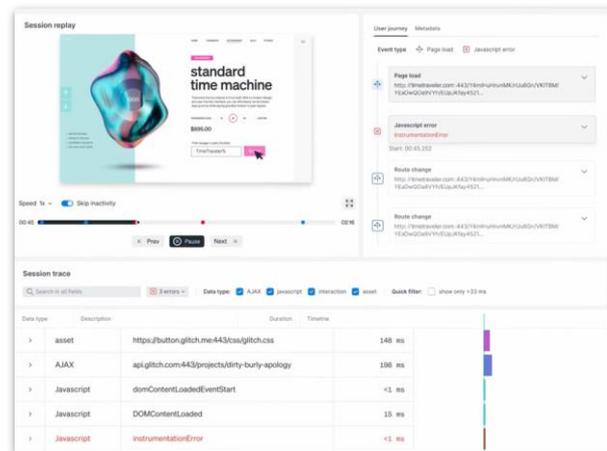
# さらにユーザー視点の”観測”を可能に

## 監視



ユーザーの手元でエラーが  
出ていることを検知する

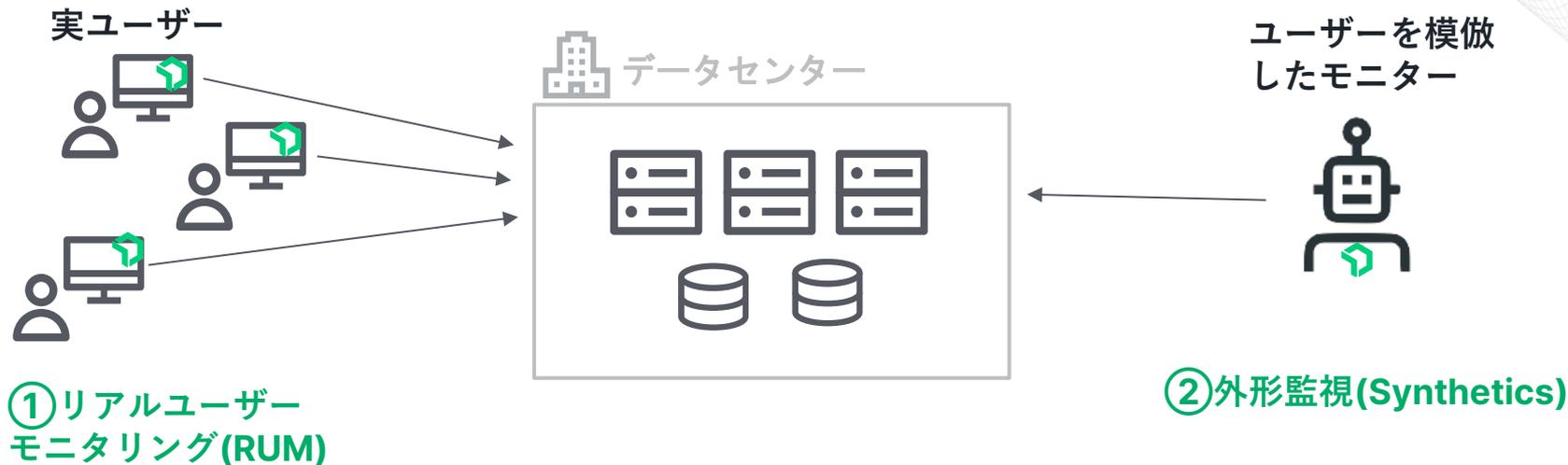
## 観測



エラーが出ていることに加え、  
なぜエラーが出たのか  
分析できる情報を得る

# 座学(2)ユーザー視点 の観測の種類とでき ること

# Webサイトにおける2種類のユーザー視点の観測



# Webサイトに2種類のユーザー視点の観測

## ①リアルユーザーモニタリング (RUM)



実際にWebサイトにアクセスしている利用ユーザーの、端末内のブラウザ上で、パフォーマンスやエラーを計測しNew Relicに送信

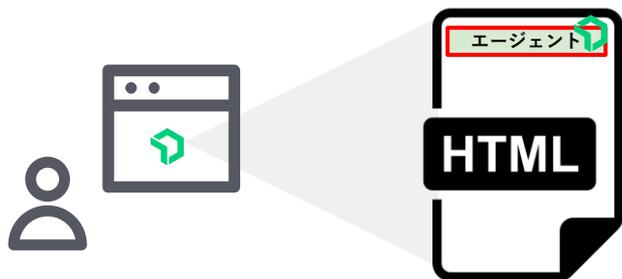
## ②外形監視(Synthetics)



ユーザーを模倣するモニターから定期的にWebサイトへアクセス、操作を実行し、その結果をNew Relicに送信

# ユーザー視点の観測の仕組み

## ①リアルユーザーモニタリング (RUM)



WebサイトのHTMLにエージェントとして動作するJavascriptスニペットを埋め込むことで、ユーザーのブラウザ上でHTMLが読み込まれた際にエージェントが起動し、計測を実施

## ②外形監視(Synthetics)



New Relicが提供する世界各地のモニターに対し、観測したいWebページのURLやユーザー操作(入力やクリック等)を設定し、そのモニターから見たアクセス結果を計測

# ユーザー視点の観測の活用ポイント

	リアルユーザーモニタリング(RUM)	外形監視(Synthetics)
特徴、優位性	<ul style="list-style-type: none"><li>● 実際のユーザー体験を知ることができる</li><li>● 環境(地理情報、ブラウザ種別等)に応じた分析が可能</li><li>● ユーザーID等を追加で付加することで、個別ユーザーの操作を追うことが可能</li></ul>	<ul style="list-style-type: none"><li>● ユーザーの操作を模倣し、問題なくサイトが稼働しているか確認することができる</li><li>● 証明書の期限切れなど目的に応じたシナリオに対応</li></ul>
注意点	<ul style="list-style-type: none"><li>● スクリプトの読み込みが前提となるのでページが表示されないような障害に気づくのは難しい(PV数の低下で間接的に把握する)</li><li>● ネットワーク環境等ユーザーごとのノイズがある</li></ul>	<ul style="list-style-type: none"><li>● 代表的なシナリオを決めて実装する必要があり、全てのユーザー操作シナリオを網羅させるのが難しい</li><li>● 実際のユーザー体験を測っているわけではない</li></ul>

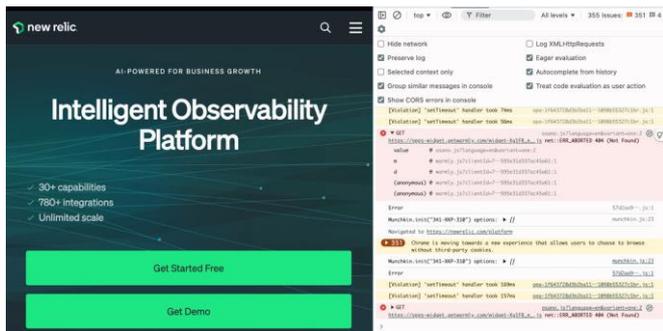
# ユーザー視点の観測で見られる情報

- 可用性
- パフォーマンス
- 個々のユーザーの属性

# 可用性

## ①リアルユーザーモニタリング (RUM)

- JavaScriptエラーによってWebページが正常に表示できない
- AJAX通信のエラー



## ②外形監視(Synthetics)

- Webページにアクセスできない
- Webページを構成するリソースがエラーを返す
- 特定の操作シナリオに失敗する (ECにおける決済等)



# パフォーマンス

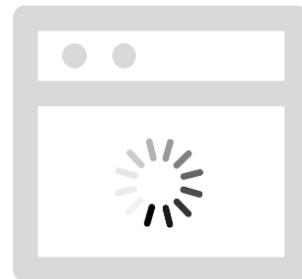
## ①リアルユーザーモニタリング (RUM)

- 全ユーザーが体験しているWebページの読み込み時間やCore web vitals(後述)
- AJAXのパフォーマンス



## ②外形監視(Synthetics)

- 重要なWebページの読み込み時間
- 上記Webページを構成するリソースごとの読み込み時間



# (参考) Browser Core Web Vitals

ページの読み込みパフォーマンス、インタラクティブ性、視覚的安定性に関する実際のユーザーエクスペリエンスを測定する一連の指標。GoogleがSEOを評価する際にも使われる。



**Largest Contentful Paint (LCP)** : 読み込みのパフォーマンスを測定します。優れたユーザーエクスペリエンスを提供するため、LCPはページの読み込みが最初に開始してから2.5秒以内に発生するようにします。

**Interaction to Next Paint (INP)** : インタラクティブ性を測定します。優れたユーザーエクスペリエンスを提供するには、ページのINPを200ミリ秒以下にする必要があります。\*過去には**FID**という指標を使っていました

**Cumulative Layout Shift (CLS)** : 視覚的な安定性を測定します。優れたユーザーエクスペリエンスを提供するには、ページのCLSを0.1以下にする必要があります。

参考 : <https://docs.newrelic.com/jp/docs/browser/browser-monitoring/getting-started/browser-summary-page/>  
<https://web.dev/articles/vitals?hl=ja#core-web-vitals>

# 個々のユーザーの属性

## ①リアルユーザーモニタリング (RUM)

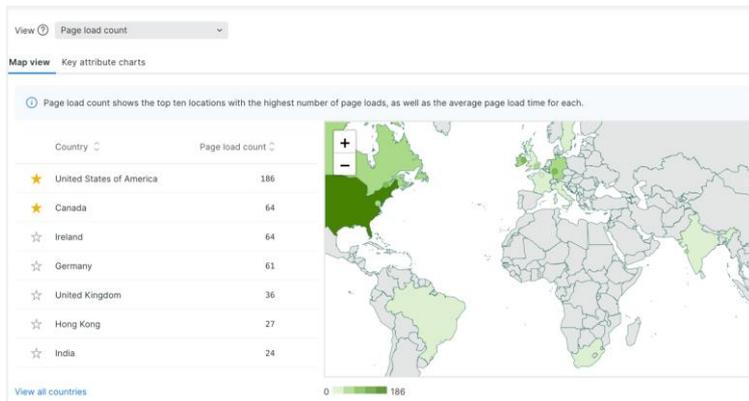
前述の可用性やパフォーマンスに関するデータに、以下のような情報を付加することができる

- 端末情報
- 地理情報
- ユーザーID等(要カスタマイズ)



## ②外形監視(Synthetics)

\*実ユーザーのアクセスではないため対象外





# ハンズオン(0) 環境を確認する



# 今回監視対象のサイト

[靴下屋さん](ECサイト)

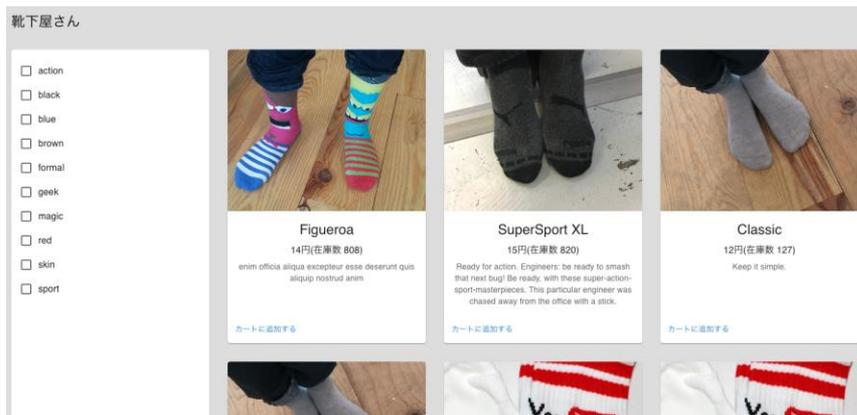
JAVA+Node.js(React)およびMySQLにより構築された靴下屋さんのECサイトをモニタリングしています

まずアクセスを試してみてください。

<https://front.demo.learn.nrkk.technology/>

User: [e100000@example.com](mailto:e100000@example.com)

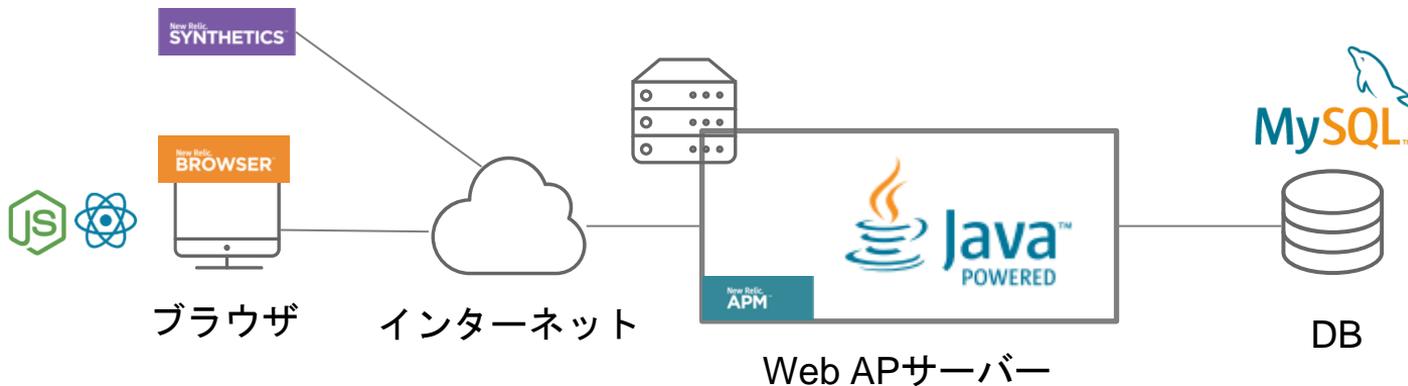
Password: password



# 今回の環境の監視構成

New Relicでモニタリング

- Synthetics（外形監視）
- Browser（フロントエンド）
- APM（バックエンド）



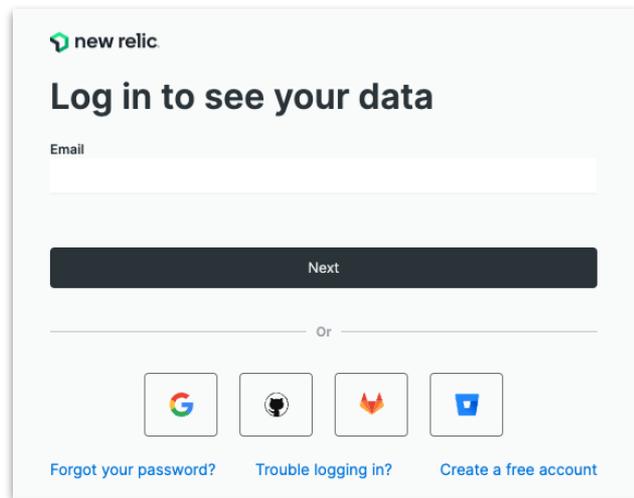
# ハンズオン環境について

## 1. New Relicをブラウザで表示

- New Relic : <https://one.newrelic.com>

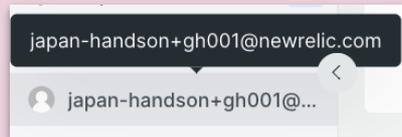
## 2. New Relicにログイン

- ユーザー: [japan-handson+gh001@newrelic.com](mailto:japan-handson+gh001@newrelic.com)
- パスワード: **rpDS!fg&hd**  
(アール、ピー、ディー、エス、!、エフ、ジー、&、エイチ、ディー)



### 下記にご注意ください：

- 普段New Relicをお使いの方は**セッションが残っている場合があります**
- ブラウザは下記のいずれかの**プライベートブラウジング**をご利用ください
  - Chrome : シークレットウィンドウ
  - Firefox : プライベートウィンドウ
  - Edge : InPrivate ウィンドウ
- ユーザー名が **“japan-handson+gh001@newrelic.com”** であることを**画面左下で確認**ください



# ハンズオン(0) UIの確認

●ログイン後の画面を確認ください

●右上のアカウント選択で、2つアカウントが表示されていることをご確認下さい

## Ogameday\_common

- アプリがあるアカウント

## Ogameday\_sandbox

- ハンズオンで設定を作成するアカウント

●APM、Browserアプリとして、front-serviceがあることをご確認ください。

The screenshot shows the New Relic 'All Entities' interface. The left sidebar contains navigation items: Quick Find, Integrations & Agents, All Capabilities, All Entities (selected), Dashboards, Logs, Query Your Data, APM & Services, Traces, Synthetic Monitoring, Alerts, Kubernetes, Browser, Mobile, Errors Inbox, Apps, Workloads, and Saved Views. The main content area is titled 'All Entities' and shows a search bar and filters. Below, there are sections for 'Your system' and 'Other entities'. The 'Services - APM' section contains a table with columns: Name, Account, Respons..., Through..., and Error rate. The 'front-service' entry is highlighted. The 'Services - OpenTelemetry' section is also visible. At the top right, an account selection dropdown is open, showing 'gameday\_sandbox' (4365310) and 'gameday\_common' (4097184) as options, with 'front-service' highlighted in the table below.

# ハンズオン(1) サービス死活監視を 設定してみる

# New RelicのSyntheticsでできること

Basic

Advanced

シンプル

ページの状態確認

ユーザー操作を再現

コードによる高度な監視

## Availability

Ping

サイトが正しい状態か(status 200を返すか)をチェックする

## Page link crawler

Broken Links Monitor

ページのリンク切れを監視する

## User step execution

Step Monitor

ユーザーの操作をステップによるノーコードで再現する

## User flow / functionality

Scripted Browser

スクリプトを使用し、より高度なユーザーのブラウザ操作を再現

## Page load performance

Simple browser

単一ページのパフォーマンスに関する詳細な情報を取得する

## SSL certificate expiration

Certificate Check

SSL証明書の有効期限と有効期間を確認する

## Endpoint availability

Scripted API

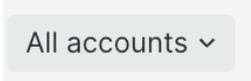
HTTPクライアントを利用してリモートエンドポイントを監視

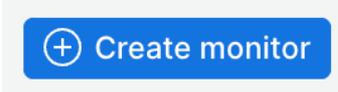
このハンズオンで扱う内容

# ハンズオン(1)-1.Pingモニターを作成する

## 事前準備: 外形監視設定画面への遷移

左メニューから  Synthetic Monitoring を選択

次に右上の  をクリックし、ドロップダウンより”gameday\_sandbox”を選択

その後右上に現れる  から外形監視の設定ができるようになります

# ハンズオン(1)-1.Pingモニターを作成する

## Pingモニターの作成

以下の要領でPingモニター(Availability)を作成してみましょう

モニター名: [お名前]\_pingtest

監視対象のURL: ご自身の会社のWebサイト、または任意のWebサイトを指定してください

監視間隔: 1分

Webページに含まれる文字列を検索する設定を入れてください(オプション)

監視元のロケーション: Tokyo, JP

# ハンズオン(1)-1. Pingモニターを作成する

## Pingモニター実行結果の確認

モニターの作成に成功すると、モニターの画面に遷移します。

1-2分で結果が表示されますので詳細を見てみましょう。

\*待っても画面が変わらない場合は、リロードしてみてください

- 左サブメニューにResultsという項目があるので、そこからモニターの実行結果を確認できます
- 複数行表示されている場合は、任意の行の"Timestamp"列のリンクをクリックしてください
- 以下の点を確認しましょう
  - モニターの実行に成功しているでしょうか、失敗しているでしょうか
  - 実行に掛かった時間はどれくらいでしょうか
  - 他にどのような情報が見えるでしょうか

# ハンズオン(1)-1. Pingモニターを作成する

## Pingモニターを失敗させる

左サブメニューのGeneralを押し、モニターの設定画面を開きます。

文字列を検索する設定(Text Validation)に、ページに存在しない文字列を入れて保存してください。

その後チェックの結果がどうなるか確認してください。また、直前のハンズオンと同様にチェック結果の詳細を見てみましょう。

- どのようなエラー情報が見られるでしょうか

# ハンズオン(1)-2.Simple Browserモニターを作成する

## Simple Browserモニターの作成

同じ要領でSimple Browserモニター(Page load performance)を作成してみましょう

モニター名: [お名前]\_simpletest

監視対象のURL: Pingモニターと同じURL

Browser: Chrome

Device Emulation: Desktop

## 監視間隔: 1日

※この設定を間違えないようにしてください。他の方がモニターを作れなくなる恐れがあります

それ以外はPingモニターと同じ設定(文字列検索設定はエラーが出る設定にしてください)

# 補足: Syntheticsのチェック回数制限について

New RelicのSyntheticsは月間のチェック回数上限があり、それを超過するとチェックができなくなります。

Pingモニターだけはチェック回数上限の対象外のため無制限に作成が可能です。

## Available synthetic monitor checks ⓘ

Your included checks	1,000,000
Checks scheduled this month	337,560
<b>Est. monthly checks for this monitor</b>	<b>30</b>

# ハンズオン(1)-2.Simple Browserモニターを作成する

## Simple Browserモニター実行結果の確認

Pingモニターと同様に、モニターの画面に遷移したら左サブメニューのResultsという項目からモニターの実行結果を確認しましょう。

- モニターの実行に成功しているでしょうか、失敗しているでしょうか
- 実行に掛かった時間はどれくらいでしょうか
- Ping Monitorと見比べて、収集できるデータの違いを確認しましょう

# ハンズオン1-(3). Synthetics Monitorの結果を元にしたアラート設定

## 【作成するアラート】

ハンズオン1-(1)で設定したPingモニター([お名前]\_pingtest)がチェックに失敗したらアラートを発報

## 【作成するための画面】

Pingモニターの確認画面を開き、右上の  **Manage alerts** アイコンを押すと、アラート作成画面に遷移します。

- Create Alert conditionを選択してアラートの作成を開始しましょう
- 2つの作成オプションが出てきますが、“Use guided mode”を使うと直感的に作成することができます

# ハンズオン(1) 手順・解説

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Pingモニターを作成

アカウント4365310  
を選択

Create monitorを  
押す

Synthetic  
Monitoringを  
選択

new relic

SYNTHETIC MONITORING

Account: 4365310 - gameday\_sandbox

Monitors

Search by entity name Entity Type = Synthetic monitor

Lookout view

	Name ↑	Monitor stat...	Success rat...	Locations fa...	Period	Monitor type	
☆	● test1_pingtest	Enabled	59.75%	1 / 1	60 s	Ping	...
☆	● test1_simpletest	Enabled	100%	-	24 h	Simple Brow...	...

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Pingモニターの作成(続き)

The screenshot shows the New Relic Synthetic Monitoring dashboard. The left sidebar contains navigation options such as 'Quick Find', 'Integrations & Agents', 'All Capabilities', 'All Entities', 'Dashboards', 'Logs', 'Query Your Data', 'APM & Services', 'Traces', 'Synthetic Monitoring' (highlighted), 'Alerts', 'Kubernetes', 'Browser', 'Mobile', 'Errors Inbox', 'Apps', 'Workloads', 'Saved Views', 'Help', and 'Add User'. The main content area is titled 'Synthetic Monitoring' and displays a 'Create monitor' wizard. The first step is '1. Select monitor type', with subsequent steps being '2. Configure monitor' and '3. Select locations'. An orange box highlights the 'Availability' monitor type, which is described as 'Ping' and 'Simple ping using cURL'. Other monitor types include 'SSL certificate expiration', 'Page link crawler', 'Page load performance', 'User flow / functionality', 'Endpoint availability', and 'User step execution'. The text 'Availability (ping) を選択' is overlaid on the 'Availability' monitor type.

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Synthetic Monitoring

### Create monitor

Ping

1. Select monitor type 2. **Configure monitor** 3. Select locations

Account: 4365310 - gameday\_sandbox ▾

Name (required)

nr\_pingtest

名前を入力

Tags

Add a new tag...

Enter tags in the form of "key:value" and

URL (required)

https://newrelic.com/jp

監視対象のURLを入力

Period

1 min ▾

Advanced options

Text validation (optional)

快適なデジタル顧客体験を。

ページに含まれる文字列を入力

987 characters left

ApdexTarget (optional)

7

Cancel

Select locations

次へ

## Pingモニターの作成(続き)

1. Select monitor type 2. Configure monitor 3. **Select locations**

To minimize false positives, we recommend choosing at least 3 locations, preferably from different regions.

Private locations (0)

Show or

Public locations (1)

test2

North America (0)

Columbus, OH, USA

Montreal, Québec, CA

Portland, OR, USA

San Francisco, CA, USA

Washington, DC, USA

South America (0)

São Paulo, BR

Europe (0)

Dublin, IE

Frankfurt, DE

London, England, UK

Milan, IT

Paris, FR

Stockholm, SE

Australia (0)

Sydney, AU

Asia (1)

Hong Kong, HK

Manama, BH

Mumbai, IN

Seoul, KR

Singapore, SG

Tokyo, JP

Africa (0)

Cape Town, ZA

Tokyo, JPを選択

Cancel

Save monitor

保存

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

Synthetic Monitoring / Synthetic monitors

nr\_pingtest ☆ Tags Metadata Workloads

Good

Since 30 minutes ago (GMT+9) Manage alerts

Summary

MONITOR

100% Success rate  
0/2 Checks failed  
0/1 Locations failing  
Everything's A-OK!  
No failures detected within the last 30 min

Activity stream

Monitor created 6:40am  
nr\_pingtest  
Create Synthetic Monitoring / Synthetic monitors

Good

nr\_pingtest ☆ Tags Metadata Workloads

Related

Summary

MONITOR

Results

Resources

TRIAGE

Diagnose

REPORTS

SLA

Service levels

SETTINGS

General

Alert conditions

MORE VIEWS

Add app

Change tracking

Events explorer

Help

Logs

Duration and availability of checks in the last 30 min

Tokyo, JP 0 failures / 2 checks ...

15.5 ms max  
14.4 ms median

Run check

January 23rd 6:40 AM January 23rd 6:40 AM

Average Median 95th Percentile

Narrow data to Search for any attribute or value.

Request & response time

5.94 ms 30 µs 2.23 ms ...  
Receive Send Wait

7 ms  
6 ms  
5 ms  
4 ms  
3 ms  
2 ms

Duration(s)

Metadata

Account

Account

Entity group

Monitor ID

Settings

General

Alert conditions

MORE VIEWS

Add app

Change tracking

Events explorer

Help

Logs

Resultsを選択

## Pingモニター実行結果の確認

Synthetic Monitoring / Synthetic monitors

nr\_pingtest ☆ Tags Metadata Workloads

Good

Since 30 minutes ago (GMT+9) Manage alerts

Summary

MONITOR

Results

Resources

TRIAGE

Diagnose

REPORTS

SLA

Service levels

SETTINGS

General

Alert conditions

MORE VIEWS

Add app

Change tracking

Events explorer

Help

Logs

Duration by location

Duration by location Network timings 18.2 ms average.duration

26 ms  
24 ms  
22 ms  
20 ms  
18 ms  
16 ms  
14 ms

6:40am 6:45am 6:50am 6:55am 7:00am 7:05am

Duration(s)

Tokyo, JP

Search results

Location = Tokyo, JP Status = Success, Failed

Result	Timestamp	Location	Executio...	Network ...	Respons...	Failure Message
SUCCESS	January 23, 2025...	Tokyo, JP	22 ms	19.1 ms	455.6 kB	-
SUCCESS	January 23, 2025...	Tokyo, JP	20 ms	17.2 ms	455.6 kB	-
SUCCESS	January 23, 2025...	Tokyo, JP	21 ms	18.4 ms	455.6 kB	-
SUCCESS	January 23, 2025...	Tokyo, JP	17 ms	14.8 ms	455.6 kB	-
SUCCESS	January 23, 2025...	Tokyo, JP	30 ms	22.9 ms	455.6 kB	-

任意のタイムスタンプを選択

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Pingモニター実行結果の確認(続き)

The screenshot shows the New Relic Synthetic Monitoring interface for a monitor named 'nr\_pingtest'. The main content area displays the following information:

- MONITOR:** Status is **Success**. Execution Duration is **22 ms**.
- Waterfall:** Shows a request for **445 kB** (Bytes transferred), **1** (Requests), and **19.1 ms** (Total time).
- Table:** A table with columns: Type, Request, Status, Size, Time, Timeline. The row shows: **html**, **https://newrelic.com/jp**, **200**, **445 kB**, **19.1 ms**, and a multi-colored bar representing the request timeline.

Annotations on the screenshot:

- 実行結果** (Execution Result) points to the 'Success' status.
- 実行に掛かった時間** (Time taken to execute) points to the '22 ms' execution duration.
- HTMLのリクエスト結果** (HTML request result) points to the 'html' row in the table.

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Pingモニターを失敗させる

Synthetic Monitoring / Synthetic monitors

nr\_pingtest ☆ Tags Metadata Workloads

User experience Good

Summary

MONITOR

Results

Resources

TRIAGE

Diagnose

REPORTS

SLA

Service levels

SETTINGS

**General**

Alert conditions

MORE VIEWS

Add app

Change tracking

Events explorer

Help

Logs

### Edit monitor

Ping

1. Configure monitor 2. Select locations

Monitor enabled

**Name (required)**  
nr\_pingtest

**Tags**  
Add a new tag...  
Enter tags in the form of "key:value" and press enter to add it to this monitor.

**URL (required)**  
https://newrelic.com/jp

**Period**  
1 min

Advanced options

**Text validation (optional)**  
存在しない文字列  
992 characters left

Generalを  
選択

Text validation に存在しない  
文字列を入れる

# ハンズオン(1)-1 ~ 解答 Pingモニターを作成する

## Pingモニターを失敗させる(続き)

**チェックが失敗している  
状況を確認できる**

61.3% Success rate  
12/30 Checks failed  
1/1 Locations failing  
Response Validation Error  
Response did not contain

実行結果

エラーの詳細

❗ Check failed with response validation error  
Error Response did not contain the expected string '存在しない文字列'

Type	Request	Status	Size	Time	Timeline
html	https://newrelic.com/jp	200	445 kB	15.7 ms	

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニターの作成

The screenshot shows the New Relic Synthetic Monitoring interface. The left sidebar contains navigation options, with 'Synthetic Monitoring' selected. The main content area is titled 'Synthetic Monitoring' and shows a 'Create monitor' wizard. The first step, 'Select monitor type', displays several options:

- Availability** (Ping): Simple ping using cURL.
- SSL certificate expiration** (Certificate Check): Check the expiration and validity period on an SSL certificate of a given domain.
- Page link crawler** (Broken Links Monitor): Test page links for success.
- Page load performance (Simple browser)** (Simple browser): Full page load of a given URL. Provides deep data insights like resource breakdowns and timelines. This option is highlighted with an orange box and text: **Page load performance (Simple browser) を選択**.
- User flow / functionality** (Scripted browser): Full scripted browser test.
- Endpoint availability** (Scripted API): Uses an HTTP client to monitor remote API endpoints (REST, healthchecks, etc).
- User step execution** (Step Monitor): Zero code monitor building.

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニターの作成(続き)

1. Select monitor type 2. **Configure monitor** 3. Select locations 4. Validation results

**Legacy runtime options removed**  
New monitors cannot be created using the Node 10 or Chrome 72 runtimes after August 26, 2024.  
[See our docs](#) [Upgrade Monitors](#)

Account: 4365310 - gameday\_sandbox ▾

**Name (required)**  
nr\_simpletest **名前を入力**

**URL (required)**  
https://newrelic.com/jp **監視対象のURLを入力**

**Period**  
1 day ▾ **1 dayを選択**

**Tags**  
Add a new tag...  
Enter tags in the form of "key:value" and press enter to add it to this monitor.

Advanced options

**Text validation (optional)**  
存在しない文字列 **存在しない文字列を入力**  
992 characters left

**ApdexTarget (optional)**  
7

Verify SSL

Enable screenshot on failure and in script

**次へ**

**Browser** (🗑)  
 Chrome **Chromeを選択**  
 Firefox **\*実行ブラウザを選べます**

**Device Emulation** (🗑)  
 Desktop (1366 × 768) **Desktopを選択**  
 Tablet landscape (1024 × 768)  
 Tablet portrait (768 × 1024)  
 Mobile landscape (800 × 400)  
 Mobile portrait (400 × 800) **\*エミュレーションする実行環境を選べます**

1. Select monitor type 2. Configure monitor 3. **Select locations** 4. Validation results

Private locations (0)  Show only

Public locations (1)

test2

North America (0)  Europe (0)  Asia (1)

Columbus, OH, USA  Dublin, IE  Hong Kong, HK

Montreal, Québec, CA  Frankfurt, DE  Manama, BH

Portland, OR, USA  London, England, UK  Mumbai, IN

San Francisco, CA, USA  Milan, IT  Seoul, KR

Washington, DC, USA  Paris, FR  Singapore, SG

South America (0)  Australia (0)  Africa (0)

São Paulo, BR  Sydney, AU  Cape Town, ZA

Tokyo, JP **Tokyo, JPを選択**

**Available synthetic monitor checks** (🗑)

Your included checks	1,000,000
Checks scheduled this month	359,160
Est. monthly checks for this monitor	30

**チェック数の上限を超えていないことを確認**

**保存**

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニター 実行結果の確認

The screenshot displays the New Relic Synthetic Monitoring interface for a monitor named 'nr\_simplestest'. The interface is divided into several sections:

- Summary:** Shows overall monitor status: 0% Success rate, 1/1 Checks failed, and 1/1 Locations failing. An 'Assertion Error' is noted: 'The page did not contain t...'. A 'Run check' button is visible.
- Activity stream:** Shows the monitor was created at 7:19am.
- Duration and availability of checks in the last 30 days:** A bar chart shows a single failure at January 23rd 7:20 AM.
- Performance timings:** A table lists various metrics: First Byte (20.1 ms), First Contentful Paint (1.59 s), Page Load (3.29 s), etc.
- Duration by location:** A line chart shows the duration of checks for Tokyo, JP from 6:55am to 7:20am.
- Related:** A sidebar menu shows the 'Results' tab is selected.
- Metadata:** Shows account information.
- Table:** A table at the bottom lists individual check results. One entry is highlighted with an orange box: 'January 23, 20...'. The failure message is 'AssertionError: The page did not cont...'. Other columns include Timestamp, Location (Tokyo, JP), Browser (Chrome 117), Executi... (9.92 s), Networ... (3.74 s), Respon... (2.8 MB), and Failure Message.

Annotations in orange text highlight the 'Results' tab in the sidebar and the 'January 23, 20...' timestamp in the table. A grey arrow points from the 'Results' tab to the table entry.

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニター 実行結果の確認(続き)

The screenshot shows the New Relic Synthetic Monitoring interface for a browser check. The status is "Failed" with 4 resources with errors and an execution duration of 9.92 s. The browser type is Chrome and the device type is Desktop. The error details indicate a "Check failed with assertion error" because the page did not contain the expected text "存在しない文字列". A screenshot of the page is displayed, showing a Japanese landing page for New Relic. The waterfall chart at the bottom shows various performance metrics.

実行結果

実行に掛かった時間 (ping monitorと比較しましょう)

チェックに失敗したときだけスクリーンショットが表示される

January 23, 2025 at 7:20:11 AM | Tokyo, JP

**Failed** 4 Resources with errors 9.92 s Execution Duration Chrome Browser Type Desktop Device Type

**Error details**

① Check failed with assertion error

Error The page did not contain the expected text: 存在しない文字列

**Screenshot**

**Waterfall** Script Log (10) Browser Log (4)

https://newrelic.com/jp

Metric	Value
6.63 MB Bytes transferred	96 Requests
3.74 s Total time	15 Long running tasks
1.59 s First paint	1.59 s First contentful paint
1.07 s DOM content loaded	3.29 s Page loaded

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニター 実行結果の確認(続き)

The screenshot displays the New Relic Synthetic Monitoring interface for a monitor named 'nr\_simpletest'. The top section shows various performance metrics: 6.63 MB Bytes transferred, 96 Requests, 3.74 s Total time, 15 Long running tasks, 1.59 s First paint, 1.59 s First contentful paint, 1.07 s DOM content loaded, and 3.29 s Page. Below this is a table of request details.

Type	Request	Status	Size	Time	Timeline
html	https://newrelic.com/jp	200	445 kB	31.4 ms	
font	https://newrelic.com/themes/...	200	32.6 kB	101 ms	
font	https://newrelic.com/themes/...	200	35 kB	100 ms	
font	https://newrelic.com/themes/...	200	37 kB	102 ms	
css	https://newrelic.com/sites/def...	200	216 kB	102 ms	
css	https://newrelic.com/sites/def...	200	423 kB	118 ms	
javascript	https://cmp.osano.com/AzZV...	200	266 kB	136 ms	
javascript	https://newrelic.com/sites/def...	200	5.84 kB	98 ms	
javascript	https://newrelic.com/libraries/...	200	1.79 kB	99.2 ms	

Annotations on the image:

- An orange box labeled "各種パフォーマンス指標" (Various performance metrics) points to the top summary metrics.
- An orange box labeled "ページロード中のログ" (Log during page loading) points to the "Browser Log (4)" section, which shows a waterfall of requests with error messages like "Failed to load resource: net:ERR\_BLOCKED\_BY\_CLIENT".
- An orange box labeled "ページの構成要素のロード時間やStatus Code" (Load time or Status Code of page components) points to the table of request details.

# ハンズオン(1)-2 ~ 解答 Simple Browser モニター を作成する

## Simple Browser モニター 実行結果の確認(続き)

[Ping モニターとの違い]

	Ping	Simple Browser
実行時間	HTMLファイルがロードされるまで	ページの構成要素(CSS, JS, image 等)がすべてロードされるまで
チェック失敗時	失敗とのみ表示	失敗の際の画面キャプチャも表示
詳細情報	HTMLファイルの読み込み時間、ステータスコードのみ	ページの構成要素ごとの読み込み時間、ステータスコード ページ読み込み時のログ

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定

Synthetic Monitoring / Synthetic monitors

nr\_pingtest ☆ Tags Metadata Workloads

Good

Since 24 hours ago (GMT+9) Manage alerts

Summary

0% Success rate 30/30 Checks failed 1/1 Locations failing Response Validation Error

Activity stream

Monitor updated 7:21am

Monitor created

Related entities

Alert Policies

Manage alertsを選択

Alerts

Account: 4365310 - gameday\_sandbox

Alert Policies

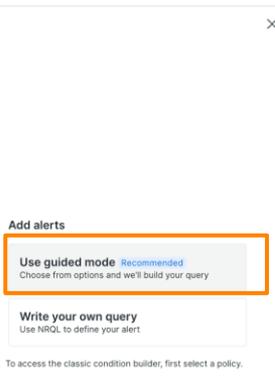
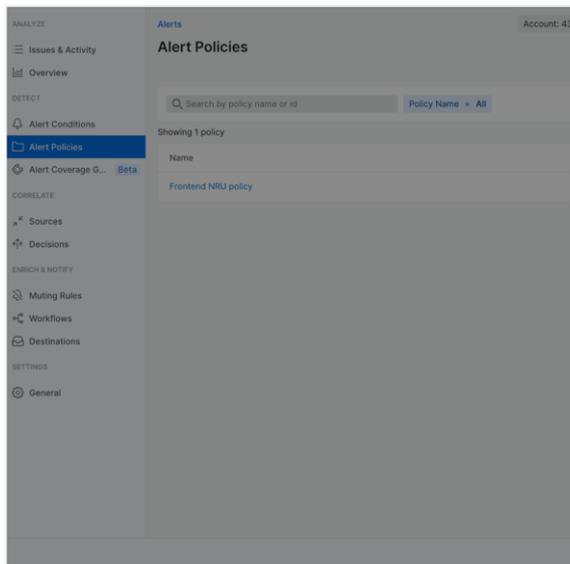
+ New alert condition + New alert policy

Showing 1 policy

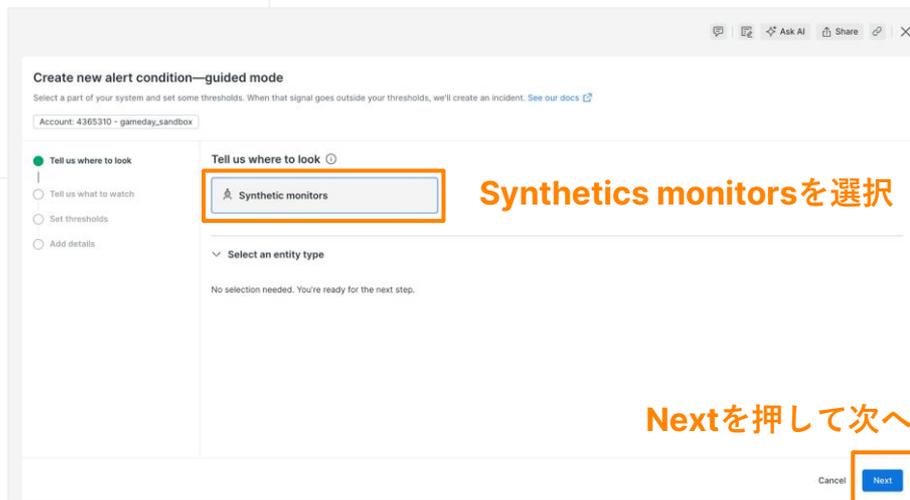
Name	Open issues	# of conditions	
Frontend NRU policy	1	1	...

New Alert Conditionを選択

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定



Use guided modeを選択



Synthetics monitorsを選択

Nextを押して次へ

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定

The screenshot displays the Synthetics Monitor configuration interface. On the left, a sidebar contains four steps: 'Tell us where to look' (completed), 'Tell us what to watch' (active), 'Set thresholds', and 'Add details'. The main area is titled 'Tell us what to watch' and includes a search bar and a list of entities. The 'nr\_pingtest' entity is selected and highlighted with an orange box, with the Japanese text 'ご自身のpingモニター' (Your own ping monitor) overlaid. Below the entity list, the 'Select a metric to monitor' section shows 'Median duration (s)' with 'Failures' selected in a dropdown menu, also highlighted with an orange box and the text 'failureを選択' (Select failure). At the bottom right, there are 'Cancel' and 'Next' buttons.

Tell us what to watch

Select the entities to watch (max 20)

Search entities by name or attributes. If you create new entities with these attributes, we'll watch those as well.

All Selected 1

Filter by name or tags

- Entities
- test\_pingtest
- nr\_pingtest **ご自身のpingモニター**
- test1\_simpletest
- nr\_simpletest

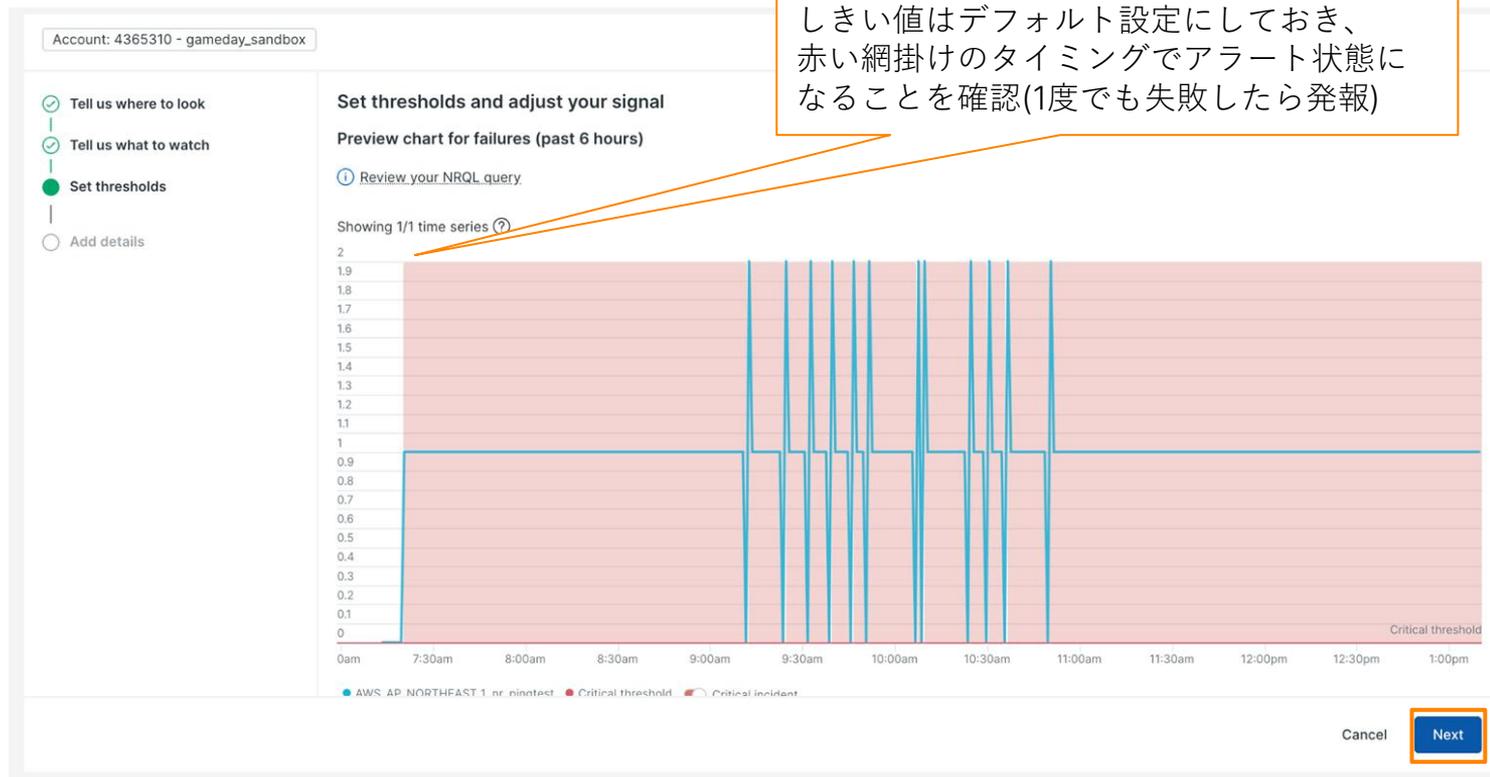
Select a metric to monitor

Median duration (s) Failures **failureを選択**

Cancel Next

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定

しきい値はデフォルト設定にしておき、赤い網掛けのタイミングでアラート状態になることを確認(1度でも失敗したら発報)



Cancel

Next

Nextを押して次へ

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定

Select a part of your system and set some thresholds. When that signal goes outside your thresholds, we'll create an incident. [See our docs](#)

Account: 4365310 - gameday\_sandbox

- ✓ Tell us where to look
- ✓ Tell us what to watch
- ✓ Set thresholds
- Add details

### Add details

Name your alert condition \*

nr\_pingmonitorアラート **任意の名前を入力**

Connect this condition to a policy \*

Existing policy  New policy

Policy \* Frontend NRU policy - ID: 5880932 **Existing policyを選択し、"Frontend NRU policy"を選択**

Close open incidents after 3 days

#### Customize incidents from this condition

Title template ⓘ

Enter a title to use for all incidents

Description template ⓘ

Enter a description to use for all incidents

Cancel <> View as code **Save condition**

**Save conditionを  
押して保存**

# ハンズオン(1)-3 ~ 解答 Synthetics Monitorの結果を元にしたアラート設定

## nr\_pingmonitorアラート

Select a part of your system and set some thresholds. When that signal goes outside your thresholds, we'll create an incident. [See our docs](#)

ID: 47577177 | Account: 4365310 - gameday\_sandbox | Policy: 5880932 - Frontend NRU policy

- ✔ Tell us where to look
- ✔ Tell us what to watch
- ✔ Set thresholds
- ✔ Add details

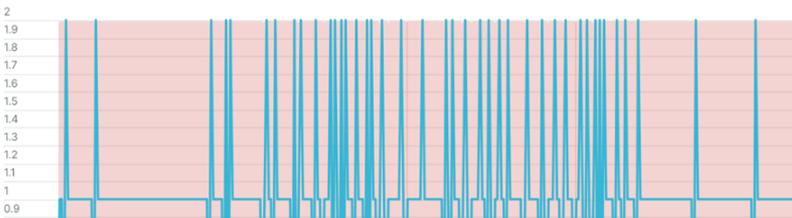
✔ **Alert condition enabled**  
To get notifications, make sure this alert condition's policy is connected to a workflow with notifications set up.

### Summary

**Query**

```
SELECT filter(count(*), WHERE result = 'FAILED') AS 'Failures' FROM SyntheticCheck WHERE entityGuid IN ('NDM2NTMxMHxTWU5USHxNT05JVE9SfDVl0TIzNjJhLTM3MDMtNDkzYS1iMjQ0LTQ5NGZiZTliOTNm0A') AND NOT isMuted FACET location, monitorName
```

Showing 1/1 time series ?



</> View as code | Edit alert condition | View policy | **Close**

アラートが作成できたことを確認してクローズ

# 座学(3)サービス死活 監視から外形監視 (Synthetics)へ

# 従来のWebアプリケーションの死活監視

Webアプリケーションが動いているかどうかをシステム外部から確認する取り組みはよく行われています。特にシンプルな、HTTPのリクエストが成功するかを確認する手法が取られてきました。

## HTTPリクエストの確認

- 特定ページのリクエストの成功(レスポンスコード200)
- 返ってきた内容に、特定の文字列が入っているか
- 返ってくるまでの時間が想定内か

### メリット

- 人が目で見て確認しなくても良い
- 問題があったことを実ユーザーと同じ程度のタイミングで気づける
- サービスの外部から、アプリケーション、DBの疎通までできる範囲で確認可能

# 死活監視に加えて求められている確認観点

近年のWebアプリケーションの構成では、HTTPのリクエスト確認だけでは確認できない観点が増えています。監視の高度化でまず初めに挙げられる観点が3つあります。

## 確認観点

- SSL証明書期限切れ
  - SSL必須化、SSLの有効期限短縮の風潮
- Javascript実行を伴う画面の描画
  - Javascriptが動かないと画面が描画されない。ユーザーにとっては動いていないと同じ
- 最低限のシナリオテスト
  - ログインしないと動いているかわからないサービスの確認

# SSL証明書期限切れ

## 課題

Webサービスを提供する上で現在では必須であり、適切に更新されるべきなのですが、作業が必要かつ簡単ではない状況もあります。

期限切れまでに、余裕を持って作業できるように準備することや、作業後に問題なく設定できたことを確認しなければいけません。

## New Relicでできること

- SSL証明書を確認したいドメインと何日前に確認したいかを設定するだけで、確認したい期日になったら通知を飛ばせるようになります。
- 複数設定できるので、確認したいドメインが多くても一元管理可能です。
- 証明書が切れるまであと何日あるかというデータを使って、可視化もできます。

# SSL証明書期限切れ

SSL証明書の更新が行われないと、ユーザーがWebサイトにアクセスできなくなります。大規模サービスで問題が発生したケースもあり、決して他人事ではありません。



ITmedia NEWS > 製品動向 > 「Microsoft Teams」に2時間ほどアクセスできない障...

## 「Microsoft Teams」に2時間ほどアクセスできない障害発生 原因はサーバ証明書の更新し忘れ

🕒 2020年02月06日 10時56分 公開 [新野淳一, ITmedia]

🖨️ 印刷 🗕 見る 📱 Share 📊 12 📧 0

PR PC画面の盗み見を防ぐ！ プライバシースクリーン搭載の新型PCを見る

PR コロナ禍のテレワーク、セキュリティ対策は大丈夫？ 専門家がポイントを指南

この記事は新野淳一氏のブログ「Publickey」に掲載された「[マイクロソフトのクラウドサービス「Microsoft Teams」がサーバ証明書を更新し忘れ、2時間のあいだユーザーからアクセスできなくなる障害発生](#)」（2020年2月6日掲載）を、ITmedia NEWS編集部で一部編集し、転載したものです。

マイクロソフトがチャットサービスとして提供する「Microsoft Teams」は、急速

参考：[「Microsoft Teams」に2時間ほどアクセスできない障害発生 原因はサーバ証明書の更新し忘れ \(ITmedia\)](#)

# Javascript実行を伴う画面の描画

## 課題

最近だと、リッチでユーザー体験の良いWebサービスを提供しようと、Single Page Applicationの採用が広がっています。ただし、構成によっては、Javascriptを実行しないと画面描画できません。

また、APIにはエラーがなくUI上だけの問題の場合では、「画面が真っ白」という状態に気付けないこともあります。

## New Relicでできること

- URLを設定し表示されていて欲しい要素や文字列をノーコードで設定します。
- 実行結果をスクリーンショット付きで確認できるようになります。
- 問題が発生して意図した動作にならなかったことを通知で受け取ることができます。

# 最低限のシナリオテスト

## 課題

ログインした後に確認できる画面や、サービス上重要な操作ができることを確認したいことがあります。Javascript実行を伴う画面と同様、実際に動かしてみて問題がなければWebサービスが動いていると判断するケースが増えています。

## New Relicでできること

- ボタンのクリックやテキスト入力など画面での操作をシミュレーションできます。
- 複雑な操作やチェックを行いたい場合はコーディングもできます。

# New RelicのSyntheticsでできること

Basic

Advanced

シンプル

ページの状態確認

ユーザー操作を再現

コードによる高度な監視

## Availability

Ping

サイトが正しい状態か(status 200を返すか)をチェックする

## Page link crawler

Broken Links Monitor

ページのリンク切れを監視する

## User step execution

Step Monitor

ユーザーの操作をステップによるノーコードで再現する

## User flow / functionality

Scripted Browser

スクリプトを使用し、より高度なユーザーのブラウザ操作を再現

## Page load performance

Simple browser

単一ページのパフォーマンスに関する詳細な情報を取得する

## SSL certificate expiration

Certificate Check

SSL証明書の有効期限と有効期間を確認する

## Endpoint availability

Scripted API

HTTPクライアントを利用してリモートエンドポイントを監視

# 死活監視に加えて求められている確認観点

近年のWebアプリケーションの構成では、HTTPのリクエスト確認だけでは確認できない観点が増えています。監視の高度化でまず初めに挙げられる観点が3つあります。

## 確認観点

- SSL証明書期限切れ
  - SSL必須化、SSLの有効期限短縮の風潮
- Javascript実行を伴う画面の描画
  - Javascriptが動かないと画面が描画されない。ユーザーにとっては動いていないと同じ
- 最低限のシナリオテスト
  - ログインしないと動いて

### SSL certificate expiration

Certificate Check

SSL証明書の有効期限と有効期間を確認する

### User step execution

Step Monitor

ユーザーの操作をステップによるノーコードで再現する

### User flow / functionality

Scripted Browser

スクリプトを使用し、より高度なユーザーのブラウザ操作を再現

# ハンズオン

## (2) Syntheticsで死 活監視以外を設定す る

# ハンズオン(2)-1.Certificate Checkモニターを作成する

## Certificate Checkモニターの作成

以下の要領でCertificate Checkモニターを作成してみましょう

モニター名: [お名前]\_certtest

監視対象のドメイン: front.demo.learn.nrkk.technology

\*要確認

監視間隔: 1日

期限までの日数: 30日

Runtime: Node \*\* (選択できるバージョン)

# SSL証明書期限切れ

## SSL certificate expiration

Certificate Check

SSL証明書の有効期限と有効期間を確認する



更新期限が近づいています。  
更新しましょう。

30日前

有効期限

# ハンズオン(2)-2.Stepモニターを作成する

## Stepモニターの作成

以下の要領で、URLのみ指定するStepモニターを作成してみましょう。リクエスト成功したWebアプリのスクリーンショットが取得されていることも確認しましょう。

モニター名: [お名前]\_steptest

監視対象のURL: <https://newrelic.com>

監視間隔: 1日

利用ブラウザ: Chrome

メモ: Simple Browserとの違い

Simple Browserは画面描画を行いますが、成功時にはスクリーンショットを取得しません。

# Javascript実行を伴う画面の描画

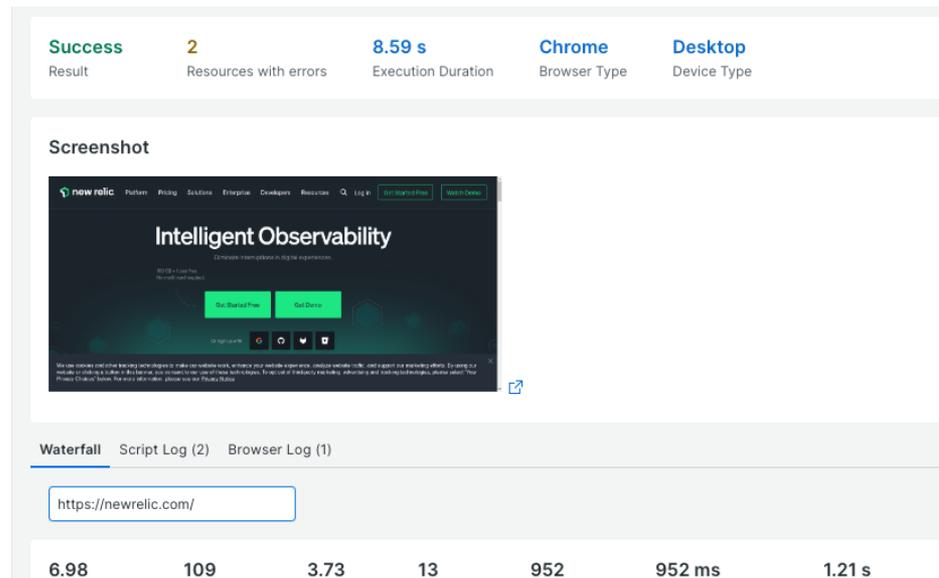
## User step execution

Step Monitor

ユーザーの操作をステップによるノーコードで再現する

設定内容例

開くURL : <https://newrelic.com>



# ハンズオン(2)-3.Scripted Browserモニターを作成する

## Scripted Browserモニターの作成

サンプルスクリプトを使って、Scripted Browserモニターを作成しましょう。リクエスト成功したWebアプリのスクリーンショットが取得されていることも確認しましょう。

モニター名: [お名前]\_scripttest

サンプルスクリプト

監視間隔: 1日

利用ブラウザ: Chrome

```
$browser.get('https://front.demo.learn.nrkk.technology/login').then(async function(){
  const userNameInput = await $browser.findElement($driver.By.xpath('//*[@id="username"]'));
  await userNameInput.sendKeys("e100000@example.com");
  const passwordInput = await $browser.findElement($driver.By.xpath('//*[@id="password"]'));
  await passwordInput.sendKeys("password");
  const loginButton = await $browser.findElement($driver.By.xpath('/html/body/div/form/button'));
  await loginButton.click();
  const category1 = await
  $browser.findElement($driver.By.xpath('//*[@id="root"]/div/div[2]/div[1]/div/div/label[1]'));
  await category1.click();
});
```

メモ: Simple Browserとの違い

Simple Browserは画面描画を行いますが、成功時にはスクリーンショットを取得しません。

# Javascript実行を伴う画面の描画

## User flow / functionality

Scripted Browser

スクリプトを使用し、より高度なユーザーのブラウザ操作を再現

Please sign in


ログイン

靴下屋さん カート (0)

- action
- black
- blue
- brown
- formal
- geek
- magic
- red
- skin
- sport



**Figueroa**  
14円(在庫数 808)  
enim officia aliquam excepteur esse deserunt quis aliquip nostrud anim

[カートに追加する](#)



**SuperSport XL**  
15円(在庫数 820)  
Ready for action. Engineers: be ready to smash that next bug! Be ready, with these super-action-sport-masterpieces. This particular engineer was chased away from the office with a stick.

[カートに追加する](#)

# ハンズオン(2)

## 手順・解説

# ハンズオン(2)-1 ~ 解答 Certificate Checkモニターを作成する

## Certificate Checkモニター の作成

アカウント4365310  
を選択

Create monitorを  
押す

Synthetic  
Monitoringを  
選択

The screenshot shows the New Relic Synthetic Monitoring interface. The left sidebar has 'Synthetic Monitoring' highlighted. The main content area shows a list of monitors:

	Name ↑	Monitor stat...	Success rat...	Locations fa...	Period	Monitor type	
☆	● test1_pingtest	Enabled	59.75%	1 / 1	60 s	Ping	...
☆	● test1_simpletest	Enabled	100%	-	24 h	Simple Brow...	...

# ハンズオン(2)-1 ~ 解答 Certificate Check モニターを作成する

## Certificate Check モニターの作成(続き)

SYNTHETIC MONITORING

Synthetic Monitoring

Synthetic Monitoring

Create monitor

SSL certificate expiration (certificate Check) を選択

1. Select monitor type 2. Configure monitor 3. Select locations

<b>Availability</b> Ping Simple ping using cURL	<b>SSL certificate expiration</b> Certificate Check Check the expiration and validity period on an SSL certificate of a given domain.	<b>Page link crawler</b> Broken Links Monitor Test page links for success	<b>Page load performance</b> Simple browser Full page load of a given URL. Provides deep data insights like resource breakdowns and timelines.
<b>User flow / functionality</b> Scripted browser Full scripted browser test.	<b>Endpoint availability</b> Scripted API Uses an HTTP client to monitor remote API endpoints (REST, healthchecks, etc).	<b>User step execution</b> Step Monitor Zero code monitor building	

# ハンズオン(2)-1 ~ 解答 Certificate Checkモニターを作成する

## Certificate Checkモニターの作成(続き)

Account: 4365310 - gameday\_sandbox ▾

Name (required) **名前を入力**  
YOURNAME\_certtest

Runtime **Node 16.10.0 ▾**

Domain (required) **監視対象のドメインを入力**  
front.demo.learn.nrkk.technology

Tags  
Add a new tag  
Enter tags in the form of "key=value" and press enter

Days remaining until expiration (required)  
30 **期限までの日数を入力**

Period **1 day ▾** **1 dayを選択**

> Advanced options

Available synthetic monitor checks ⓘ

Your included checks	1,000,000
Checks scheduled this month	359,190
Est. monthly checks for this monitor	0

1. Select monitor type 2. Configure monitor 3. Select locations

ⓘ To minimize false positives, we recommend choosing at least 3 locations, preferably from different regions.  Private locations (0)  Show or hide private locations

Public locations (1)

test2

<input type="checkbox"/> North America (0)	<input type="checkbox"/> Europe (0)	<input checked="" type="checkbox"/> Asia (1)
<input type="checkbox"/> Columbus, OH, USA	<input type="checkbox"/> Dublin, IE	<input type="checkbox"/> Hong Kong, HK
<input type="checkbox"/> Montreal, Québec, CA	<input type="checkbox"/> Frankfurt, DE	<input type="checkbox"/> Manama, BH
<input type="checkbox"/> Portland, OR, USA	<input type="checkbox"/> London, England, UK	<input type="checkbox"/> Mumbai, IN
<input type="checkbox"/> San Francisco, CA, USA	<input type="checkbox"/> Milan, IT	<input type="checkbox"/> Seoul, KR
<input type="checkbox"/> Washington, DC, USA	<input type="checkbox"/> Paris, FR	<input type="checkbox"/> Singapore, SG
	<input type="checkbox"/> Stockholm, SE	<input checked="" type="checkbox"/> Tokyo, JP
<input type="checkbox"/> South America (0)	<input type="checkbox"/> Australia (0)	<input type="checkbox"/> Africa (0)
<input type="checkbox"/> São Paulo, BR	<input type="checkbox"/> Sydney, AU	<input type="checkbox"/> Cape Town, ZA

**Tokyo, JPを選択**

Validate Cancel **Select locations** **次へ**

Cancel **Save monitor** **保存**

# ハンズオン(2)-1 ~ 解答 Certificate Checkモニターを作成する

## Certificate Checkモニター実行結果の確認

Synthetic Monitoring / Synthetic monitors

YOURNAME\_certtest ☆ Tags Metadata Workloads

Good from Sep 19, 10:2

Summary

MONITOR

100% Success rate  
0/1 Checks failed  
0/1 Locations failing  
Everything's A-OK!  
No failures detected within the last 30 days

Results **Resultsを選択**

Resources

TRiage

Diagnose

REPORTS

SLA

Service levels

SETTINGS

General

Alert conditions

Duration and availability of checks in the last 30 days

Tokyo, JP 0 failures / 1 checks

579 ms max  
579 ms median

January 24th 9:23 AM January 24th 9:23 AM

Run check

Average Median 95th Percentile

Narrow data to Search for any attribute or value.

Resources

TRiage

Diagnose

REPORTS

SLA

Service levels

SETTINGS

General

Alert conditions

MORE VIEWS

Add app

Change tracking

Events explorer

Help

Logs

Duration by location

Duration by location Network timings

580 ms  
579.8 ms  
579.6 ms  
579.4 ms  
579.2 ms  
579 ms  
578.8 ms  
578.6 ms  
578.4 ms  
578.2 ms  
578 ms

8:25am 8:30am 8:35am 8:40am 8:45am 8:50am 8:55am 9:00am 9:05am 9:10am 9:15am 9:20am 9:25am

Tokyo, JP

Search results Location = Tokyo, JP Status = Success, Failed

Result	Timestamp	location	Executio...	Network...	Respons...	Failure Message
SUCCESS	January 24, 202...	tokyo, JP	3.24 s	579 ms	-1	-

任意のタイムスタンプを選択

# ハンズオン(2)-1 ~ 解答 Certificate Checkモニターを作成する

## Certificate Checkモニター実行結果の確認(続き)

**実行結果**

January 24, 2025 at 9:23:57 AM | Tokyo, JP

**Success** - 3.24 s  
Result Resources with errors Execution Duration

**Waterfall**

- 1 579 ms  
Bytes transferred Requests Total time

Search requests Type = application, css, ... and 7 more State = Success, Error

Type	Request	Status	Size	Time	Timeline
not provid...	SSL Network Duration	200	0 B	579 ms	

**実行に掛かった時間**

**SSLリクエスト結果**

# ハンズオン(2)-1 ~ 解答 Certificate Check モニターを作成する

## Certificate Check モニター 実行結果 SSL失効までの日数の確認方法

The screenshot shows the New Relic Events Explorer interface. On the left sidebar, the 'Events explorer' option is highlighted. In the main event list, 'SyntheticCheck' is selected. The event details on the right show an NRQL query and a JSON response. Annotations include:

- A callout box pointing to the 'SyntheticCheck' event type in the list: **SyntheticCheck を選択**
- A callout box pointing to the code editor icon in the event details toolbar: **JSON表示を選択**
- A callout box pointing to the 'websiteDaysUntilExpiration' field in the JSON response: **SSL失効までの日数**

```
NRQL SELECT * FROM SyntheticCheck WHERE entityGuid = 'NDM2NTMxMHxTWU5USHxNT05JVE9SfDljZjQ2M2NkLTIyMjUtNGY1ZS04Yjc2LTRkN2QyNzEwZDQ5NQ' SINCE 60 MINUTES AGO LIMIT 10
```

```
{
  "totalResponseBodySize": -1,
  "totalResponseHeaderSize": -1,
  "totalResponseSize": -2,
  "type": "SCRIPT_API",
  "typeLabel": "Scripted API",
  "userDaysUntilExpiration": 30,
  "websiteDaysUntilExpiration": 387
}
```

Events explorer を選択

# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニターの作成

アカウント4365310  
を選択

Synthetic  
Monitoringを  
選択

Account: 4365310 - gameday\_sandbox

SYNTHETIC MONITORING

Monitors

Search by entity name Entity Type = Synthetic monitor

Lookout view

	Name ↑	Monitor stat...	Success rat...	Locations fa...	Period	Monitor type	
☆	● test1_pingtest	Enabled	59.75%	1 / 1	60 s	Ping	...
☆	● test1_simpletest	Enabled	100%	-	24 h	Simple Brow...	...

Create monitorを  
押す

# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニターの作成(続き)

The screenshot displays the New Relic Synthetic Monitoring dashboard. On the left is a navigation sidebar with icons for various monitoring features. The main content area is titled 'Synthetic Monitoring' and shows a progress bar with three steps: '1. Select monitor type', '2. Configure monitor', and '3. Select locations'. Below the progress bar, there are eight monitor type cards arranged in a 2x4 grid. The 'User step execution' card, which includes the sub-option 'Step Monitor', is highlighted with an orange border. The other cards include: Availability (Ping), SSL certificate expiration (Certificate Check), Page link crawler (Broken Links Monitor), Page load performance (Simple browser), User flow / functionality (Scripted browser), and Endpoint availability (Scripted API).

**1. Select monitor type** 2. Configure monitor 3. Select locations

- Availability**  
Ping  
Simple ping using cURL
- SSL certificate expiration**  
Certificate Check  
Check the expiration and validity period on an SSL certificate of a given domain.
- Page link crawler**  
Broken Links Monitor  
Test page links for success
- Page load performance**  
Simple browser  
Full page load of a given URL. Provides deep data insights like resource breakdowns and timelines.
- User flow / functionality**  
Scripted browser  
Full scripted browser test.
- Endpoint availability**  
Scripted API  
Uses an HTTP client to monitor remote API endpoints (REST, healthchecks, etc).
- User step execution**  
Step Monitor  
Zero code monitor building

# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニターの作成(続き)

Account: 4365310 - gameday\_sandbox ▾

Name (required)  
YOURNAME\_step1test **名前を入力**

Browser  Chrome  Firefox **Chromeのみを選択**

Device Emulation  Desktop (1366 × 768)  
 Tablet landscape (1024 × 768)  
 Tablet portrait (768 × 1024)  
 Mobile landscape (800 × 400)  
 Mobile portrait (400 × 800)

Period  
1 day ▾ **1 dayを選択**

Tags  
Add a new tag...  
Enter tags in the form of "key:value" and press enter to add it to this monitor.

> Advanced options

Available synthetic monitor checks ⓘ  
Your included checks 1,000,000  
Checks scheduled this month 359,220  
Est. monthly checks for this monitor 0

Validate Cancel **Select locations** **次へ**

1. Select monitor type 2. Configure monitor **3. Select locations** 4. Define steps

① To minimize false positives, we recommend choosing at least 3 locations, preferably from different regions.  Private locations

Public locations (1)  test2

<input type="checkbox"/> North America (0)	<input type="checkbox"/> Europe (0)	<input checked="" type="checkbox"/> Asia (1)
<input type="checkbox"/> Columbus, OH, USA	<input type="checkbox"/> Dublin, IE	<input type="checkbox"/> Hong Kong, HK
<input type="checkbox"/> Montreal, Québec, CA	<input type="checkbox"/> Frankfurt, DE	<input type="checkbox"/> Manama, BH
<input type="checkbox"/> Portland, OR, USA	<input type="checkbox"/> London, England, UK	<input type="checkbox"/> Mumbai, IN
<input type="checkbox"/> San Francisco, CA, USA	<input type="checkbox"/> Milan, IT	<input type="checkbox"/> Seoul, KR
<input type="checkbox"/> Washington, DC, USA	<input type="checkbox"/> Paris, FR	<input type="checkbox"/> Singapore, SG
	<input type="checkbox"/> Stockholm, SE	<input checked="" type="checkbox"/> Tokyo, JP
<input type="checkbox"/> South America (0)	<input type="checkbox"/> Australia (0)	<input type="checkbox"/> Africa (0)
<input type="checkbox"/> São Paulo, BR	<input type="checkbox"/> Sydney, AU	<input type="checkbox"/> Cape Town, ZA

Validate Cancel **Define steps** **次へ**



# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニターの実成(続き)

Synthetic Monitoring

Synthetic Monitoring

Step builder Validation results

Click the Add button to add the step to your monitor. Use the up and down arrows on each step to rearrange steps and the trash icon to remove steps from your monitor.

> Navigate

> Click Element

> Double Click Element

> Select From Dropdown

1 Navigate:

確認したいサービスのURLを入力

Validate   保存

# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニター実行結果の確認

The screenshot displays the New Relic Synthetic Monitoring interface for a monitor named 'YOURNAME\_steptest'. The status is 'Good'. The summary shows a 100% success rate, 0/1 checks failed, and 0/1 locations failing. The 'Results' tab is selected, showing a bar chart for 'Duration and availability of checks in the last 30 days' for 'Tokyo, JP' with 0 failures and 1 check. A 'Run check' button is visible. Below the chart, the 'Request & response time' section shows a breakdown: 5.89 ms Receive, 720 µs Send, and 33.3 ms Wait. An arrow points from this section to a table of results. The table has columns for Result, Timestamp, Location, Execution time, Network latency, Response time, and Failure Message. A row shows a 'SUCCESS' result at 'January 24, 2025...' in 'tokyo, JP' with an execution time of 3.24 s and network latency of 579 ms. The 'Timestamp' cell in the table is highlighted with an orange box. To the right, a 'Duration by location' chart shows a single data point for Tokyo, JP at approximately 579 ms.

**Resultsを選択**

**任意のタイムスタンプを選択**

Result	Timestamp	Location	Executio...	Network...	Respons...	Failure Message
SUCCESS	January 24, 2025...	tokyo, JP	3.24 s	579 ms	-1	-

# ハンズオン(2)-2 ~ 解答 Stepモニターを作成する

## Stepモニター実行結果の確認(続き)

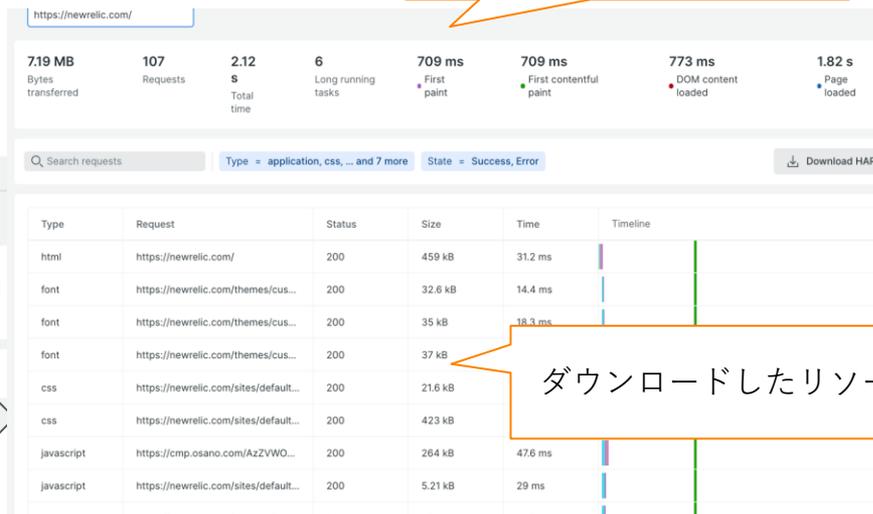
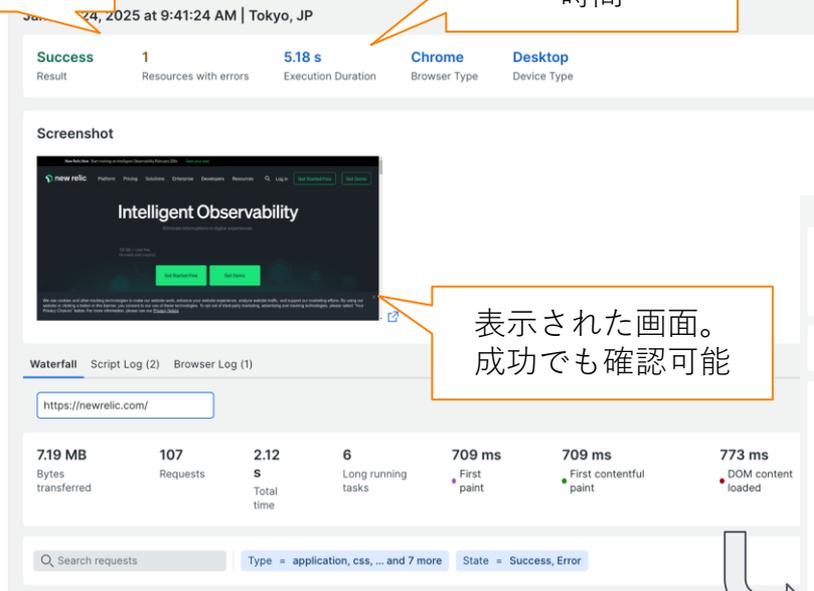
実行結果

実行に掛かった  
時間

画面描画のパフォーマンス

表示された画面。  
成功でも確認可能

ダウンロードしたリソース



# ハンズオン(2)-3 ~ 解答 Scripted Browserモニターを作成する

## Scripted Browserモニターの作成

アカウント4365310  
を選択

Create monitorを  
押す

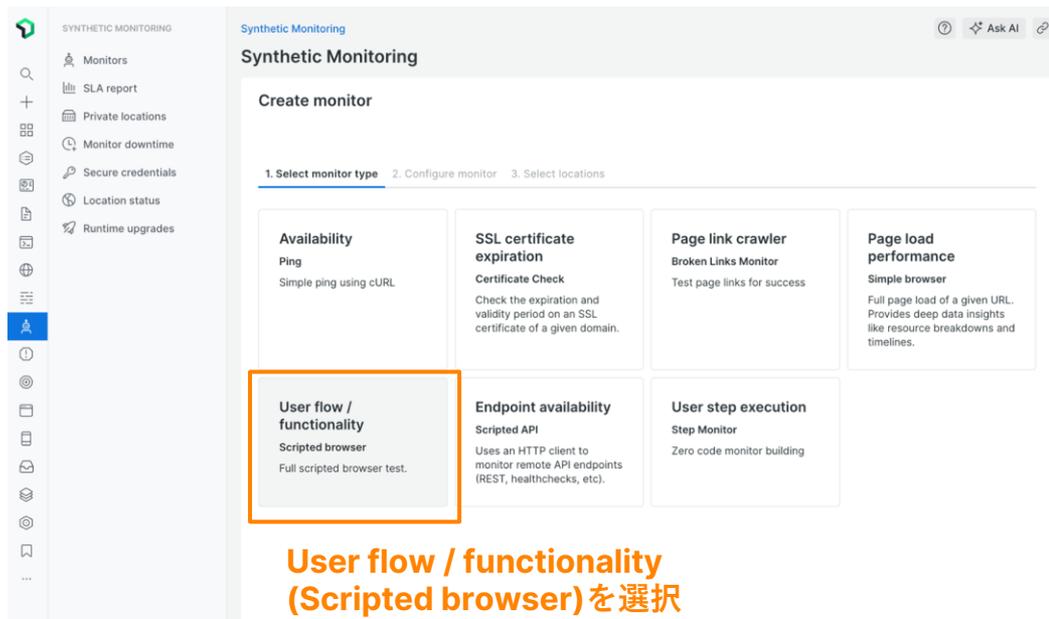
Synthetic  
Monitoringを  
選択

The screenshot shows the New Relic Synthetic Monitoring interface. The left sidebar has 'Synthetic Monitoring' highlighted. The main content area shows a list of monitors:

	Name ↑	Monitor stat...	Success rat...	Locations fa...	Period	Monitor type	
☆	● test1_pingtest	Enabled	59.75%	1 / 1	60 s	Ping	...
☆	● test1_simpletest	Enabled	100%	-	24 h	Simple Brow...	...

# ハンズオン(2)-3 ~ 解答 Scripted Browser モニターを作成する

## Scripted Browser モニターの作成(続き)



The screenshot shows the 'Synthetic Monitoring' interface in the New Relic console. The main area is titled 'Create monitor' and is divided into three steps: '1. Select monitor type', '2. Configure monitor', and '3. Select locations'. Under step 1, there are seven monitor type options:

- Availability**  
Ping  
Simple ping using cURL
- SSL certificate expiration**  
Certificate Check  
Check the expiration and validity period on an SSL certificate of a given domain.
- Page link crawler**  
Broken Links Monitor  
Test page links for success
- Page load performance**  
Simple browser  
Full page load of a given URL. Provides deep data insights like resource breakdowns and timelines.
- User flow / functionality**  
Scripted browser  
Full scripted browser test. (This option is highlighted with an orange box in the image)
- Endpoint availability**  
Scripted API  
Uses an HTTP client to monitor remote API endpoints (REST, healthchecks, etc).
- User step execution**  
Step Monitor  
Zero code monitor building

Below the screenshot, the text 'User flow / functionality (Scripted browser) を選択' is written in orange.

# ハンズオン(2)-3 ~ 解答 Scripted Browserモニター ーを作成する

## Scripted Browserモニターの作成(続き)

Account: 4365310 - gameday\_sandbox

Name (required)  
YOURNAME\_scripttest **名前を入力**

Browser **Chromeのみを選択**  
 Chrome  
 Firefox

Device Emulation  
 Desktop (1366 × 768)  
 Tablet landscape (1024 × 768)  
 Tablet portrait (768 × 1024)  
 Mobile landscape (800 × 400)  
 Mobile portrait (400 × 800)

Period **1 dayを選択**  
1 day

Tags  
Add a new tag...  
Enter tags in the form of "key:value" and press enter to add it to this monitor.

> Advanced options

Available synthetic monitor checks  
Your included checks 1,000,000  
Checks scheduled this month 359,250  
Est. monthly checks for this monitor 0

Validate Cancel **Select locations** **次へ**

1. Select monitor type 2. Configure monitor 3. Select locations 4. Write script

To minimize false positives, we recommend choosing at least 3 locations, preferably from different regions.  Private locations

Public locations (1)  test2

<input type="checkbox"/> North America (0)	<input type="checkbox"/> Europe (0)	<input checked="" type="checkbox"/> Asia (1)
<input type="checkbox"/> Columbus, OH, USA	<input type="checkbox"/> Dublin, IE	<input type="checkbox"/> Hong Kong, HK
<input type="checkbox"/> Montreal, Québec, CA	<input type="checkbox"/> Frankfurt, DE	<input type="checkbox"/> Manama, BH
<input type="checkbox"/> Portland, OR, USA	<input type="checkbox"/> London, England, UK	<input type="checkbox"/> Mumbai, IN
<input type="checkbox"/> San Francisco, CA, USA	<input type="checkbox"/> Milan, IT	<input type="checkbox"/> Seoul, KR
<input type="checkbox"/> Washington, DC, USA	<input type="checkbox"/> Paris, FR	<input type="checkbox"/> Singapore, SG
<input type="checkbox"/> São Paulo, BR	<input type="checkbox"/> Stockholm, SE	<input checked="" type="checkbox"/> Tokyo, JP
<input type="checkbox"/> Sydney, AU	<input type="checkbox"/> Cape Town, ZA	

Available synthetic monitor checks

Validate Cancel **Write script** **次へ**

# ハンズオン(2)-3 ~ 解答 Scripted Browserモニターを作成する

## Scripted Browserモニターの作成(続き)

1. Select monitor type 2. Configure monitor 3. Select locations 4. Write script

Script editor Validation results

Insert secure credential

```
1 $browser.get('https://front.demo.learn.nrkk.technology/login').then(async function(){
2   const userNameInput = await $browser.findElement($driver.By.xpath('//*[@id="username"]'));
3   await userNameInput.sendKeys("e100000@example.com");
4   const passwordInput = await $browser.findElement($driver.By.xpath('//*[@id="password"]'));
5   await passwordInput.sendKeys("password");
6   const loginButton = await $browser.findElement($driver.By.xpath('/html/body/div/form/button'));
7   await loginButton.click();
8   const category1 = await $browser.findElement($driver.By.xpath('//*[@id="root"]/div/div[2]/div'));
9   await category1.click();
10 });
11
12
```

サンプルのスクリプトを貼り付ける

Validate

Cancel

Save monitor

スクリプトを検証



ハンズオン中、サービスが意図的に高負荷になることがあります。実行に失敗しても次に進みましょう。

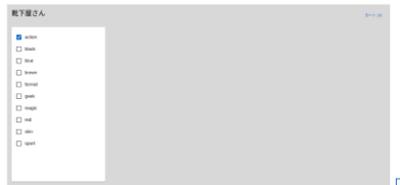
Synthetic Monitoring

Script editor Validation

January 24, 2024 at 9:50 AM UTC+9, 0

Success 14 6.15 s  
Result Resources with errors Execution Duration

Screenshot



Waterfall Browser Log (1)

https://front.demo.learn.nrkk.t... https://front.demo.learn.nrkk.t...

Validate

Cancel

Save monitor

スクリプトを保存

# ハンズオン(2)-3 ~ 解答 Scripted Browserモニターを作成する

## Scripted Browserモニターの実行結果の確認

Synthetic Monitoring / Synthetic monitors

YOURNAME\_scripttest ☆ Tags Metadata Workloads

Good

Summary

MONITOR

100% Success rate 0/1 Checks failed 0/1 Locations failing Everything's A-OK! No failures detected within the last 30 days

Results

Resources

TRiage

Diagnose

REPORTS

SLA

Service levels

SETTINGS

General

Alert conditions

MORE VIEWS

Duration and availability of checks in the last 30 days

Tokyo, JP 0 failures / 1 checks ...

3.77 s max 3.77 s median

Run check

January 24th 9:56 AM January 24th 9:56 AM

Average Median 95th Percentile

Narrow data to Search for any attribute or value.

Resultsを選択

Duration by location

Duration by location Network timings average.duration 3.77 ...

3.766 ms 3.765.8 ms 3.765.6 ms 3.765.4 ms 3.765.2 ms 3.765 ms 3.764.8 ms 3.764.6 ms 3.764.4 ms 3.764.2 ms 3.764 ms

am 9:00am 9:05am 9:10am 9:15am 9:20am 9:25am 9:30am 9:35am 9:40am 9:45am 9:50am 9:55am

Tokyo, JP

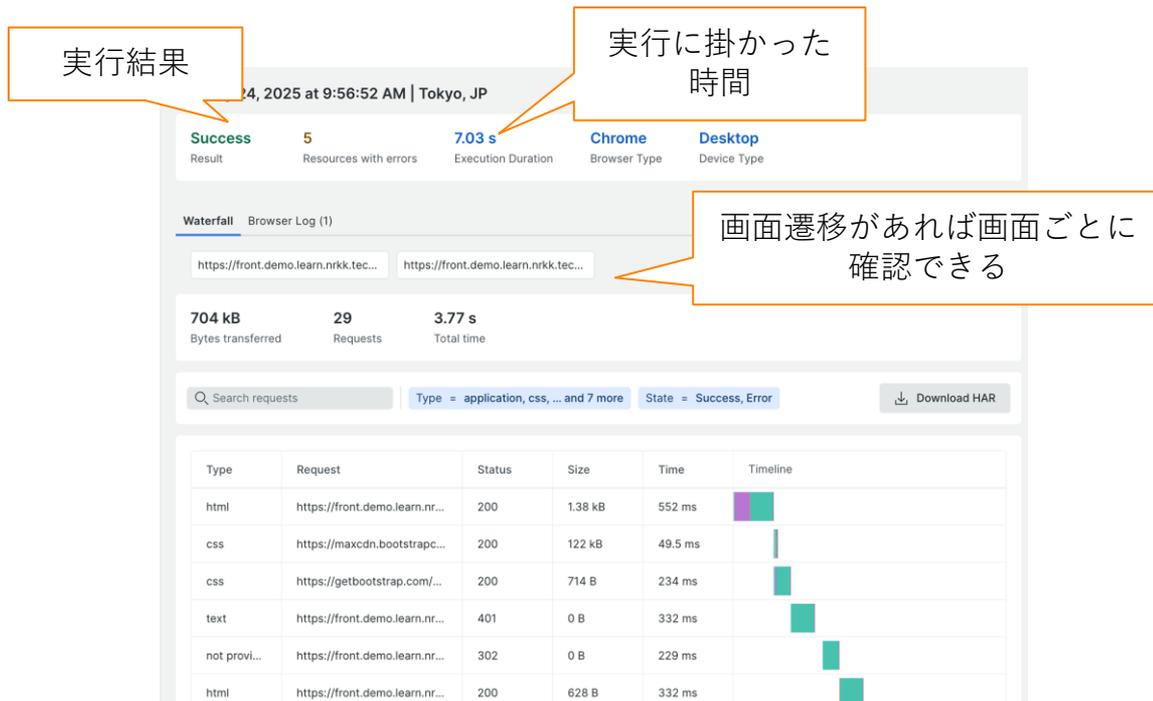
Search results Location = Tokyo, JP Status = Success, Failed

Result	Timestamp	Location	Browser	Execu...	Netw...	Resp...	Failure Message
SUCC...	January 24, ...	Toky...JP	Chrome 117	7.03 s	3.77 s	283.4...	-

任意のタイムスタンプを選択

# ハンズオン(2)-3 ~ 解答 Scripted Browserモニター ーを作成する

## Scripted Browserモニターの実行結果の確認



# 座学(4)

## リアルユーザーの **Web** パフォーマンス の分析

# New Relic Browserの主要機能



早期解決  
プロアクティブな対応

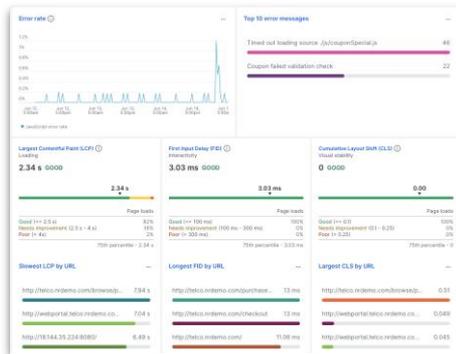


予期せぬ課題に  
高速にアプローチ

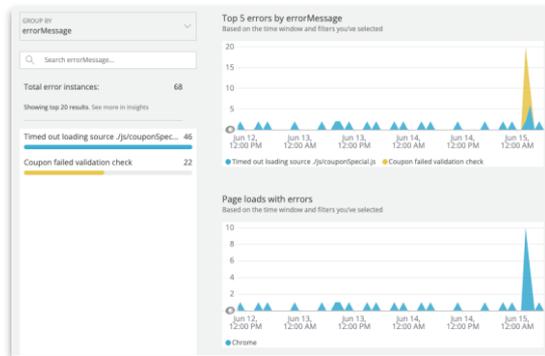


顧客体験とシステムの  
因果関係を把握し理解

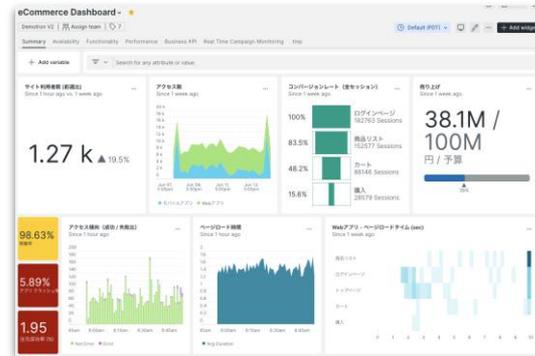
## パフォーマンス概要の把握



## エラー分析



## 行動分析



# 運用保守によくある課題



障害認知はユーザー任せ



状況確認はヒアリング頼み



顧客体験や速度改善など、ユーザーへの効果が見えない

# リアルユーザーモニタリングの効果



障害認知はユーザー任せ

早期解決・プロアクティブな対応



状況確認はヒアリング頼み

予期せぬ課題に高速にアプローチ



顧客体験や速度改善など、ユーザーへの効果が見えない

顧客体験とシステムの因果関係を把握し、理解を得る

# ユースケース

- 各ページのユーザー体験を改善し、SEO対策
- 地理や環境ごとの状況分析
- JavaScript Errorの内容調査
- ユーザーIDでカスタマーサポートが個別IDごとの状況分析

# ハンズオン(3)： リアルユーザーの **Web** パフォーマンスの分析

# ハンズオン(3)Webパフォーマンスの分析

**Browserアプリケーション[front-service]を選択し**

**以下のユースケースに沿ってWebサイトのパフォーマンスを調査してください。**

## 1. 時間の設定

- データの表示期間を、過去 3時間に設定してください。

## 1. 各ページのユーザー体験を改善し、SEO対策

### - Core Web Vitals

- 最大コンテンツの描画時間 (LCP) が最も遅いWebページはどれでしょうか。

### - Ajax

- 最も応答時間がかかるAjax (非同期通信) はどれでしょうか。また、該当AjaxのJavaScriptの実行状況を調査してみましょう。

## 2. 地理や環境ごとの状況分析

- どの国から最もアクセスされているでしょうか。
  - その国では、どのブラウザから最もアクセスされているでしょうか。

# ハンズオン(3)Webパフォーマンスの分析

**Browserアプリケーション[front-service]を選択し**

**以下のユースケースに沿ってWebサイトのパフォーマンスを調査してください。**

## 4. JavaScript Errorの内容調査

### – Errors

- 最もユーザー影響の多いエラーはなんでしょうか。またエラーの詳細を確認してみましょう。

## 5. ユーザーIDでカスタマーサポートが個別IDごとの状況分析

### – Session Replay

- user idが[usr id2]のお客様から問い合わせがありました。  
Session Replayを確認して[usr id2]の最新セッションの状況を確認しましょう

# 手順・解説

# ハンズオン(3)-1 ~ 解答 時間の設定

new relic

Browser Save view

Account: 4097184 - gameday\_common

gameday\_common  
を選択

Search by entity name Entity Type = Browser application, Micro frontend

Entities

	Name ↑	Account	Throughput (p...	Largest conten...	Interaction to ...	Cumulative lay...	Errors	Pageload time ...	Ajax throughp...	
☆	catalogue-web	gameday_com...	79.5 page/s	836 ms	24 ms	0	328	4254 ms	2.57 rpm	...
☆	front-service	gameday_com...	0.27 page/s	1990 ms	8 ms	0.009	-	1628 ms	0.47 rpm	...

front-serviceを選択

Browserを選択

# ハンズオン(3)-1 ~ 解答 時間の設定

The screenshot shows the New Relic One dashboard for an application named 'front-service'. The dashboard is divided into several sections: Summary, Errors, Monitor, Performance (LCP, INP, CLS), and Activity stream. The Activity stream section is highlighted with an orange box, and a dropdown menu is open, showing a list of time ranges. The '3 hours' option is selected and highlighted with an orange box. An orange arrow points from the '3 hours' option in the dropdown to the 'Since 3 hours ago (GMT+9)' text in the Activity stream header. Another orange box highlights the 'Since 3 hours ago (GMT+9)' text in the header.

3 hoursを選択

過去3時間のデータが表示されます。

# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Core web vitals

The screenshot shows the New Relic monitoring dashboard for a service named 'front-service'. The interface includes a left-hand navigation menu with categories like ERRORS, MONITOR, TRIAGE, USERS, and SETTINGS. The 'Web vitals' option is highlighted with an orange box and a callout bubble that says 'Web Vitalsを選択' (Select Web Vitals).

The main dashboard area displays several key performance indicators (KPIs) and charts:

- Error rate:** A line chart showing the JavaScript error rate over time, with a peak around 3:20pm. The overall status is 'Good'.
- Top 10 error messages:** A list of the most frequent errors, all being 'Unhandled Promise Rejection: GET /...'. Each error has a count of 4.
- Session replay:** A section showing recorded sessions with their duration and error counts. Three sessions are listed, all with 0 errors.
- Largest Contentful Paint (LCP):** A loading performance metric with a value of 4.22 s, labeled as 'POOR'. A horizontal bar chart shows the distribution of page loads, with the 75th percentile at 4.22 s.
- Interaction To Next Paint (INP):** A responsiveness metric with a value of 90 ms, labeled as 'GOOD'. A horizontal bar chart shows the distribution of page loads, with the 75th percentile at 90 ms.
- Cumulative Layout Shift (CLS):** A visual stability metric with a value of 0.00, labeled as 'GOOD'. A horizontal bar chart shows the distribution of page loads, with the 75th percentile at 0.00.
- Slowest LCP by URL:** A list of URLs with their LCP values. The slowest is 'https://front.demo.learn.nrkk.tech...' with a value of 4203.13 ms.
- Longest INP by URL:** A list of URLs with their INP values. The longest is 'https://front.demo.learn.nrkk.tech...' with a value of 448 ms.
- Largest CLS by URL:** A list of URLs with their CLS values. The largest is 'https://front.demo.learn.nrkk.tech...' with a value of 0.

# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Core web vitals

Browser / applications

front-service ☆ Tags Metadata ...

User experience **Good** Services **Good** Engineering operations **100% critical alerts** Since 30 minutes ago (GMT+9)

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

**Web vitals**

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay

Browsers

Geography

SETTINGS

Application settings

Filter by Search for any attribute or value.

### Web vitals

**Largest Contentful Paint (LCP)** ⓪  
Loading performance  
**3.92 s NEEDS IMPROVEMENT**

**Interaction To Next Paint (INP)** ⓪  
Responsiveness  
**90 ms GOOD**

**Cumulative Layout Shift (CLS)** ⓪  
Visual stability  
**0 GOOD**

75th percentile - 3.92 s

75th percentile - 90 ms

75th percentile - 0

Category	Good (<=)	Needs improvement	Poor (>)
LCP (s)	57%	18%	25%
INP (ms)	75%	25%	0%
CLS	100%	0%	0%

Page loads

### Page URLs

Search URLs

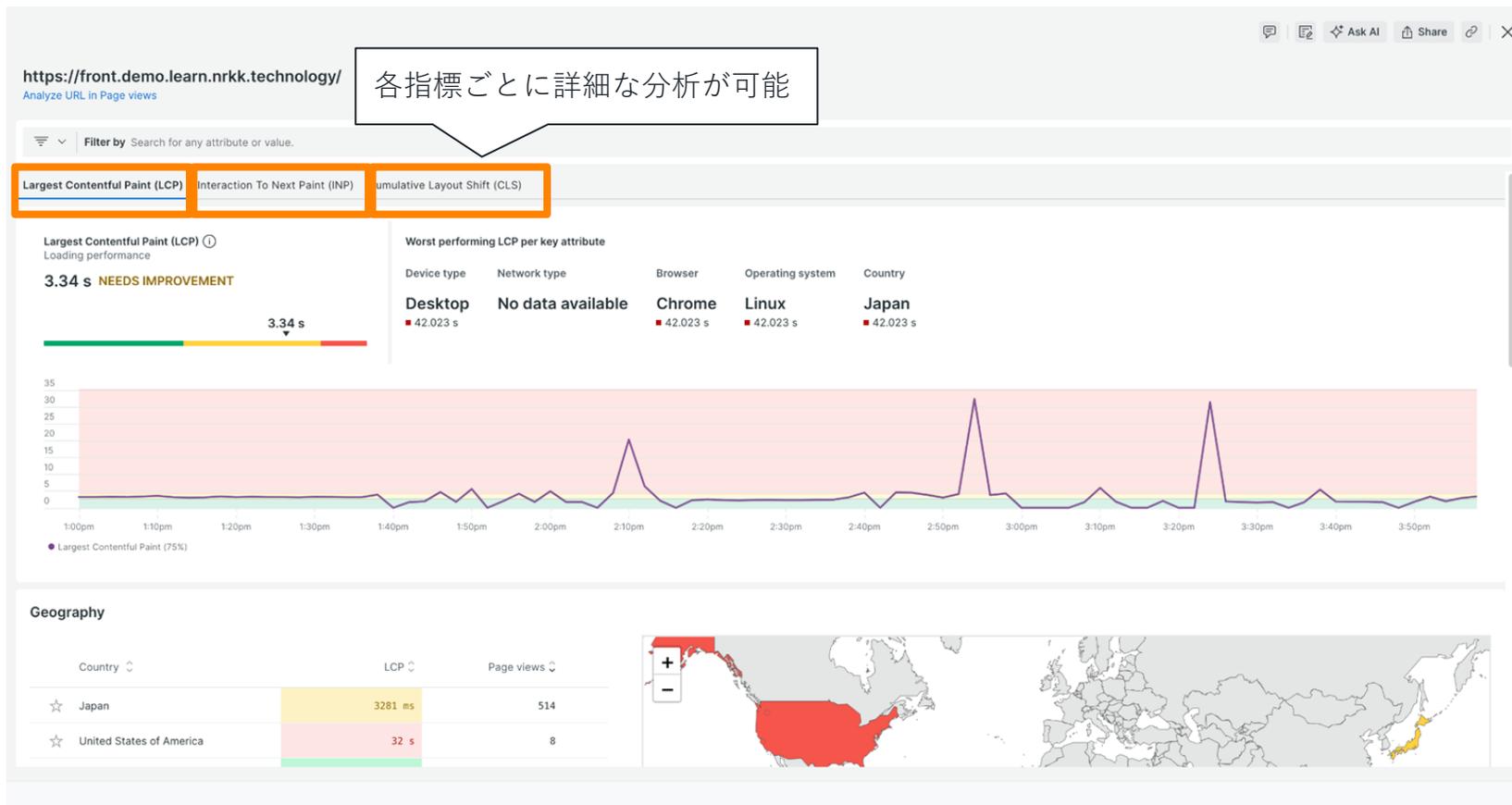
Page URL

**最もLCPの遅いページを選択**

ページごとのCore Web Vitals

Page URL	LCP	INP	CLS	Page views
https://front.demo.learn.nrkk.technology/	4203 ms	448 ms	0	45
https://front.demo.learn.nrkk.technology/product/808a2de1-1aaa-4c25-a9b9-6612e8f29a38	3922 ms	88 ms	0	6
https://front.demo.learn.nrkk.technology/product/6d62d909-f957-430e-8689-b5129c0bb75e	3883 ms	16 ms	0	4
https://front.demo.learn.nrkk.technology/product/a0a4f044-b040-410d-8ead-4de0446aec7e	2609 ms	64 ms	0	8

# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Core web vitals



# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Ajax

The screenshot displays the New Relic APM interface for a service named 'front-service'. The left sidebar shows navigation options, with 'AJAX' selected. The main area shows a table of requests sorted by response time. The top two requests are highlighted with orange boxes and arrows pointing to a dropdown menu. The dropdown menu is open, showing 'Response time' as the selected sort option. Below the table, a throughput chart shows request volume over time.

**Sort by:** Response time (selected) | **Group by:** requestUrl

**Showing:** Total time percentage, Response time (selected), Throughput (rpm), Average data transfer, Average callback time

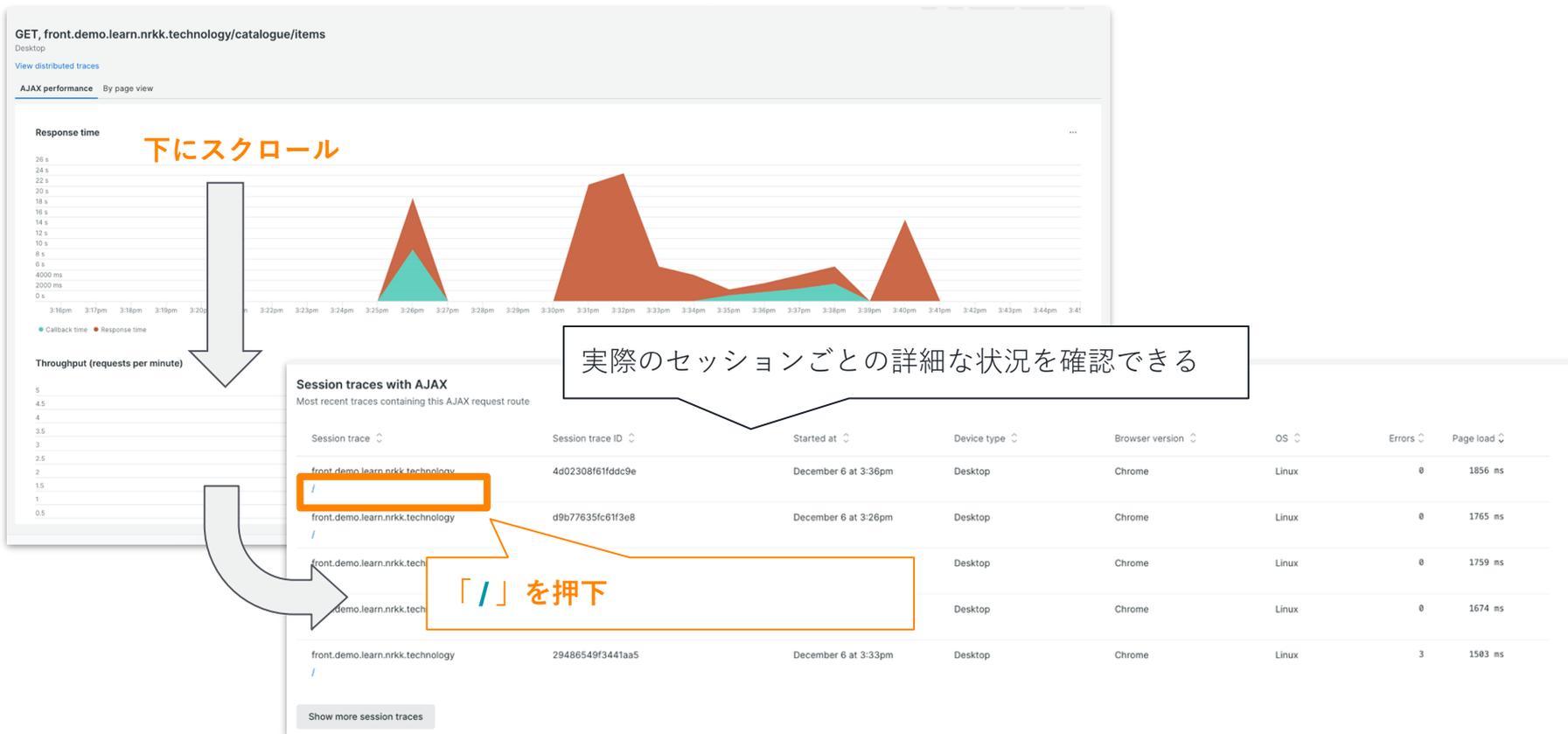
Request	Response time
front.demo.learn.nrkk.technology/catalog... GET /catalogue/items	9 s
front.demo.learn.nrkk.technology/catalog... GET /catalogue/item/808a2de1-1aaa-4c...	9 s
front.demo.learn.nrkk.technology/catalog... GET /catalogue/item/6d62d909-f957-43...	2574 ms
front.demo.learn.nrkk.technology/catalog... GET /catalogue/item/a0a4f044-b040-41...	1798 ms
front.demo.learn.nrkk.technology/order GET /order	1798 ms
front.demo.learn.nrkk.technology/order/c... POST /order/confirm	1785 ms
front.demo.learn.nrkk.technology/order/p... POST /order/purchase	1547 ms
front.demo.learn.nrkk.technology/catalog... GET /catalogue/tags	1194 ms
front.demo.learn.nrkk.technology/cart GET /cart	1091 ms

**Throughput (requests per minute)**

**Annotations:**

- Response timeを選択し、応答時間でソート
- ソートされたAJAXで最も遅いものを選択

# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Ajax



# ハンズオン(3)-2 ~解答 各ページのユーザー体験を改善し、SEO対策 - Ajax

ページ描画の各パフォーマンス

/  
Linux Chrome 129 | December 6 at 3:38

<b>535 ms</b> Backend	<b>874 ms</b> DOM processing	<b>1856 ms</b> Page load	<b>1691 ms</b> Waiting on AJAX	-- First interaction	-- Interaction to next paint	<b>3595 ms</b> Largest contentful paint	<b>7 s</b> Total duration
--------------------------	---------------------------------	-----------------------------	-----------------------------------	-------------------------	---------------------------------	--	------------------------------

## Session trace

🔍 Search in all fields

Data type = All

Duration = All durations

Timestamp	Data type	Description	Timeline
> 00:00	JavaScript	document.cis	
> 00:00	JavaScript	document.startTime	
> 00:00	JavaScript	document.firstInterimResponseStart	
> 00:00	JavaScript	document.workerStart	
> 00:00	JavaScript	document.activationStart	
> 00:00	JavaScript	document.criticalCSSStart	
> 00:00	JavaScript	document.criticalCSSStart	
> 00:00	JavaScript	document.activationStart	
> 00:00	JavaScript	document.workerStart	
> 00:00	JavaScript	document.firstInterimResponseStart	
> 00:00	JavaScript	document.startTime	
> 00:00	JavaScript	document.cis	

時系列上でJavaScript実行状況を分析することができる

# ハンズオン(3)-3 ~解答 地理や環境ごとの状況分析

Browser / applications

front-service ☆ Tags Metadata ...

User experience Good Services Good Engineering operations Good

Since 3 hours ago (GMT+9)

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

Web vitals

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay

Browsers

**Geography**

SETTINGS

Application settings

Alert conditions

Domain conditions

View Page load count

Map view Key attribute charts

Page load count shows the top ten locations with the highest number of page loads

Country Page load count

☆ Japan	204
☆ United States of America	32
☆ Australia	3
☆ Ireland	3

最もアクセスの多い国を選択



0 204

Page load performance Historical performance

Average page load time 2500 ms

ic

# ハンズオン(3)-3 ~解答 地理や環境ごとの状況分析

Browser / applications

front-service

User experience Good Services Good Engineering operations Good

Since 3 hours ago (GMT+9)

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

Web vitals

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay

Browsers

**Geography**

SETTINGS

Application settings

Country code = JP

View Page load count

**Key attribute charts** を選択

Map view

Page load count shows the top ten locations with the highest number of page loads, as well as the average page load time for each.

Region	Page load count
Tokyo	199

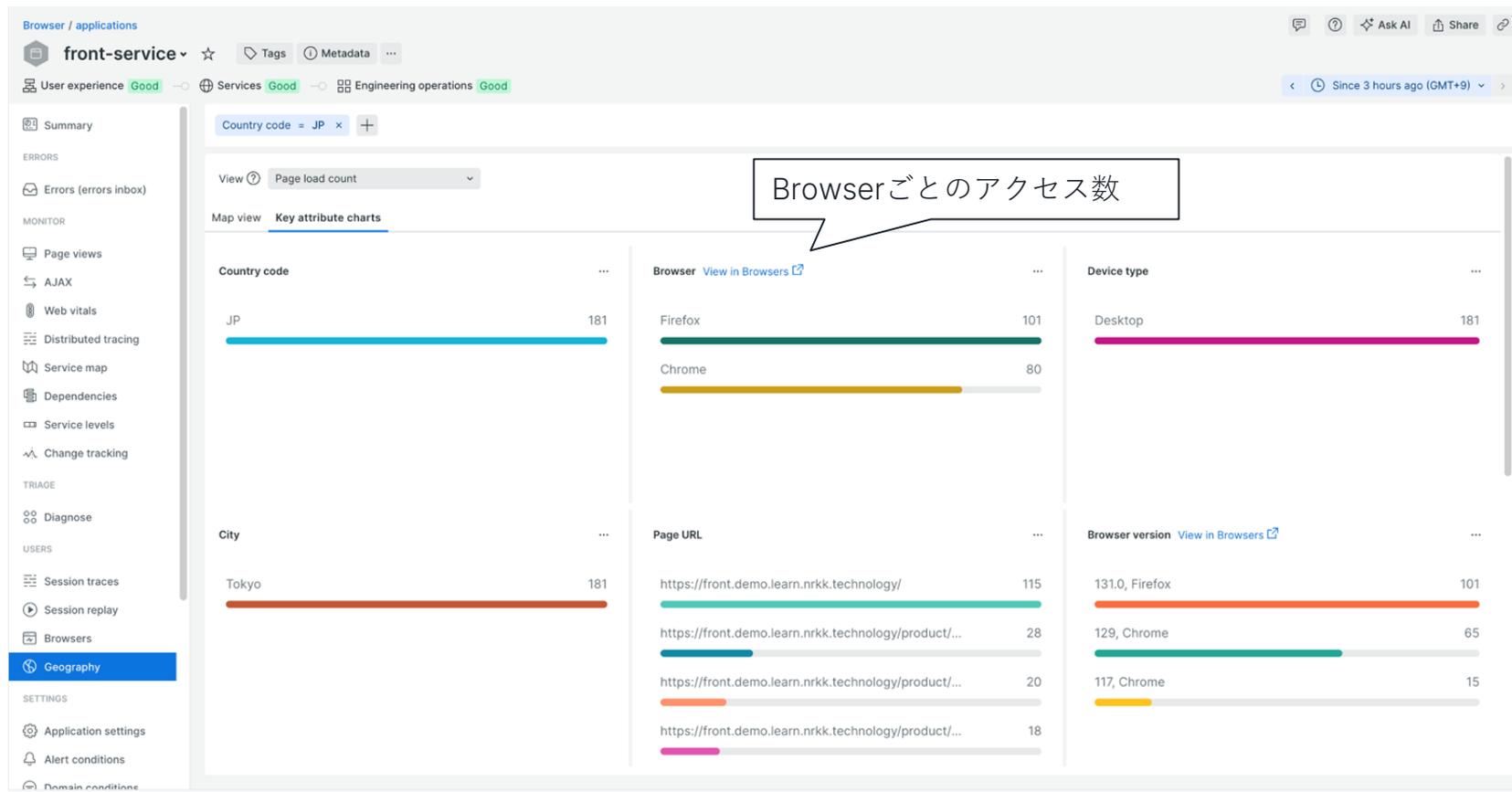
選択した国で絞り込まれ  
さらに都市ごとのアクセス数が表示される

0 199

Page load performance Historical performance

Average page load time

# ハンズオン(3)-3 ~解答 地理や環境ごとの状況分析



# ハンズオン(3)-4 ~解答 Java Script Errorの内容調査

The screenshot displays the New Relic error monitoring interface for the application 'front-service'. The interface is divided into several sections:

- Summary:** Shows overall service health with 'User experience Good' and '100% critical alerts' for both 'Service' and 'Engineering operations'.
- Errors:** A sidebar on the left contains navigation options: 'Errors (errors inbox)', 'MONITOR', 'Page view', 'AJAX', 'Web vitals', 'Distributed tracing', 'Service map', 'Dependencies', 'Service levels', 'Change tracking', 'TRIAGE', 'Diagnose', 'USERS', 'Session traces', 'Session replay', 'Browsers', 'Geography', 'SETTINGS', 'Application settings', and 'Alert conditions'. The 'Errors (errors inbox)' option is highlighted with an orange box and labeled 'Errorsを選択'.
- Triage:** The main content area has tabs for 'Triage' and 'Group errors'. The 'Triage' tab is selected and highlighted with an orange box and labeled 'Triageを選択'. Below the tabs is a search bar for filtering errors.
- Visualizations:** Two charts are displayed:
  - Error count:** A line chart showing the top 5 error messages over time. The legend includes 'Unhandled Promise Rejection: GET /catalogue/...'.
  - Page load errors:** A grouped bar chart showing errors grouped by browser type (Chrome and Firefox) over time.
- Table:** A table at the bottom lists error occurrences with columns for 'errorMessage', 'Occurrences', 'First occurred', and 'Last occurred'. The table shows four entries for 'Unhandled Promise Rejection' errors.

# ハンズオン(3)-4 ~解答 Java Script Errorの内容調査

Browser / applications

front-service

User experience Good Services 100% critical alerts Engineering operations 100% critical alerts

from Feb 4, 01:00 pm to Feb 4, 04:00 pm (GMT+9)

Summary

ERRORS

Errors (errors inbox)

MONITOR

Page views

AJAX

Web vitals

Distributed tracing

Service map

Dependencies

Service levels

Change tracking

TRIAGE

Diagnose

USERS

Session traces

Session replay

Browsers

Geography

SETTINGS

Application settings

Alert conditions

Triage Group errors

error.group.metadata.state = Unresolved

Sort by Occurrences

Go to global inbox

Errors overview

Error count  
Top 5 by error group

Page load errors  
Grouped by browser type

影響ユーザー数

エラー発生数

最も影響ユーザー数の多いエラーを選択

Unhandled Promise Rejection: GET /catalogue/items?tags=action is failed  
front-service | 3 months old - last seen about 6 hours ago  
Session replays 1

Unhandled Promise Rejection: GET /catalogue/items?tags=action,black is failed  
front-service | 3 months old - last seen about 6 hours ago  
Session replays 1

# ハンズオン(3)-4 ~解答 JavaScript Errorの内容調査

The screenshot displays the New Relic error monitoring interface. At the top, the error type is identified as `JavaScriptError`. The error message is `Unhandled Promise Rejection: GET /catalogue/items?tags= is failed`. The occurrence details show the error occurred on December 6, 2024, at 2:54pm, from a Chrome browser on a Linux desktop. The session replay section is highlighted with a callout box containing the text `エラー発生時のセッションリプレイ`. A large downward-pointing arrow on the right side of the interface is labeled `下にスクロール` (Scroll down).

**Error**

front-service

Error message:  
Unhandled Promise Rejection: GET /catalogue/items?tags= is failed

Start a new discussion

**Error group metrics**

Users impacted 13 users impacted since 24 hours ago  
— users impacted since 1 month ago

First seen 28 days ago in  
0.0.39

Last seen about 1 hour ago in  
0.0.39

account:gameday\_common accountId:4097184

**Occurrences** 30 occurrences

Time	Occurrences
Dec 05, 3:00pm	0
Dec 05, 6:00pm	0
Dec 05, 9:00pm	0
Dec 06, 12:00am	0
Dec 06, 3:00am	0
Dec 06, 6:00am	0
Dec 06, 9:00am	11
Dec 06, 12:00pm	0
Dec 06, 3:00pm	5

**Occurrence details**

Exception type Error  
Timestamp December 6, 2024, 2:54pm  
URL /  
Browser Chrome  
Device type Desktop  
Operating system Linux

Show more attributes

**Session replay**

エラー発生時のセッションリプレイ

Send us feedback

NEW RELIC

# ハンズオン(3)-4 ~解答 JavaScript Errorの内容調査

The screenshot displays the New Relic error monitoring interface. On the left, there is a performance chart and a 'Profiles' section showing deviceType (Desktop 100.0%), requestUri (/ 100.0%), and userAgentName (Chrome 100.0%). The main area shows a 'Stack trace' for an 'Error: Unhandled Promise Rejection: GET /catalogue/items?tags=action is failed'. The stack trace points to 'utils/api.ts' at line 18:20. The code snippet shows a catch block where an error is thrown. Below the stack trace is an 'Event trail' section showing a 'Javascript error' event at 13:48:59.418 and a 'Browser interaction' event at 13:48:23.444. A callout box points to the stack trace with the text: 'Stack trace ※ソースマップをNew Relicに連携することでエラー該当箇所を表示可能'. Another callout box points to the error message with the text: 'エラー発生までの導線'. A large grey arrow on the right side points downwards with the text: '下にスクロール'. The New Relic logo is visible in the bottom right corner.

Stack trace  
※ソースマップをNew Relicに連携することでエラー該当箇所を表示可能

下にスクロール

エラー発生までの導線

# ハンズオン(3)-4 ~解答 Java Script Errorの内容調査

エラー発生時の詳細な属性情報

エージェントが標準で取得する属性

- ・端末種類（デスクトップ・モバイル）
- ・実行ブラウザ
- ・OSの種類 など

コードのカスタマイズにより

任意の属性を追加で送信することが可能

- ・ユーザーを識別する一意の値（userid等）を連携することで、エラーの影響ユーザー数を表示できる

## Attributes

Search attributes

Name ↑	Value
accountId	4097184
agentVersion	1.263.0
application.version	0.0.41
asn	16509
asnLatitude	35.6893
asnLongitude	139.6899
asnOrganization	AMAZON-02
browserInteractionId	cda63153-1f20-4cb1-9689-f05337ec799d
browserStackHash	1103674554
city	Tokyo
countryCode	JP
deviceType	Desktop
enduser.id	uid_38
firstErrorInSession	true
firstOccurrenceTimestamp	1738644539418
hasReplay	true

# ハンズオン(3)-5 ~解答 ユーザーIDでカスタマーサポートが個別IDごとの状況分析

The screenshot shows a web analytics dashboard with a table of error logs. The table has columns for 'Starting URL', 'Started At', 'Device Type', 'Browser', and 'Errors'. The 'Errors' column shows a count of 1 for a specific entry. A filter is applied to 'User IDs' with the value 'uid\_2'. The 'Session replay' option is selected in the left sidebar.

**User IDsが「uid\_2」のセッションだけを絞り込む**

**Session Replayを選択**

**Edit filter**

Filter: User IDs

Operator: =

Value: uid\_2

Search...  
 All  
 uid\_8  
 uid\_38  
 uid\_43  
 uid\_19  
 uid\_2  
 uid\_63  
 uid\_35  
 uid\_26

Reset Cancel Apply

Close

# ハンズオン(3)-5 ~解答 ユーザーIDでカスタマーサポートが個別IDごとの状況分析

Browser / applications

front-service

User experience Good Services 100% critical alerts Engineering operations 100% critical alerts

from Feb 4, 01:00 pm to Feb 4, 04:00 pm (GMT+9)

Send us feedback

User IDs = uid\_2 URLs viewed = All Duration >= 5 seconds

SESSIONS

User ID	Starting URL	Duration	Started At ↓	Device Type	Browser	Errors
uid_2	front.demo.learn.nrkk.technology /	2 m 41 s	February 4 at 3:44pm	Desktop	Chrome	0
uid_2	front.demo.learn.nrkk.technology /	1 m 58 s	February 4 at 2:47pm	Desktop	Chrome	0
uid_2	front.demo.learn.nrkk.technology /	2 m 23 s	February 4 at 2:26pm	Desktop	Firefox	0
uid_2	front.demo.learn.nrkk.technology /	2 m 17 s	February 4 at 2:02pm	Desktop	Firefox	0
uid_2	front.demo.learn.nrkk.technology /	2 m 11 s	February 4 at 1:31pm	Desktop	Firefox	0
uid_2	front.demo.learn.nrkk.technology /	2 m 13 s	February 4 at 1:20pm	Desktop	Firefox	0

Errors (errors inbox)

MONITOR

- Page views
- AJAX
- Web vitals
- Distributed tracing
- Service map
- Dependencies
- Service levels
- Change tracking

TRIAGE

- Diagnose

USERS

- Session traces
- Session replay**
- Browsers
- Geography

SETTINGS

- Application settings
- Alert conditions
- Domain conditions

セッション開始時刻が最新のセッションを選択

# ハンズオン(3)-5 ~解答 ユーザーIDでカスタマーサポートが個別IDごとの状況分析

Session replay

uid\_2 February 4 at 3:44pm Metadata Send us feedback

実際のセッション状況を動画で再生

Event trail

Search in all fields Event type = All

1874 ms

00:28 > Unhandled Promise Rejection: GET...

00:34 > /product/a0a4f044-b040-410d-8ead-4de0446...

1809 ms > /product/808a2de1-1aaa-4c25-a9b9-6612e8f...

00:42 > /product/808a2de1-1aaa-4c25-a9b9-6612e8f...

1815 ms

01:30 > /

2164 ms

01:39 > /

2241 ms

01:47

Play 01:01 / 02:41

1x Skip inactivity

Session trace /

Search in all fields Data type = All Duration = All durations

Timestamp	Data type	Description
> 00:15	Interaction	change input:focus-action-PrivateSwitchBase-Input-css-189df3
> 00:15	JavaScript	animationstart span_MuiTouchRipple-chi1d_MuiTouchRipple-chi1dLeaving
> 00:15	JavaScript	animationstart span-css-12n3ag_MuiTouchRipple-Figgle-MuiTouchRipple-FiggleF131e
> 00:15	JavaScript	pointerout input:focus-action-PrivateSwitchBase-Input-css-189df3
> 00:15	JavaScript	pointerover input:focus-action-PrivateSwitchBase-Input-css-189df3
> 00:15	JavaScript	input:focus-action-PrivateSwitchBase-Input-css-189df3

- ・テキストと入力内容は保存されません。マスクして表示されます。
- ・実際のスクリーンショットやビデオがキャプチャされているわけではありません。
- ・描画に必要なDOMと状態変更を記録し、再構築して動画形式で表示します。そのため、保存されるデータ量は動画と比べ大幅に削減されます。

# 補足: Browser Agentのインストール

導入手順は3パターン

- HTMLのHEADタグ内にJavascriptのスニペットを挿入
- NPMパッケージによるインストール
- APMのオプションで自動導入

セッションリプレイは別途有効化が必要です。

- [有効化手順](#)

# 補足: Browser Agentのインストール(参考資料)



## [New Relic ファーストステップガイド \(導入資料集\)](#)



## [New Relic Browser インストール手順](#)

# まとめ

# まとめ

- New Relic Synthetic を利用して **ユーザー体験観測の設定** を体験
- New Relic Browser を利用して **どのようなユーザー体験観測ができるか** を体験
- ダッシュボードによる横断的な分析や  
ユーザー体験指標をもとにしたアラート設定については  
以下のハンズオンを合わせて受講いただくことで、より詳しく理解できます
  - NRU300 - ダッシュボード開発と NRQL の基本
  - NRU300 - アラート設計の基本と活用

# NRU GUIDE

NRU300

# New Relic University

## <https://newrelic.com/jp/learn>

New Relic について基本から応用まで学べるコンテンツです

## New Relic University

New Relic One を学ぶ

自分のレベルに合わせて学びはじめよう

弊社では、New Relic One について学べるコンテンツを New Relic University として無償で公開しています。これから使い始める方も、既に習熟されている方も、お客様のペースで、お好きなところから始めて頂けます。



### インストール方法を学ぶ

New Relic One を実際にインストールして実証するための手順書や必要な情報について学びます。



### New Relic One の全体概要を学ぶ (1月中旬公開予定)

New Relic One の全体像やライセンス形態をオンデマンドの動画視聴で学びます。



### New Relic One の主要機能を学ぶ

New Relic One に含まれる TDP/FSON/JAI の3レイヤーにおける主要機能を動画で学びます。



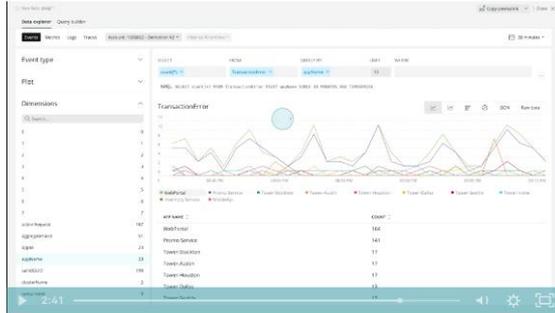
### New Relic One の実践方法を学ぶ

New Relic のエンジニアが提供するハンズオントレーニングを通じて実践操作方法を学びます。

## IN THIS ARTICLE

- ▶ Telemetry Data Platform >
- ▶ Event & Metrics >
- ▶ OSS Data Explorer >
- ▶ Dashboard >
- ▶ Full Stack Observability >
- ▶ Explorer >
- ▶ APM >
- ▶ APM for Serverless >
- ▶ Browser >
- ▶ Mobile >
- ▶ Synthetics >
- ▶ Infrastructure >
- ▶ Distributed Tracing >
- ▶ Workloads >
- ▶ Errors Inbox >
- ▶ Performance >

## Event & Metrics



**Data Explorer でデータを確認する**

New Relic へ送信されたデータを確認する方法の一つであるデータエクスプローラーをご紹介します。データエクスプローラーは New Relic One の画面上で、確認したいデータや条件を選択することで、New Relic にあるデータを簡単に表示することが出来る機能です。

# New Relic University(NRU)とは

New Relicの基礎から応用までを学べ、認定資格も取得できるセルフラーニングコンテンツです

Install	NRU 100	NRU 200	NRU 300/400	Exam
<p><b>New Relicを使い始める</b></p> <p>New Relic へのサインアップやエージェントインストールの方法などのガイドを提供</p> <p><a href="#">APM / Browser / Infrastructure / Logs / Mobile (iOS/Android) / AWS統合 / Azure 統合 / GCP統合</a> インストール手順</p>	<p><b>Observability/New Relicを知る</b></p> <p>New Relic やオブザーバビリティに関する基礎知識を座学にて学習</p> <p><b>NRU Practitioner</b> オブザーバビリティ入門</p> <p><b>NRU 101</b> New Relic 入門</p> <p><b>Hands on for Beginners</b> 初心者向けハンズオン(オフライン)</p>	<p><b>New Relicの主要機能を学ぶ</b></p> <p>New Relicに含まれる主要機能に含まれる54の機能群を動画で説明</p> <p>フロントエンド、バックエンド、アラート、データ分析、ユースケースに応じた機能</p>	<p><b>New Relicの使い方を体感する</b></p> <p>New Relic を実際に操作し、主要機能を利用できる状態にするためのトレーニング</p> <p><b>NRU 300</b></p> <p>アプリケーションとインフラ性能観測の基本</p> <p>Webサイト上のユーザー体験観測の基本</p> <p>ダッシュボード開発とNRQLの基本</p> <p>アラート設計の基本と応用</p> <p><b>NRU 400</b></p> <p>IDEと連携し、問題解決を加速するNew Relic活用の実践</p> <p>SLI/SLO設計の基本</p>	<p>資格を得る</p> <p>New Relicの知識を有していることを証明するための試験、合格すると資格バッジを授与</p> <p><b>NRU for Exam</b> New Relic Full-Stack Observability 認定試験対策講座</p> <p>フルスタックオブザーバビリティ認定試験</p>
<p>▶サインアップ方法</p> <p><a href="https://newrelic.com/jp/blog/how-to-relic/create-new-account">https://newrelic.com/jp/blog/how-to-relic/create-new-account</a></p> <p>▶インストールガイド</p> <p><a href="https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide">https://newrelic.com/jp/blog/how-to-relic/new-relic-faststep-guide</a></p>	<p>▶オンデマンドセミナー</p> <p><a href="https://newrelic.com/jp/resources/presentations/nru-practitioner2022">https://newrelic.com/jp/resources/presentations/nru-practitioner2022</a></p> <p><a href="https://newrelic.com/jp/resources/presentations/nru101-2022">https://newrelic.com/jp/resources/presentations/nru101-2022</a></p>	<p>▶主要機能解説動画</p> <p><a href="https://newrelic.com/jp/learn">https://newrelic.com/jp/learn</a></p>	<p>▶開催スケジュール</p> <p><a href="https://newrelic.com/jp/events">https://newrelic.com/jp/events</a></p>	<p>▶受験サイト</p> <p><a href="https://learn.newrelic.com/full-stack-observability-exam-jp">https://learn.newrelic.com/full-stack-observability-exam-jp</a></p>

本セッションはこちら

# 2025年1-3月期開催の資格取得向けセミナー

## **New Relic Full Stack Observability Practitioner 認定試験対策講座**

開催日程：2025年3月5日（水）15:00-17:00

開催形態：Zoomを使用したオンラインの座学形式

参加費：無料

▼お申し込みはこちらから

<https://newrelic.com/jp/events/2025-03-05/nrufsoexm>

※試験対策講座の内容をより理解いただき、実践力を身につけていただくために、後述のラーニングパスにそった学習を推奨しています

**後述の認定試験の合格に向けた最短の道になります！**

# Observabilityのスペシャリストを目指せ！ Full Stack Observability Practitioner認定試験

## 【この認定試験を通じて身につくスキル】

- Observabilityの実現のためにNew Relicが取得するデータの理解と、目的に応じたデータ分析やアラート設定
- バックエンドおよびフロントエンドの問題発見とトラブルシューティング

## 【認定試験に向けた準備】

- New Relicの基本的な操作経験
- ラーニングパスに沿った学習

## 【合格者特典】

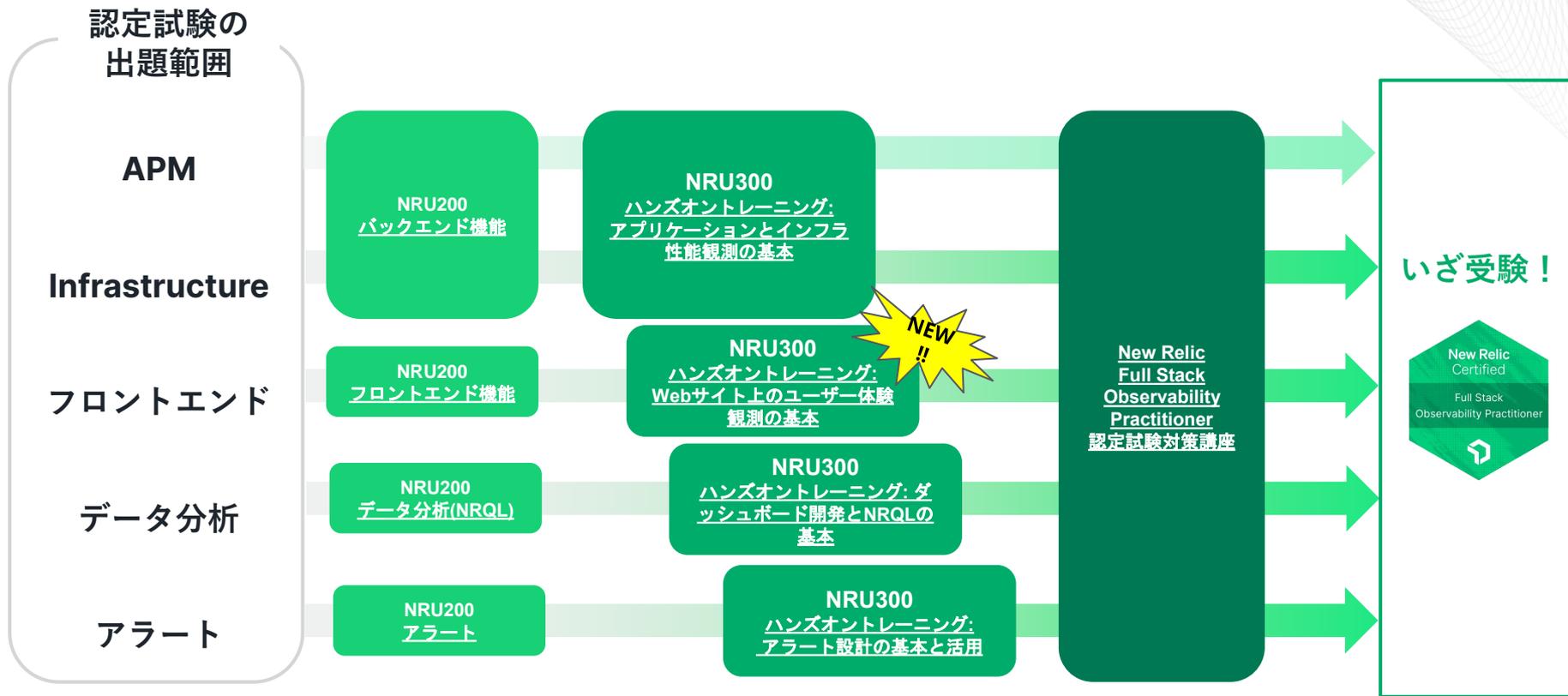
- デジタル認定証とバッジ
  - 合格者限定ノベルティの送付
- ★今だけ認定者限定デザインTシャツ！



認定試験受験  
サイトは [こちら](#)

- 受験料: 無料
- 試験の言語: 日本語
- 受験形態: オンライン(いつでも受験可能)

# 認定資格取得までのラーニングパス(出題範囲別)



# ロール別New Relic ラーニングパス

フロントエンド  
開発エンジニア

サーバーサイド  
開発エンジニア

インフラ運用  
エンジニア

SRE

ネットワーク  
エンジニア

セキュリティ  
エンジニア

プロジェクト  
マネージャー

プロダクト  
マネージャー

Observability  
/New Relicを  
知る

Observability入門

New Relic入門

サインアップ(アカウント開設)

New Relicを  
使い始める

エージェント導入を体験

APMハンズオン  
(Progate社提供学習タスク)

Java バッチの計測ハンズオン

ネットワークモニタリングハンズオン

初心者向けハンズオン ※オフライン開催

ご自身の環境へのエージェント導入

New Relicの  
主要機能を  
学ぶ

フロントエンド機能 ※サーバーサイド/インフラ担当は特にSynthetics受講を推奨

(興味に応じて左記から受講)

バックエンド機能 ※フロントエンド担当は興味に応じて受講

分散トレーシング、Errors Inbox、  
CodeStream

セキュリティ  
機能(作成中)

データ分析(NRQL) / アラート / サービスレベルマネジメント

New Relicの  
使い方を体  
感する

ハンズオントレーニング: アプリケーションとインフラ性能観測の基本  
ハンズオントレーニング: Webサイト上のユーザー体験観測の基本  
ハンズオントレーニング: ダッシュボード開発とNRQLの基本

ハンズオントレーニング: アラート設計の基本と活用

ハンズオントレーニング: IDEと連携し、  
問題解決を加速するNew Relic活用の実践

ハンズオントレーニング: SLI/SLO設計の基本

資格を得る

フルスタックオブザーバビリティ認定試験(受験に向けた講座あり) ※また、受験のためには上記コンテンツをできるだけ網羅的に学習することを推奨

# New Relic サインアップのご案内

- 全機能が無料で使い放題 (1名) !
- 転送データが100GB/月まで無料 !
- 必要事項記入ですぐに利用開始 !
- クレジットカード記入不要 !
- 利用期限なし !
- 抽選でTシャツがプレゼント !

※ [New Relic フリープランで始めるオブザーバビリティ!](#)

応募ステップは下記①と②だけ!

- ① New Relic 無料サインアップ ([Link](#))
- ② New Relic Tシャツ応募フォーム ([Link](#))



[newrelic.com/jp/sign-up-japan](https://newrelic.com/jp/sign-up-japan)

# NRUG

## ぬるぐで学ぶ

### New Relic User Group

New Relic ユーザーが集い、実践事例や最新機能紹介などを実施。初心者支部や SRE 支部などが形成されており、エンジニア同士でのネットワーキングや信頼性の高い情報交換が可能。



NRUG 本部



NRUG SRE 支部



NRUG 沖縄

New!!

NRUG 沖縄支部



以上、お疲れさまでした  
ご質問があればチャットにご記入ください  
アンケートにご協力お願いいたします

Thank you.



以上、お疲れさまでした  
ご質問があればチャットにご記入ください  
アンケートにご協力お願いいたします

Thank you.

