

FUJITSU Hybrid IT Service FJcloud-O

API Management

機能説明書

第 1.9 版

富士通株式会社

Copyright 2016-2024 FUJITSU LIMITED

FUJITSU Hybrid IT Service FJcloud-O
API Management 機能説明書 第 1.9 版

発行日 2021 年 6 月

All Right Reserved, Copyright© 富士通株式会社

●本書の無断複製・転載を禁じます。

更新履歴

| 版数 | 作成日 | 更新内容 |
|-----|------------|---------------------------------|
| 1.0 | 2016/2/1 | 新規作成 |
| 1.1 | 2016/7/1 | Standard 提供に伴う更新 |
| 1.2 | 2016/9/16 | バージョンアップに伴う機能・画面の更新 |
| 1.3 | 2016/11/25 | ゲートウェイ拡張提供に伴う更新 |
| 1.4 | 2017/1/23 | WebAPI 機能およびフルアナリティクスの提供に伴う更新 |
| 1.5 | 2018/9/14 | JWT ポリシー追加に伴う更新 |
| 1.6 | 2018/11/9 | Analytics データ保持についての記載を現在の仕様に修正 |
| 1.7 | 2019/12/2 | 東日本リージョン3、西日本リージョン3 対応に伴う更新 |
| 1.8 | 2020/8/13 | サービス名称を変更 |
| 1.9 | 2021/6/1 | バージョンアップに伴う機能・画面の更新 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

はじめに

本書は、FUJITSU Hybrid IT Service FJcloud-O API Management(以下、本サービス)を利用していただくための、操作方法や使用例について記載したマニュアル(以下、本マニュアル)です。本サービスは、様々な Web サービスの API を管理する統合 API プラットフォームです。

本サービスでは、プラン及びオプションによってご利用いただける機能が異なります。本書では、特定のプラン及びオプションでご利用いただける機能については、以下のアイコンで表現しています。

Pro

Pro のプランでご利用いただける機能です。

(東日本リージョン 1 のみ)

フルアナリティクス

フルアナリティクスオプションでご利用いただける機能です。

(東日本リージョン 3、西日本リージョン 3 のみ)

※ アイコンがない機能については、すべてのプラン及びオプションでご利用いただけます。

お願い

- 本書の内容について予告無く変更を行うことがありますのでご了承ください。

目次

| | | |
|------|-----------------------|----|
| 第1章 | 基礎情報 | 6 |
| 第1節 | 固有概念 | 7 |
| 第2節 | 構成 | 8 |
| 第3節 | バックエンドサービス | 9 |
| 第4節 | 機能概要 | 10 |
| 第2章 | Dashboard 機能 | 11 |
| 第1節 | Dashboard 機能について | 12 |
| 第3章 | APIs 機能 | 14 |
| 第1節 | Policy | 15 |
| 第2節 | Flow | 18 |
| 第3節 | Resource | 19 |
| 第4節 | API Proxy 開発のライフサイクル | 20 |
| 第5節 | トレース機能 | 22 |
| 第4章 | Publish 機能 | 25 |
| 第1節 | Publish 機能について | 26 |
| 第2節 | Products | 27 |
| 第3節 | Developers | 29 |
| 第4節 | Developer Apps | 31 |
| 第5章 | Analytics 機能 | 34 |
| 第1節 | Proxy Performance | 35 |
| 第2節 | Target Performance | 37 |
| 第3節 | Cache Performance | 39 |
| 第4節 | Latency Analysis | 41 |
| 第5節 | Error Analysis | 42 |
| 第6節 | Developer Engagement | 44 |
| 第7節 | Traffic Composition | 47 |
| 第8節 | Business Transactions | 51 |
| 第9節 | Devices | 53 |
| 第10節 | Custom Report | 55 |
| 第6章 | Admin 機能 | 59 |
| 第1節 | Admin 機能について | 60 |
| 第2節 | Organization History | 63 |

第1章

基礎情報

ここでは、API Management の機能を知るにあたって
必要となる情報について説明します。

第1節. 固有概念

機能の理解に必要なAPI Management 固有の概念について説明します。

- API Proxy

API Proxy は API Management のコアとなる要素で、バックエンドサービスとそれを利用するアプリとの間に位置し、アプリからのリクエストをバックエンドサービスに渡すタイミング及びバックエンドサービスからのレスポンスをアプリに返すタイミングで作用します。

API Management の機能を利用するためにはまず API Proxy を作成し、作成した API Proxy に対して Policy (バックエンドサービスへの付加機能) を設定します。

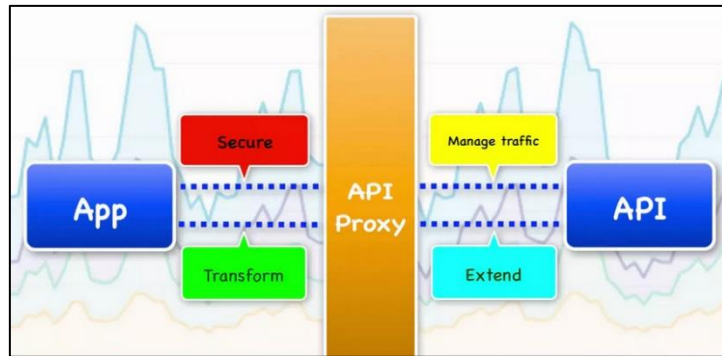


図 1-1 : API Proxy

- Policy

Policy はバックエンドサービスに対する付加機能であり、API Proxy に対して設定します。認証、Traffic 制限、キャッシュ、レスポンスの形式変換などの機能 (Policy 参照) をコーディングなしでバックエンドサービスに追加できます。

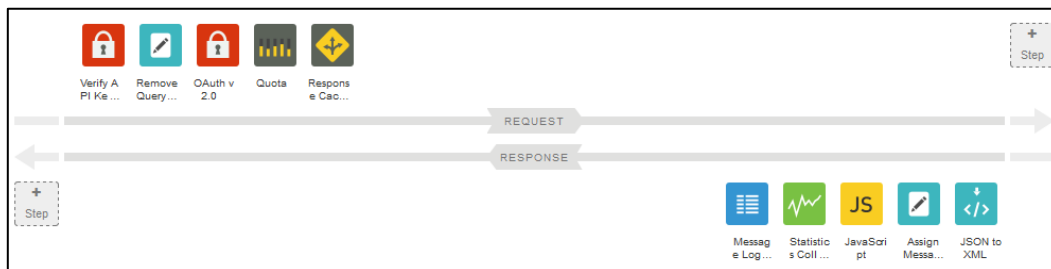


図 1-2 : Policy

第2節. 構成

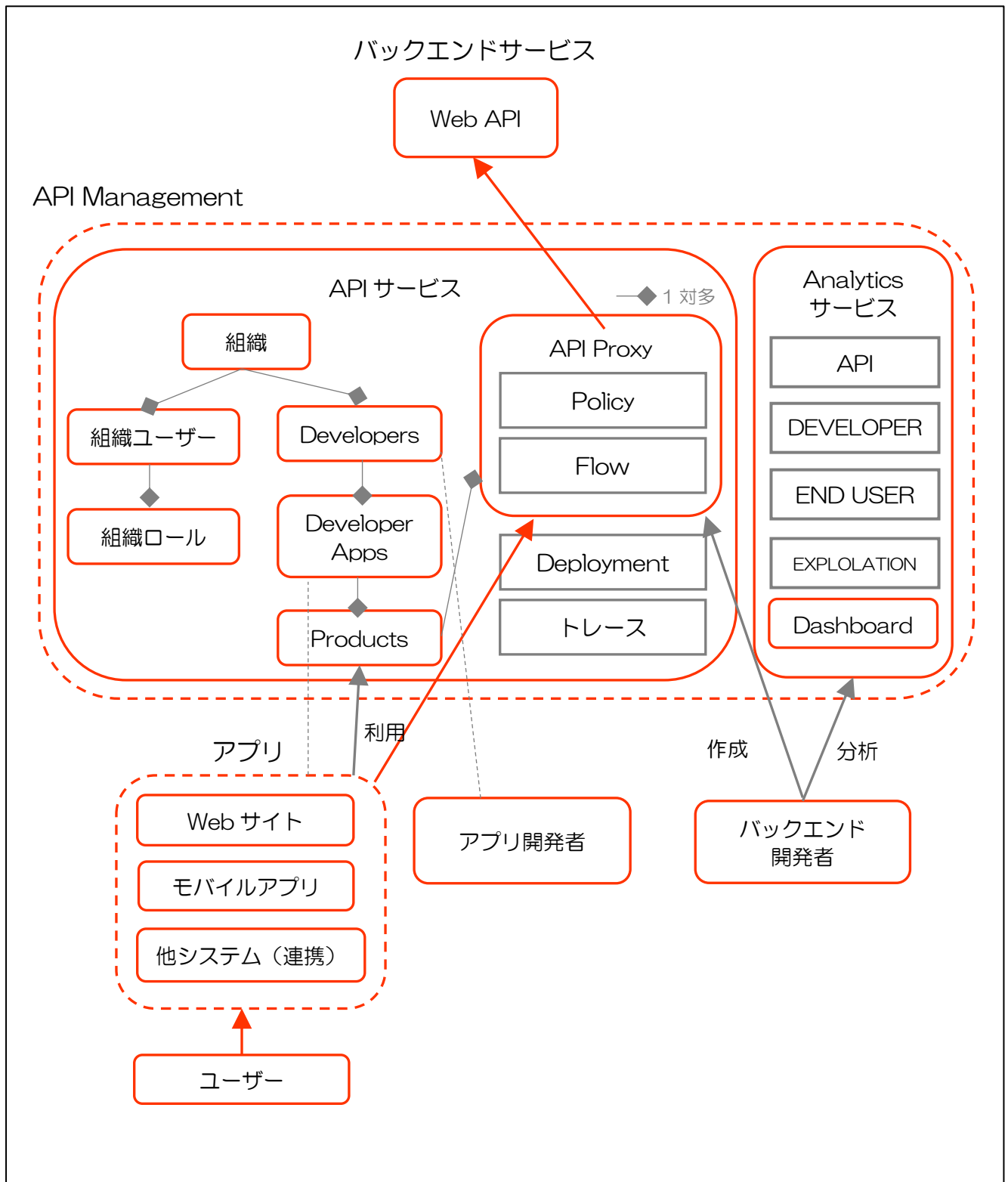


図 1-3 : 主要なコンポーネント及び機能の構成図

第3節. バックエンドサービス

API Management のバックエンドに指定できるサービス (バックエンドサービス) について説明します。

Web アクセスが可能なサービスであれば、バックエンドサービスとして指定することが可能です。EXTENSION Policy や Node.js を使用することで、複数のバックエンドサービス呼び出しによるデータのマッシュアップや、Web API 以外のバックエンドサービスの呼び出しが可能になります。なお、単一の Web API であれば、EXTENSION Policy や Node.js を使用せずとも呼び出し可能です。また、Node.js により API Management 上にバックエンドサービスを構築することも可能です。

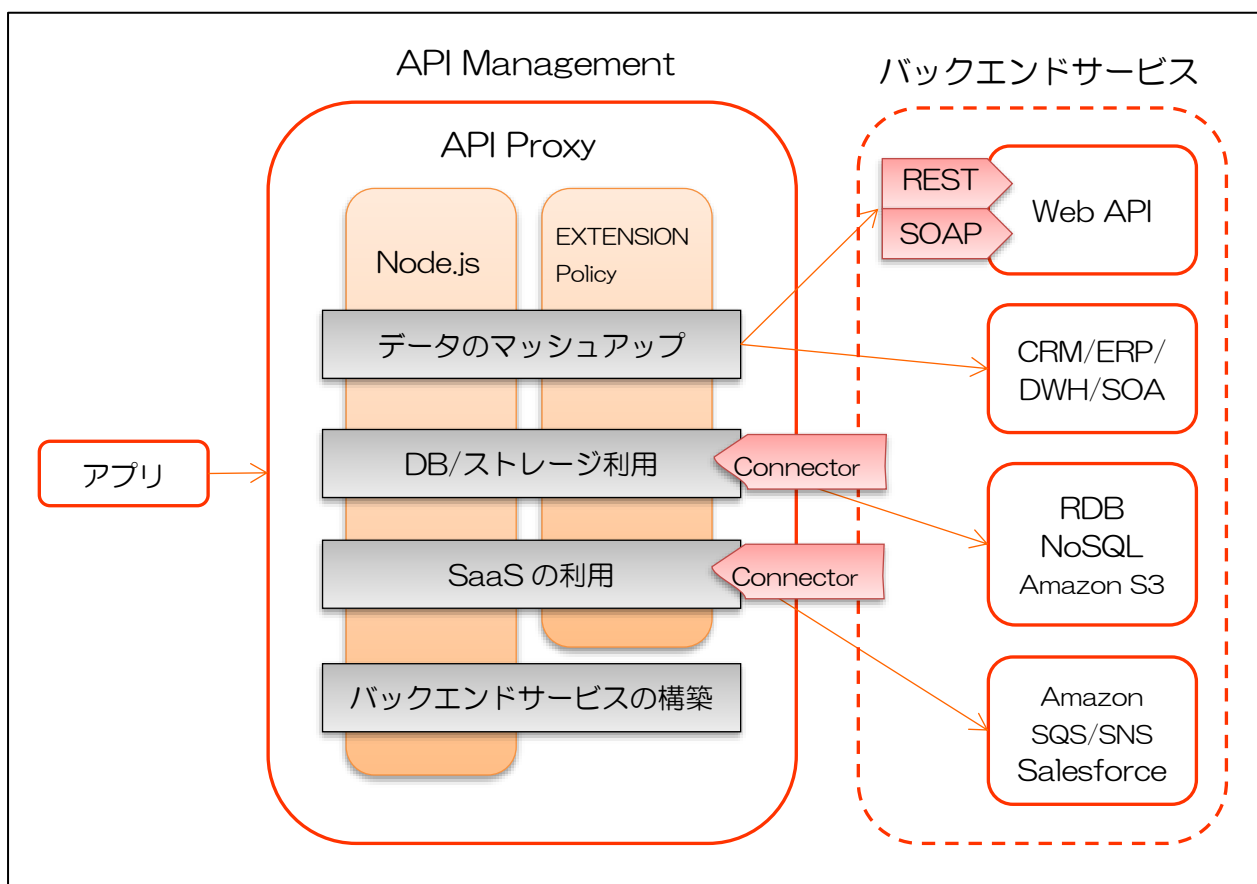


図 1-4 : バックエンドサービス

第4節. 機能概要

API Management は大きく5つの機能に分かれます。各機能の概要は下記の通りです。

表 1-1 : API Management の機能一覧

| 機能 | 概要 |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| APIs | API Proxy の作成がメインの機能であり、作成した API Proxy に Policy、Flow を設定できます。API Proxy 開発のプロセスを最適化するために利用できる、バージョン管理機能やデプロイ機能などの開発ライフサイクル機能や、API Proxy をデバッグするためのトレース機能も備えています。 |
| Publish | アプリから API を利用できるようにする API 公開用の機能です。 API を公開するプロセスは以下の通りです。 1. Products 作成 2. Developers 登録 3. Developer Apps 登録 各コンポーネントの概要は以下の通りです。 <ul style="list-style-type: none">● Product API Proxy を束ねるコレクションです。Product に含まれる API Proxy へのアクセス権（Read/Write）や流量制限の設定が可能であるため、Read のみの Free 版としてや、Read/Write は許すが流量制限をかけるライトユーザー版としての公開が可能です。API の種類や販売プランにより複数の Product を作成できます。● Developers API Proxy（Product）の利用を希望する開発者を氏名やメールアドレスで登録します。Analytics 機能により開発者毎の API 利用量などの確認も可能です。● Developer Apps Developers に登録されたアプリ開発者は、自身が開発したアプリを登録することができます。また、アプリを登録する際に、利用する Product を選択することで API Key が払い出されます。API Key をリクエストに含めることでアプリ開発者は、API の利用が可能になります。 |
| Analytics | リクエストの分析を行う機能です。 Metrics とは、リクエスト数、レスポンス時間、キャッシュのヒット率、リクエストエラー率などを示し、分析のタイプとしてアプリ開発者やアクセス元毎の分析、中央値やパーセンタイル値を用いた API 性能の可視化に加えて、Custom Report でオリジナルの分析レポートを作成することが可能です。 |
| Dashboard | Analytics 機能のグラフの一部が表示されており、API の利用状況をすばやく把握するのに役立ちます。 |
| Admin | API Management のユーザーやロールを管理（作成、変更、削除）できます。 また、組織に対するユーザーの操作履歴の確認が可能です。 |

詳細につきましては、次章以降を参照ください。

第2章

Dashboard 機能

ここでは、API Management の Dashboard 機能について説明します。

第1節. Dashboard 機能について

Dashboard 機能は、API の利用状況を素早く把握したい場合に役立ちます。Dashboard 画面では、利用状況を確認する上で有効な Analytics 機能のグラフ (Proxy Traffic、Developer Engagement、Developer Apps) を図 2-1 のように 1 画面で表示します。

Dashboard 画面に表示される各グラフは、See more details in <graph name> のリンクをクリックすることで、Analytics 機能の該当する詳細ページを表示します。詳細ページにつきましては、第5章 Analytics 機能を参照ください。

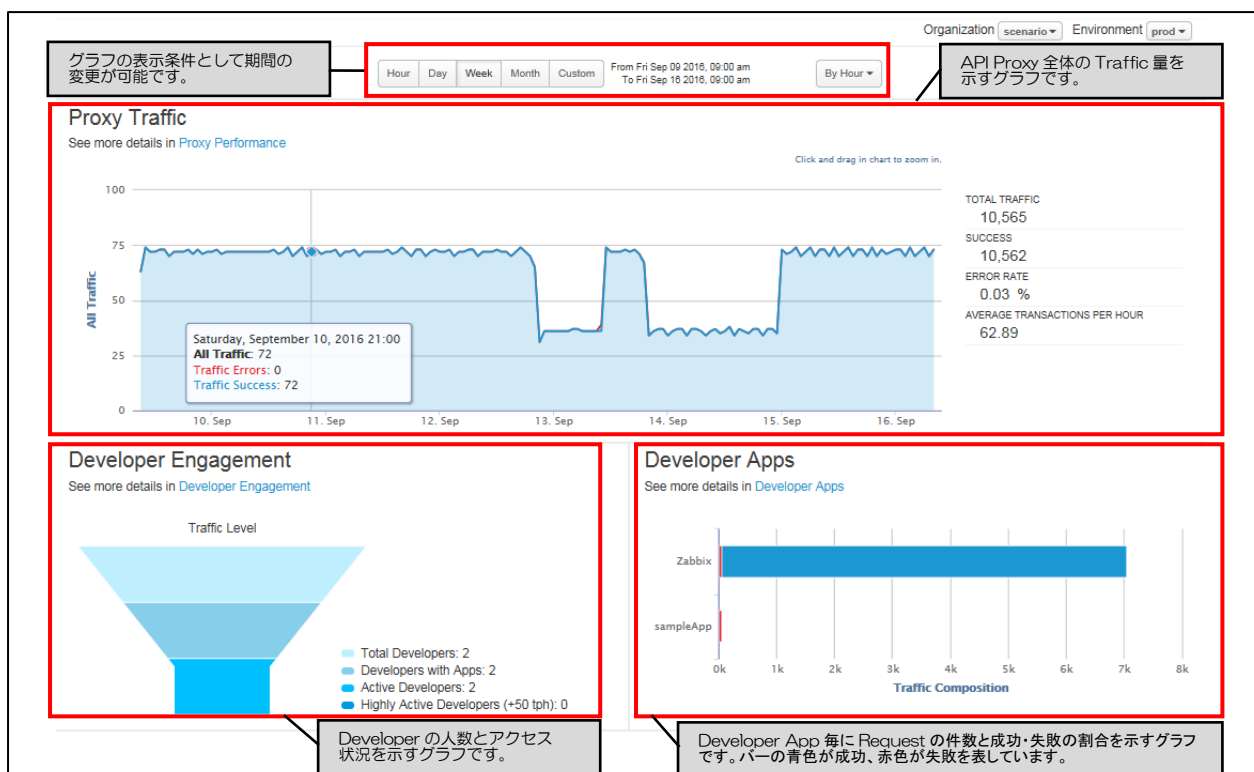


図 2-1 : Dashboard 画面

表 2-1 : Proxy Traffic の項目一覧

| 項目 | 説明 |
|------------------------------|----------------------------|
| TOTAL TRAFFIC | API Proxy 全体のリクエスト数を表示します。 |
| SUCCESS | 成功したリクエスト数を表示します。 |
| ERROR RATE | 失敗したリクエストの割合を表示します。 |
| AVERAGE TRANSACTION PER HOUR | 1 時間あたりの平均リクエスト・レスポンス数 |

表 2-2 : Developer Engagement の項目一覧

| 項目 | 説明 |
|--------------------------|---------------------------------------------------|
| Total Developers | 組織内の API Proxy に関連する Developer の合計人数を示します。 |
| Developers with Apps | 組織内のアプリに関連する Developer の合計人数を示します。 |
| Active Developers | API Proxy 経由でリクエストを送信した Developer の人数を示します。 |
| Highly Active Developers | 1 時間あたりの Transaction 数が 50 以上の Developer 人数を示します。 |

表 2-3 : Developer Apps の項目一覧

| 項目 | 説明 |
|------------|------------------------------------|
| <App name> | アプリの名前と Traffic 量（成功・失敗を含む）を表示します。 |

Reports Dashboard 画面につきましては、第5章 Analytics 機能を参照ください。

第3章

APIs 機能

ここでは、API ManagementのAPIs機能について説明します。

第1節. Policy

Policy は API Management で用意されたモジュールのようなものです。Policy を設定することでプログラムコードを記述することなく、セキュリティ、Traffic の制限、データの変換等を行えます。API Management で用意されている Policy については、下表を参照ください。

表 3-1 : Policy (TRAFFIC MANAGEMENT) 一覧









| Policy | 説明 |
|---------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|  Quota | 時間の単位（月、日、時、分、秒）と回数を指定することで、アプリのリクエスト件数を制限します。 |
|  Spike Arrest | アプリからバックエンドサービスに対する、秒間あたりのリクエスト件数を指定して制限します。 例) 毎分 30 件 (30pm) 許可する設定にした場合、2 秒毎に 1 件許可します。2 秒以内に 2 件来た場合、2 件目は処理されません。 |
|  Concurrent Rate Limit | バックエンドサービスに同時接続できるアプリの数を制限します。 |
|  Response Cache | バックエンドサービスからのレスポンスをキャッシュします。 |
|  Lookup Cache | Populate Cache でキャッシュしたデータを取得します。 |
|  Populate Cache | セッション ID や認証情報等、任意のデータをキャッシュします。 |
|  Invalidate Cache | 条件を指定して Populate Cache でキャッシュしたデータを削除します。 |
|  Reset Quota | Quota でカウントしたリクエスト件数を指定した値でリセットします。 |

表 3-2 : Policy (MEDIATION) 一覧

















| Policy | 説明 |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|  JSON to XML | JSON 形式を XML 形式に変換します。 |
|  XML to JSON | XML 形式を JSON 形式に変換します。 |
|  Raise Fault | ステータスコードに応じてカスタムメッセージが出力できます。 |
|  XSL Transform | XML 形式を HTML やプレーンテキスト等の別フォーマットに変換します。 |
|  SOAP Message Validation | 受信した SOAP メッセージが、XSD スキーマまたは WSDL に準拠していない場合は拒否します。 |
|  Assign Message | HTTP Request または Response メッセージの作成・修正が可能です。 |
|  Extract Variables | リクエストまたはレスポンスが指定した条件（URI Path や Query Param 等）と一致した場合、変数を指定して追加することができます。 |
|  Access Entity | API Management のデータストアから取得した情報（アプリ、API Product、Developer 等）を変数に設定します。 |
|  Key Value Map Operations | PUT、GET、DELETE メソッドで Key/Value のペアを保存・検索・削除することができます。 |

表 3-3 : Policy (SECURITY) 一覧

| Policy | 説明 |
|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
|  Basic Authentication | Basic 認証 (Base64 のエンコードまたはデコード) の設定ができます。 |
|  XML Threat Protection | XML の脆弱性に対する攻撃を防ぐ設定ができます。 |
|  JSON Threat Protection | JSON の脆弱性に対する攻撃を防ぐ設定ができます。 |
|  Regular Expression Protection | 正規表現でリクエストを拒否することができます。 |
|  OAuth v2.0 | OAuth2.0 のエンドポイントに対する設定 (アクセストークンの生成やチェック等) ことができます。 |
|  Get OAuth v2.0 Info | OAuth2.0 のアクセストークンや認証コード等の情報を取得することができます。 |
|  Set OAuth v2.0 Info | OAuth2.0 のアクセストークンに関連付けられたカスタム属性を追加・更新することができます。 |
|  Delete OAuth v2.0 Info | OAuth2.0 のアクセストークンや認証コード等の情報を削除することができます。 |
|  OAuth v1.0a | OAuth1.0a のエンドポイントに対する設定 (アクセストークンの生成やチェック等) ことができます。 |
|  Get OAuth v1.0a Info | OAuth1.0a のアクセストークンや認証コード等の情報を取得することができます。 |
|  Delete OAuth v1.0 Info | OAuth1.0a のアクセストークンや認証コード等の情報を削除することができます。 |
|  Verify API Key | アクセスを許可する API Key を設定することができます。 |
|  Access Control | IP アドレスによるアクセス許可・拒否設定ができます。 |
|  LDAP | LDAP 認証の設定ができます。 |
|  Generate SAML Assesion | 送信する XML Request に SAML Assertion を追加します。 |
|  Validate SAML Assertion | 受信した SOAP Request に添付されている SAML Assertion をチェックし、無効なメッセージの場合は拒否します。 |
|  Generate JWT | 署名付き JWT を生成することができます。 |
|  Verify JWT | 署名付き JWT を検証することができます。 |
|  Decode JWT | 署名付き JWT を検証せずにデコードすることができます。 |
|  Generate JWS *1 | 署名付き JWS を生成することができます。 |
|  Verify JWS *1 | 署名付き JWS を検証することができます。 |
|  Decode JWS *1 | 署名付き JWS を検証せずにデコードすることができます。 |

*1 東日本リージョン 3/西日本リージョン 3 のみ利用可能

表 3-4 : Policy (EXTENTION) 一覧

| Policy | 説明 |
|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Java Callout | <p>ゲートウェイ拡張プランまたはバックエンドセキュア接続プランのみの機能です。Java を使ったカスタム Policy を実装することができます。</p> <p>※Java Callout Policy は API Management の画面から追加することができず、追加用の WebAPI にリクエストを送信する必要があります。詳細は、別紙「API Management WebAPI リファレンス」をご参照下さい。</p> |
|  Python | Python を実行します。 |
|  JavaScript | JavaScript を実行します。 |
|  Service Callout | 外部サービスを呼び出します。 |
|  Flow Callout | Shared Flow を呼び出します。 |
|  Statistics Collector | Analytics 用にメッセージ内のデータ (Product ID、価格、ターゲット URL 等) を収集することができます。 |
|  Message Logging | <p>syslog サーバーにメッセージログを記録することができます。</p> <p>※ローカルディスクへの保存はご利用いただけません。</p> |

第2節. Flow

API Proxy に Policy の実行タイミングを設定することにより、API の振る舞いをプログラムすることができる機能です。

API Proxy はリクエストやレスポンスをパイプライン処理します。パイプラインは API Proxy に定義された一連の Flow から構成され、各 Flow には Policy をアタッチできます。

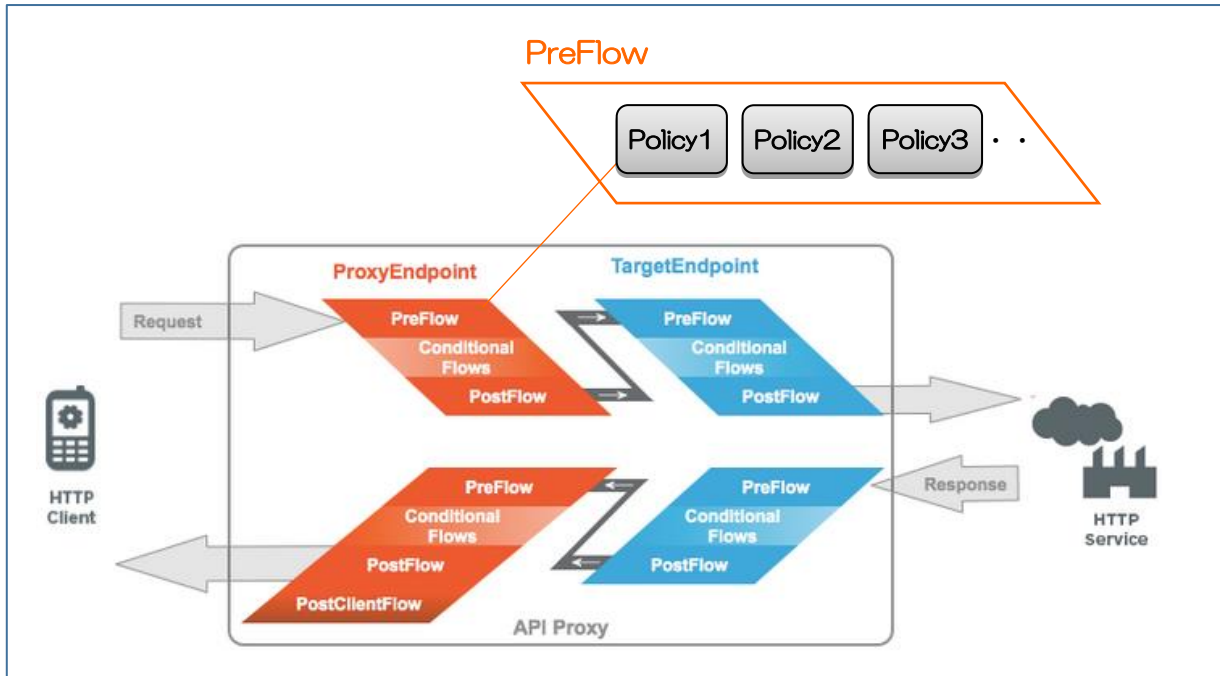


図 3-1 : Flow の実行順

表 3-5 : 各 Flow の概要

| No. | ステージ | 内容 |
|-----|-------------------|--------------------------------------------------------------------------------------------------------------------|
| 1 | PreFlow | Conditional Flow の前に実行されるフロー。 API Proxy の全リクエストに対して実行したい Policy をアタッチします。 アタッチ対象の Policy 例：認証、クォータ制限 |
| 2 | Conditional Flows | 1 つ以上のフローを定義可能。指定された条件が true になる場合にのみ実行されます。 条件の例としては、モバイルアプリの場合、特定のリソースへのアクセスの場合、リクエストボディの内容に応じて処理する場合などがあります。 |
| 3 | Post Flow | Conditional Flow の後に実行されます。 アタッチ対象の Policy 例：ロギング、イベント通知、メッセージ形式の変換 |
| 4 | PostClientFlow | クライアントアプリにレスポンスが返った後に実行されるフローです。主に、クライアントへのレスポンス送信の開始と終了のインターバルを計る用途で使われます。 ※MessageLogging Policy のみアタッチ可能 |

第3節. Resource

Resource を作成することで、Proxy Endpoint の Conditional Flows（条件付きフロー）に定義を追加することができます。追加した定義は、Resource を作成する際に指定する条件（表 3-6 の通り）とクライアントから送信されるリクエストのパターンが一致した場合に処理されます。

表 3-6 : Resource で設定可能な条件

| 条件 | 説明 |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| Path | 対象となるバックエンドサービスのリソースパスを指定します。バックエンドサービスの URL が下記の場合、「/reports」がリソースパスです。 > http://mygreatweatherforecast.com/ reports |
| Method | リクエストに含まれる HTTP メソッドを指定することができます。指定可能な Method は下記の通りです。 > GET > PUT > POST > DELETE > All |

Resource は、作成が完了すると図 3-2 の通り Proxy Endpoint の Conditional Flows に配置され、Policy のアタッチが可能な状態になります。

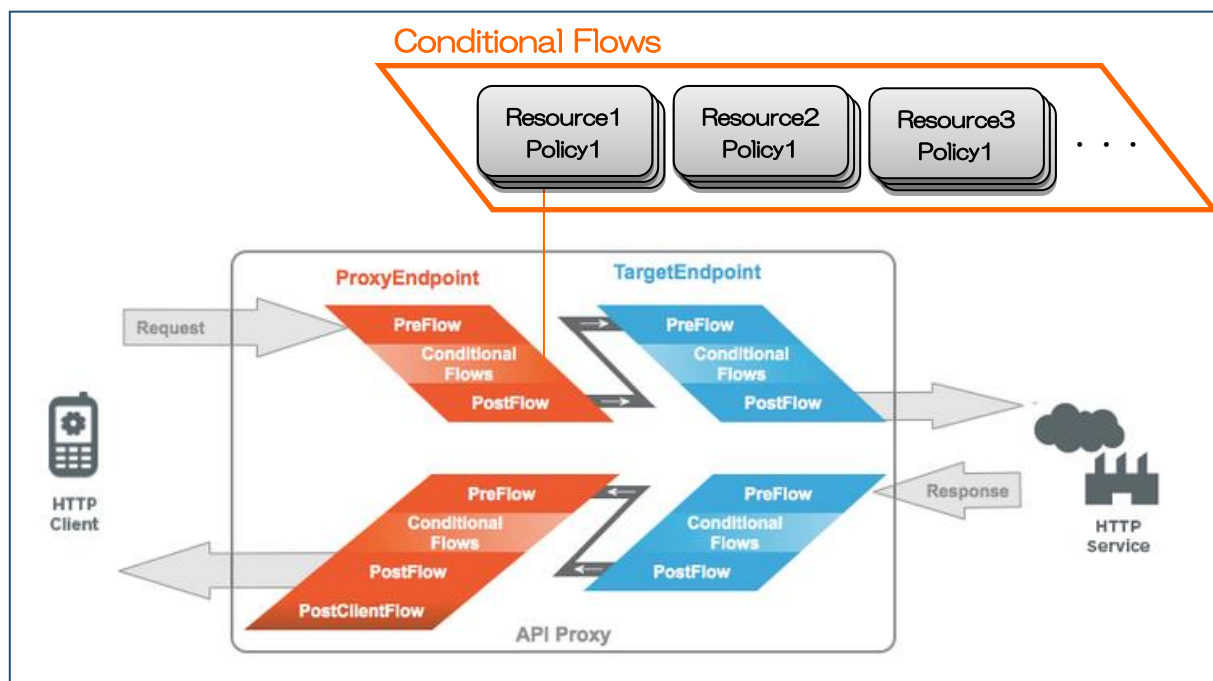


図 3-2 : Resource の概念図

第4節. API Proxy 開発のライフサイクル

API Proxy 開発のライフサイクルの中で利用できる機能について説明します。

本機能により、開発者の組織におけるソフトウェア開発ライフサイクルに沿った API Proxy 開発が可能になります。

表 3-7 : API Proxy 開発のライフサイクルで利用できる機能

| No. | 機能 | | 説明 |
|-----|------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------|
| 1 | API Proxy の デプロイ環境 | 開発フェーズ毎の環境 | 開発環境、テスト環境、性能測定環境、本番環境など開発で必要になる環境を複数利用できます。 |
| | | HTTP/HTTPS | 各環境で HTTP アクセス用と HTTPS アクセス用の2つのエンドポイントが利用できます。 ※基本的には prod 環境では HTTP アクセス用を無効にします。 |
| 2 | API Proxy の デプロイ | シームレスなデプロイ | アプリやユーザーへの影響を最小限に抑えながらデプロイ可能です。 |
| | | 複数バージョンのデプロイ | バージョン違いの API Proxy を同一環境にデプロイ可能なため、マルチバージョン環境を実現可能です。 デプロイ後のバージョン戻しも可能です。 |
| 3 | API Proxy の バージョン管理 | バージョン | アプリ側に改修が必要なる規模の変更が API Proxy に入る際などにバージョンをインクリメントします。 API Proxy のベース URI にバージョン番号が含まれます。 例) api.company.com/v1/... |
| | | リビジョン | バージョンより細かい改訂番号であるリビジョン単位でも管理可能です。 |

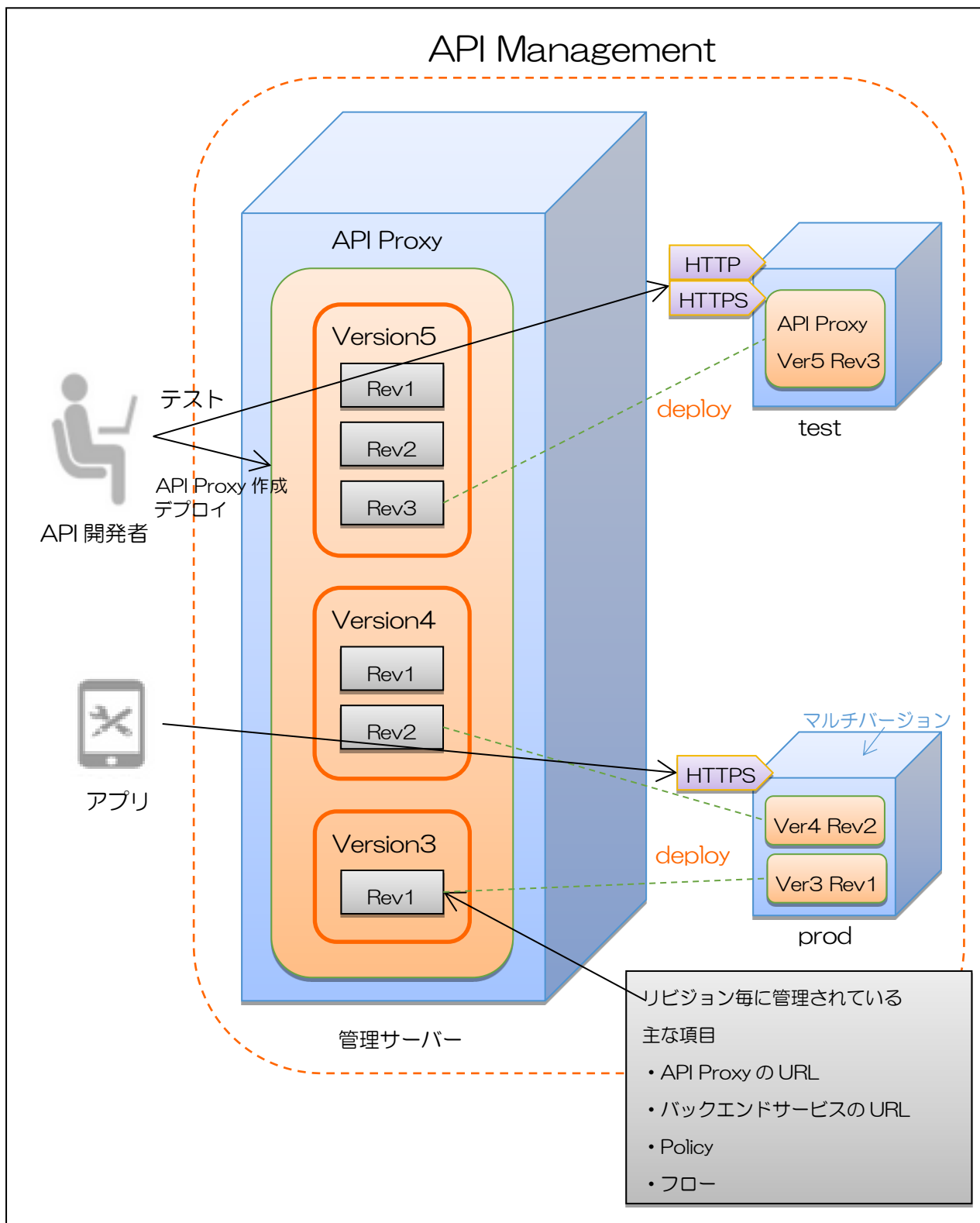


図 3-3 : API Proxy 開発のライフサイクル

第5節. トレース機能

トレース機能は、トラブルシューティングやデバッグ用に提供されているツールです。トレース機能を使うことで、API Proxy に送信されるリクエストおよびレスポンスの Traffic をキャプチャし、Transaction として可視化することができます。トレース機能の画面説明は図 3-4 を参照ください。

The screenshot shows the API Proxy tracing interface. The top navigation bar includes tabs for OVERVIEW, DEVELOP, TRACE, and PERFORMANCE. The main interface is divided into several sections:

- Deployment to Trace:** Environment prod, Revision 1
- Stop Trace Session:** Remaining Time: 09:02
- Download Trace Session** and **Node.js Logs** buttons
- Request の送信フォーム:** Method: GET, URL: http://test2-prod.apimng.com:10080/docs, Status: 200
- Filters:** Name and Value columns for HTTP Header and Query Parameter.
- Transactions:** A table listing transactions with columns for Status, Method, URI, and Elapsed time.

| Status | Method | URI | Elapsed |
|--------|--------|--------|---------|
| 3 200 | GET | /docs/ | 22 ms |
| 2 200 | GET | /docs/ | 21 ms |
| 1 200 | GET | /docs | 20 ms |
- Transaction Map:** A visual timeline of the transaction flow.
- Phase Details:** Target Request Flow Started and Request sent to target server (GET http://172.23.131.2/docs).
- Target Endpoint:** Name: default
- Request Headers:**

| Header | Value |
|-------------------|--------------------------------------|
| Accept | */* |
| Accept-Encoding | gzip, deflate |
| Date | Fri, 16 09 2016 00:54:40 GMT |
| Host | test2-prod.apimng.com:10080 |
| User-Agent | NING/1.0 |
| X-Apigee-Trace-Id | 3295b8d0-69da-405f-a7ac-36f2bdbc968f |
| X-Forwarded-For | 172.23.131.2 |
| X-Forwarded-Port | 10080 |
- View Options:** Checkboxes for Transaction Map (Show Disabled Policies, Show Skipped Phases, Show All FlowInfos), Phase Details (Automatically Compare Selected Phase, Show Variables, Show Properties).

Numbered callouts in the image identify the following features:

- ① トレース対象の選択欄
- ② トレースの開始・停止ボタン
- ③ トレース結果のダウンロードボタン
- ④ Node.js のログ表示ボタン
- ⑤ フィルタの設定欄
- ⑥ Transaction の一覧
- ⑦ トレース結果の表示欄
- ⑧ トレース結果の表示オプション

図 3-4 : トレース画面の概要図

表 3-8 : トレース画面の項目一覧

| 項目 | 説明 |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| ①トレース対象の選択欄 | トレースを行う Environment を選択します。 |
| ②トレースの開始・停止ボタン | Start Trace Session ボタンを押下するとトレースを開始します。トレースが開始されると、ボタンは Stop Trace Session に変わり、押下する事でトレースを停止する事ができます。 ※トレース開始後 10 分を経過するとトレースは自動的に停止されます。 |
| ③トレース結果のダウンロードボタン | トレースした結果を xml 形式でダウンロードできます。 |
| ④Node.js のログ表示ボタン | Node.js のログ表示画面に遷移します。 |
| ⑤フィルタの設定欄 | HTTP Header と Query Parameter の変数と値でキャプチャする Traffic をフィルタします。 ※トレース中にフィルタの設定を変更しても反映されません。トレースを開始する前にフィルタを設定する必要があります。 |
| ⑥Transaction の一覧 | トレース中にキャプチャされた Transaction が表示されます。ここで選択中の Transaction をトレース結果として表示します。 |
| ⑦Request の送信フォーム | 指定した URL に GET リクエストの送信を行います。右側の Status には、送信結果のステータスコードが表示されます。 |
| ⑧トレース結果の表示欄 | 表示欄の上部には Transaction Map (Transaction をアイコンで可視化した図) が表示され、図中のアイコンをクリックすると、その時の詳細情報が表示欄下部の Phase Details に表示されます。 |
| ⑨トレース結果の表示オプション | トレース結果の Transaction Map および Phase Details の表示条件を設定できます。 |

表 3-9 : Transaction Map のアイコン一覧

| アイコン | 説明 |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | リクエストを送信するクライアントアプリです。 |
|  | エンドポイントの切り替わりで表示されるアイコンです。表示されるタイミングは下記の通りです。 <ul style="list-style-type: none"> ➤ クライアントからのリクエストを受信したタイミング ➤ バックエンドサービスにリクエストを送信したタイミング ➤ バックエンドサービスからの Response を受信したタイミング ➤ クライアントに Response を送信したタイミング |
|  | 小さいバーと大きいバーの 2 種類があります。小さいバーは Flow の中で Properties に変化があったことを示し、大きいバーは API Proxy の各 Flow で処理が開始されるタイミングを示します。 |
|  | Analytics のアクションが発生したことを示します。 |
|  | Conditional Flow の条件と一致した場合に表示されます。 |
|  | Conditional Flow の条件と一致しなかった場合に表示されます。 |
|  | 各 Flow で設定された Policy のアイコンです。 |
|  | バックエンドサービスが、Node.js の場合に表示されます。 |
|  | リクエストを受信し、レスポンスを返すバックエンドサービスです。 |
|  | Transaction Map でクリックしたアイコンの処理時間（ミリ秒）をピンク色のタイムラインで示します。 |
|  | Policy が無効になっている場合、該当する Policy のアイコン上に表示されます。 |
|  | Policy でエラーが発生した場合もしくは、RaiseFault Policy が実行された場合、該当する Policy のアイコン上に表示されます。 |
|  | Policy が処理条件に一致せずスキップされた場合、該当する Policy のアイコン上に表示されます。 |

表 3-10 : トレース結果の表示オプション一覧

| Option | 説明 |
|--------------------------------------|----------------------------------------------|
| Show Disabled Policies | チェックを入れると無効になっている Policy を表示します。 |
| Show Skipped Phases | チェックを入れると処理がスキップされた Policy を表示します。 |
| Show all FlowInfos | チェックを入れると FlowInfo を表示します。 |
| Automatically Compare Selected Phase | チェックを入れると処理前後の比較を Phase Details に表示します。 |
| Show Variables | チェックを入れると変数を Phase Details に表示します。 |
| Show Properties | チェックを入れると Properties を Phase Details に表示します。 |

第4章

Publish 機能

ここでは、API Management の Publish 機能について説明します。

第1節. Publish 機能について

Publish 機能では、Products (API Proxy のグループ)、Developer (アプリ開発者)、Developer Apps (アプリ) の設定が可能です。設定を行うことで、アプリ開発者がアプリと API Proxy を通して API を利用できるようになります。

アプリ開発者からバックエンドサービスまでの関係を図 4-1 に示します。

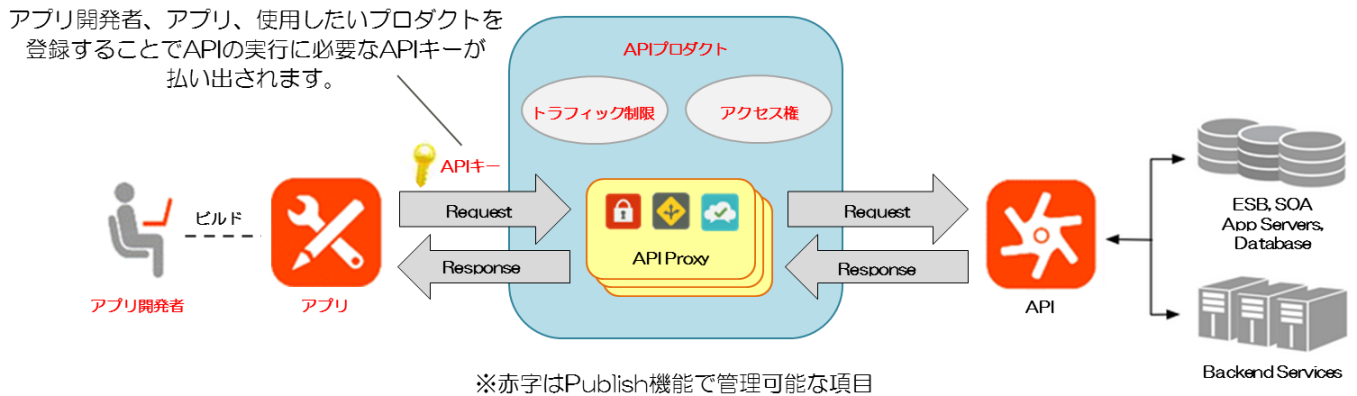


図 4-1 : App developer からバックエンドサービスまでの関係

各コンポーネントの詳細については、次節以降を参照ください。

第2節. Products

Publish 機能の Products では、API Product の作成・修正・削除と Products に関する分析グラフ (Analytics) が確認できます。

API Product とは、複数の API Proxy をグルーピングしたもので、そのグループの情報や分析、操作履歴の確認などを行えます。また、API Product 毎に、トラフィック制限や、アクセス権等の設定が可能です。API Product の概念は図 4-2 の通りです。

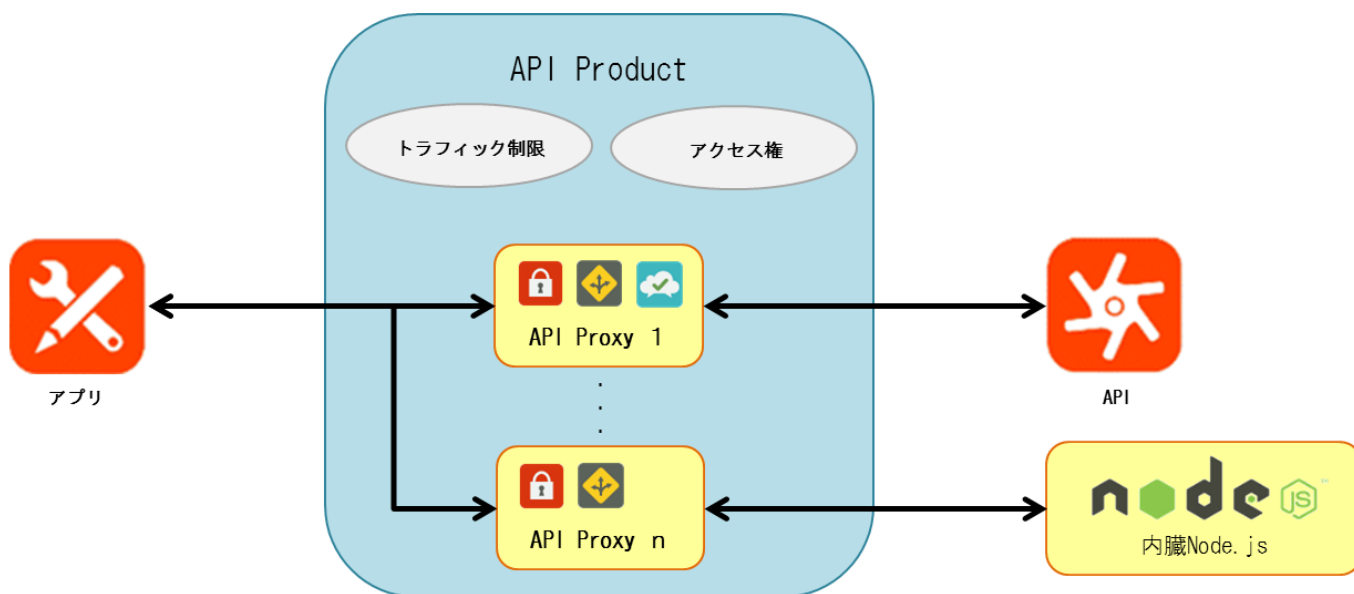


図 4-2 : API Product の概念図

Products の List タブでは、作成した Products の一覧を確認することができます。また、Actions のボタンから Product の History (操作履歴) 表示や Product 削除、Product のロール設定が行えます。

The screenshot shows a web dashboard for 'Products'. At the top right, there are filters for 'Organization' (scenario) and 'Environment' (prod). Below the title 'Products', there are tabs for 'List' and 'Analytics'. A search bar is present with a dropdown menu set to 'All'. On the right, it shows '1-2 of 2' with navigation arrows and a '+ Product' button. The main content is a table with the following columns: Product, Keys, Created, Modified, and Actions.

| Product | Keys | Created | Modified | Actions |
|----------------------------------------|------|--------------------------|--------------------------|--------------------------------------------------------------------------|
| sampleProduct | 1 | Sep 13, 2016 10:40:24 AM | Sep 13, 2016 10:40:24 AM | History Delete Roles (1) |
| web-monitoring Product | 1 | Sep 1, 2016 10:20:03 AM | Sep 1, 2016 10:20:03 AM | History Delete Roles |

図 4-3 : Products の一覧画面

表 4-1 : Products 一覧画面のボタン説明

| ボタン名 | 説明 |
|-----------|-------------------------------------|
| + Product | Product の作成画面を表示します。 |
| History | 同じ行の操作履歴（作成・修正日時、作成・修正者等の情報）を表示します。 |
| × Delete | 同じ行の Product を削除します。 |
| Roles | 同じ行の Product に対する Roles 設定画面を表示します。 |

Products の Analytics タブ(分析グラフ)では、登録されている API Product 毎に Metric (表 4-2 の通り) の合計を示したグラフを表示します。

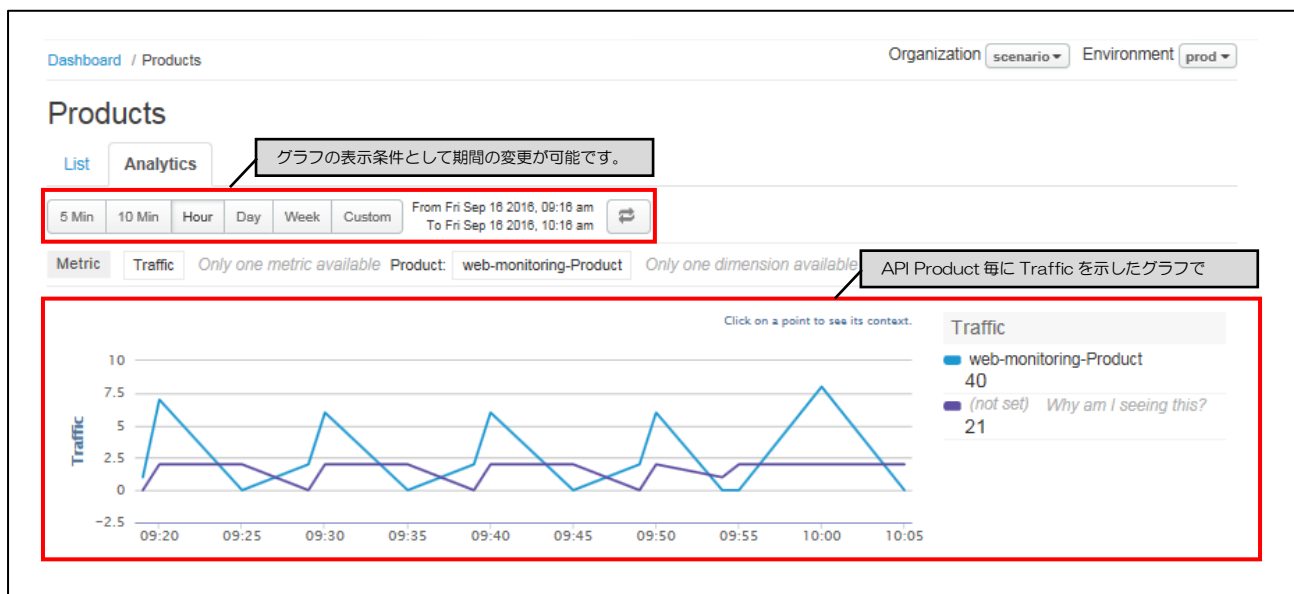


図 4-4 : Products の Analytics 画面

表 4-2 : Products の Metric について

| Metric | 説明 |
|---------|-------------------------------------------------------------------------|
| Traffic | グラフに Traffic (リクエスト数) が表示されます。グラフ右側の欄には Product 名と、Traffic の合計値が表示されます。 |

第3節. Developers

Publish 機能の Developers では、Developer の作成・修正・削除と Developer に関する分析グラフ (Analytics) が確認できます。

Developer を作成し、Developer App に割り当てることで固有の API Key が生成され、この API Key を利用して API Product にアクセスさせることで、Developer の特定とセキュリティの確保を行います。Developer の概念は図 4-5 の通りです。

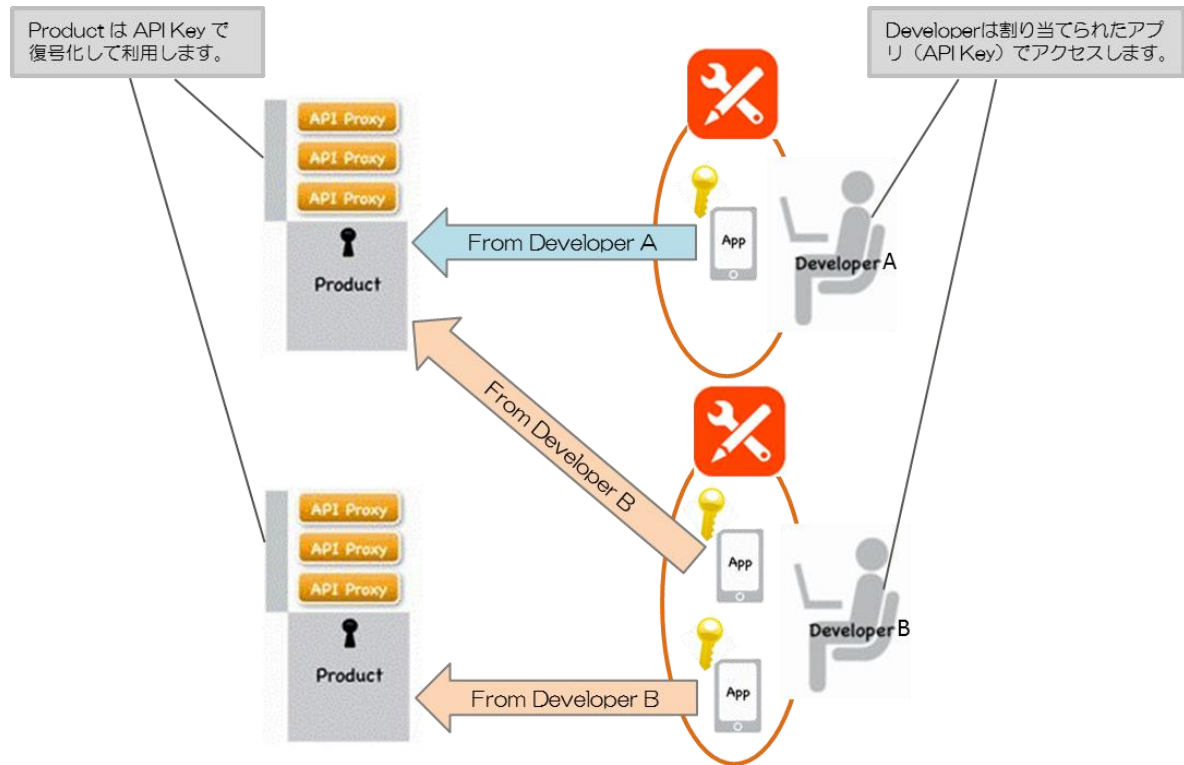


図 4-5 : Developer の概念図

Developers の List タブでは、作成した Developer の一覧を確認することができます。一覧画面の各種ボタンについては、表 4-3 を参照ください。

| Developer | Email | Username | Apps | Keys | Actions |
|---------------|--------------------------|--------------|------|------|----------|
| Fujitsu Taro | tesla2@weathersample.com | Fujitsu Taro | 0 | 0 | × Delete |
| Nikolai Tesla | tesla@weathersample.com | Nikolai | 2 | 2 | × Delete |

図 4-6 : Developers の一覧画面

表 4-3 : Developers 一覧画面のボタン説明

| ボタン名 | 説明 |
|-------------|-----------------------------------------------------------|
| Export | Developers 単体または、Developers、Apps、Products を CSV 形式で出力します。 |
| + Developer | Developer の作成画面を表示します。 |
| × Delete | 同じ行の Developer を削除します。 |

Developers の Analytics タブ (分析グラフ) では、登録されている Developer 毎に選択中の Metric (表 4-4 の通り) の合計値を示したグラフを表示します。

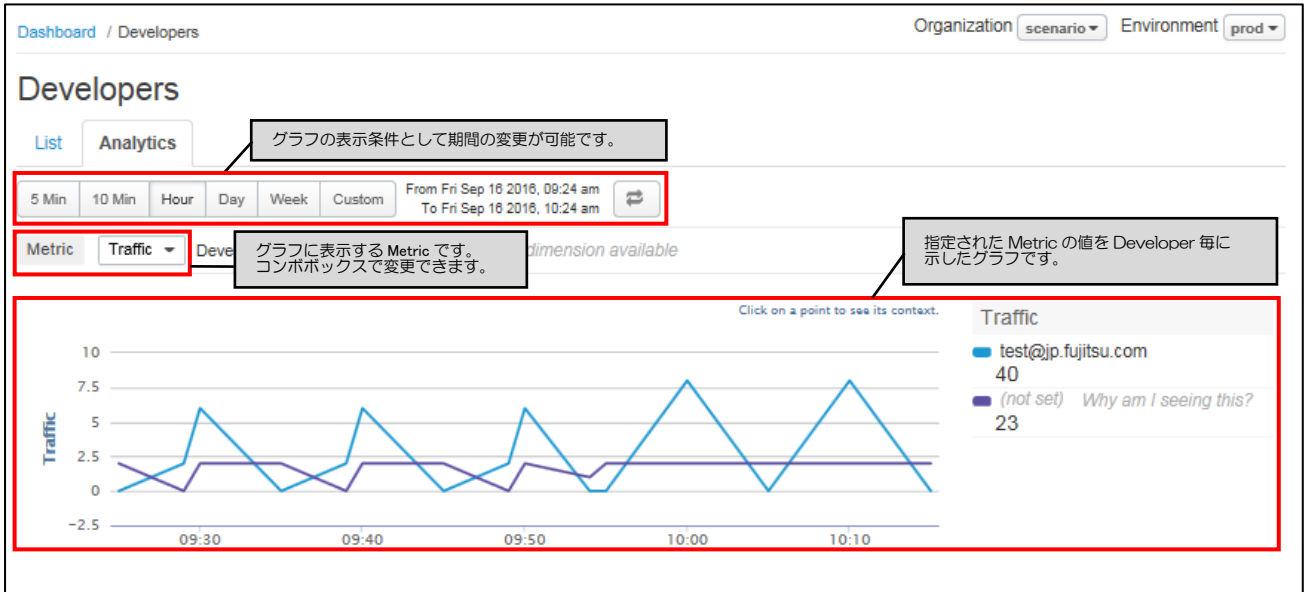


図 4-7 : Developers の Analytics 画面

表 4-4 : Developers の Metric について

| Metric | 説明 |
|---------|----------------------------------------------------------------------------------|
| Traffic | グラフに Traffic (リクエスト数) が表示されます。グラフ右側の欄には Developer のメールアドレスと、Traffic の合計値が表示されます。 |
| Errors | グラフに Errors (エラー数) が表示されます。グラフの右側の欄には Developer のメールアドレスと、Error の合計値が表示されます。 |

第4節. Developer Apps

Publish 機能の Developer Apps では、アプリの作成・修正・削除とアプリに関する分析グラフ (Analytics) が確認できます。

Developer App を作成では、1つのアプリに対して1人の Developer と1つ以上の Product を割り当てる必要があります。アプリの作成が完了した時点で API Key が生成されます。この時、割り当てた Product の Key Approval Type (鍵認証方式) によって、API Key のステータスが異なります。アプリと API Key の概念および API Key のステータス詳細については下記を参照ください。

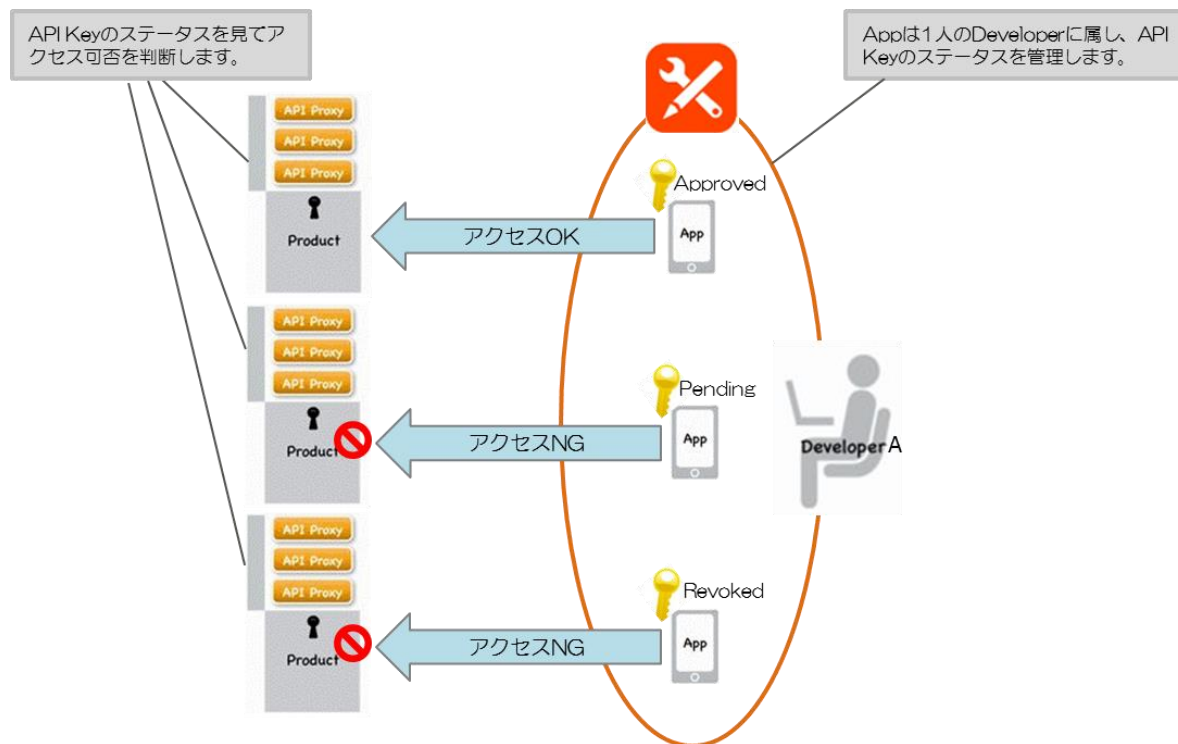


図 4-8 : アプリと API Key の概念図

表 4-5 : API Key のステータス一覧

| API Key のステータス | 説明 |
|----------------|--------------------------------------------------------------------------------------------------------------------------------|
| Approved | API Key が承認された状態 (利用可能) です。 Product の Key Approval Type が Automatic の場合、Developer App を作成すると API Key は Approved の状態で生成されます。 |
| Pending | API Key が未承認の状態 (利用不可) です。 Product の Key Approval Type が Manual の場合、Developer App を作成すると API Key は Pending の状態で生成されます。 |
| Revoked | API Key が無効な状態 (利用不可) です。 Approved 状態の API Key を Revoke (無効) に設定すると状態が Revoked になります。 |

Developer Apps の List タブでは、作成した Developer App の一覧を確認することができます。



図 4-9 : Developer Apps の一覧画面

表 4-6 : Developer Apps 一覧画面のボタン説明

| ボタン名 | 説明 |
|-----------------|-----------------|
| + Developer App | アプリの作成画面を表示します。 |
| × Delete | 同じ行のアプリを削除します。 |

Developer Apps の Analytics タブ（分析グラフ）では、登録されているアプリ毎に Metric（図 4-10 の通り）の合計値を示したグラフを表示します。

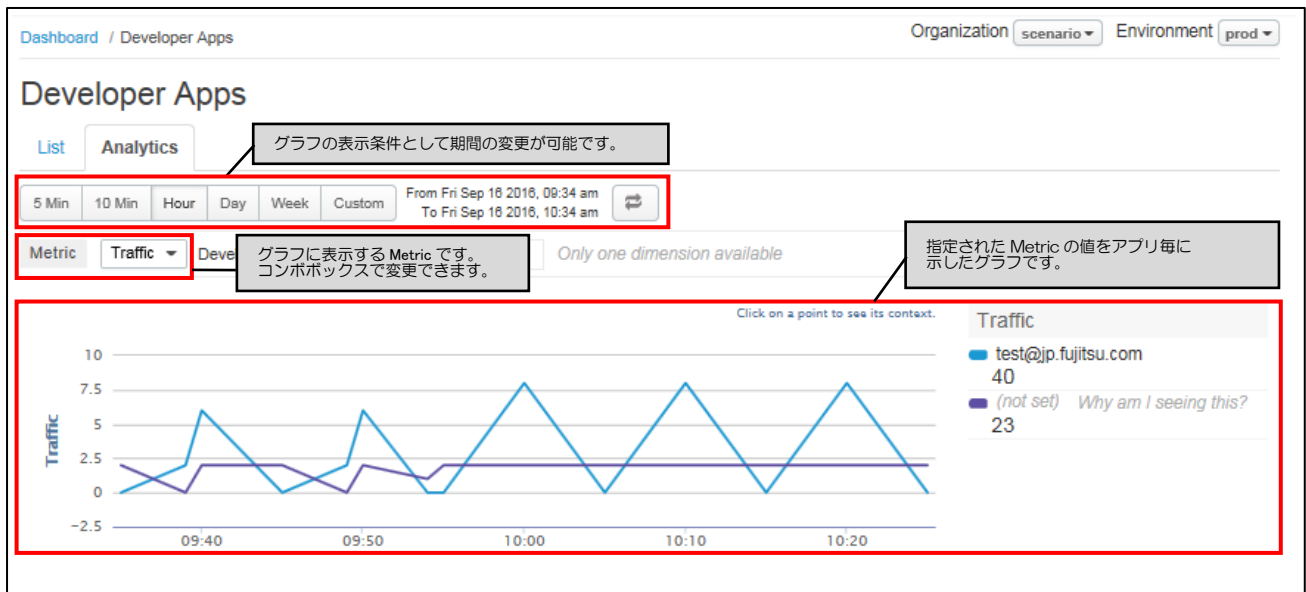


図 4-10 : Developer Apps の Analytics 画面

表 4-7 : Developer Apps の Metric について

| Metric | 説明 |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Traffic | グラフに Traffic (リクエスト数) を表示されます。グラフ右側の欄にはアプリ名と、Traffic の合計値が表示されます。 |
| Errors | グラフに Errors (エラー数) が表示されます。グラフの右側の欄にはアプリ名と、Error の合計値が表示されます。 |
| Average Target Response Time | グラフに Average Target Response Time (バックエンドサービスからのレスポンスを受け取ってからクライアントに返すまでの平均時間) を示したグラフが表示されます。グラフ右側の欄にはアプリ名と Average Target Response Time 全体の平均時間が表示されます。 |
| Average Response Time | グラフに Average Response Time (クライアントからリクエストを受け取ってからレスポンスを返すまでの平均時間) を示したグラフが表示されます。グラフ右側の欄にはアプリ名と Average Response Time 全体の平均時間が表示されます。 |
| Maximum Response Time | グラフに Maximum Response Time (クライアントからリクエストを受け取ってからレスポンスを返すまでの最大時間) を示したグラフが表示されます。グラフの右側の欄にはアプリ名と Maximum Response Time グラフで表示している期間内の最大値) が表示されます。 |

第5章

Analytics 機能

ここでは、API Management の Analytics 機能について説明します。

第1節. Proxy Performance

Analytics 機能の Proxy Performance では、API Proxy の Traffic (リクエスト数) と Average Response Time (平均応答時間) を全体と個別それぞれに集計したグラフを表示します。

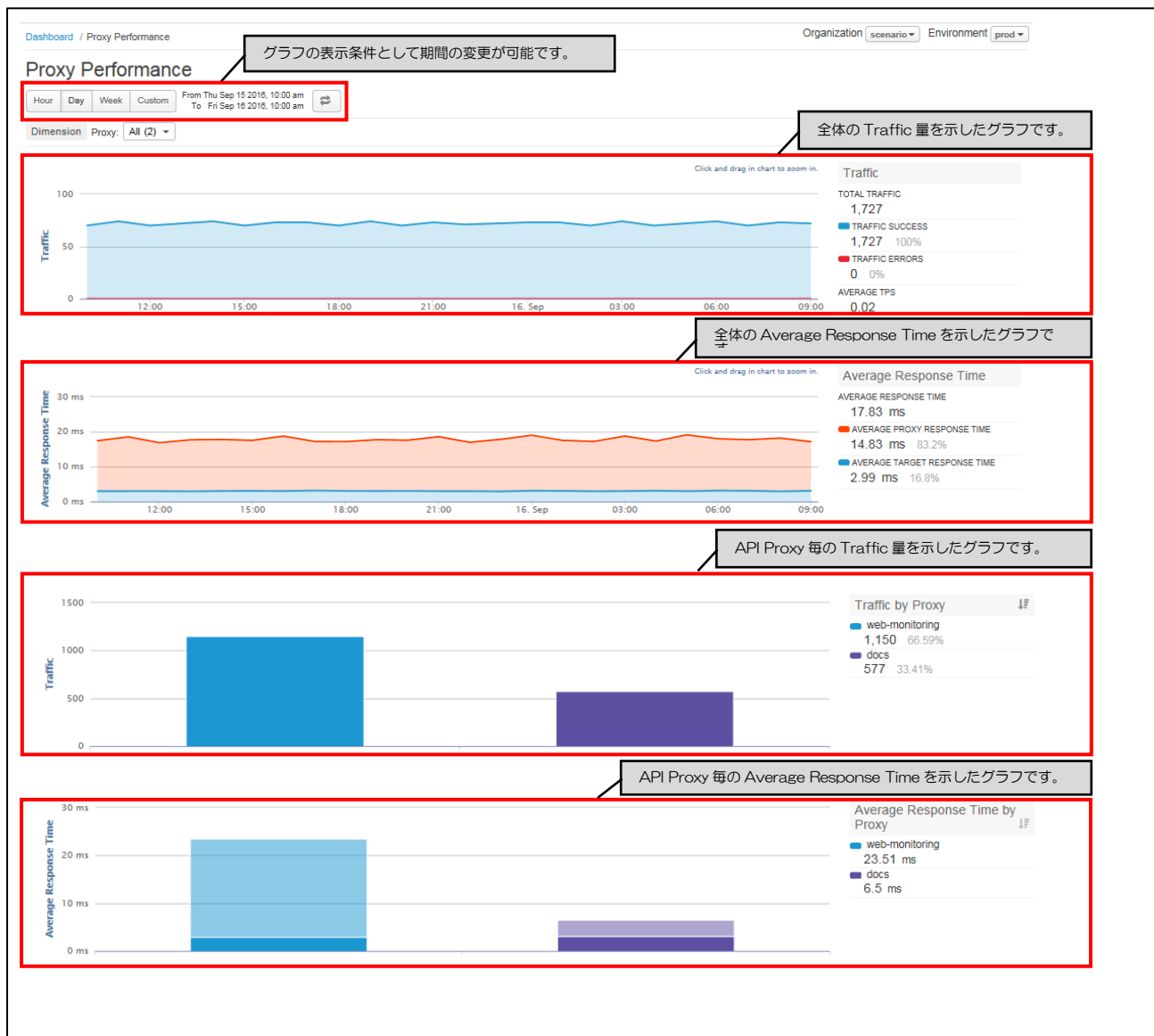


図 5-1 : Proxy Performance の画面

表 5-1 : Proxy Performance の Traffic について

| 項目名 | 説明 |
|-----------------|----------------------------------------------|
| TOTAL TRAFFIC | API Proxy のリクエスト数 (全体の合計) が表示されます。 |
| TRAFFIC SUCCESS | API Proxy で成功したリクエスト数 (全体の合計) が表示されます。 |
| TRAFFIC ERRORS | API Proxy で発生したリクエストのエラー数 (全体の合計) が表示されます。 |
| AVERAGE TPS | API Proxy のリクエスト・レスポンス数の平均値 (全体の平均) が表示されます。 |

表 5-2 : Proxy Performance の Average Response Time について

| 項目名 | 説明 |
|------------------------------|---------------------------------------------------------------------------|
| AVERAGE RESPONSE TIME | API Proxy がクライアントからリクエストを受けとり、クライアントにレスポンスを返すまでの平均時間（全体の平均）が表示されます。 |
| AVERAGE PROXY RESPONCE TIME | API Proxy がクライアントからリクエストを受けとり、バックエンドサービスにリクエストを送信するまでの平均時間（全体の平均）が表示されます。 |
| AVERAGE TARGET RESPONCE TIME | API Proxy がバックエンドサービスからレスポンスを受けとり、クライアントにレスポンスを返すまでの平均時間（全体の平均）が表示されます。 |

表 5-3 : Proxy Performance の Traffic by Proxy について

| 項目名 | 説明 |
|----------------|------------------------------------------------|
| < Proxy name > | API Proxy 毎に名前とその API Proxy に対するリクエスト数が表示されます。 |

表 5-4 : Proxy Performance の Average Response Time by Proxy について

| 項目名 | 説明 |
|----------------|------------------------------------------------------------------------|
| < Proxy name > | API Proxy 毎に名前とその API Proxy がリクエストを受けとり、クライアントにレスポンスを返すまでの平均時間が表示されます。 |

第2節. Target Performance

Analytics 機能の Target Performance では、バックエンドサービスへの Traffic 量とリクエストの成功・失敗件数、応答時間、レスポンスの成功・失敗件数、Payload Size をグラフに表示します。



図 5-2 : Target Performance の画面

表 5-5 : Target Performance の Traffic by Proxy について

| 項目名 | 説明 |
|------------|-----------------------------------------------------|
| < Target > | バックエンドサービス毎に IP アドレスとそのバックエンドサービスに対するリクエスト数が表示されます。 |

表 5-6 : Target Performance の Traffic について

| 項目名 | 説明 |
|---------------|------------------------------|
| TOTAL TRAFFIC | バックエンドサービスに対するリクエスト数が表示されます。 |
| ERRORS | リクエストの失敗件数が表示されます。 |
| SUCCESS | リクエストの成功件数が表示されます。 |

表 5-7 : Target Performance の Response Time について

| 項目名 | 説明 |
|---------------------|----------------------------------------------------------------|
| AVERAGE TIME | API Proxy がクライアントからリクエストを受けとり、クライアントにレスポンスを返すまでの平均時間が表示されます。 |
| AVERAGE TARGET TIME | API Proxy がバックエンドサービスへのリクエスト送信完了後からレスポンスを受け取るまでの平均時間が表示されます。 |
| AVERAGE PROXY TIME | API Proxy がクライアントからリクエストを受信してから、バックエンドサービスへ送信するまでの平均時間が表示されます。 |

表 5-8 : Target Performance の Target Errors について

| 項目名 | 説明 |
|--------------|---------------------------------|
| TOTAL ERRORS | レスポンスの失敗件数が表示されます。 |
| 3XX ERRORS | ステータスコードが 300 系のレスポンス件数が表示されます。 |
| 4XX ERRORS | ステータスコードが 400 系のレスポンス件数が表示されます。 |
| 5XX ERRORS | ステータスコードが 500 系のレスポンス件数が表示されます。 |

表 5-9 : Target Performance の Payload Size について

| 項目名 | 説明 |
|-----------------------|--------------------------------------|
| TOTAL PAYLOAD SIZE | リクエストとレスポンスの Payload Size 合計が表示されます。 |
| REQUEST PAYLOAD SIZE | リクエストの Payload Size 合計が表示されます。 |
| RESPONSE PAYLOAD SIZE | レスポンスの Payload Size 合計が表示されます。 |

第3節. Cache Performance

Pro

フルアナリティクス

Analytics 機能の Cache Performance では、Response Cache Policy のキャッシュヒット率や件数、応答時間をグラフに表示します。

※データ保持期間は Pro とフルアナリティクスオプションが 3 か月、Standard は保持されません。

Cache Performance のグラフを表示させるには、Response Cache Policy を Request の ProxyEndpoint PreFlow と Response の TargetEndpoint PostFlow の両方に設定し、通信をキャッシュさせる必要があります。



図 5-3 : Cache Performance の画面

表 5-10 : Cache Performance の Cache Hit Rate について

| 項目名 | 説明 |
|------------------------|----------------------|
| AVERAGE CACHE HIT RATE | 全体のキャッシュヒット率が表示されます。 |

表 5-11 : Cache Performance の Cache Hits について

| 項目名 | 説明 |
|----------------|-----------------------------------|
| ALL CACHE HITS | 全体のキャッシュヒット件数が表示されます。 |
| < Proxy name > | API Proxy 毎に名前とキャッシュヒット件数が表示されます。 |

表 5-12 : Cache Performance の Response Time について

| 項目名 | 説明 |
|----------------------------|-----------------------------|
| AVERAGE TIME WITH CACHE | キャッシュがあった場合の平均応答時間が表示されます。 |
| AVERAGE TIME WITHOUT CACHE | キャッシュをなかった場合の平均応答時間が表示されます。 |
| CACHE IMPROVEMENT | キャッシュの有り無しでの速度の違いを倍率で表示します。 |

表 5-13 : Cache Performance の Cache Hits by App について

| 項目名 | 説明 |
|----------------|---------------------------|
| ALL CACHE HITS | 全体のキャッシュヒット件数が表示されます。 |
| < App name > | アプリの名前とキャッシュヒット件数が表示されます。 |

第4節. Latency Analysis

Analytics 機能の Latency Analysis では、API Proxy の処理時間や、バックエンドサービスの応答時間をグラフに表示します。



図 5-4 : Latency Analysis の画面

表 5-14 : Latency Analysis の各グラフ項目について

| 項目名 | 説明 |
|-----------------|------------------------------------------------------------------|
| Median | API Proxy がクライアントからリクエストを受け、レスポンスを返すまでの時間に対する中央値が表示されます。 |
| 95th Percentile | API Proxy がクライアントからリクエストを受け、レスポンスを返すまでの時間に対する95パーセンタイルの値が表示されます。 |
| 99th Percentile | API Proxy がクライアントからリクエストを受け、レスポンスを返すまでの時間に対する99パーセンタイルの値が表示されます。 |

第5節. Error Analysis

Analytics 機能の Error Analysis では、API Proxy が処理するリクエストおよび Response の Transaction 中に発生したエラーの情報（件数やステータスコード等）をグラフ表示します。

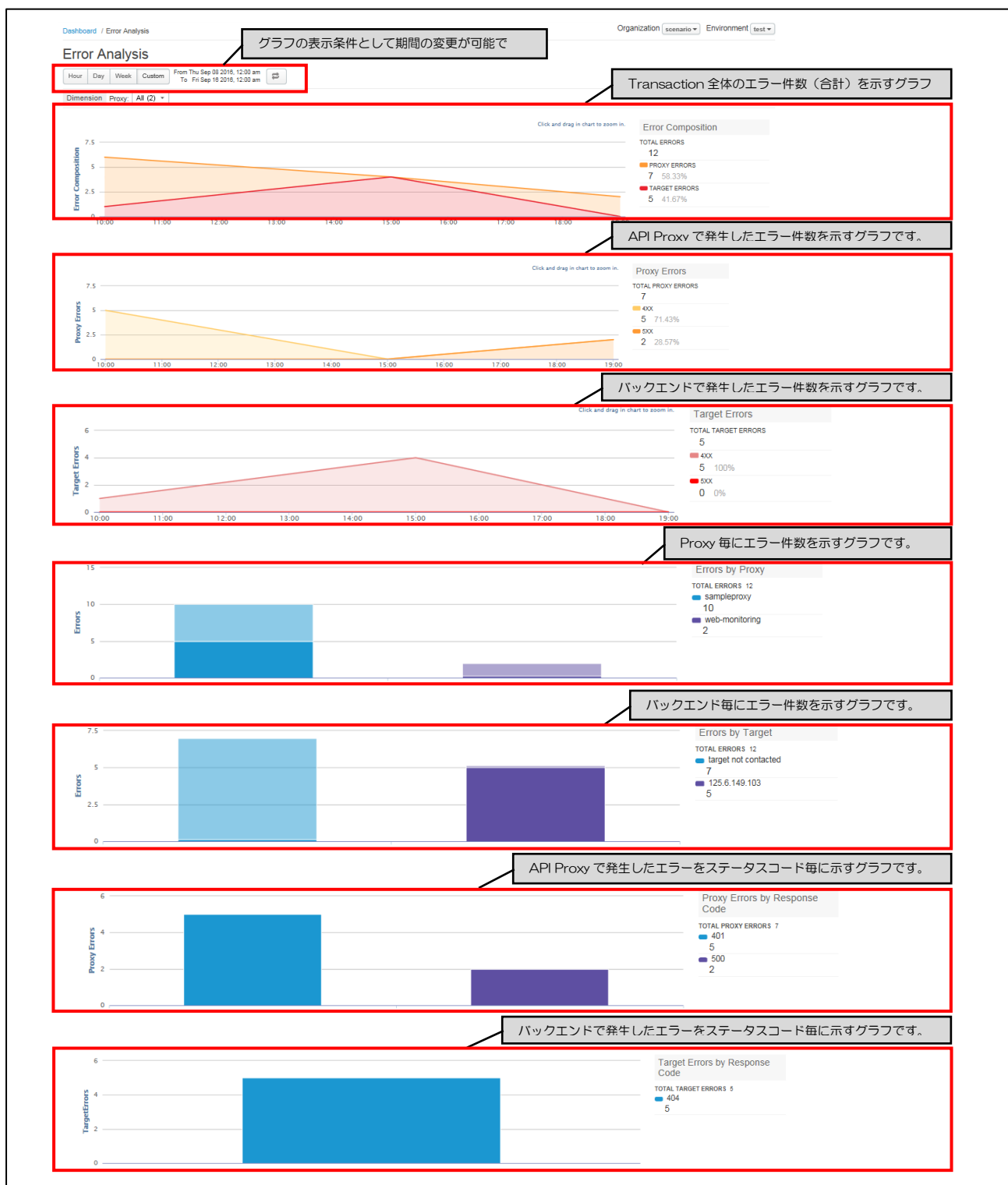


図 5-5 : Error Analysis の画面

表 5-15 : Error Analysis の Error Composition について

| 項目名 | 説明 |
|---------------|---------------------------------|
| Total Errors | 全体のエラー発生件数が表示されます。 |
| Proxy Errors | API Proxy で発生したエラーの合計件数が表示されます。 |
| Target Errors | バックエンドサービスで発生したエラーの合計件数が表示されます。 |

表 5-16 : Error Analysis の Proxy Errors について

| 項目名 | 説明 |
|--------------------|-------------------------------------------------------|
| TOTAL PROXY ERRORS | API Proxy で発生したエラーの合計件数が表示されます。 |
| 4XX | API Proxy で HTTP ステータスコードが 4XX 台となっているエラーの発生件数を表示します。 |
| 5XX | API Proxy で HTTP ステータスコードが 5XX 台となっているエラーの発生件数を表示します。 |

表 5-17 : Error Analysis の Target Errors について

| 項目名 | 説明 |
|---------------------|-------------------------------------------------------|
| TOTAL TARGET ERRORS | バックエンドサービスで発生したエラーの合計件数が表示されます。 |
| 4XX | バックエンドサービスで HTTP ステータスコードが 4XX 台となっているエラーの発生件数を表示します。 |
| 5XX | バックエンドサービスで HTTP ステータスコードが 5XX 台となっているエラーの発生件数を表示します。 |

表 5-18 : Error Analysis の Errors by Proxy について

| 項目名 | 説明 |
|--------------|------------------------------|
| TOTAL ERRORS | 全体のエラー発生件数が表示されます。 |
| <Proxy name> | Proxy 毎に名前と発生したエラー件数が表示されます。 |

表 5-19 : Error Analysis の Errors by Target について

| 項目名 | 説明 |
|---------------|----------------------------------------|
| TOTAL ERRORS | 全体のエラー発生件数が表示されます。 |
| <Target name> | バックエンドサービス毎に IP アドレスと発生したエラー件数が表示されます。 |

表 5-20 : Error Analysis の Proxy Errors by Response Code について

| 項目名 | 説明 |
|--------------------|-------------------------------------------|
| TOTAL PROXY ERRORS | API Proxy で発生したエラーの合計件数が表示されます。 |
| <Target name> | API Proxy で発生したエラーのステータスコードとエラー件数が表示されます。 |

表 5-21 : Error Analysis の Target Errors by Response Code について

| 項目名 | 説明 |
|--------------------|----------------------------------------|
| TOTAL PROXY ERRORS | エンドポイントで発生したエラーの合計件数が表示されます。 |
| <Target name> | エンドポイントで発生したエラーのステータスコードとエラー件数が表示されます。 |

第6節. Developer Engagement

Pro

フルアナリティクス

Analytics 機能の Developer Engagement では、Developer が生成した Traffic 情報を収集することで、Developer の人数やアクセス状況を示す図（Engagement）と、Traffic 量、エラー率を集計した表（Active Developers）を生成します。

生成された図と表は下記の通りです。

※データ保持期間は Pro とフルアナリティクスオプションが 3 か月、Standard は保持されません。

Developer の詳細については、第4章 Publish 機能を参照ください。

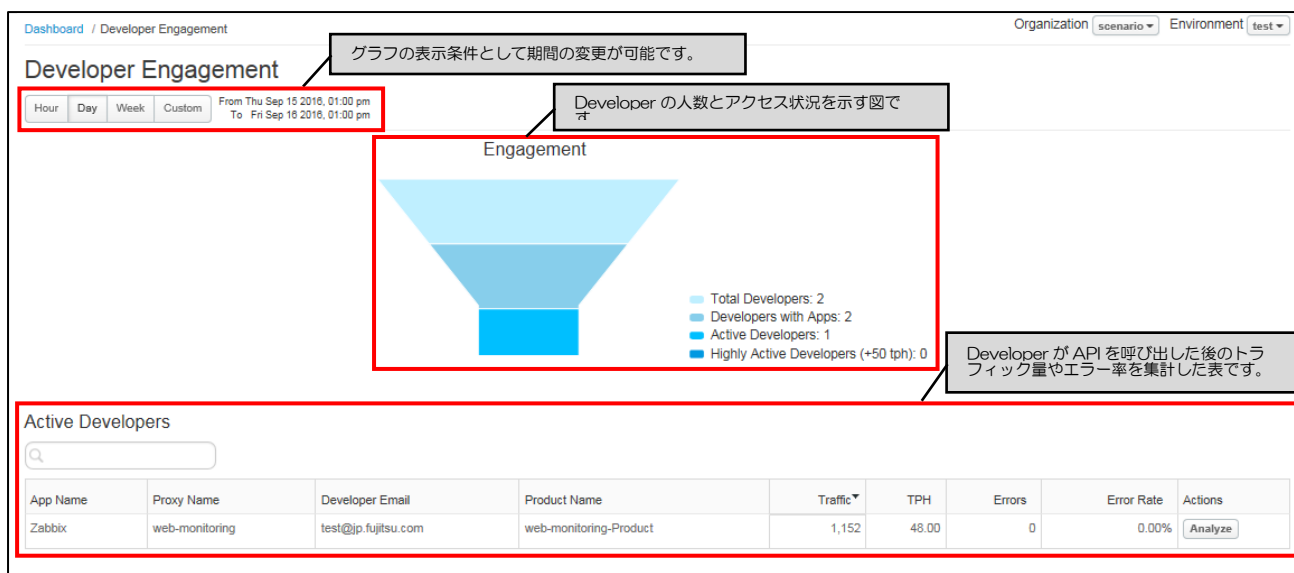


図 5-6 : Developer Engagement の画面

表 5-22 : Developer Engagement の Engagement について

| 項目名 | 説明 |
|------------------------------------|---------------------------------------------------|
| Total Developers | 組織内の API Proxy に関連する Developer の合計人数を示します。 |
| Developers with Apps | 組織内のアプリに関連する Developer の合計人数を示します。 |
| Active Developers | API Proxy 経由でリクエストを送信した Developer の人数を示します。 |
| Highly Active Developers (+50 tph) | 1 時間あたりの Transaction 数が 50 以上の Developer 人数を示します。 |

表 5-23 : Developer Engagement の Active Developers について

| 項目名 | 説明 |
|-----------------|------------------------------------|
| App Name | アプリの名前を表示します。 |
| Proxy Name | アプリに関連する Proxy の名前を表示します。 |
| Developer Email | アプリを登録した Developer のメールアドレスを表示します。 |
| Product Name | アプリに関連する Product の名前を表示します。 |
| Traffic | アプリが生成した Traffic 量を表示します。 |
| TPH | 1 時間あたりの Transaction 生成数を表示します。 |
| Errors | アプリで発生したエラー数を表示します。 |
| Error Rate | アプリで発生したエラーの割合（エラー数/リクエスト数）を表示します。 |
| Actions | Analyze ボタンを押下するとアプリの詳細画面が表示されます。 |

Active Developers の Analyze ボタンを押下すると、アプリの詳細画面が表示されます。アプリの詳細画面では、指定した Metric（表 5-24 の通り）のアプリに対する値をグラフ表示します。

表 5-24 : Developer Engagement の Metric について

| 項目名 | 説明 |
|---------------|--------------------------|
| Traffic | Traffic 量（リクエスト数）を表示します。 |
| Errors | エラー数を表示します。 |
| Response Time | 応答時間を表示します。 |



図 5-7 : Developer Engagement のアプリ詳細を表示する画面

表 5-25 : アプリ詳細画面の情報について

| 項目名 | 説明 |
|-------------------------|-------------------------------------------------|
| App | 現在表示中のアプリ名を表示します。 |
| Created | アプリが作成された日付を表示します。 |
| Developer Name | アプリを作成した Developer の名前を表示します。 |
| Developer Email | アプリを作成した Developer のメールアドレスを表示します。 |
| Proxies Used by App | Proxies Used by App でグラフに表示している Proxy の数です。 |
| Apps by Developer | Apps by Developer でグラフに表示しているアプリの数です。 |
| Apps Using Same Proxies | Apps Using the Same Proxies でグラフに表示しているアプリの数です。 |

表 5-26 : アプリ詳細画面の Traffic Level of Other Apps Using Same Proxies について

| 項目名 | 説明 |
|--------|----------------------------------------------------|
| High | 1 時間あたりの Transaction 数が 500 以上のアプリ数を表示します。 |
| Medium | 1 時間あたりの Transaction 数が 50 以上かつ 500 未満のアプリ数を表示します。 |
| Low | 1 時間あたりの Transaction 数が 50 未満のアプリ数を表示します。 |

表 5-27 : アプリ詳細画面の Proxies used by App について

| 項目名 | 説明 |
|---------------------|-------------------------------------|
| ALL PROXIES TRAFFIC | API Proxy 全体の Metric に対する合計値を表示します。 |
| <Proxy name> | API Proxy の名前と Metric に対する値を表示します。 |

表 5-28 : アプリ詳細画面の Other Apps by Developer について

| 項目名 | 説明 |
|------------------|-----------------------------------------------------------|
| ALL APPS TRAFFIC | 表示中のアプリを作成した Developer が作成する全てのアプリの Metric に対する合計値を表示します。 |
| <App name> | アプリの名前と Metric に対する値を表示します。 |

表 5-29 : アプリ詳細画面の Other Apps using the same Proxies について

| 項目名 | 説明 |
|------------------|------------------------------------------------------------|
| ALL APPS TRAFFIC | 表示中のアプリが利用している API Proxy に対する Metric の合計値（他のアプリを含む）を表示します。 |
| <App name> | アプリの名前と Metric に対する値を表示します。 |

第7節. Traffic Composition

Analytics 機能の Traffic Composition では、API Proxy、App、Product、Developer それぞれの Traffic 量 Top10 を示すグラフが図 5-8 の通り表示されます。



図 5-8 : Traffic Composition の Overview 画面

表 5-30 : Traffic Composition の Top 10 Proxies Traffic について

| 項目名 | 説明 |
|--------------|---------------------------------|
| <Proxy name> | API Proxy の名前と Traffic 量を表示します。 |

表 5-31 : Traffic Composition の Top 10 Apps Traffic について

| 項目名 | 説明 |
|------------|--------------------------|
| <App name> | アプリの名前と Traffic 量を表示します。 |

表 5-32 : Traffic Composition の Top 10 Products Traffic について

| 項目名 | 説明 |
|----------------|-------------------------------|
| <Product name> | Product の名前と Traffic 量を表示します。 |

表 5-33 : Traffic Composition の Top 10 Developers Traffic について

| 項目名 | 説明 |
|----------------|---------------------------------|
| <Product name> | Developer の名前と Traffic 量を表示します。 |

各グラフには Dimension でドリルダウン可能な画面が用意されており、Traffic Composition 画面左上の By Dimension ボタンを押下することで遷移できます。ドリルダウンする際に指定可能な Dimension は下記の通りです。

表 5-34 : Traffic Composition の Dimension 一覧

| Metric | 内容 |
|----------------|-----------------------------------------------|
| Proxies | APIs 機能で作成した API Proxy 毎に Traffic 量を表示します。 |
| Developer Apps | Publish 機能で作成したアプリ毎に Traffic 量を表示します。 |
| Developers | Publish 機能で作成した Developer 毎に Traffic 量を表示します。 |
| Products | Publish 機能で作成した Product 毎に Traffic 量を表示します。 |

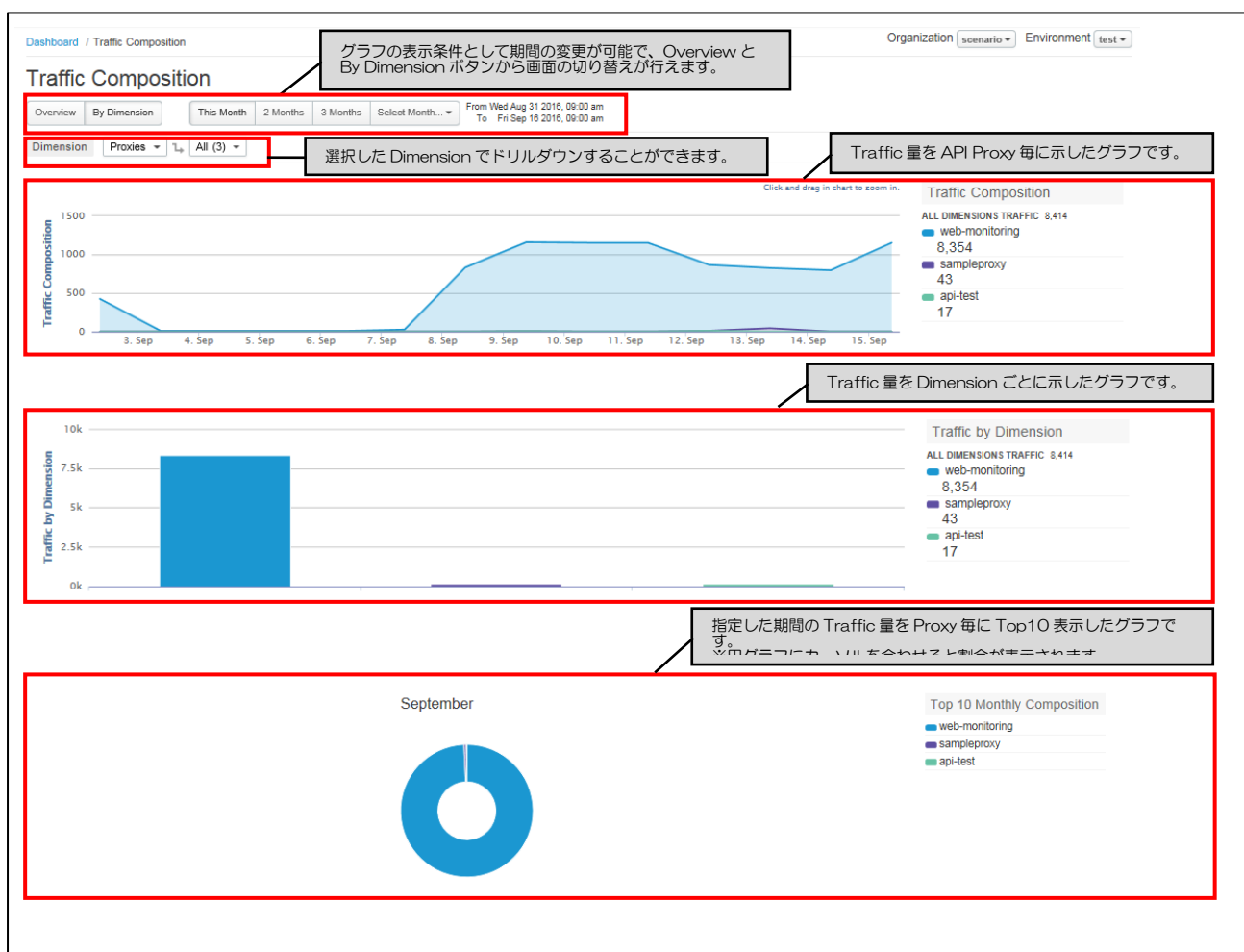


図 5-9 : Traffic Composition の By Dimension 画面

表 5-35 : Traffic Composition について

| 項目名 | 説明 |
|------------------------|---------------------------------|
| ALL DIMENSIONS TRAFFIC | Traffic 量の合計を表示します。 |
| <Proxy name> | API Proxy の名前と Traffic 量を表示します。 |

表 5-36 : Traffic Composition の Traffic by Dimension について

| 項目名 | 説明 |
|------------------------|---------------------------------|
| ALL DIMENSIONS TRAFFIC | Traffic 量の合計を表示します。 |
| <Proxy name> | API Proxy の名前と Traffic 量を表示します。 |

表 5-37 : Traffic Composition の Top 10 Monthly Composition について

| 項目名 | 説明 |
|--------------|-----------------------|
| <Proxy name> | API Proxy の名前が表示されます。 |

第8節. Business Transactions

本機能は現バージョンではご利用できません。

Business Transactions 画面では、指定したパターンと URI が合致するリクエストの統計を確認することができます。また、複数の API Proxy 及び複数パターン（API Proxy 毎に複数指定可能）の統計が一つのグラフ上に表示されるため、パターン毎の比較を容易に行えます。
※データ保持期間は Pro とフルアナリティクスオプションが 3 か月、Standard は保持されません。

● URI パターンの指定

URI パターンは、リクエスト URI のパス部分を半角スラッシュ「/」から始まる文字列で指定します。また、文字列中にアスタリスク「*」ワイルドカードを含めることができます。

例) 下記リクエストの統計を確認したい場合には、パターンを「/abc/*/dec」と指定します。

➢ https://example.com/v1/inventory/abc/123/dec

➢ https://example.com/v1/inventory/abc/456/dec

※パターンの指定は APIs > API Proxies > 設定対象の API Proxy > PERFORMANCE > Configure ボタンで表示される画面（下図の通り）で設定できます。

| Pattern | Verbs | Environments |
|--------------|-------|--------------|
| /* | GET | test |
| /abc/123/dec | GET | prod |
| /abc/456/dec | GET | prod |

図 5-10 : URI パターンの指定

● Metrics

➢ グラフ（時系列）

Business Transactions 画面では、下記 Metrics を時系列で確認することができます。

表 5-38 : Business Transactions の Metrics（グラフ）について

| Metric | 内容 |
|-----------------------|------------------------------------------------------------------------------------------------------|
| Traffic | Traffic 量（リクエスト数）を表示します。 |
| Average Response Time | 単位時間あたりの平均レスポンス時間を表示します。 ※レスポンス時間：API Management がリクエストを受け取り終わってから、クライアントにレスポンスを返却し始めるまでの時間を示します。 |
| Error Rate | エラーになったリクエスト率を表示します。 |
| Average Data Exchange | 単位時間あたりの通信量の平均を表示します。 ※通信量とは Traffic リクエストデータサイズとレスポンスデータサイズの合計です。 |

➤ 統計値

表 5-39 : Business Transactions の Metrics (統計値) について

| Metric | 内容 |
|--------------------|-------------------------------------------------|
| MOST TRAFFIC | 単位時間あたりに最も Traffic が多かったパターンの Traffic 数を表示します。 |
| LEAST TRAFFIC | 単位時間あたりに最も Traffic が少なかったパターンの Traffic 数を表示します。 |
| HIGHEST ERROR RATE | エラー率が最も多かったパターンのエラー率を表示します。 |
| LOWEST ERROR RATE | エラー率が最も少なかったパターンのエラー率を表示します。 |

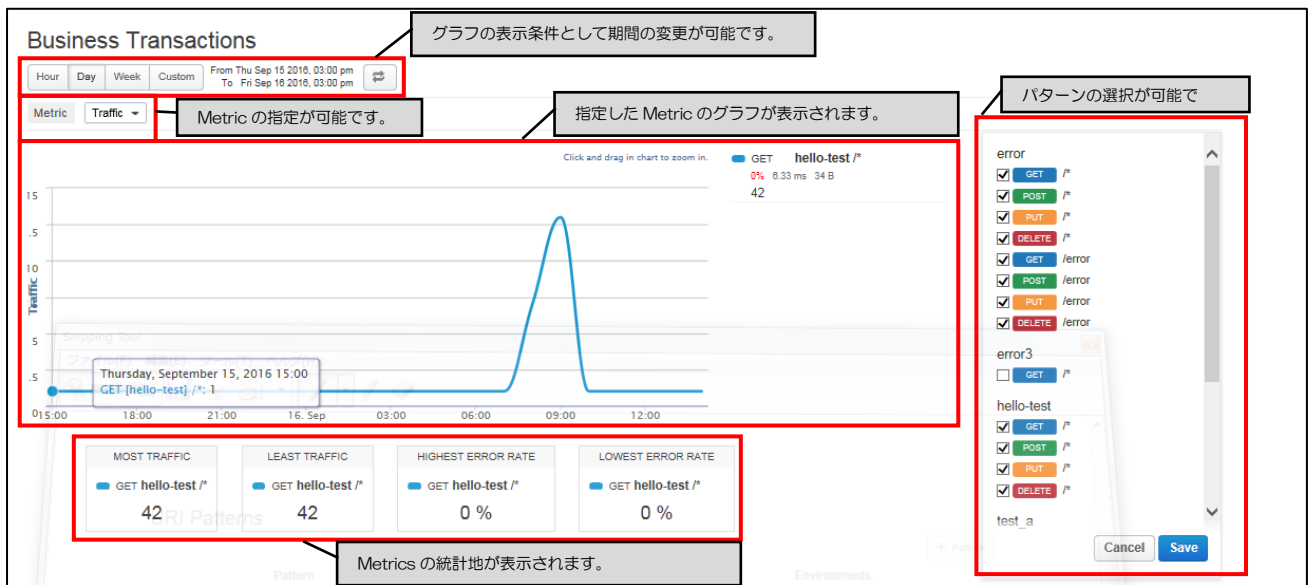


図 5-11 : Business Transactions 画面

第9節. Devices

Analytics 機能の Devices では、API に対するアクセス元のデバイス情報 (Platform、Agent、Device Type、OS Families) をグラフ化して表示します。また、表示しているグラフは API Proxy でドリルダウンすることができます。



図 5-12 : Devices の画面

表 5-40 : Devices の Traffic by Platform について

| 項目名 | 説明 |
|-----------------------|----------------------------------------------------------------------------------------------------------|
| ALL PLATFORMS TRAFFIC | Platform 全体の Traffic 量を表示します。 |
| <Platform name> | アクセス元が利用している Platform の名称 (Google Chrome、Safari、FireFox、cURL、IE、その他 Platform、not set) と Traffic 量を表示します。 |

表 5-41 : Devices の Traffic by Agent について

| 項目名 | 説明 |
|--------------------|-------------------------------------------------------------------------------------|
| ALL AGENTS TRAFFIC | Agent 全体の Traffic 量を表示します。 |
| <Agent name> | アクセス元が利用している Agent の名称 (browser、robot、library、その他 Agent、not set) と Traffic 量を表示します。 |

表 5-42 : Devices の Traffic by Device Type について

| 項目名 | 説明 |
|-------------------|-----------------------------------------------------------------------------------------------------|
| ALL TYPES TRAFFIC | Device Type 全体の Traffic 量を表示します。 |
| <Type name> | アクセス元が利用している Device Type の名称 (Personal computer、mobile device、その他の Type、not set) と Traffic 量を表示します。 |

第10節. Custom Report

Pro

フルアナリティクス

Custom Report では、Y 軸となる Metrics と X 軸となる Dimensions または時間を自由に選択してグラフを作成することができます。また、Dimensions は複数指定することでドリルダウンが可能となっており、Filter 機能でデータを絞り込むことが可能です。

Custom Report の作成画面と概要は下記を参照ください。

※データ保持期間は Pro とフルアナリティクスオプションが3か月、Standard は保持されません。

The screenshot shows the 'New Custom Report' interface. It is divided into several sections, each highlighted with a red border and a callout box:

- Basics:** Includes fields for 'Report Name', 'Report Description', and 'Chart Type' (Column or Line). A callout box states: 'Basics (基本情報) の設定欄です。'
- Metrics:** A table with columns for 'Metric', 'Aggregate Function', and 'Actions'. A callout box states: 'Metrics の設定欄です。'
- Dimensions:** Includes a 'Dimension' field and 'Actions'. A callout box states: 'Dimensions の設定欄です。'
- Filter:** A table for 'Filter Conditions' with columns for 'Connector', 'Name', 'Operator', 'Value', and 'Actions'. A callout box states: 'Filter の設定欄です。'

At the bottom, there are 'Cancel' and 'Save' buttons.

図 5-13 : Custom Report の作成画面

表 5-43 : Custom Report 作成画面の項目一覧

| 項目 | 説明 |
|------------|----------------------------------------------------------------------------|
| Basics | レポート名、レポートの説明、X 軸の種別（Column を指定した場合は Dimension、Line を指定した場合は時刻を表示）を指定できます。 |
| Data | グラフのデータ集計間隔（日毎、時毎、分毎のいずれか）と集計対象の Environment を指定できます。 |
| Metrics | Y 軸に表示する Metrics を選択します。複数選択した場合は選択した数だけのグラフがレポートの表示画面に出力されます。 |
| Dimensions | X 軸に表示する Dimension を選択します。複数選択した場合は、欄左側の番号順にドリルダウンすることができます。 |
| Filter | Metrics または Dimension の条件を指定することで、グラフデータのフィルタリングをおこないます。 |

表 5-44 : Custom Report で設定可能な Metric 一覧

| 項目 | 説明 |
|---------------------------------|---------------------------------------------------------------|
| Average transactions per second | 1 秒あたりのリクエストとレスポンス数を表します。 |
| Cache hit | キャッシュにヒットしたリクエスト数を表します。キャッシュにヒットしたリクエストは、バックエンドサービスには転送されません。 |
| L1 Cache Elements Count | トランザクション中における L1 キャッシュ上の要素の数を表します。 |
| Policy errors | Policy でエラーとなったリクエストの数を表します。 |
| Proxy errors | Proxy でエラーとなったリクエストの数を表します。 |
| Request processing latency | Proxy がリクエストを処理する時間を表します。 |
| Request size | リクエストのデータサイズ (KB) を表します。 |
| Response Cache Executed | Response Cache Policy の実行数を表します。 |
| Response processing latency | Proxy がレスポンスを処理する時間を表します。 |
| Response size | レスポンスのデータサイズ (KB) を表します。 |
| Target errors | バックエンドサービスでエラーとなったリクエストの数を表します。 |
| Target response time | Proxy がバックエンドサービスにリクエストを送信してからクライアントにレスポンスを返し始めるまでの時間を表します。 |
| Total response time | Proxy がクライアントからリクエストを受け取ってからレスポンスを返し始めるまでの時間を表します。 |
| Traffic | リクエストとレスポンスの件数を表します。 |

表 5-45 : Custom Report で設定可能な Dimension 一覧

| 項目 | 説明 |
|----------------------|----------------------------------------------------------------------|
| Access Token | OAuth のアクセストークン (OAuth Policy を使用した際) を表示します。 |
| Developer App | リクエスト元のアプリの名前を表示します。 ※Developer App 画面で登録したアプリの名前です。 |
| Developer ID | リクエスト元のアプリ開発者を表示します。 ※Developers 画面で登録した Developer の内部 ID です。 |
| Flow Resource | リクエスト対象のバックエンドサービスのリソースを表示します。 ※Conditional Flow 作成時に指定するリソース名です。 |
| Proxy | API Proxy 名を表示します。 |
| Proxy Client IP | API Proxy の IP アドレスを表示します。 |
| Proxy Path Suffix | API Proxy URI のベースパスより後ろの文字列を表示します。 |
| Request Verb | リクエストメソッドを表示します。 |
| Response Status Code | レスポンスコードを表示します。 |
| Target Response Code | バックエンドサービスのレスポンスコードを表示します。 |
| Target URL | バックエンドサービスの URL を表示します。 |
| User Agent | User-Agent ヘッダの値を表示します。 |
| X Forwarded For | X Forwarded For ヘッダの値を表示します。 |
| Custom Dimensions | Collect Statistics Request Policy で設定した Statics Collector を表示します。 |
| API Product | リクエストを受け取った API Proxy をメンバにもつ Product を表示します。 |

| 項目 | 説明 |
|--------------------------|---------------------------------------------------------------------------------|
| Business Unit ID | 事業 ID を表示します。 |
| Cache Key | Cache Key を表示します。 |
| Cache Name | Cache Name を表示します。 |
| Cache Source | Cache Source を表示します。 |
| Channel ID | チャンネル ID を表示します。 |
| City | リクエスト元の都市名を表示します。 |
| Client Application Name | クライアントアプリの名前を表示します。 |
| Client Host | クライアントアプリをホスティングしているコンピュータの名前を表示します。 |
| Client ID | API Key を表示します。 ※Developers 画面で発行した API Key です。 |
| Client IP Address | リクエスト元の IP アドレスを表示します。 |
| Client Organization Name | クライアントの組織名を表示します。 |
| Client Request ID | クライアントのリクエスト ID を表示します。 |
| Continent | リクエスト元の大陸名を表示します。 |
| Country | リクエスト元の国名を表示します。 |
| Day of week | リクエストした曜日を表示します。 |
| Developer Email | 開発者のメールアドレスを表示します。 ※Developers 画面で登録した Developer のメールアドレスです。 |
| Device Category | デバイスの種類 (Personal computer 等) を表示します。 ※User-Agent に含まれる情報です。 |
| Device ID | デバイス ID を表示します。 |
| Environment | API Proxy がデプロイされている環境の名前 (test、prod 等) を表示します。 |
| Flow Name on Error | エラーが発生したフロー名を表示します。 |
| Flow State on Error | エラーが発生したフローの状態を表示します。 |
| Gateway Flow ID | ゲートウェイフローの ID を表示します。(VM-8D5-S-0011_BTnU7OVE_RouterProxy-8-1047_6 等) |
| Gateway Source | ゲートウェイ (router、message_processor 等) を表示します。 |
| Market ID | マーケット ID を表示します。 |
| Month | リクエストした月を表示します。 |
| Organization | 組織名を表示します。 |
| OS Family | OS の種類 (Windows) を表示します。 ※User-Agent に含まれている情報です。 |
| OS Version | 接続元の OS バージョンを表示します。 例) 6.1 (Windows7 のコードバージョン) ※User-Agent に含まれている情報です。 |
| Partner ID | パートナー ID を表示します。 |
| Policy Name on Error | エラーが発生した Policy の名前を表示します。 |
| Proxy | Proxy Endpoint 名を表示します。 |
| Proxy Base Path | API Proxy の Base Path を表示します。 |
| Proxy Revision | API Proxy の Revision(1, 2, 3 … 等) を表示します。 |
| Referred Client IP | 参照されたクライアント IP を表示します。 |
| Region | 地域を表示します。 |

| 項目 | 説明 |
|---------------------|----------------------------------------------|
| Request Path | リクエスト URI のパス部分を表示します。 |
| Request URI | リクエスト URI を表示します。 |
| Session ID | セッション ID を表示します。 |
| Target | Target Endpoint 名を表示します。 |
| Target Base Path | バックエンドサービスの Base Path を表示します。 |
| Target Host | バックエンドサービスのドメイン名を表示します。 |
| Target IP Address | バックエンドサービスの IP アドレスを表示します。 |
| Time of Day | リクエスト時刻を表示します。 |
| Time Zone | リクエスト元のタイムゾーンを表示します。 |
| Traffic Referral ID | 参照元の ID を表示します。 |
| User Agent Family | エージェントのカテゴリ (Chrome、IE 等) を表示します。 |
| User Agent Type | エージェントの型 (Browser 等) を表示します。 |
| User Agent Version | エージェントのバージョン (Chrome→43.0、IE→11.0 等) を表示します。 |
| Virtual Host | Virtual Host 名を表示します。 |
| Week of Month | リクエストした週を表示します。 |
| ax_isp | ax_isp を表示します。 |

作成画面で Custom Report の作成が完了すると、Custom Report の表示画面 (図 5-14 の通り) に遷移します。Custom Report の表示画面では、ドリルダウンやデータの期間等、グラフの表示条件を変更することができます。



図 5-14 : Custom Report の表示画面

第6章

Admin 機能

ここでは、API Management の Admin 機能について説明します。

第1節 Admin 機能について

Admin 機能では、API Management のユーザー作成とロール（アクセス権限）の設定が可能です。ユーザー（Organization User）は、組織に所属させ権限設定を行ったロール（Organization Role）を割り当てることで、組織内のコンポーネントに対するアクセスを制限します。

ユーザー作成直後は、アカウントがアクティブ化されていません。ユーザー作成時に登録したメールアドレス宛にアクティブ化するためのメールが送信されますので、メールに従ってアクティブ化してください。

ロールを使ったアクセス権限の設定で利用可能なコンポーネントは次の通りです。

- APIs 機能の API Proxies
- Publish 機能の Products と Developers and Apps
- Analytics 機能の Custom Reports

ロールの設定はコンポーネントの要素毎（API Proxy、Caches、Product、Custom Reports）に一括して権限（Create、View、Edit、Delete、Trace、Deploy）を設定する方法と、個別に権限設定を行う方法の2つがあります。また、デフォルトで表 6-1 のロールが用意されています。※個別に設定する方法は、作成済みの要素を対象とするため Create 権限は付与できません。

表 6-1 : デフォルトのロール一覧

| ロール名 | 概要 |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Business User | Publish 機能と Analytics 機能の変更およびトレースの操作ができるロールです。Admin 機能と APIs 機能の Environment Configuration は画面に表示されず、その他機能は読込のみに制限されます。 |
| Operations Administrator | API のデプロイとトレースの操作ができるロールです。Admin 機能と APIs 機能の Environment Configuration は画面に表示されず、その他機能は読込のみに制限されます。 |
| Organization Administrator | 管理者向けのロールで、組織内全ての機能を変更できます。 |
| User | API Proxy の変更とトレースの操作ができるロールです。Admin 機能と APIs 機能の Environment Configuration は画面に表示されず、その他機能は読込のみに制限されます。 |

Admin 機能の概要およびロールで設定可能な機能一覧については、図 6-1 を参照ください。

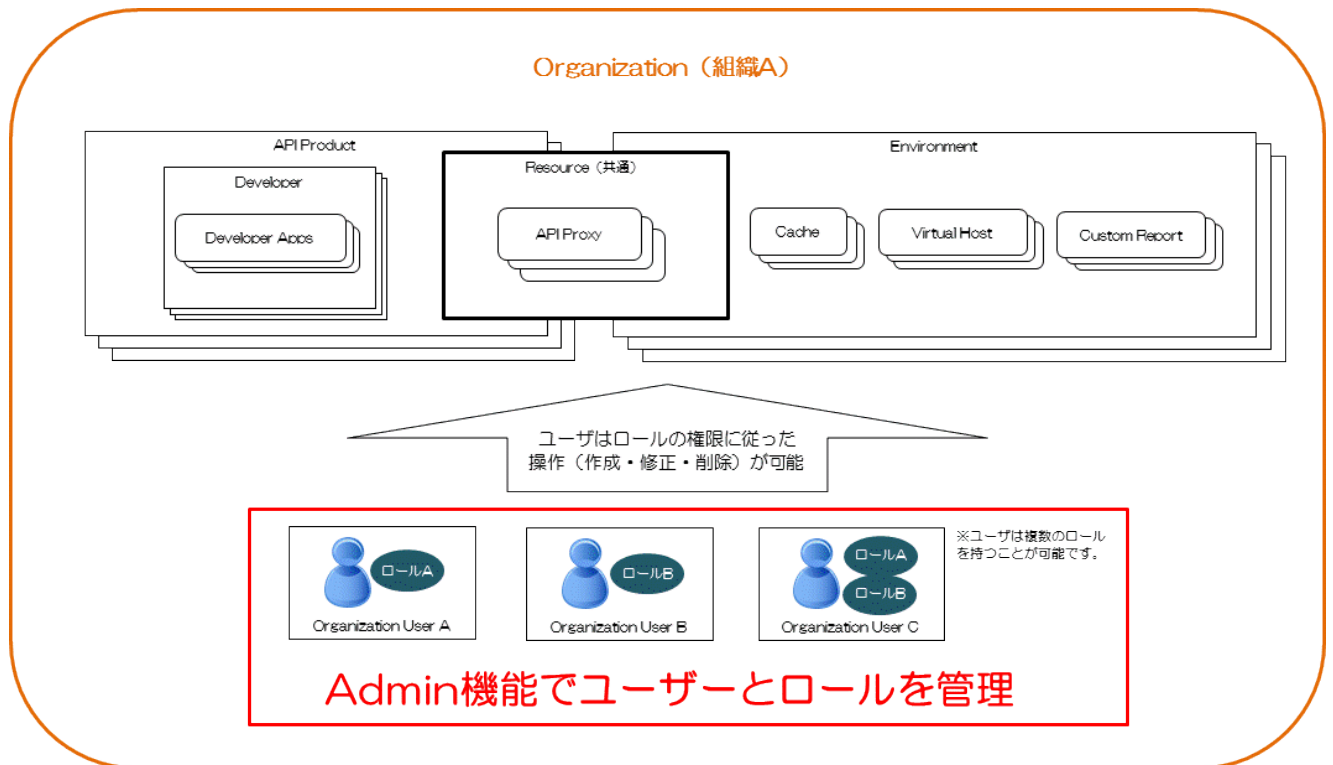


図 6-1 : Admin 機能の概要図

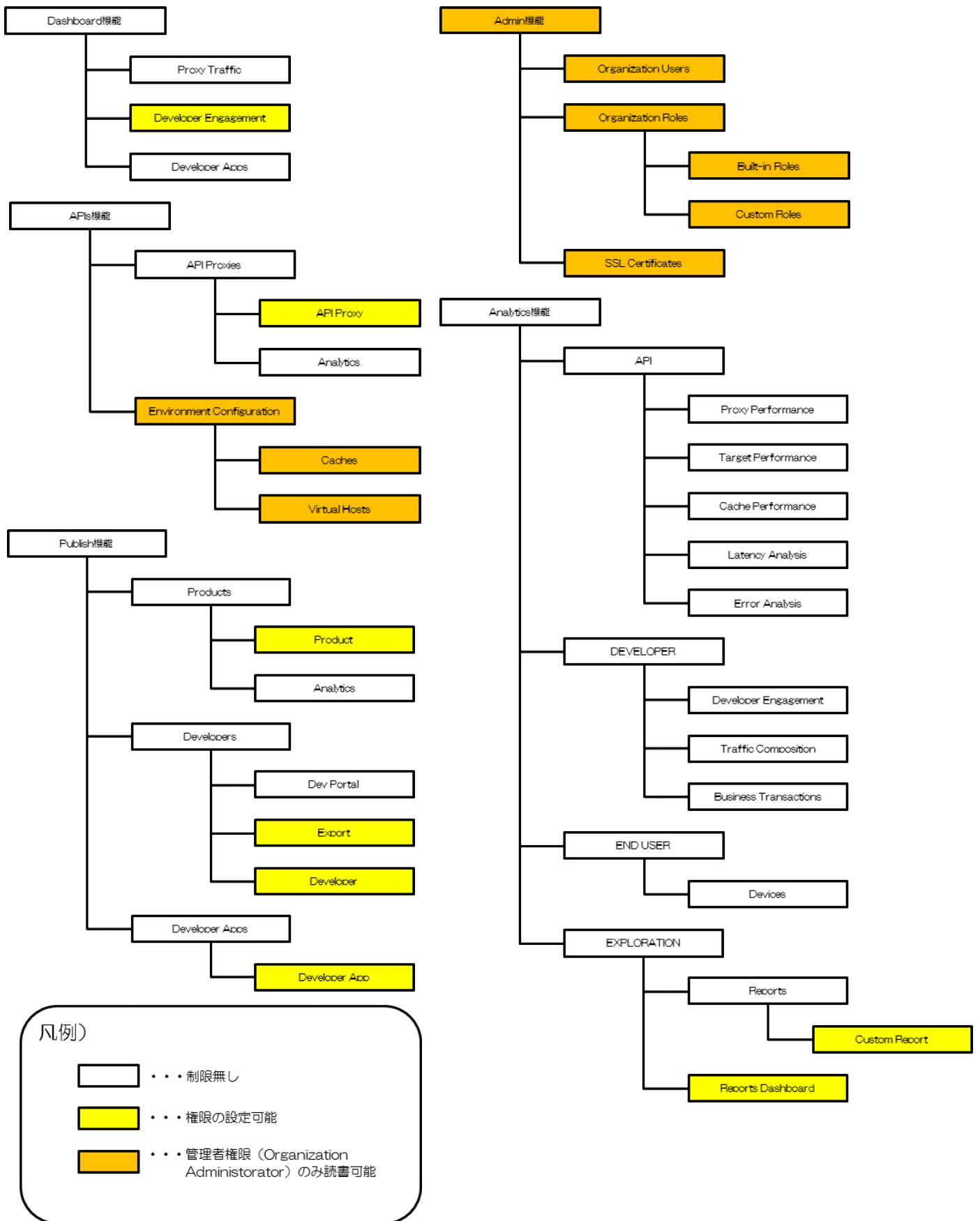


図 6-2 : ロールで設定可能な権限

第2節 Organization History

Admin 機能の Organization History では、組織に対する操作履歴（API コールの情報）を表示します。

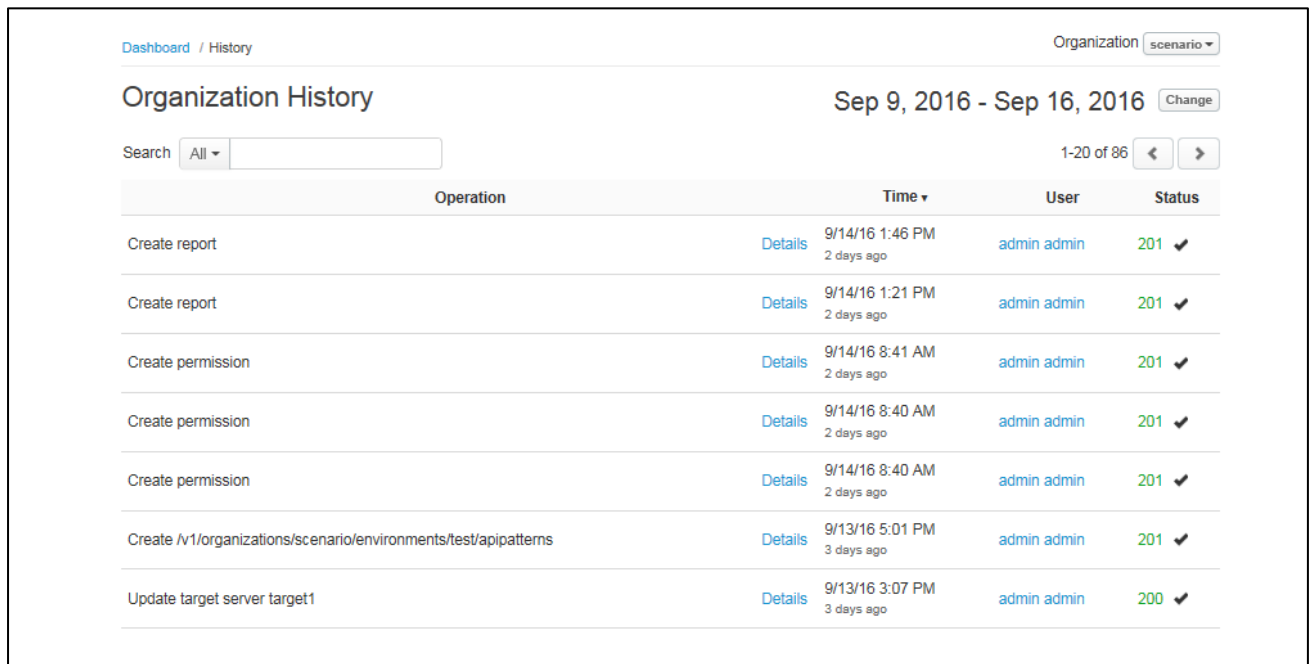


図 6-3 : Organization History の一覧画面

表 6-2 : Organization History 画面の項目

| 項目名 | 説明 |
|-----------|------------------------------------------------------------------------------------|
| Operation | 操作履歴を表示します。Details のリンクをクリックすることで、図 6-4 : Organization History の詳細画面より詳細な内容を表示します。 |
| Time | 操作を行った日時を表示します。 |
| User | 操作を行ったユーザーの名前を表示します。 |
| Status | API コールに対するレスポンスコードを表示します。レスポンスコードは、正常系の場合は緑文字、異常系の場合は赤文字で表示されます。 |



図 6-4 : Organization History の詳細画面

表 6-3 : Organization History 詳細画面の項目

| 項目名 | 説明 |
|-------------|-----------------------------------------------------------------|
| Operation | 操作の種別（CREATE、UPDATE、DELETE）を表示します。 |
| Time | 操作を行った日時を表示します。 |
| Request URI | リクエストの送信先 URI を表示します。 |
| User | 操作を行ったユーザーの名前を表示します。 |
| Request | リクエストの内容を表示します。 |
| Status | API コールに対するリターンコードを表示します。リターンコードは、正常系の場合は緑文字、異常系の場合は赤文字で表示されます。 |