

# Eclipse MicroProfile提供機能に関する考察

三菱UFJインフォメーションテクノロジー  
ITプロデュース部  
高橋 博実

## 三菱UFJインフォメーションテクノロジー(MUIT)

三菱UFJ銀行をはじめとしたMUFGのIT戦略を支える！

システム開発・保守・運用を支援

三菱UFJフィナンシャル・グループ

三菱UFJ銀行

三菱UFJ信託銀行

三菱UFJモルガン・スタンレー証券

三菱UFJニコス

などグループ会社を  
多数サポート

# 会社紹介

◆ **設立** 1988年6月 ※詳しい沿革は、当社HPをご覧ください

◆ **事業内容**

検索ワード：  
MUIT

キャリア採用について：  
HPの採用情報を参照

主として三菱UFJ銀行、並びに三菱UFJフィナンシャル・グループ  
各社の業務等に関する、システムの企画・設計・開発・保守・運用

◆ **本社** 東京都中央区晴海二丁目1-40

◆ **資本金** 181百万円  
(三菱UFJ銀行 85%、三菱UFJフィナンシャル・グループ 15%)

◆ **従業員** 1,972名(2019年4月現在)

# 目次

- 現状のサーバーサイドWebアプリケーションの作り方
- Eclipse MicroProfileとは？
- Eclipse MicroProfileの代表機能の紹介
- 今後の期待とトライしたいこと

# 目次

- 現状のサーバーサイドWebアプリケーションの作り方
- Eclipse MicroProfileとは？
- Eclipse MicroProfileの代表機能の紹介
- 今後の期待とトライしたいこと

# 現状のサーバーサイドWebアプリケーションの作り方

当社におけるWebアプリケーション開発について現状は以下

- SoR (System of Record) 領域についてはJava EEベースのインハウスフレームワークを開発、利用し安定性を享受

**安定性**

**品質**

**長期利用**



- SoE (System of Engagement) 領域についてはSpringBootに社内固有開発部分を追加し、俊敏性を享受

**スピード**

**変更容易性**

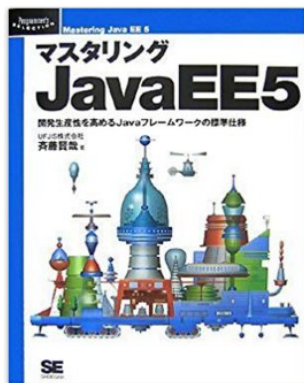
**クラウド向け機能**



- 現状をどう改善していくかが課題
  - 膨大なJava EEベースの既存資産（業務アプリ/共通部品）をどうするか？
    - 一つ一つのシステムが大きく、複雑なため改修しにくい
  - SpringBootについては、初回開発時は生産性が高いがバージョンアップに追従していくコスト/体制/仕組みが必要

# 当社とJava EEの関係

- 2005年のJavaOneに参加した当社員がJava EE5の開発生産性（テスト容易性）に感銘を受けた
- 当時社内標準化されておらず、各協力会社固有の開発手法だった頃からJava EE標準のインハウスフレームワーク整備を企画、開発
- 金融という特性上、ベンダーロックインはNGでありJava EE標準という複数アプリケーションサーバー上で動作するというメリットはマッチした
- 当時日本語書籍が無かったため書籍を出し、日本におけるJava EEの普及に貢献
  - 第1版ではGlassfishベースのサンプルコードだったが、第2版でJBossAS対応



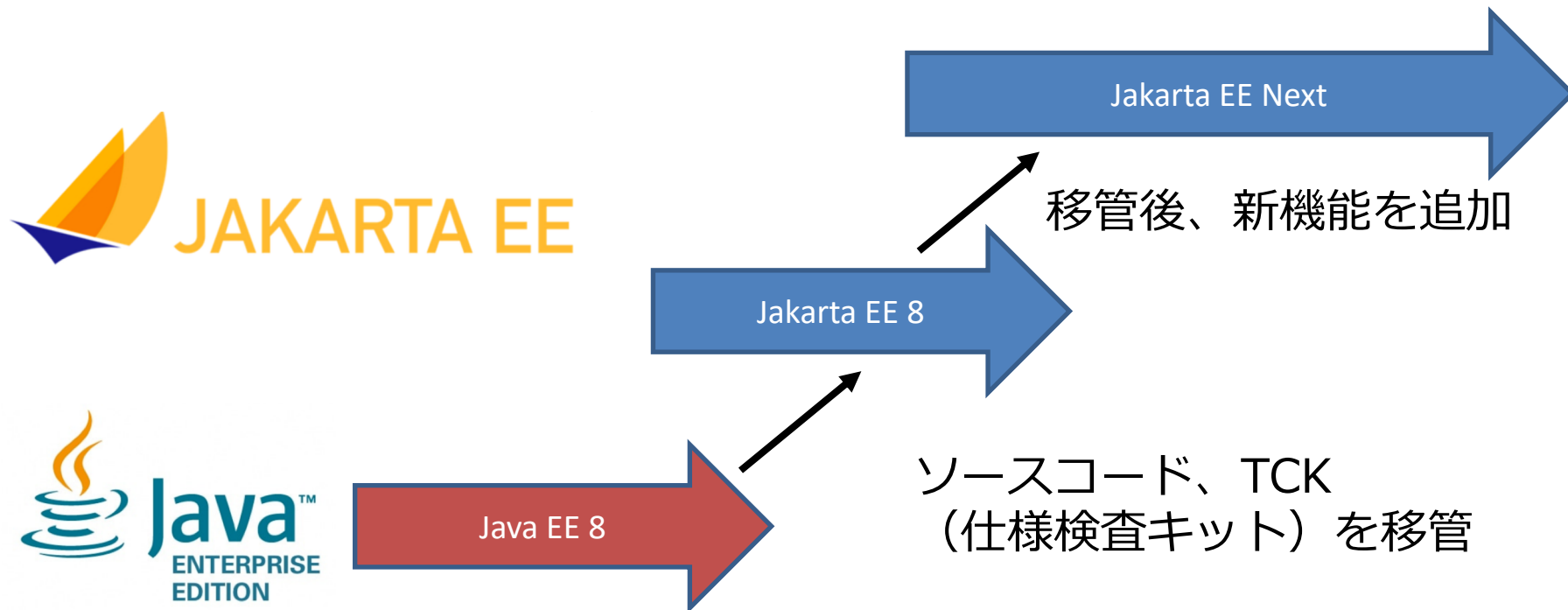
# Java EEの直近の状況

- （皆様ご存知の通り）Java EEはOracle主導を離れEclipse Foundationに移行
- 移行後はJakarta EEというブランド名になり、現在は移管作業中（詳細は後述）
- 単純にこれを待っているだけでは、Java EEネイティブな企業としては望ましくない
- 同じEclipse主導のサーバーサイドJava仕様である、Eclipse MicroProfileについて調査、検討を実施



# Jakarta EEのロードマップ

Jakarta EE8 がリリースされた後、今後の計画がより明確になってくると思われる。

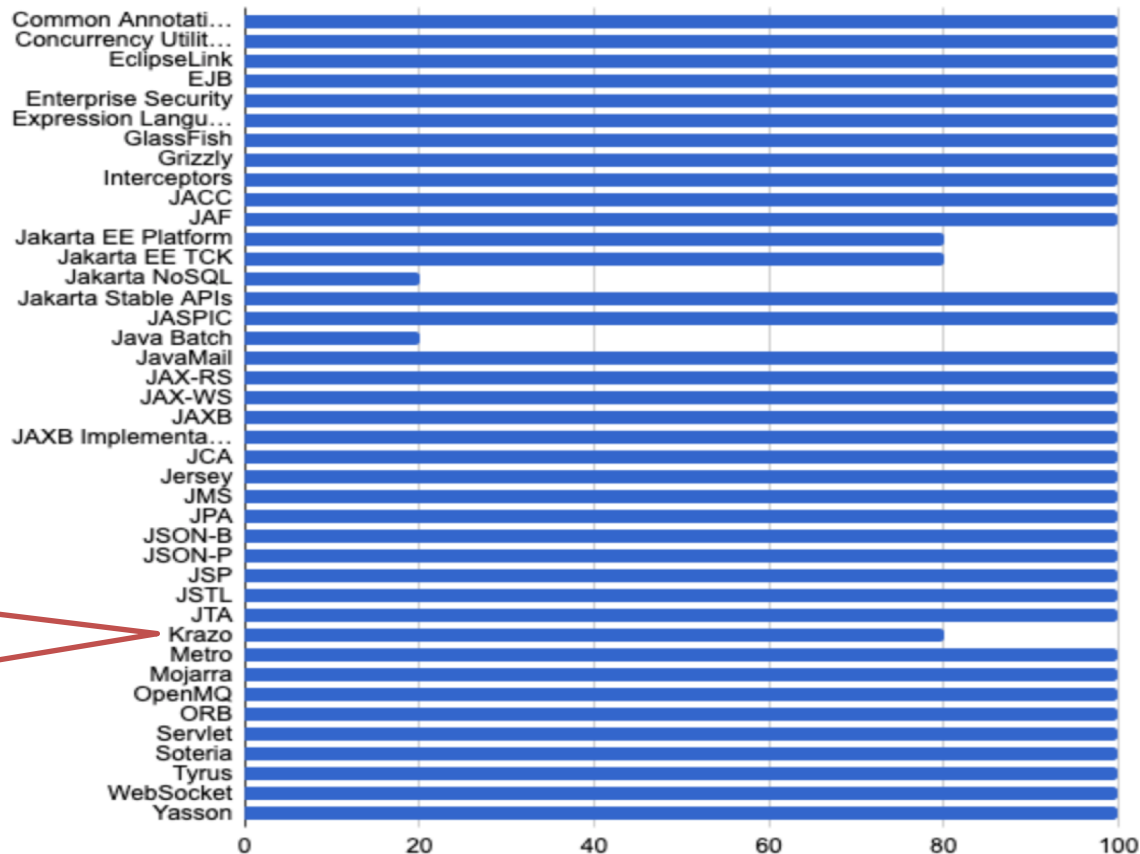


# (参考) Java EE仕様の移管状況

## ■ 以下サイトにて公開

- <https://www.eclipse.org/ee4j/status.php>

## ■ 右のグラフは 2019/04/21時点の もの

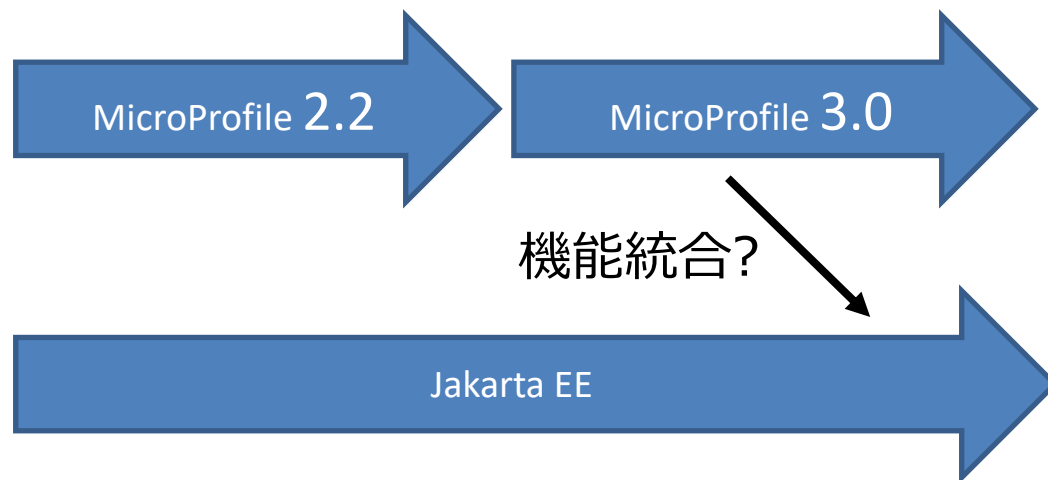


MVC1.0、  
旧Ozarkである  
Krazoへの期待大

# Eclipse MicroProfileとJakarta EEの関係

- 両者は目的が似ており管理主体も同じEclipse Foundation
- 今後は統合をされる見込みであるが、Eclipse Foundationは「まだその時期ではない」とコメント

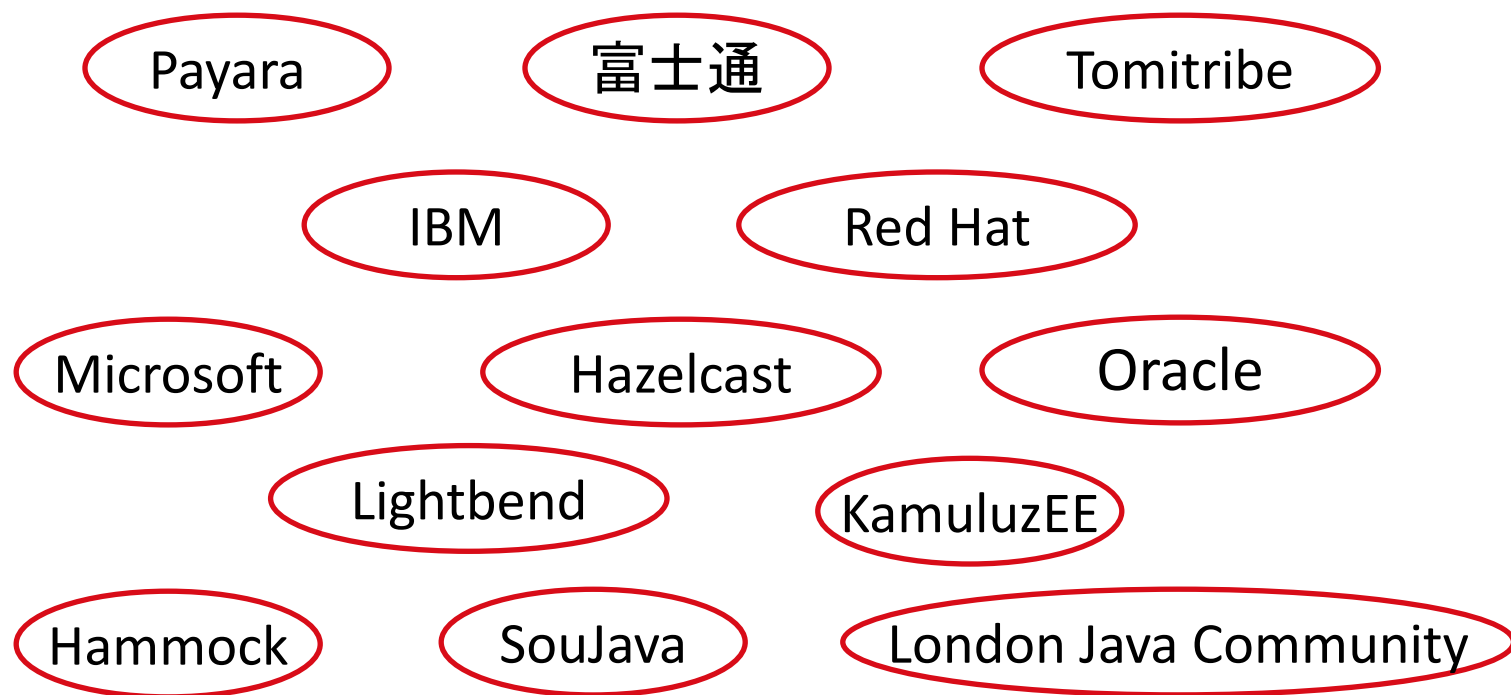
2019/06リリース予定



# 目次

- 現状のサーバーサイドWebアプリケーションの作り方
- **Eclipse MicroProfileとは？**
- Eclipse MicroProfileの代表機能の紹介
- 今後の期待とトライしたいこと

- Java EE技術をベースとしたマイクロサービス向けの仕様群
- 仕様策定メンバーとしては、アプリケーションサーバーベンダーおよびユーザーグループで構成されている



# Java EEとの機能差異

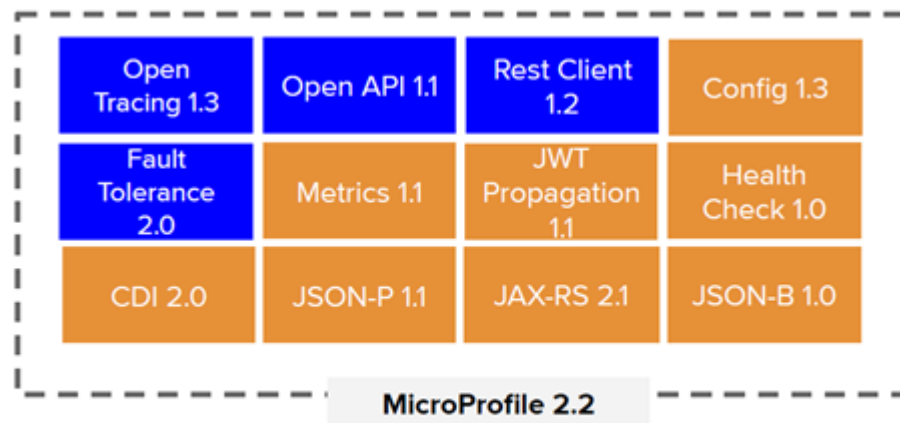
- Eclipse MicroProfileはJava EEのバージョンアップの進化が遅いという課題から誕生した経緯もあり、以下の状況となっている

- Java EEの一部機能のみをサポート

- JAX-RS
- CDI
- JSON-P 等

- Eclipse MicroProfile独自の機能をスピード感を持って拡充。以下は代表例  
詳細は後述

- Config
- Fault Tolerance
- Health Check
- Open Tracing



※<https://microprofile.io/> より画像引用

■ = Updated  
■ = No change from last release (MicroProfile 2.1)

# スタンドアロン仕様について

- 先ほど表記の仕様以外に独立されたMicroProfile仕様
- 2019年6月リリースでは以下を含む方向で議論中
  - Reactive Messaging
    - 今年2月にリリースされたリアクティブ関連仕様のアップデート版
    - <https://github.com/eclipse/microprofile-reactive-messaging>
  - Long Running Actions
    - ロック無く、グローバルに一貫した結果を保証する方法で長期実行アクティビティを調整するための疎結合サービス用のAPI
    - <https://github.com/eclipse/microprofile-lra>
  - GraphQL
    - RESTを置き換える形で提案されているWebAPIの通信仕様。SQLのWeb版
    - <https://github.com/eclipse/microprofile-graphql>

# 簡単に試すには

- 2019年2月にβ版として公開された、MicroProfile Starterが有用
  - 見た目どおり、Spring Initializrに提供されている機能は似ている
  - テンプレートソースプロジェクトのダウンロードが可能

The screenshot shows the MicroProfile Starter web application. The header includes the MicroProfile logo and the text "MicroProfile Starter 'Beta'" and "Generate MicroProfile Maven Project with Examples". The main form contains the following fields and options:

- groupid \***: Input field with "com.example".
- artifactid \***: Input field with "demo".
- MicroProfile Version \***: Dropdown menu.
- Java SE Version**: Dropdown menu with "Java 8" selected.
- Project Options**:
  - MicroProfile Server \***: Dropdown menu.
  - beans.xml \***: Dropdown menu with "implicit" selected.
- Examples for specifications**: A grid of checkboxes for various features:
  - ☒ Config
  - ☒ Fault Tolerance
  - ☒ JWT Auth
  - ☒ Health Metrics
  - ☒ Health Checks
  - ☒ OpenAPI
  - ☒ OpenTracing
  - ☒ TypeSafe Rest Client

At the bottom, there is a "DOWNLOAD" button and a warning message: "⚠ 'implicit' -> no beans.xml, 'annotated' and 'all' are the values for 'bean-discovery-mode'"

MicroProfileの  
バージョンを選定  
可能

実装サーバーを選定  
(TomEE、Payara Micro、  
Open Libertyなど)

Java SEの  
バージョンを選定

MicroProfile  
各機能の利用  
有無を選定



# 目次

- 現状のサーバーサイドWebアプリケーションの作り方
- Eclipse MicroProfileとは？
- **Eclipse MicroProfileの代表機能の紹介**
- 今後の期待とトライしたいこと

# Eclipse MicroProfileの代表機能の紹介

■ Eclipse MicroProfile独自の機能をスピード感を持って拡充。以下は代表例。

- Config

- デフォルト値をシステムプロパティや環境変数で上書き可能  
SpringのConfiguration機能と類似
- 詳細は後述

- Health Checks

- サービスの状態を統一的なフォーマット、REST等で返却  
Spring Boot Actuatorと類似
- 詳細は後述

- Fault Tolerance

- サービス間連携の障害伝播を防ぐために設定に応じてサービス呼び出しを閉塞  
Spring CloudのCircuit Braker(Hystrix/Resilience4j) と類似

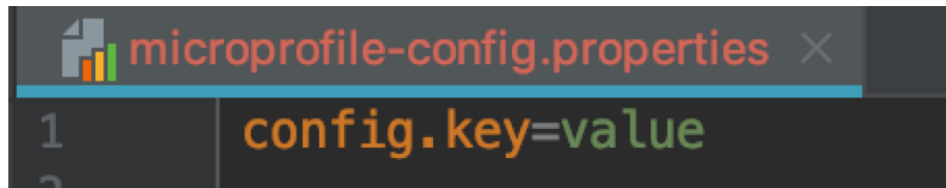
- Open Tracing

- サービスの境界を越えてリクエストをトレーシング可能とする  
Spring Cloud Sluethと類似

# MicroProfile Config

## ■ 設定情報をAPIやインジェクションの形で取得する仕様

### ● ファイル設定例



```
microprofile-config.properties
1 config.key=value
```

### ● LookUpのAPIによる設定値の取得方法

```
Config config = ConfigProvider.getConfig();
String value = config.getValue("config.key", String.class);
```

### ● インジェクションによる設定値の取得方法

```
@Inject
@ConfigProperty(name = "config.key", defaultValue = "default")
private String configValue;
```

## ■ 直感的なAPIで習得が容易

# 設定元と優先順位

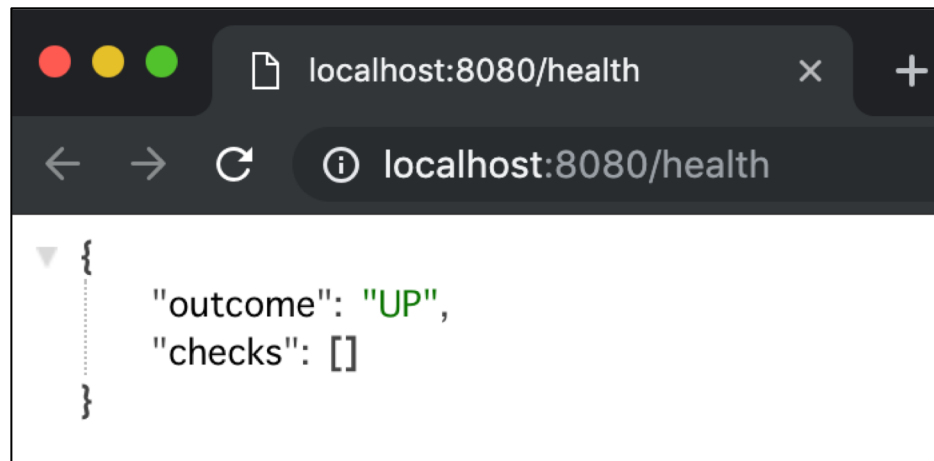
- SpringBootのConfigと同様、設定方法は複数デフォルトで用意されており、同じキーがあった場合の優先度が決められている
  - 優先度は変更することも可能
  - 設定ファイルのデフォルト配置場所は以下
    - META-INF/microprofile-config.properties



	設定例(赤字が適応される結果)
起動時の引数	-Dkey1=val1
システム環境変数	key1=val2 key2=val3
設定ファイル	key2=val4 key3=val5

# MicroProfile HealthCheck

- サービスの妥当性を評価するための仕様
- Kubernetes の liveness チェックに利用されることを想定
- インストールされたマイクロサービスの 状況 (UP または DOWN のいずれか) を表す /health エンドポイントを提供
- デフォルトチェック成功時の結果は右記の通り



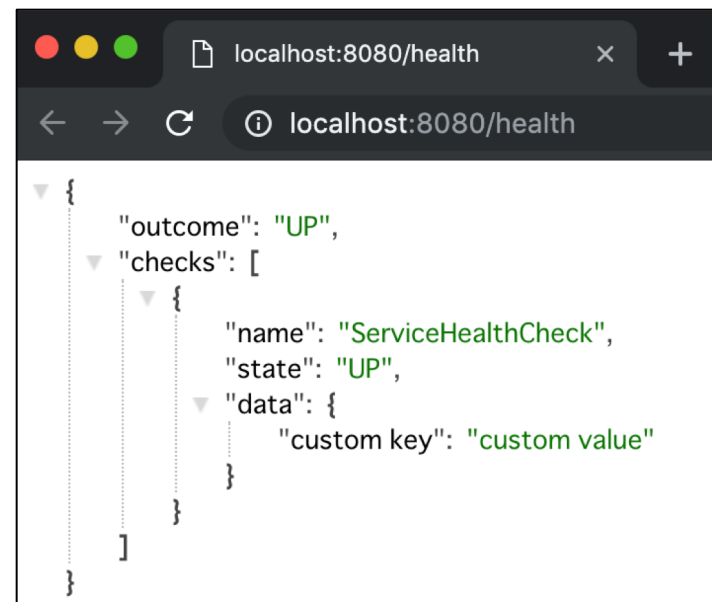
# MicroProfile HealthCheckのカスタマイズ

- 確認内容と成功時の返却値をカスタマイズ可能
- こちらもインターフェースがシンプルで直感的に実装

```
@Health
@ApplicationScoped
public class ServiceHealthCheck implements HealthCheck {

    @Override
    public HealthCheckResponse call() {

        return HealthCheckResponse.named(ServiceHealthCheck.class
            .getSimpleName())
            .withData("custom key", "custom value")
            .up()
            .build();
    }
}
```



```
{
  "outcome": "UP",
  "checks": [
    {
      "name": "ServiceHealthCheck",
      "state": "UP",
      "data": {
        "custom key": "custom value"
      }
    }
  ]
}
```

# 開発スキルセットについての考察

- Java EE ⇒ Eclipse MicroProfile  
同一サブセットのためスキル習得容易
- Spring ⇒ Eclipse MicroProfile  
前述のとおり類似機能があるため、スキル習得容易
- 上記より、現在どちらかしかやっていない人は身構えずに  
試してみることをお勧めする
- 新規でJavaでのWebアプリケーションを学ぶなら、担当がSoR領域になるか、SoE領域になるかでスキル拾得パターンは変わるものと想定

# 目次

- 現状のサーバーサイドWebアプリケーションの作り方
- Eclipse MicroProfileとは？
- Eclipse MicroProfileの代表機能の紹介
- 今後の期待とトライしたいこと



# 今後の期待とトライしたいこと

## ■ Jakarta EEとMicroProfileの関係についての考察

- 次期Java EEであるJakarta EEと今回紹介したMicroProfileは何れ統合される旨が発表されている
  - 今後 Java EE資産も多数あり同時メンテナンスしなければならない状況であれば、Jakarta EEベースの商用アプリケーションサーバーが登場するまでの間は、MicroProfileで継続学習、情報収集していくのも一手ではないか、と考えている

## ■ SoR（トラディショナル）な領域についての移行パスの本格検討

- MicroProfileは上記のとおり一時的な検討材料の一つと位置付け
- 本当に期待しているのはMicroProfileの俊敏性を持ったJakarta EEが進み、それをサポート、実装されるアプリケーションサーバーが続々と出てきて広がること

## ■ Jakarta EE / MicroProfileの継続情報収集

- 来るべきJava EE8以降の移行プラン検討に備える

**皆さんも是非Eclipse MicroProfileをお試してください**

**ご清聴ありがとうございました**

- 本プレゼンテーションにより、貴社と三菱UFJインフォメーションテクノロジー株式会社の間には何ら委任その他の契約関係が発生するものではなく、当社が一切法的な義務・責任を負うものではありません。
- 本資料は信頼できると考えられる各種データに基づいて作成されていますが、当社はその正確性、完全性を保証するものではありません。
- ここに示したすべての内容は、当社の現時点での判断を示しているに過ぎません。
- また、本資料に関連して生じた一切の損害については、当社は責任を負いません。
- その他専門的知識に係る問題については、必ず貴社の弁護士、税理士、公認会計士等の専門家にご相談の上ご確認下さい。