

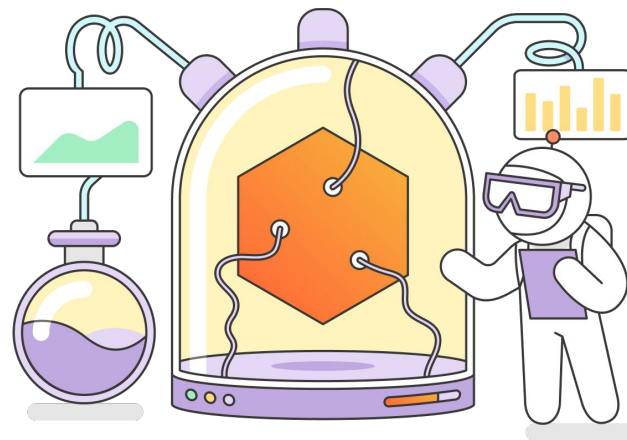


徹底解説！Postmanで APIパフォーマンステスト

Presentation slides for Postman Tokyo Meetup 2023.10



川崎庸市
Postman株式会社

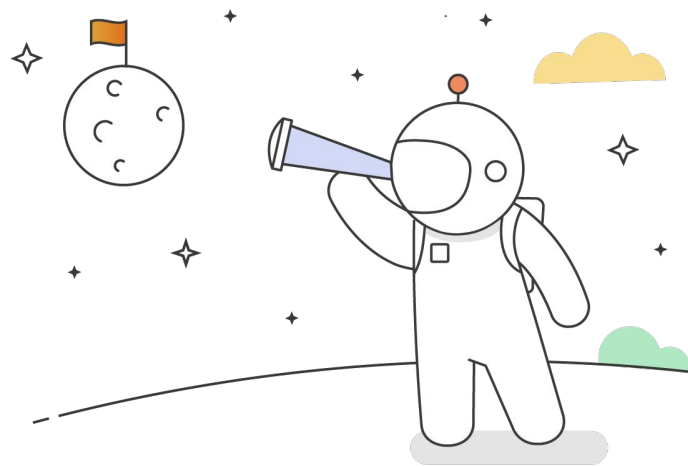




川崎 庸市

テクノロジーエバンジェリスト
Postman 株式会社

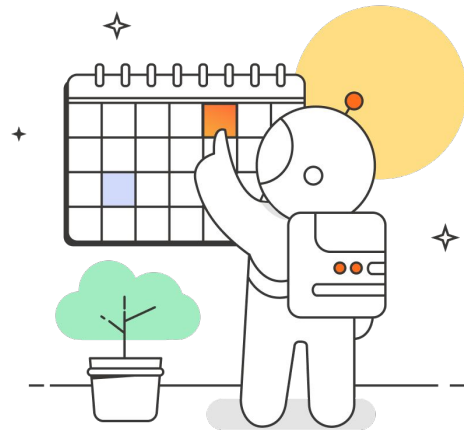
   @yokawasa





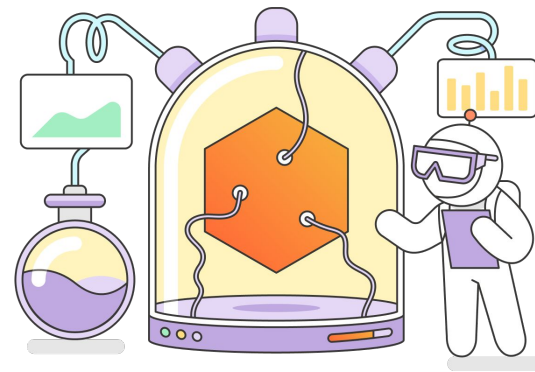
アジェンダ

- パフォーマンステスト概要
- Postmanパフォーマンステスト概要
- Postmanパフォーマンステストデモ



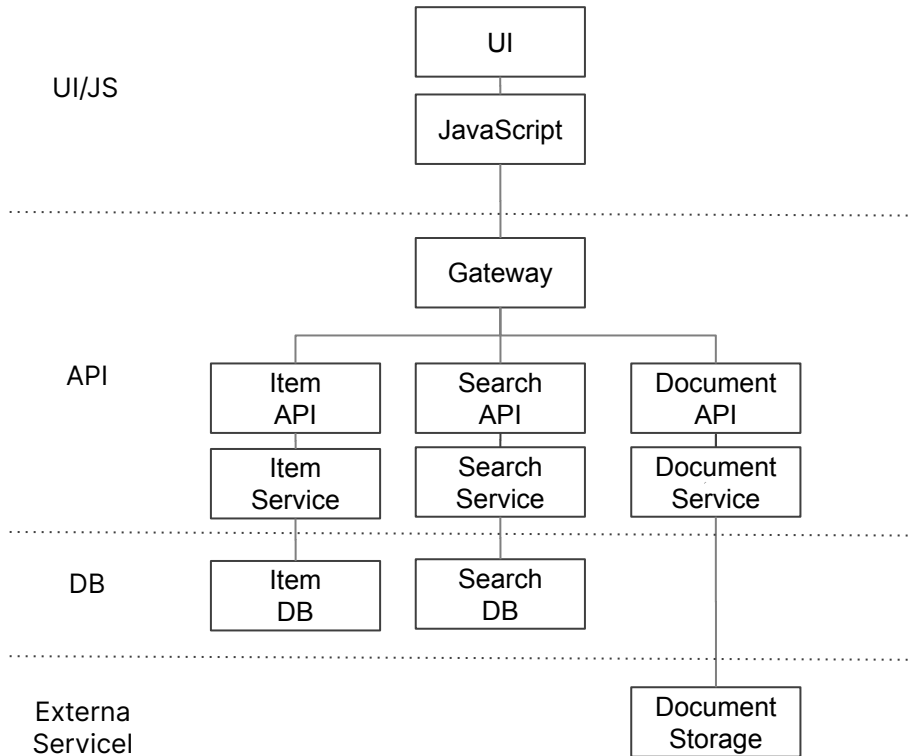


パフォーマンステストとは何か？ なぜ必要か？





APIシステムは複雑である



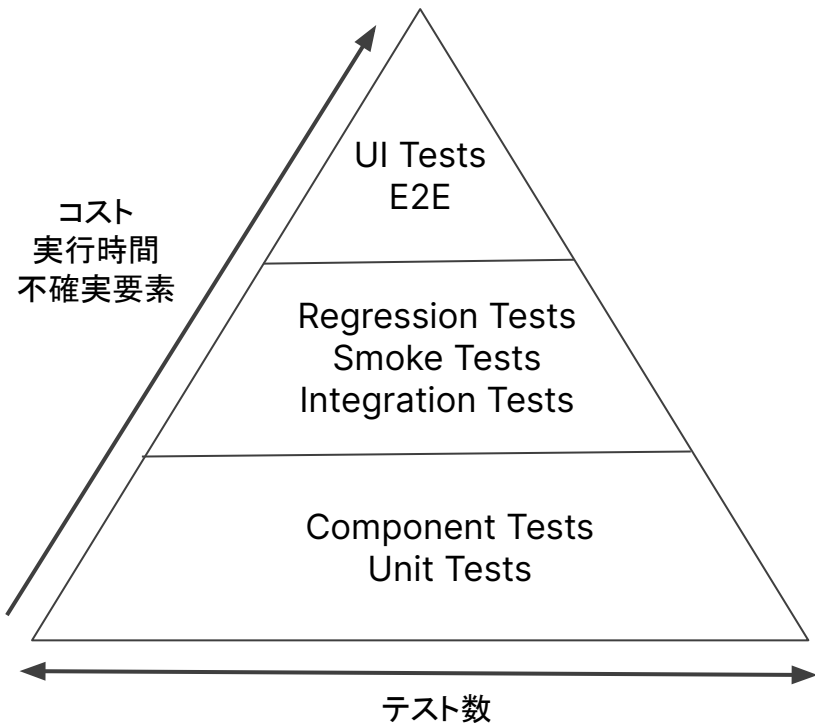
APIシステムは複数要素で構成

- 複数レイヤ(層)で構成
- 複数コンポーネントで構成
- 外部サービスも利用
- エンドユーザーは複数のAPIを組み合わせて利用



品質担保のためには質の高いテストが必要

テストのピラミッド



機能要件テスト

- UIテスト
- E2Eテスト
- リグレッションテスト
- スモークテスト
- インテグレーションテスト
- コントラクトテスト
- コンポーネントテスト
- ユニットテスト

非機能要件テスト

- セキュリティテスト
- ユーザビリティテスト
- パフォーマンステスト



***"Testing shows the presence,
not the absence of bugs."***

Edsger W. Dijkstra at NATO Software Engineering Conference 1969

(意識)テストによってバグの存在が明らかになる

つまり、網羅性のある質の高いテスト戦略が品質保証の鍵となる

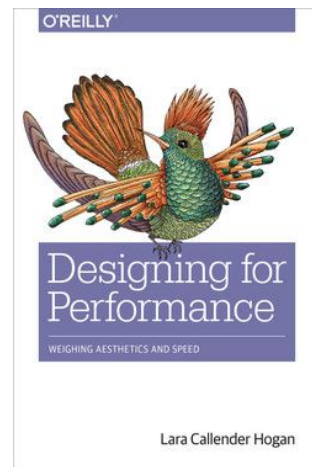


パフォーマンスはユーザー体験である

ソフトウェアやウェブアプリケーションのパフォーマンスはユーザーエクスペリエンスに直結しているとても重要な要素

Web performance is user experience. As you design and develop a new site, you'll consider many components of its user experience: layout, hierarchy, intuitiveness, ease of use, and more. Your site's experience determines how much your audience trusts your brand, returns to your site, and shares it with others.

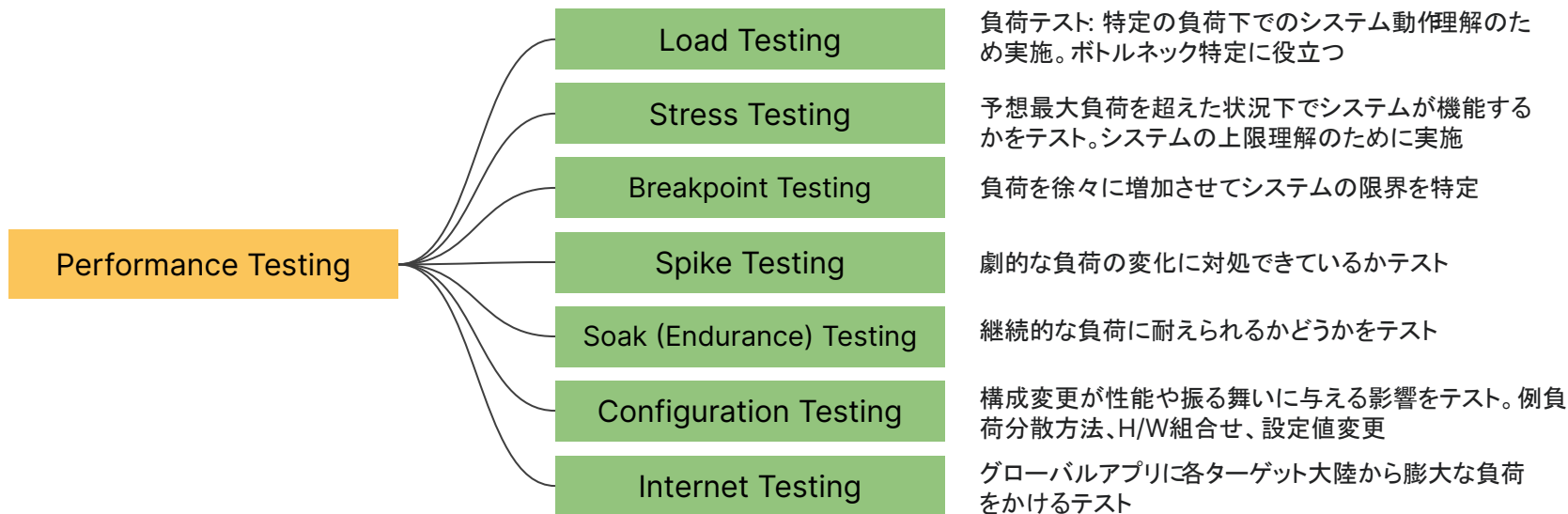
Designing for performance, by Lara Callender Hogan, Dec 2014
<https://www.oreilly.com/library/view/designing-for-performance/9781491903704/>
<https://designingforperformance.com/index.html>





パフォーマンステストとは？

パフォーマンステストは一般に、特定の作業負荷のもとで、システムが応答性と安定性の面でどのように動作するかを判断するために実施されるテストである。また、スケーラビリティ、信頼性、リソース使用量など、システムの他の品質属性を調査、測定、検証、確認することもある



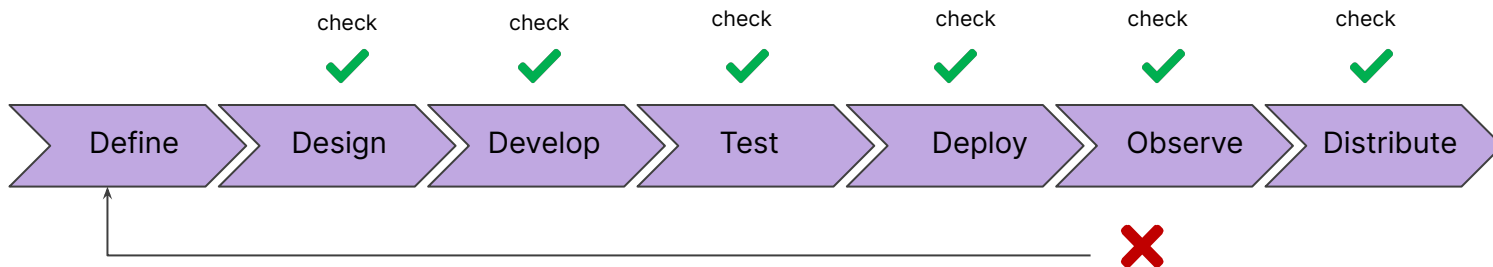


パフォーマンスに関する取り組みはできるだけ早く

他のテスト同様にライフサイクルにおいてできるだけ早い段階(シフトレフト) から 継続的に実施することが重要。 **Test frequently, Fail early**

- パフォーマンス要件
- アーキテクチャ分析
- メトリクス設計・開発

- パフォーマンス監視
- ボトルネック検知・分析
- フィードバック

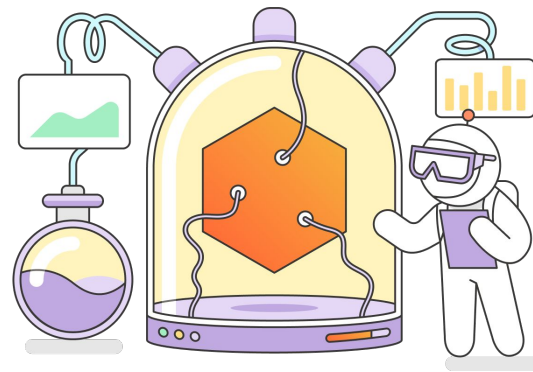


後ろのフェーズになればなるほど改修コストが大きい



Postman

パフォーマンステスト





Postmanのテスト関連機能・コンポーネント

- 変数・スクリプト
 - 変数とスクリプト を活用してAPIリクエストごとの自動処理やテストの記述可能
 - リクエスト送信前(pre-request)とレスポンス取得後(test)の2フェーズで実行可能
- コレクションランナー
 - コレクションやフォルダーのAPI群のテストをまとめて実行するコンポーネント
 - 実行可能テストタイプ
 - Functional : 機能テスト(手動 or スケジュール実行)
 - Performance : パフォーマンステスト
- モックサーバー
 - モック用コンポーネント
- モニター
 - 監視・観測用コンポーネント。スケジュール実行でコレクションテストを実行し結果を可視化
- Postman CLI / Newman
 - コマンドライン・インターフェース。シェルやプログラム実行、CI/CDなど継続的自動実行用



Postmanパフォーマンステストとは？

コレクションもしくはフォルダーに登録されているAPIリクエスト群に対して、手軽にAPIクライアントから負荷をかけ、APIの性能フィードバックをリアルタイムで得られる機能

コレクションランナー実行タイプ設定

Functional **Performance**

Set up your performance test

Virtual users

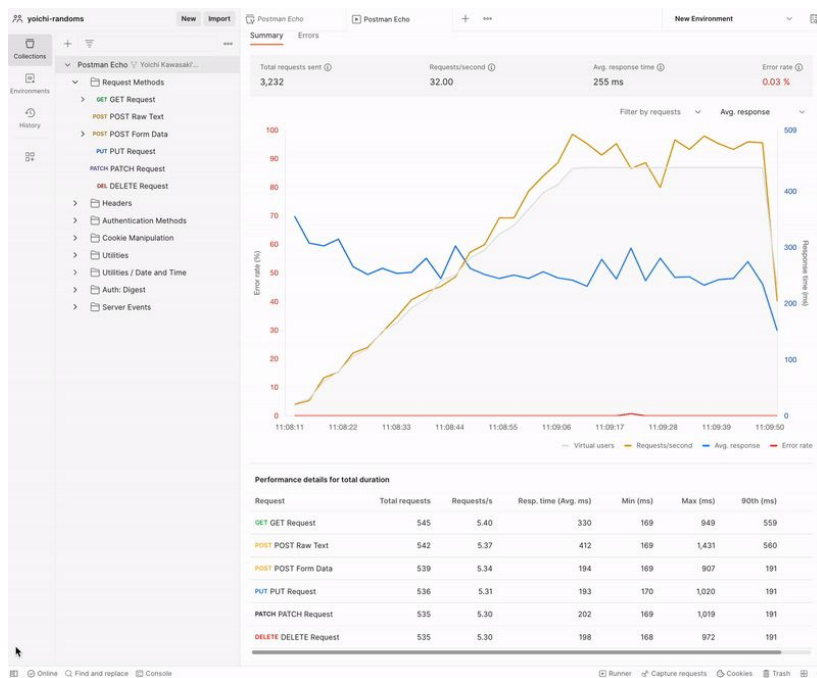
Test duration mins

Load Profile Fixed Ramp up

Ramp up duration mins

Run

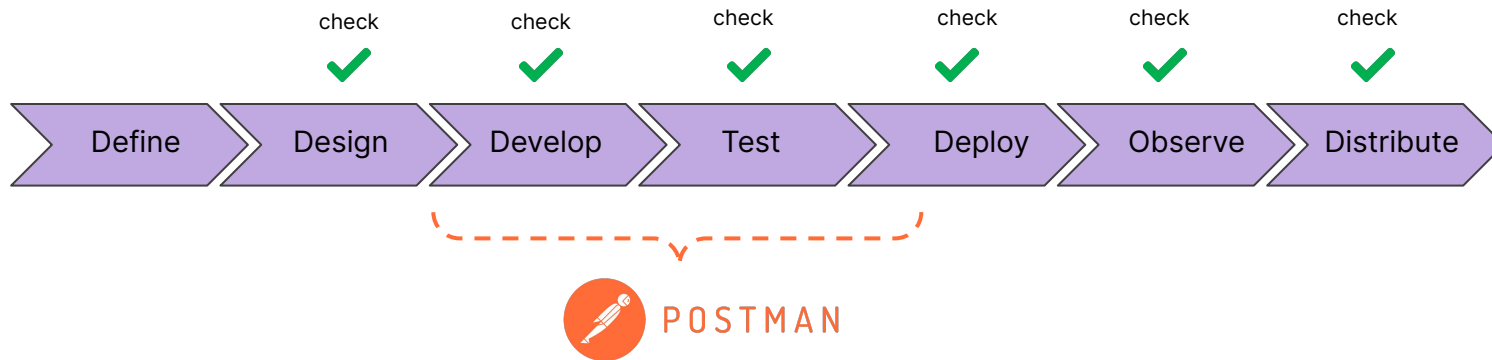
Preview
Gradually increase the number of parallel users to 20 in the first minute of the test.





Postmanパフォーマンステストのスイートスポット

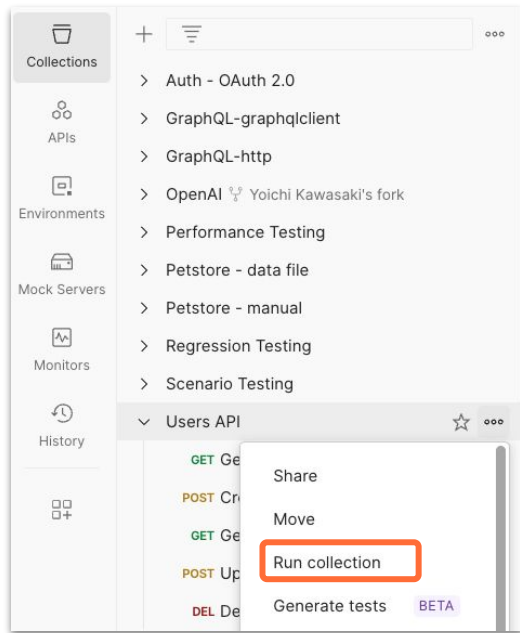
- 大規模向けではなく、小～中規模向けパフォーマンステスト
 - APIクライアントからのテストであり実現可能な出力には限界がある
- 開発ライフサイクルの比較的早い段階で実施するパフォーマンステスト
 - APIの機能テスト用と同じツールセットで、敷居低くカジュアルに実行できる



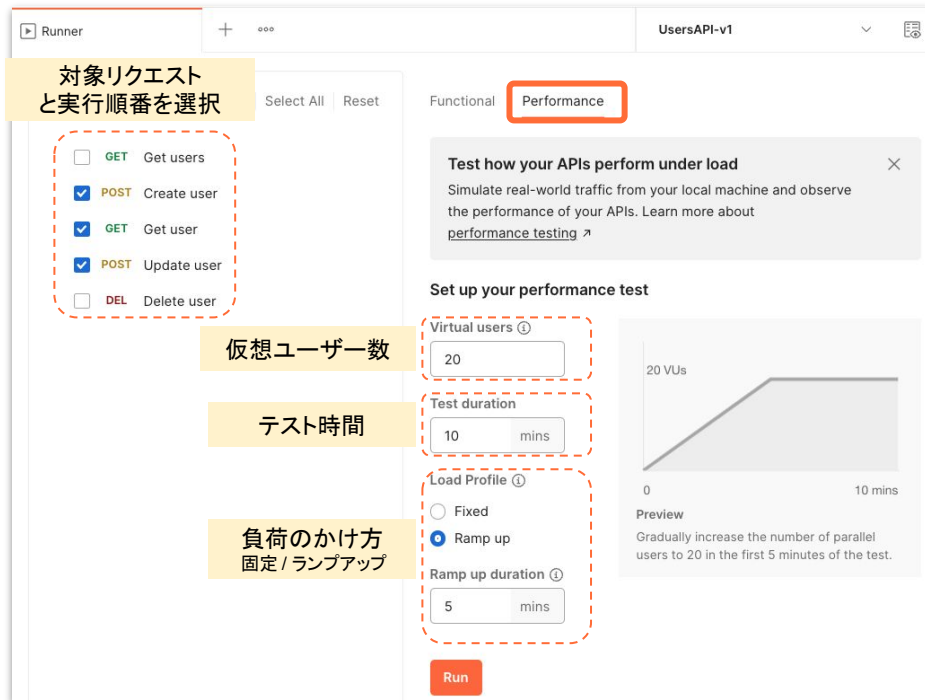


パフォーマンステストのセットアップ

(1) コレクションランナー立ち上げ



(2) パフォーマンステストの設定





リアルタイムメトリクス表示



Search

- Avg. response
- ✓ Min response
- Max response
- 99th percentile
- 95th percentile
- 90th percentile

レイテンシーの
種類切り替え

メトリクスのリアルタイム描画

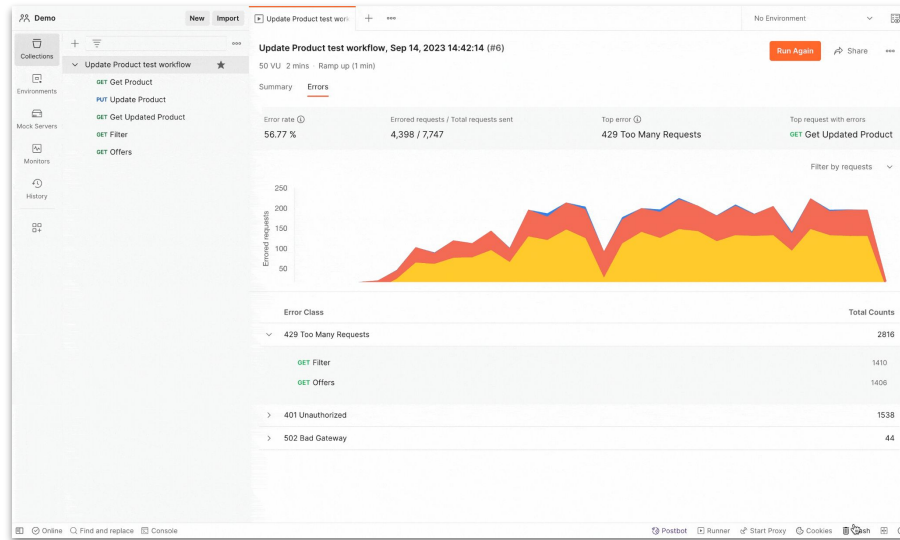
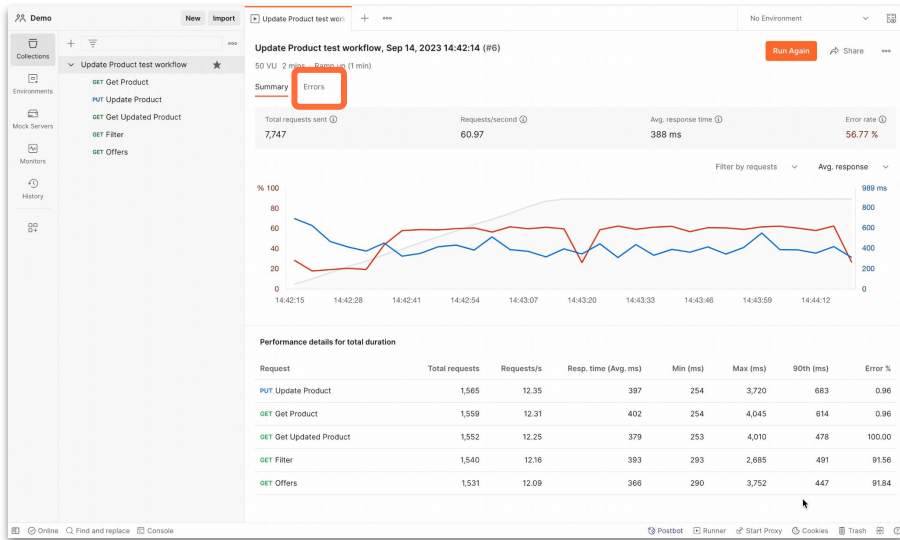
- スループット (req/sec)
- レイテンシー (ms)
- エラー率 (%)

リクエストごとのサマリー一覧



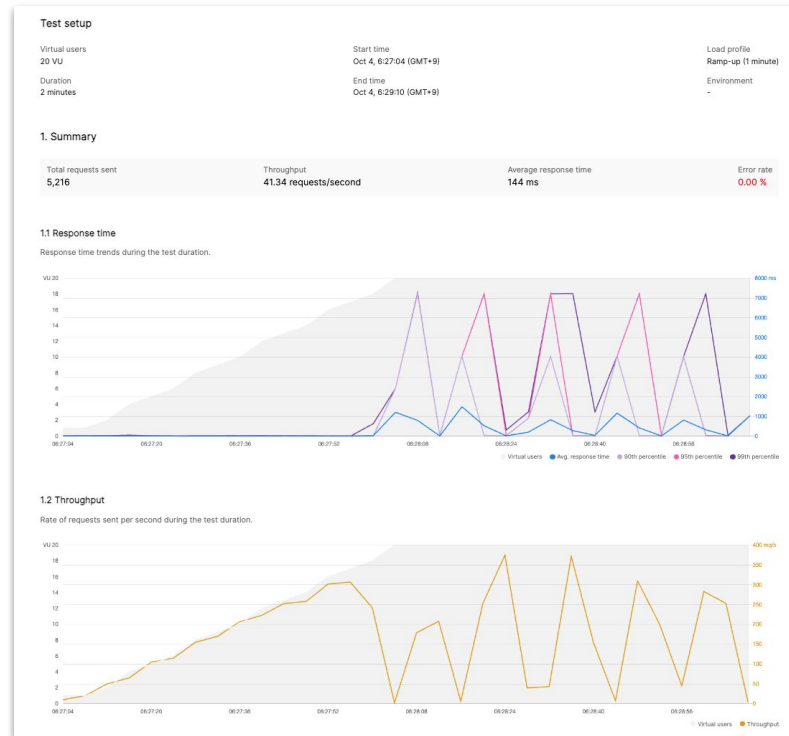
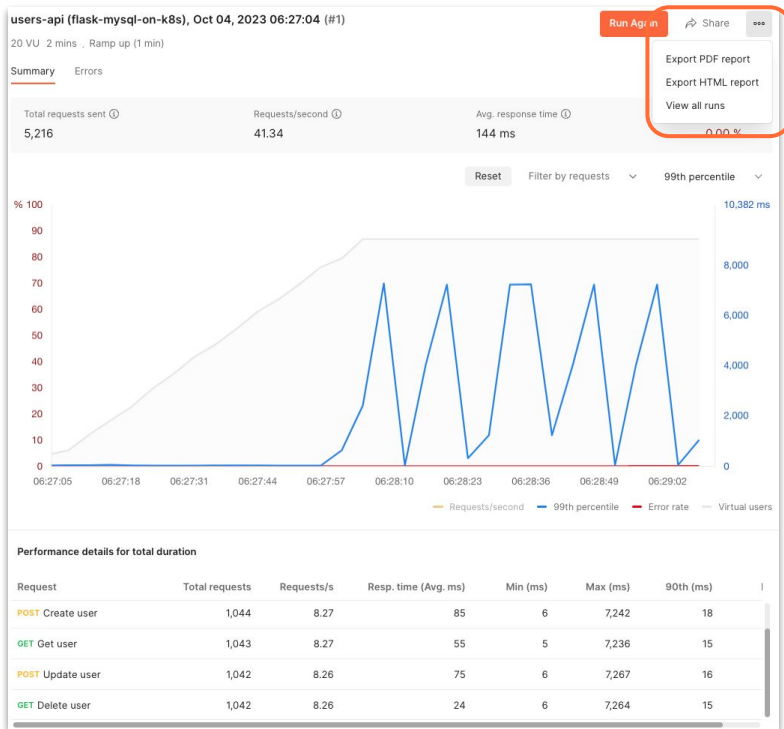
エラーリクエストのドリルダウン

発生したエラーリクエストのステータスと時間の確認、また個別のエラーリクエストの詳細についてドリルダウンが可能





結果レポート - リンク共有 or PDF / HTMLファイル出力





過去のテスト結果一覧

過去の結果は、コレクションのRunsタブ選択 → Performanceタブ選択で確認可能

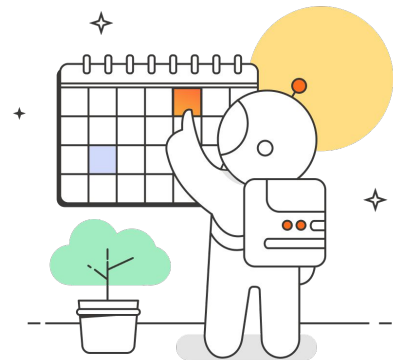
The screenshot shows the Postman interface for a collection named 'Postman Echo'. The 'Runs' tab is selected, and the 'Performance' sub-tab is active. A table displays the performance test results for this collection.

Run	Start time	VUs	Duration	Total requests	Requests/s	Resp. time (Avg ms)	Error %
#9	Jun 20, 2023, 11:08:09	20	2m	4,418	34.79	257	0.02
#8	Jun 20, 2023, 11:05:10	20	2m	4,405	34.70	258	0.02
#7	Jun 20, 2023, 11:00:41	10	2m	1,138	13.71	268	0.09
#6	Jun 19, 2023, 23:32:31	20	2m	3,085	24.30	246	0.03
#5	Jun 19, 2023, 23:29:03	20	2m	4,525	35.63	247	0.02
#4	Jun 19, 2023, 23:06:27	20	2m	2,969	23.69	247	0
#3	Jun 19, 2023, 22:56:49	20	3m	4,306	23.16	255	0.14
#2	Jun 19, 2023, 22:38:23	5	1m	427	6.39	261	0
#1	Jun 19, 2023, 22:28:05	10	2m	1,443	11.53	271	0



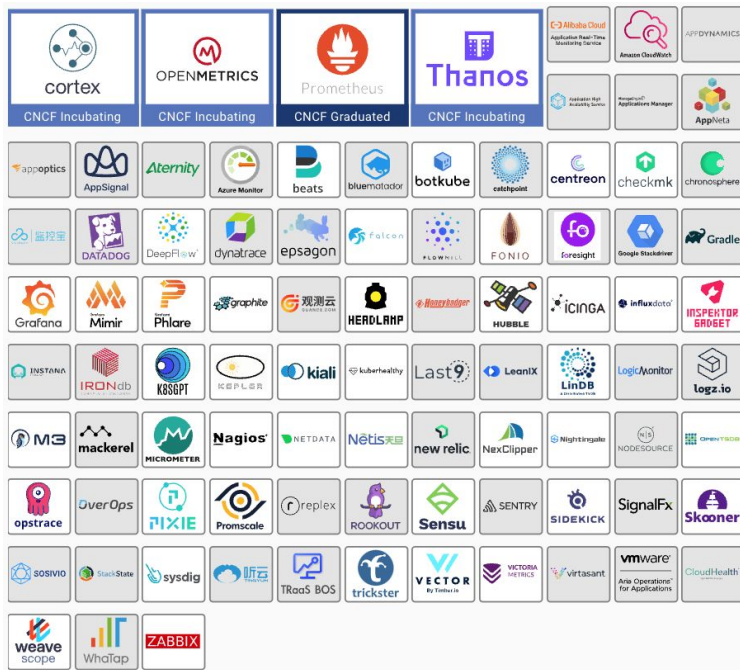
Postmanパフォーマンステストの制約、注意点

- Postmanデスクトップアプリからのみ実行可能
 - CLIからの実行ができず、CI/CDパイプラインなどからの実行が難しい
- プランごとの仮想ユーザー数の制限
 - Postmanプランごとに使用できる仮想ユーザー数に制限がある。無料プランの場合は最大 100 仮想ユーザー。それ以上の仮想ユーザー数は[有料プラン](#)へのアップグレードが必要
- シミュレーション可能な仮想ユーザー数に影響を与える要素
 - 実行するクライアントマシンのシステムリソース(CPU、メモリーなど)
 - マシン性能と仮想ユーザー数の目安については [こちら](#)
 - テストで使用するコレクションの内容(複雑度)
 - 各APIリクエストのpre-requestやtestスクリプトなど



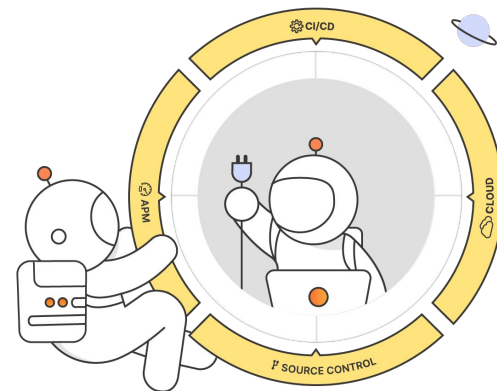


アプリケーション内部の状況やボトルネックの詳細調査には APMなどオブザーバビリティツールの活用が必要



CNCF landscape - observability & analysis
<https://landscape.cncf.io/guide#observability-and-analysis>

どこでエラーになっているのか？
どれだけ時間がかかっているのか？
どれだけリソース消費されているのか？



デモ

ユーザー情報 **CRUD API**の簡易シナリオに対して
Postmanパフォーマンステストを実行し性能やボトルネックを確認

デモ環境

- アプリフレームワーク: Python Flask
- データベース: MySQL
- デプロイメント環境: ローカルKubernetes (KIND)

<https://github.com/yokawasa/users-api-flask-mysql>





ご清聴いただき、ありがとうございました

