



メディア & エンタテインメント業界における お役立ち新機能のご紹介

2023/5/31

New Relic 株式会社

Solution Consultant 竹澤 拓子



NRUG M&E支部の主要なテーマ

ハイトラフィック

コンテンツの配信開始に伴うユーザーの大量流入や、SNSでバズった時の予測できない突発的なトラフィックに対し、New Relicを駆使してどのように対処したのか。もしくは対処できなかった反省や今後の改善など。

ユーザー体験

多彩なデバイスで提供するWebサービスやコンテンツに対し、ユーザーの満足度をNew Relicでデータとして計測し、どのようにサービスの改善に繋がられたのかという取り組み。

トラブル対応

メディア・エンタメ業界といえばトラブルがつきもの。メトリクスやイベントといったテレメトリーデータに基づいた予測といった取り組みや、事後で対応することになってしまった反省や振り返りなど。

開発体験

リリースサイクルの速い業界において、DevOpsやDevSecOpsの考えに基づき、New Relicを活用していかに関験や開発速度の向上に繋がられたのか。

本日の新機能紹介

今年(2023年1月～)発表された新機能の中から、NRUG M&E支部のテーマに特に役立ちそうなものを抜粋して改めて紹介します。
今日からは是非試してみましよう！

①

ユーザー体験



②

開発体験



③

ハイトラフィック
／トラブル対応



① ユーザ体験

サービスレベルを元にした 3つのアラートを簡単設定

- 複雑なサービスレベルのアラート条件を数クリックで設定できるようになりました
- ファストバーンレート、エラーバジェット消費、SLOコンプライアンスの3つのアラートタイプを選ぶことができます

参照：[バーンレートやエラーバジェット、SLI達成に対するアラート設定が可能になりました](#)

📉ファストバーンレート

- エラーバジェットを1時間以内に2%消費するとアラート
- 消費量が突然大きく変化したことを検知する設定

📊エラーバジェット消費

- エラーバジェットの80%を消費するとアラート通知
- エラーバジェットが無くなりそうなことを検知する設定

⚠️SLOコンプライアンス

- SLOを下回るとアラート通知をする設定
- SLOを遵守できているか知りたい場合に設定

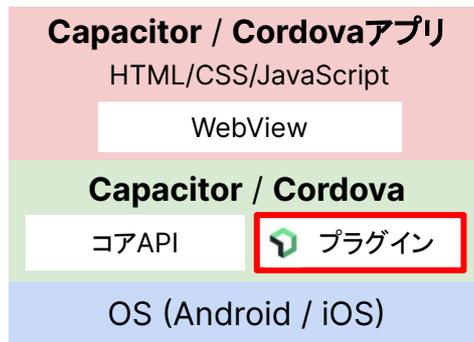
アラートタイプを選ぶだけで、
ビジネスへの影響を考慮した
アラート設定が可能！

Mobileのクロスプラットフォームサポート(1)

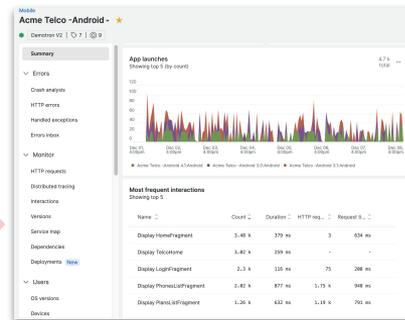
- **Capacitor プラグイン** がリリースされ、ネイティブの New Relic Android / iOS Agent を使用して、Ionic Capacitor 環境を計測可能
- **Cordova プラグイン** の機能強化がされ、Cordova プラグインのアップグレードにより、アプリの Cold/Hot の起動時間の取得が可能

参照:

- [Introduction to New Relic for Capacitor](#)
- [Monitor your Cordova application](#)
- [Capacitor Plugin v1.0.0](#)
- [Capacitor Plugin v1.1.0](#)
- [Cordova Plugin v6.1.0](#)
- [Cordova Plugin v6.0.1](#)



New Relic - Mobile UI



capacitor New!

- JavaScriptエラーのキャプチャ
- 分散トレースの生成
- promise 拒否の追跡
- インタラクションとそのシーケンスのキャプチャ
- ユーザセッションの追跡
- ネイティブ iOS エージェント 7.4.3 にアップグレード
- ネイティブ Android エージェント 6.10.0 にアップグレード



CORDOVA

Update

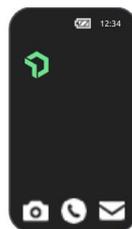
- アプリの Cold/Hot の起動時間の取得
- ネイティブ iOS エージェント 7.4.3 にアップグレード
- ネイティブ Android エージェント 6.10.0 にアップグレード

Mobileのクロスプラットフォームサポート(2)

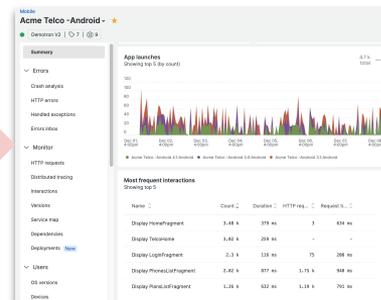
- **Flutterエージェント**がリリースされ、iOS, Androidをサポートする flutter パッケージを提供開始
- **React Nativeエージェント**がv1になり、TypeScriptサポートによるコーディング環境改善や JavaScriptエラーの捕捉が可能に

参照:

- [Monitor your Flutter app](#)
- [Monitor your React Native application](#)
- [Flutter agent release notes](#)
- [pub.dev : newrelic_mobile](#)
- [React Native agent release notes](#)
- [npm package :newrelic-react-native-agent](#)



New Relic - Mobile UI



Flutter **New!**

- dependenciesで指定
- 端末情報や回線情報など標準的な Mobile情報の取得
- httpリクエスト、エラーの捕捉
- Dart Errorの捕捉



React **Update**

- npmで導入
- 端末情報や回線情報など標準的な Mobile情報の取得
- [React Navigation](#), [React Native Navigation](#) に標準対応し、Interactionとして自動計装
- httpリクエスト、エラーの捕捉
- JavaScript Errorの捕捉

②開発体験

変更とその影響を追跡してリリースの迅速化と安定化を両立

- デプロイや機能追加などの変更を記録し、変更による品質影響(性能劣化、エラー、停止)を即座に把握
- アプリだけでなく、インフラやクライアントなどスタック全体での変更をトラッキング
- APIやCI/CDツールのプラグインで簡単にCI/CDプロセスに組み込み可
- リリースの影響を直ぐに把握できるため、精神的安定性を獲得してアグレッシブな変更を実現できます

参照: [公式ドキュメント](#)



Code Level Metrics をIDE上で確認

- お使いのIDE上でパフォーマンス情報にダイレクトアクセス
- メソッドレベルで過去 30分間を元にしたメトリクスを確認可能 (Response Time / Throughput / Error rate)
- **.NET、Python、Ruby、Go、Java、PHP、Node.js[†]**の各種 Agent で使用可能

参照:

- [リリースノート](#)
- [IDE 上に表示されるゴールドシグナルでアプリのパフォーマンスを改善する](#)

[†].Net、Ruby、Java、**PHP**はデフォルトで有効化

New!

メソッドレベルでメトリクスを表示！

```

44 avg duration: 0.010ms | throughput: 0.433rpm | error rate: 0.433epm - since 30 minutes ago
45 @app.route("/external/error")
46 def external_error():
47     req = requests.get("http://localhost:8000/error")
48     req.raise_for_status()
49
50     return req.text
51
  
```

▼ OBSERVABILITY workloads-api-java

- > cheese-spread (staging) Workloads (APM)
- ▼ cheese-spread (production) Workloads (APM)
 - ▼ Golden Metrics ⓘ

Response time (ms) ⓘ	145.34 ms
Throughput ⓘ	43.67 rpm
Error rate ⓘ	0 avg
 - ▼ Related Services
 - ▼ Calls
 - > cheese-service1 Workloads (APM)
 - > cheese-service2 Workloads (APM)
 - ▼ cheesy-web Workloads (APM)

Response time (ms) ⓘ	105.37 ms
Throughput ⓘ	2110.37 rpm
Error rate ⓘ	0 avg

関連しているサービスの情報も
IDE上で可視化！！

③ ハイトラフィック／トラブル対応

分散トレーシングへのアクセス方法が多様化

- Errors Inbox、Logs から分散トレーシングを表示できるようになりました
- 対象のトレースを見失わずに、少ない操作で掘り下げて調査することが可能です

参照: [公式ドキュメント](#)

Errors Inbox

Logs

分散トレーシング

ログに含まれるIPアドレスから位置情報を抽出

- IPアドレスから **位置情報を抽出** するログ解析ルールが追加
- 抽出した位置情報を使用して **国や市などの分析** が可能
- 不正アクセスなどの検出で利用可能

参照: [公式ドキュメント](#)、[Geolocating IP addresses](#)

Parsing rule and output

Add a parsing rule using Grok and see the results in the output

ログ解析ルール

```
%{IP:client_ip:geo({"lookup": ["city","region","regionName","countryCode","countryName","postalCode","latitude","longitude"]})}
```

抽出結果

```
client_ip.region: 13
client_ip.longitude: 139.6899
client_ip.latitude: 35.6893
client_ip.city: Tokyo
client_ip: 54.238.159.145
client_ip.countryCode: JP
client_ip.regionName: Tokyo
client_ip.postalCode: 151-0053
client_ip.countryName: Japan
}
```

ログに含まれるIPアドレスから抽出したい位置情報をParsingルールで1つ以上指定

抽出された位置情報はそれぞれ個別の属性として抽出される

対象の位置情報

- city: 都市名
- countryCode: 国の略称
- countryName: 国名
- latitude: 緯度
- longitude: 経度
- postalCode: 郵便番号、郵便番号など
- region: 州、州、または地域の略語
- regionName: 州、県、または地域の名前

Grok

業界初の生成AIオブザーバビリティアシスタント

New Relic Grok

業界初の生成AIオブザーバビリティアシスタント。チャットインターフェースを介した自然言語とAIを使用して、あらゆるテレメトリデータから深いインサイトを引き出しましょう。

リミテッドプレビュー参加へご登録ください。リミテッドプレビュー開始時にご案内いたします。

リミテッドプレビューに登録



New Relic Grokに聞けることは？



計装およびオンボード

- ✓ 計装（インストールメンテーション）ギャップを特定
- ✓ アラートの対象範囲を改善
- ✓ AWSインテグレーション
- ✓ 新規ユーザーを追加



問題の特定と修正

- ✓ このCPUの急増の原因は何か？
- ✓ カート落ちの問題を修正
- ✓ アプリがデータ送信を停止したか？
- ✓ このエラーの修正方法は？



すばやくインサイトを得る

- ✓ 直近の問題を要約する
- ✓ 先週のアクティブ数は？
- ✓ K8sのデータインテグレーションを見積る
- ✓ 前回のデプロイメント結果は？



翻訳とコラボレーション

- ✓ クエリ結果を翻訳する
- ✓ 直近の異常レポートを得る
- ✓ レポート概要を作成する
- ✓ チャットにメンバーを追加する

<参考URL>

- [New Relic Grok](#)
- [業界初の生成AIオブザーバビリティアシスタント、New Relic Grokのご紹介](#)



Thank you!

