

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】令和3年10月7日(2021.10.7)

【公表番号】特表2020-536320(P2020-536320A)

【公表日】令和2年12月10日(2020.12.10)

【年通号数】公開・登録公報2020-050

【出願番号】特願2020-518519(P2020-518519)

【国際特許分類】

G 06 F 8/60 (2018.01)

G 06 F 9/455 (2006.01)

【F I】

G 06 F 8/60

G 06 F 9/455 1 5 0

【手続補正書】

【提出日】令和3年8月25日(2021.8.25)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

コンテナ環境内の登録されたサービスエンドポイントに対するアプリケーションプログラミングインターフェイス(API)関数を与える方法であって、前記方法は、

APIレジストリにおいてAPI定義を受けるステップを含み、前記API定義は、

コンテナ環境にデプロイされたコンテナにカプセル化された第1のサービスのエンドポイントと、

1つ以上のAPI関数とを含み、前記方法はさらに、

前記APIレジストリが、前記1つ以上のAPI関数と前記サービスのエンドポイントとの間のバインディングを作成するステップと、

前記APIレジストリが、前記第1のサービスの使用を求める要求を第2のサービスから受けるステップと、

前記APIレジストリが、前記1つ以上のAPI関数を前記第2のサービスに与えるステップとを含む、方法。

【請求項2】

前記1つ以上のAPI関数を含む前記第2のサービスのためのクライアントライブラリを生成するステップをさらに含む、請求項1に記載の方法。

【請求項3】

前記クライアントライブラリは、前記API関数を実現するメンバ関数とともにオブジェクトクラスを含む、請求項2に記載の方法。

【請求項4】

前記クライアントライブラリは、前記第1のサービスのエンドポイントに対するダイレクトコールを含む、請求項2または3に記載の方法。

【請求項5】

前記クライアントライブラリは、前記1つ以上のAPI関数のうちの1つが前記第2のサービスによってコールされたときに前記第1のサービスのエンドポイントを実行時にアップデートするコードを含む、請求項2から4のいずれか1項に記載の方法。

【請求項6】

前記クライアントライブラリは、前記1つ以上のA P I関数の1つ以上のパラメータとして受けたデータを、前記第1のサービスのエンドポイントに対するH T T Pコールに再フォーマットする、請求項2から5のいずれか1項に記載の方法。

【請求項7】

前記クライアントライブラリは、前記第1のサービスのエンドポイントの前記H T T Pコールに加えるための追加情報を第3のサービスに要求する、請求項6に記載の方法。

【請求項8】

前記クライアントライブラリは、前記第1のサービスに対する失敗コールをリトライするコードを含む、請求項2から7のいずれか1項に記載の方法。

【請求項9】

前記クライアントライブラリは、結果値を前記第1のサービスの結果セットから抽出し前記結果値を前記1つ以上のA P I関数に対するリターン値としてパッケージングするコードを含む、請求項2から8のいずれか1項に記載の方法。

【請求項10】

1つ以上のプロセッサによって実行されると前記1つ以上のプロセッサに動作を実行させる命令を含むプログラムであって、前記動作は、

A P IレジストリにおいてA P I定義を受けることを含み、前記A P I定義は、

コンテナ環境にデプロイされたコンテナにカプセル化された第1のサービスのエンドポイントと、

1つ以上のA P I関数とを含み、前記動作はさらに、

前記A P Iレジストリが、前記1つ以上のA P I関数と前記サービスのエンドポイントとの間のバインディングを作成することと、

前記A P Iレジストリが、前記第1のサービスの使用を求める要求を第2のサービスから受けることと、

前記A P Iレジストリが、前記1つ以上のA P I関数を前記第2のサービスに与えることを含む、プログラム。

【請求項11】

前記第1のサービスのエンドポイントは、

I Pアドレスと、

ポート番号とを含む、請求項10に記載のプログラム。

【請求項12】

前記コンテナ環境は、コンテナにカプセル化された複数のサービスを含む、請求項10または11に記載のプログラム。

【請求項13】

前記A P Iレジストリは、前記コンテナ環境内のコンテナにカプセル化されたサービスとしてデプロイされる、請求項10から12のいずれか1項に記載のプログラム。

【請求項14】

前記A P Iレジストリは、

統合開発環境（I D E）において開発中のサービスと、

前記コンテナ環境において既にデプロイされているサービスとが、利用できる、請求項10から13のいずれか1項に記載のプログラム。

【請求項15】

前記動作はさらに、前記A P Iレジストリが、前記コンテナ環境にデプロイされているコンテナ化されたサービスの任意の利用できるエンドポイントを識別することを含む、請求項10から14のいずれか1項に記載のプログラム。

【請求項16】

1つ以上のプロセッサと、

1つ以上のメモリデバイスとを備えるシステムであって、前記1つ以上のメモリデバイスは、前記1つ以上のプロセッサによって実行されると前記1つ以上のプロセッサに動作を実行させる命令を含み、前記動作は、

A P I レジストリにおいて A P I 定義を受けることを含み、前記 A P I 定義は、コンテナ環境にデプロイされたコンテナにカプセル化された第 1 のサービスのエンドポイントと、

1 つ以上の A P I 関数とを含み、前記動作はさらに、

前記 A P I レジストリが、前記 1 つ以上の A P I 関数と前記サービスのエンドポイントとの間のバインディングを作成することと、

前記 A P I レジストリが、前記第 1 のサービスの使用を求める要求を第 2 のサービスから受けることと、

前記 A P I レジストリが、前記 1 つ以上の A P I 関数を前記第 2 のサービスに与えることとを含む、システム。

【請求項 1 7】

前記動作はさらに、前記 A P I レジストリに登録されたすべてのサービスに対する A P I 関数のリストをユーザインターフェイスに表示させることを含む、請求項 1 6 に記載のシステム。

【請求項 1 8】

前記第 1 のサービスの使用を求める前記第 2 のサービスからの要求は、前記ユーザインターフェイスに表示された前記 1 つ以上の A P I 関数のうちの 1 つの選択を受けることによって受ける、請求項 1 7 に記載のシステム。

【請求項 1 9】

前記ユーザインターフェイスはコマンドラインインターフェイスを含む、請求項 1 7 または 1 8 に記載のシステム。

【請求項 2 0】

前記ユーザインターフェイスはグラフィカルユーザインターフェイスを含む、請求項 1 7 または 1 8 に記載のシステム。