

SONY



Vision and Sensing Application SDK AIモデル・PPLインポート 機能仕様書

Copyright 2023 Sony Semiconductor Solutions Corporation

Version 0.2.0

2023 - 1 - 30

AITRIOS™、およびそのロゴは、ソニーグループ株式会社またはその関連会社の登録商標または商標です。

目次

1. 更新履歴	1
2. 用語・略語	2
3. 参照資料	3
4. 想定ユースケース	4
5. 機能概要、アルゴリズム	5
6. 操作性仕様、画面仕様（AIモデルインポート）	17
7. 操作性仕様、画面仕様（AIモデル削除）	22
8. 操作性仕様、画面仕様（PPLインポート）	24
9. 操作性仕様、画面仕様（PPL削除）	27
10. 目標性能	30
11. 制限事項	31
12. その他特記事項	32
13. 未決定事項	33

1.更新履歷

Date	What/Why
2023/01/30	初版作成

2. 用語・略語

Terms/Abbreviations	Meaning
PPL	エッジAIデバイスのAIモデルの出力(Output Tensor)を加工処理するモジュール
SAS	Shared Access Signatures Azure Storage リソースへの制限付きアクセス権を付与する URI

3. 参照資料

- Reference/Related documents（関連資料）
 - APIリファレンス
 - <https://developer.aitrios.sony-semicon.com/development-guides/reference/api-references/>
 - Console Access Library機能仕様書
 - <https://developer.aitrios.sony-semicon.com/development-guides/documents/specifications/>

4. 想定ユースケース

- 自分の環境で作成したAIモデルをConsole for AITRIOSにインポートしたい
- 作成したPPLをConsole for AITRIOSにインポートしたい
- AIモデル、またはPPLのインポート状況を確認したい
- Console for AITRIOSにインポートしたAIモデルを、エッジAIデバイスへデプロイできる形式に変換したい
- AIモデル変換状況を確認したい
- Console for AITRIOSにインポート済みのAIモデル、またはPPLを削除したい

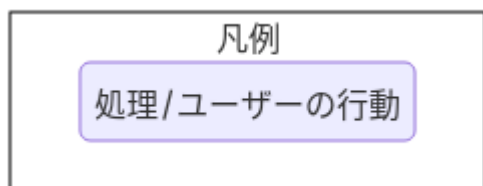
5. 機能概要、アルゴリズム

Functional Overview

- SDKのDev Container(Local PCまたはCodespaces)において、Console Access Libraryを扱うことができる
 - Console Access Library経由で実行できることは下記の通り
 - AIモデルをConsole for AITRIOSにインポートできる
 - 使用できるAIモデルは [Console for AITRIOSの動作仕様](#) に従う
 - Console for AITRIOSへのインポート元はAzure Blob Storageの為、AIモデルはAzure Blob Storageへの保存が必要
 - Console for AITRIOSにインポートされたAIモデルを変換できる
 - PPLをConsole for AITRIOSにインポートできる
 - 使用できるPPLファイルは下記の通り
 - SDKでは「.wasm」（未AOTコンパイルファイル）をサポートする

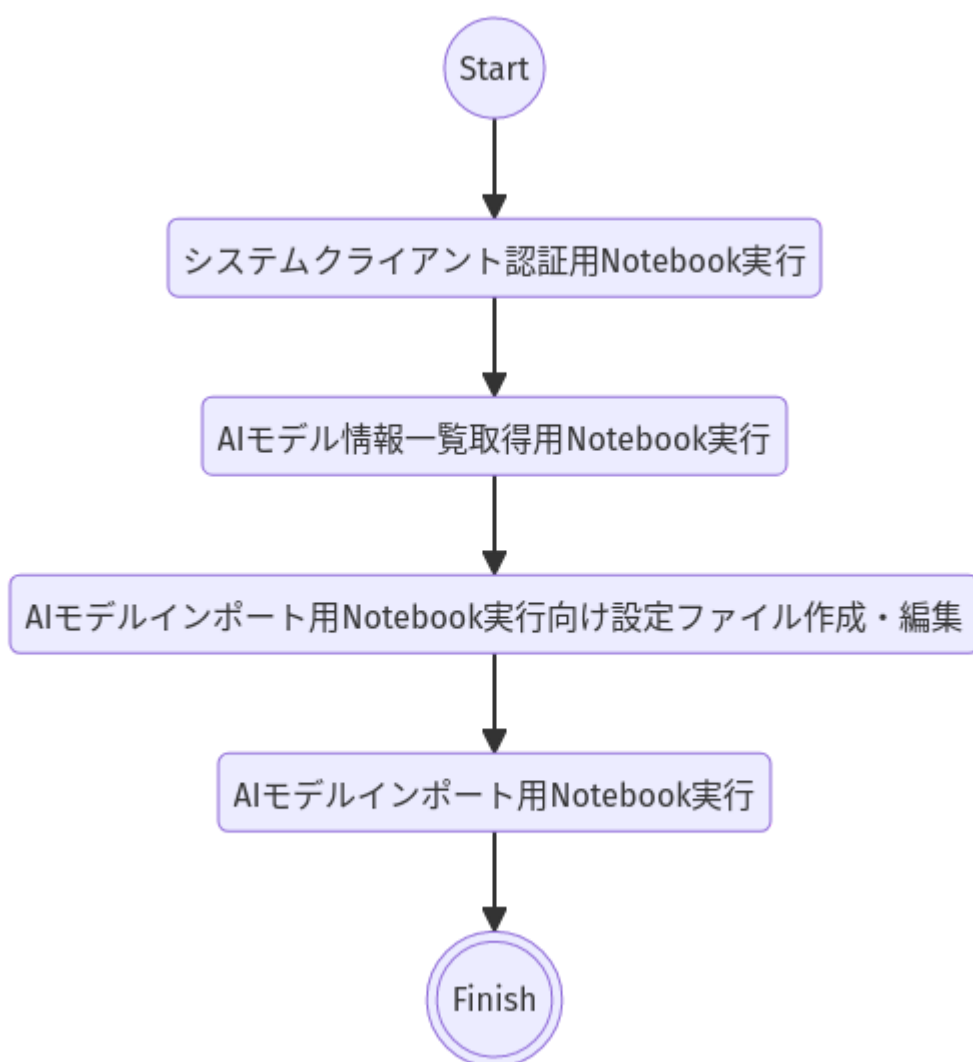
Console Access Libraryがサポートするインポートが可能なPPLの拡張子	SDKにおけるサポート
.wasm(未AOTコンパイルファイル)	Yes
.aot(AOTコンパイル済みファイル)	No

フロー



AIモデルインポート

- フロー



- フロー詳細

1. システムクライアント認証用Notebook実行

2. AIモデル情報一覧取得用Notebook実行

- Console for AITRIOSにインポート済みの `model_id` が必要な場合にAIモデル情報一覧取得用Notebookを実行する
 - 下記のケースを想定
 - Console for AITRIOSへインポート済みのAIモデルをバージョンアップしたい
 - Console for AITRIOSのAIモデルインポート状態を確認したい
 - Console for AITRIOSのAIモデルの変換状況を確認したい

3. AIモデルインポート用Notebook実行向け設定ファイル作成・編集

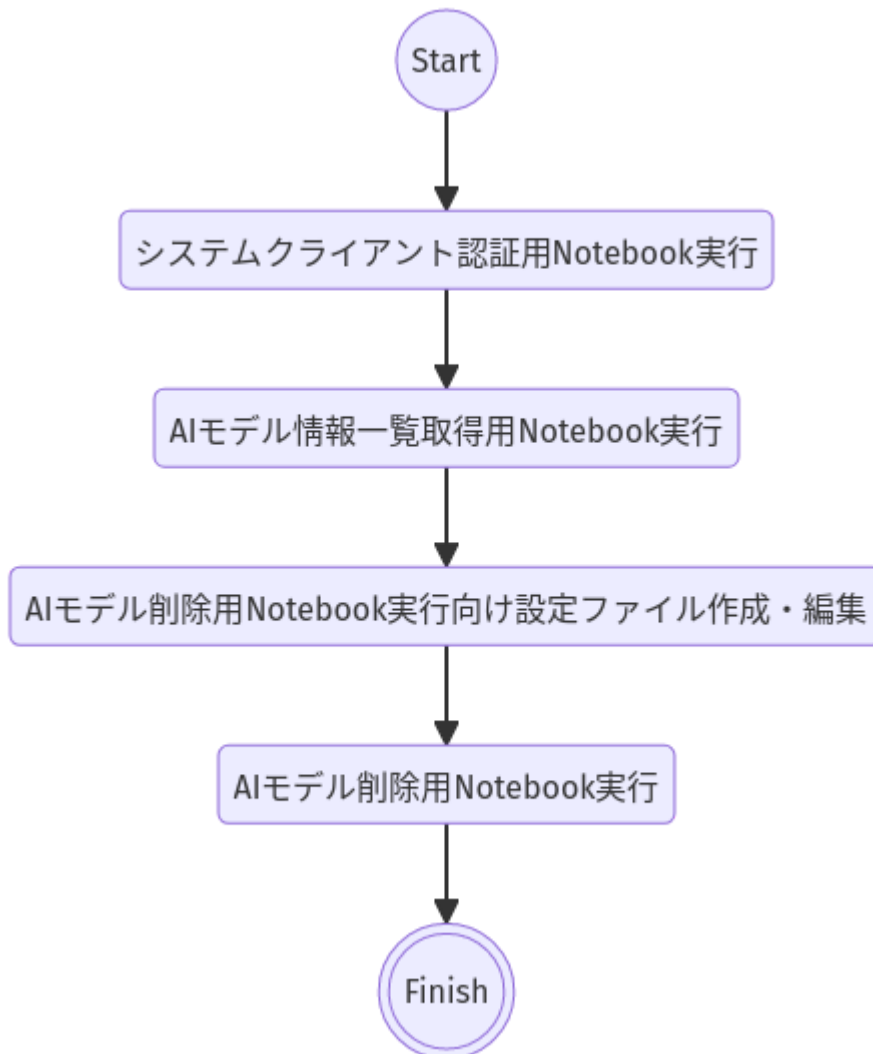
- 設定ファイル`configuration.json`を作成、編集してNotebook実行時の設定を行う

4. AIモデルインポート用Notebook実行

- 下記の機能を持つNotebookを実行する
 - AIモデルをConsole for AITRIOSへインポート
 - Console for AITRIOSのAIモデルインポート状態を確認
 - Console for AITRIOSへインポートしたAIモデルを変換
 - AIモデルの変換状況を確認

AIモデル削除

- フロー

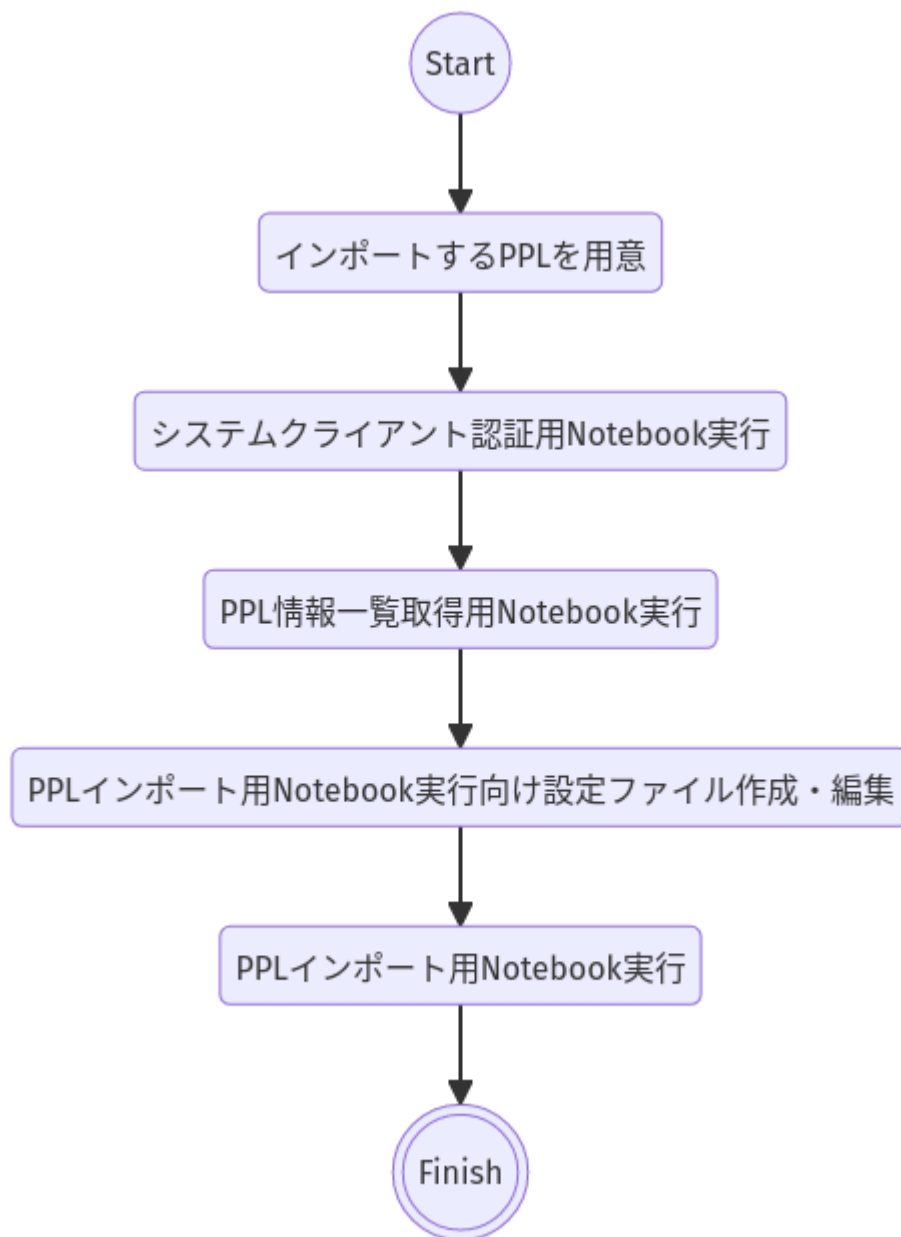


- フロー詳細

1. システムクライアント認証用Notebook実行
2. AIモデル情報一覧取得用Notebook実行
 - AIモデル情報一覧取得用Notebookを実行して、Console for AITRIOSへインポート済みの **model_id** を取得する
3. AIモデル削除用Notebook実行向け設定ファイル作成・編集
 - 設定ファイル **configuration.json** を作成、編集してNotebook実行時の設定を行う
4. AIモデル削除用Notebook実行
 - AIモデル削除用Notebook実行して、Console for AITRIOSから対象のAIモデルを削除する

PPLインポート

- フロー



- フロー詳細

1. インポートするPPLを用意

- インポートするPPLを、SDK実行環境へ格納する

2. システムクライアント認証用Notebook実行

3. PPL情報一覧取得用Notebook実行

- Console for AITRIOSにインポート済みの **app_name** 、 **version_number** が必要な場合にPPL情報一覧取得用Notebookを実行する
- 下記のケースを想定
 - Console for AITRIOS上のPPLインポート状態を確認したい

4. PPLインポート用Notebook実行向け設定ファイル作成・編集

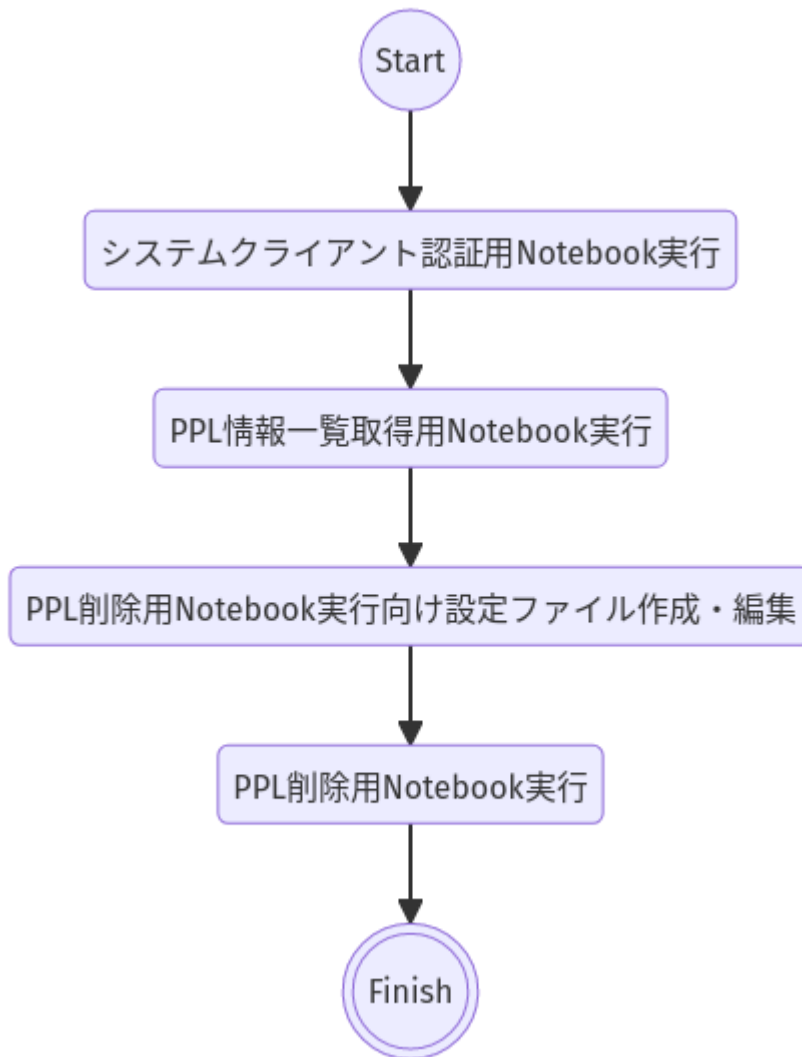
- 設定ファイル`configuration.json`を作成、編集してNotebook実行時の設定を行う

5. PPLインポート用Notebook実行

- 下記の機能を持つNotebookを実行する
 - PPLをBase64形式でエンコード
 - PPLをConsole for AITRIOSへインポート
 - Console for AITRIOS上のPPLインポート状態を確認

PPL削除

- フロー

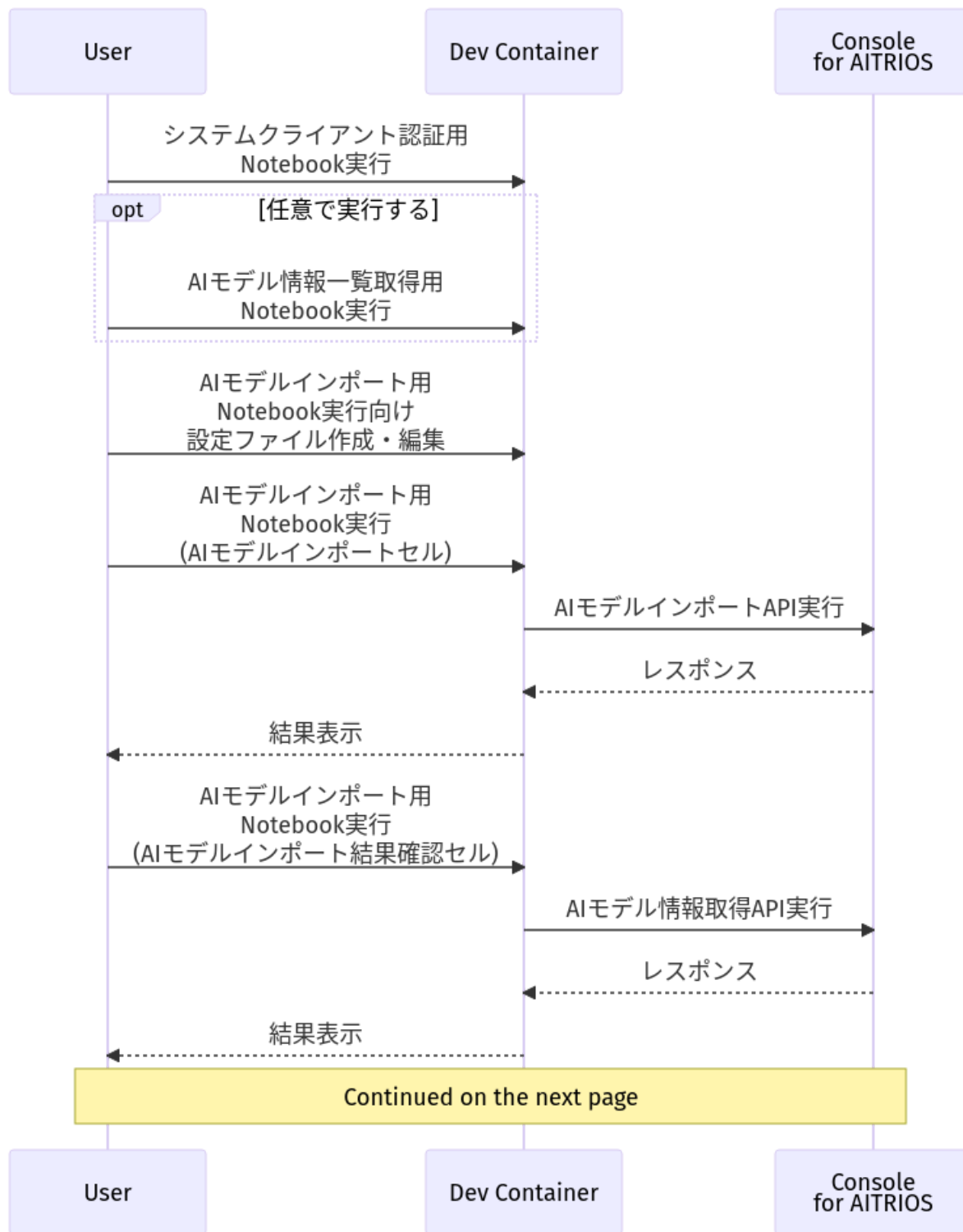


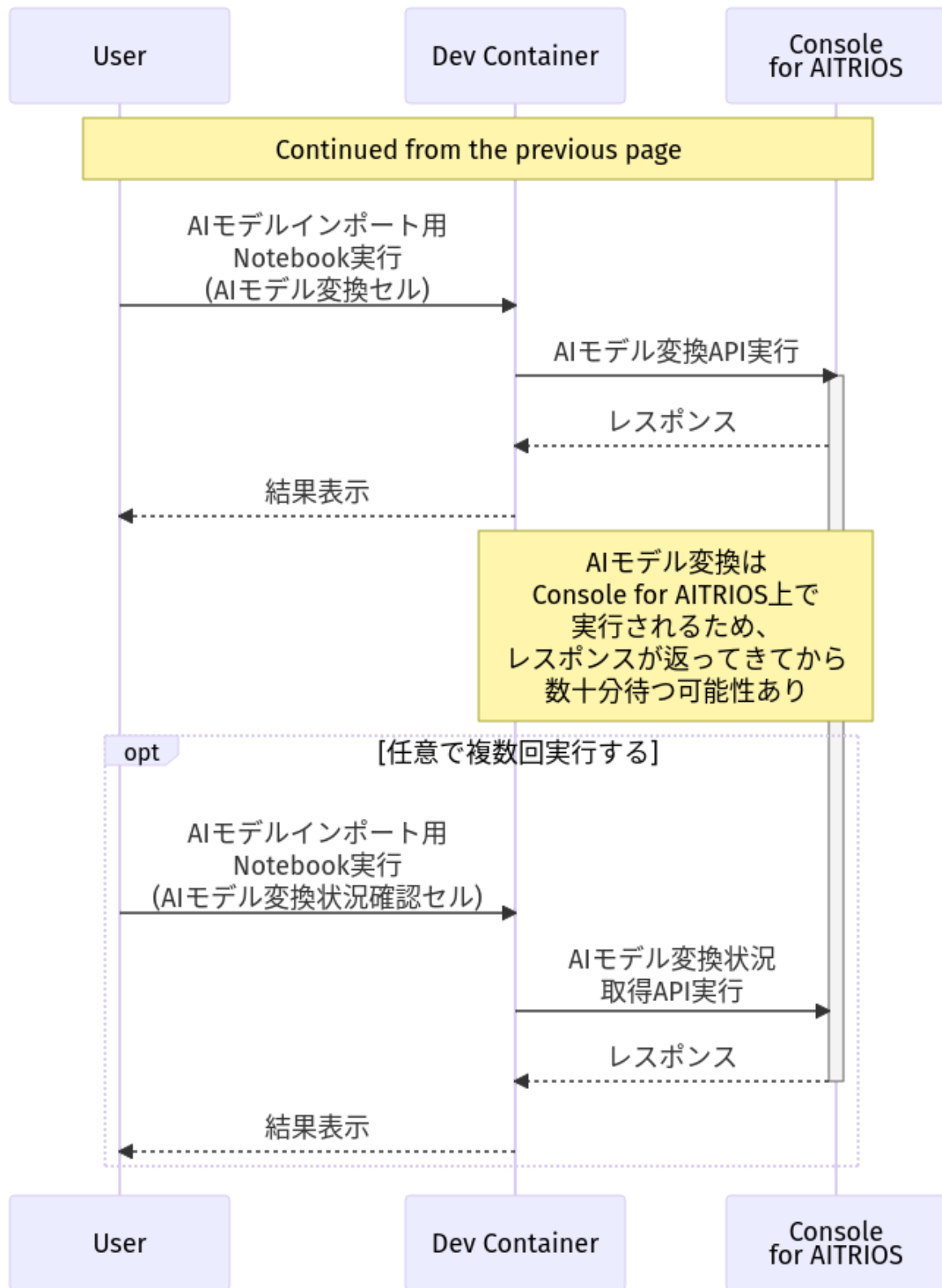
- フロー詳細

1. システムクライアント認証用Notebook実行
2. PPL情報一覧取得用Notebook実行
 - PPL情報一覧取得用Notebookを実行して、Console for AITRIOSへインポート済みの **app_name** 、 **version_number** を取得する
3. PPL削除用Notebook実行向け設定ファイル作成・編集
 - 設定ファイル **configuration.json** を作成、編集してNotebook実行時の設定を行う
4. PPL削除用Notebook実行
 - PPL削除用Notebook実行して、Console for AITRIOSから対象のPPLを削除する

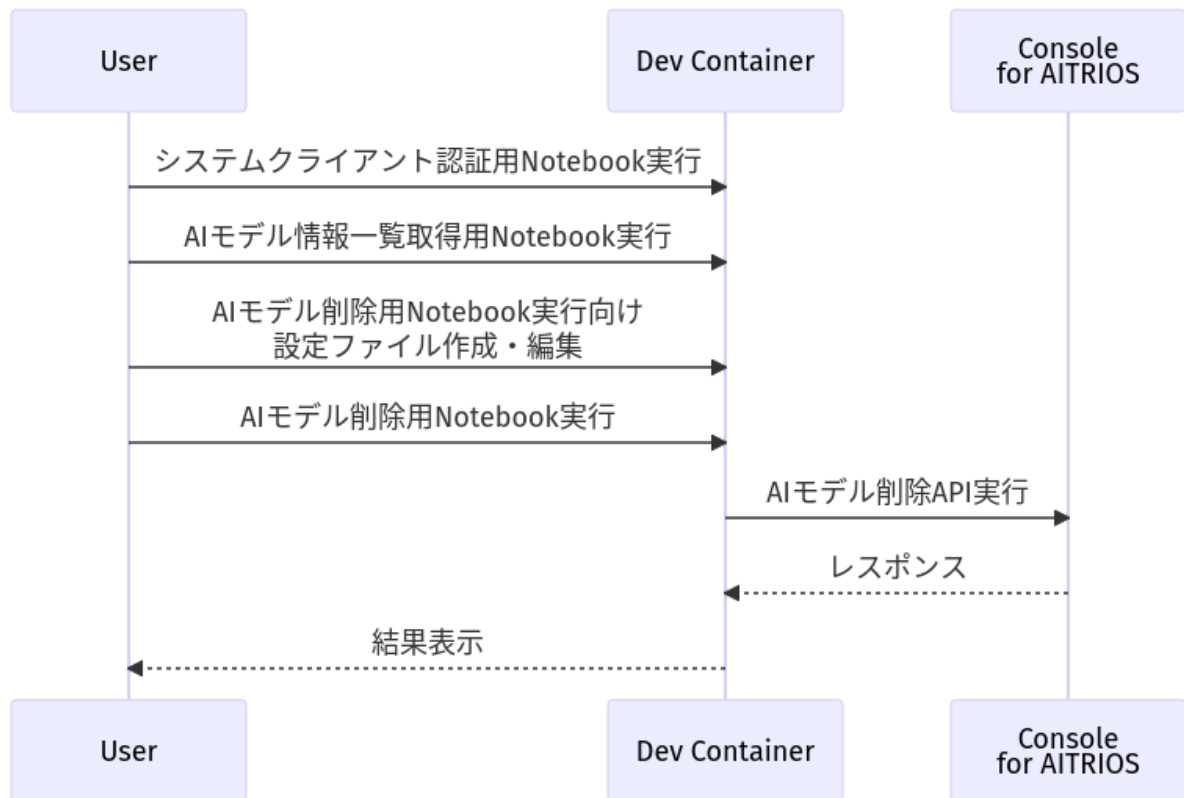
シーケンス

AIモデルインポート

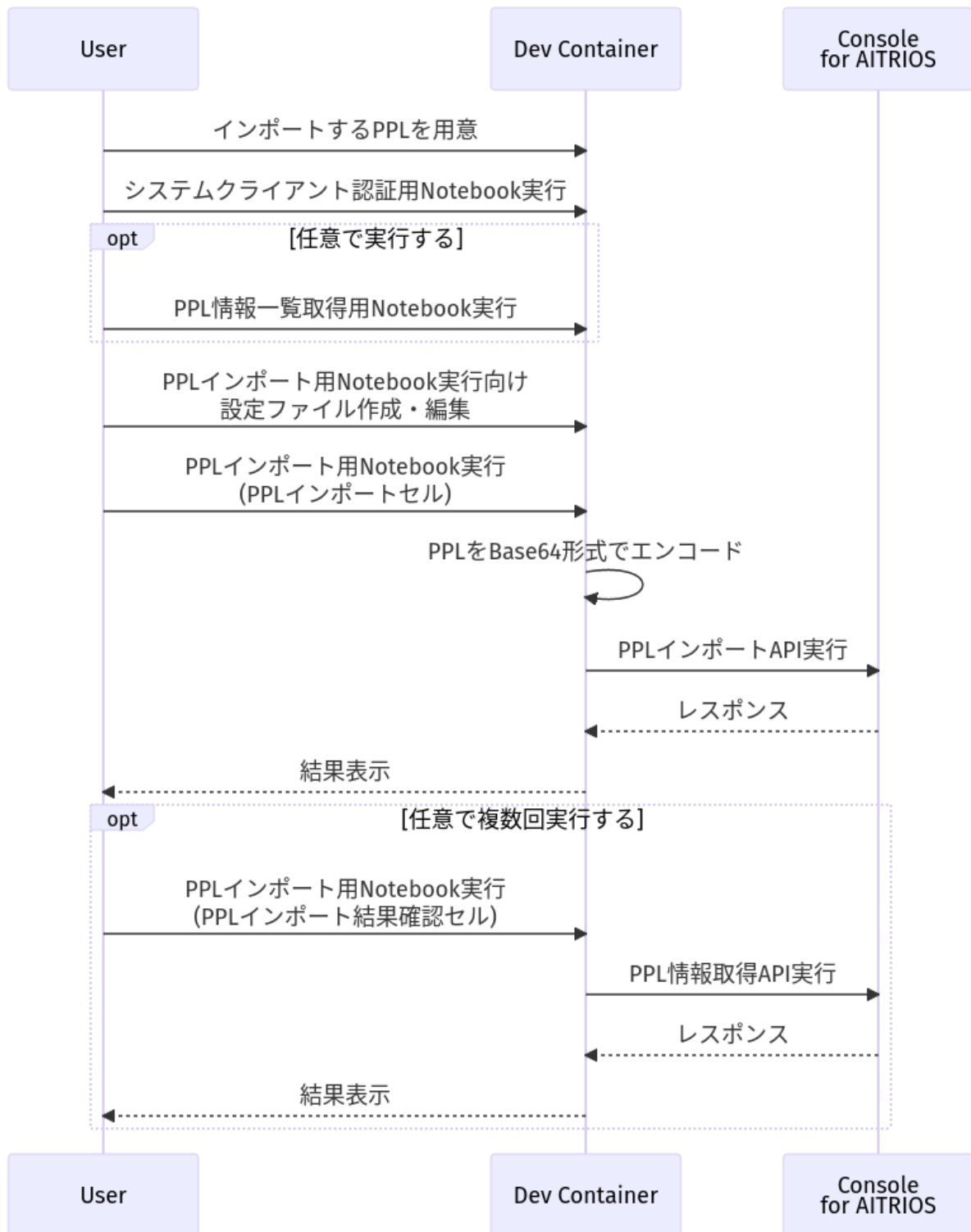




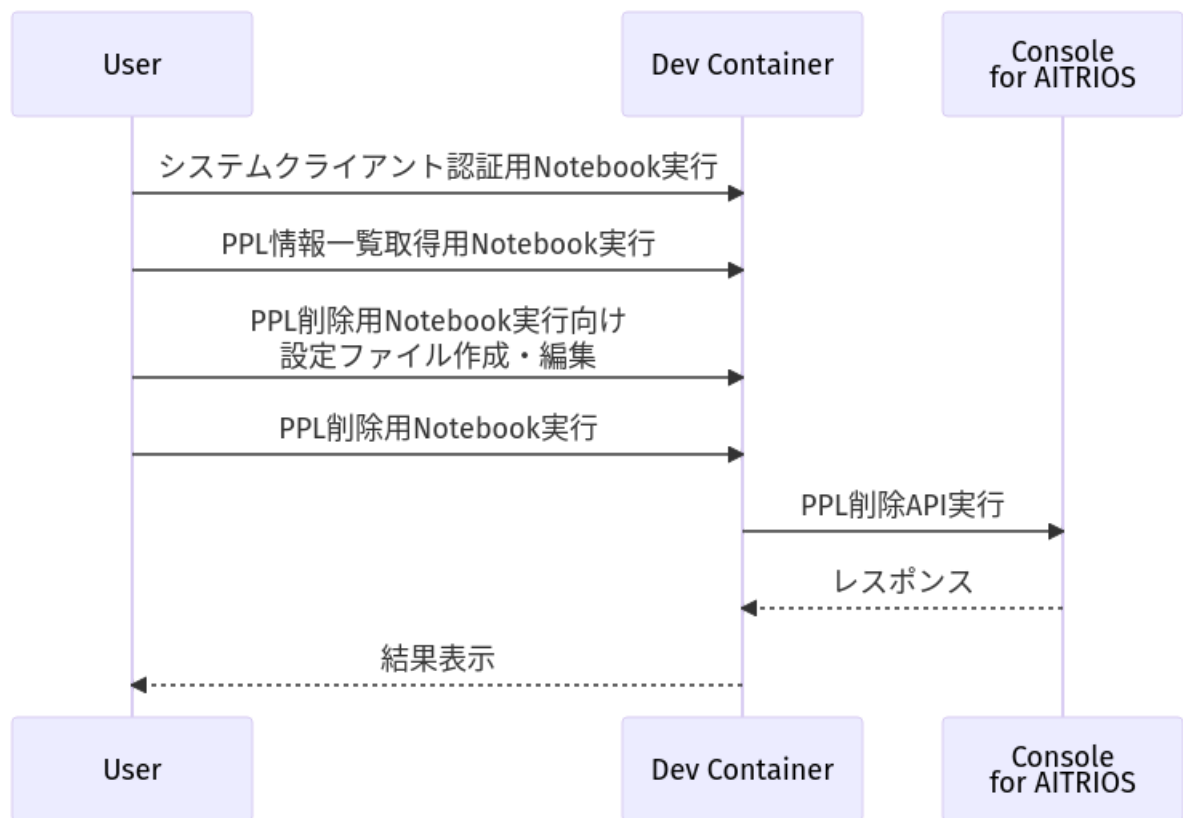
AIモデル削除



PPLインポート



PPL削除



6. 操作性仕様、画面仕様（AIモデルインポート）

前提条件

- Portal for AITRIOSからユーザー登録し、AITRIOSのプロジェクトに参加していること
- AIモデルを用意していること
- AIモデルをAzure Blob Storageへアップロードし、SAS URIを取得していること

How to start each function

1. SDK環境を立ち上げ、Topの **README.md** をプレビュー表示する
2. SDK環境Topの **README.md** に含まれるハイパーリンクから、**tutorials** ディレクトリの **README.md** にジャンプする
3. **tutorials** ディレクトリの **README.md** に含まれるハイパーリンクから、**3_prepare_model** ディレクトリの **README.md** にジャンプする
4. **3_prepare_model** ディレクトリの **README.md** に含まれるハイパーリンクから、**develop_on_sdk** ディレクトリの **README.md** にジャンプする
5. **develop_on_sdk** ディレクトリの **README.md** に含まれるハイパーリンクから、**3_import_to_console** ディレクトリの **README.md** にジャンプする
6. **3_import_to_console** ディレクトリの各ファイルから各機能に遷移する

システムクライアント認証用Notebook実行

1. **3_import_to_console** ディレクトリの **README.md** に含まれるハイパーリンクから、**set_up_console_client** ディレクトリの **README.md** にジャンプする
2. **set_up_console_client** ディレクトリのシステムクライアント認証用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

AIモデル情報一覧取得用Notebook実行

1. **3_import_to_console** ディレクトリの **README.md** に含まれるハイパーリンクから、**get_model_list** ディレクトリの **README.md** にジャンプする
2. **get_model_list** ディレクトリのAIモデル情報一覧取得用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

AIモデルインポート用Notebook実行向け設定ファイル作成・編集



特別な記載がある場合を除き、原則として省略は不可。



Console Access Library APIに渡されるパラメータについては、[Console Access Library API](#)の仕様に従う。

1. 実行ディレクトリに設定ファイル(**configuration.json**)を作成し、編集する

Configuration	Meaning	Range	Remarks
model_id	インポートするAIモデルのID 新しい model_id の場合は新規登録、システムに既に登録されている model_id を指定した場合はバージョンアップを行う	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code> ・ <code>ai_model.ai_model.AIModel.get_base_model_status</code> ・ <code>ai_model.ai_model.AIModel.publish_model</code>
model	インポートするAIモデル用 SAS URI	SAS URI形式 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>

Configuration	Meaning	Range	Remarks
converted	変換済みフラグ	true or false 詳細はConsole Access Library APIの仕様に従う。	省略可 省略した場合、false指定と同じ動作となる 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>
vendor_name	ベンダー名(新規登録の場合に指定)	文字列 詳細はConsole Access Library APIの仕様に従う。	省略可 省略した場合、ベンダー名なし 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>
comment	AIモデルとバージョンの説明 新規登録時はAIモデルとバージョンの説明、バージョンアップ時はバージョンの説明として設定	文字列 詳細はConsole Access Library APIの仕様に従う。	省略可 省略した場合、説明なしとして設定 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>

Configuration	Meaning	Range	Remarks
network_type	ネットワーク種別	文字列 詳細はConsole Access Library APIの仕様に従う。	省略可 新規登録の場合のみ有効 省略した場合、"1"指定と同じ動作となる 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>
labels	ラベル名 Custom Visionの場合、AIモデルファイルに付属するlabel.txtファイルの内容を設定する	["label01","label02","label03"] 詳細はConsole Access Library APIの仕様に従う。	省略可 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model.AIModel.import_base_model</code>

AIモデルインポート用Notebook実行

1. `3_import_to_console` の `import_to_console.ipynb` を開き、その中のPythonスクリプトを実行する
 - その後下記の動作をする
 - `3_import_to_console` ディレクトリの`configuration.json`の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
 - `configuration.json`の内容をチェックする
 - エラー発生時はその内容を表示し、中断する
 - AIモデルインポートAPIを実行する
 - インポートに成功した場合は、`import_to_console.ipynb` 内の出力に成功した旨のメッセージを表示する
 - AIモデルインポート結果確認APIを実行する
 - AIモデルの情報取得に成功した場合は、`import_to_console.ipynb` 内の出力に成功した旨のメッセージと取得したAIモデルのステータスを表示する
 - AIモデル変換APIを実行する
 - API実行に成功した場合は、`import_to_console.ipynb` 内の出力に成功した旨のメッセージを表示する
 - AIモデルの変換完了までは数十分程度の時間がかかるため、次に動作する「AIモデル変換状況確認APIを実行する」で状況の確認が必要
 - AIモデル変換状況確認APIを実行する
 - AIモデルの変換状況取得に成功した場合は、`import_to_console.ipynb` 内の出力に成功した旨のメッセージと取得したAIモデルのステータスを表示する
 - エラー発生時は `import_to_console.ipynb` 内の出力にエラー内容を表示し、中断する
 - エラーや応答時間の詳細については、[Cloud SDK Console Access Library\(Python\) 機能仕様書](#) 参照

7. 操作性仕様、画面仕様（AIモデル削除）

前提条件

- Portal for AITRIOSからユーザー登録し、AITRIOSのプロジェクトに参加していること
- AIモデルをConsole for AITRIOSへインポートしていること

How to start each function

1. SDK環境を立ち上げ、Topの **README.md** をプレビュー表示する
2. SDK環境Topの **README.md** に含まれるハイパーリンクから、 **tutorials** ディレクトリの **README.md** にジャンプする
3. **tutorials** ディレクトリの **README.md** に含まれるハイパーリンクから、 **3_prepare_model** ディレクトリの **README.md** にジャンプする
4. **3_prepare_model** ディレクトリの **README.md** に含まれるハイパーリンクから、 **develop_on_sdk** ディレクトリの **README.md** にジャンプする
5. **develop_on_sdk** ディレクトリの **README.md** に含まれるハイパーリンクから、 **delete_model_on_console** ディレクトリの **README.md** にジャンプする
6. **delete_model_on_console** ディレクトリの各ファイルから各機能に遷移する

システムクライアント認証用Notebook実行

1. **delete_model_on_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **set_up_console_client** ディレクトリの **README.md** にジャンプする
2. **set_up_console_client** ディレクトリのシステムクライアント認証用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

AIモデル情報一覧取得用Notebook実行

1. **delete_model_on_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **get_model_list** ディレクトリの **README.md** にジャンプする
2. **get_model_list** ディレクトリのAIモデル情報一覧取得用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

AIモデル削除用Notebook実行向け設定ファイル作成・編集



特別な記載がある場合を除き、原則として省略は不可。



Console Access Library APIに渡されるパラメータについては、[Console Access Library API](#)の仕様に従う。

1. 実行ディレクトリに設定ファイル(`configuration.json`)を作成し、編集する

Configuration	Meaning	Range	Remarks
<code>model_id</code>	削除するAIモデルのID	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ <code>ai_model.ai_model</code> <code>.AIModel.delete_model</code>

AIモデル削除用Notebook実行

1. `delete_model_on_console` の `delete_model_on_console.ipynb` を開き、その中のPythonスクリプトを実行する

○ その後下記の動作をする

- `delete_model_on_console` ディレクトリの[configuration.json](#)の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
- [configuration.json](#)の内容をチェックする
 - エラー発生時はその内容を表示し、中断する
- AIモデル削除APIを実行する
 - 削除に成功した場合は、`delete_model_on_console.ipynb` 内の出力に成功した旨のメッセージを表示する

○ エラー発生時は `delete_model_on_console.ipynb` 内の出力にエラー内容を表示し、中断する

- エラーや応答時間の詳細については、[Cloud SDK Console Access Library\(Python\) 機能仕様書](#) 参照

8. 操作性仕様、画面仕様（PPLインポート）

前提条件

- Portal for AITRIOSからユーザー登録し、AITRIOSのプロジェクトに参加していること
- PPLを用意していること

How to start each function

1. SDK環境を立ち上げ、Topの **README.md** をプレビュー表示する
2. SDK環境Topの **README.md** に含まれるハイパーリンクから、 **tutorials** ディレクトリの **README.md** にジャンプする
3. **tutorials** ディレクトリの **README.md** に含まれるハイパーリンクから、 **4_prepare_application** ディレクトリの **README.md** にジャンプする
4. **4_prepare_application** ディレクトリの **README.md** に含まれるハイパーリンクから、 **2_import_to_console** ディレクトリの **README.md** にジャンプする
5. **2_import_to_console** ディレクトリの各ファイルから各機能に遷移する

インポートするPPLを用意

1. インポート対象のPPLを用意して、任意のディレクトリへ格納する

システムクライアント認証用Notebook実行

1. **2_import_to_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **set_up_console_client** ディレクトリの **README.md** にジャンプする
2. **set_up_console_client** ディレクトリのシステムクライアント認証用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

PPL情報一覧取得用Notebook実行

1. **2_import_to_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **get_application_list** ディレクトリの **README.md** にジャンプする
2. **get_application_list** ディレクトリのPPL情報一覧取得用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

PPLインポート用Notebook実行向け設定ファイル作成・編集



特別な記載がある場合を除き、原則として省略は不可。



原則としてシンボリックリンクのフォルダパス、ファイルパスは使用不可。



Console Access Library APIに渡されるパラメータについては、[Console Access Library API](#)の仕様に従う。

1. 実行ディレクトリに設定ファイル(`configuration.json`)を作成し、編集する

Configuration	Meaning	Range	Remarks
<code>app_name</code>	PPL名	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ <code>deployment.deployment.Deployment.import_device_app</code>
<code>version_number</code>	PPLバージョン	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ <code>deployment.deployment.Deployment.import_device_app</code>
<code>ppl_file</code>	PPLファイルのパス	絶対パスまたは Notebook(*.ipynb)からの相対パス	省略不可

Configuration	Meaning	Range	Remarks
comment	PPL説明	文字列 詳細はConsole Access Library APIの仕様に従う。	省略可 指定なしの場合、コメントなし 下記のConsole Access Library APIに使用される。 ・ deployment.deploy ment.Deployment.im port_device_app

PPLインポート用Notebook実行

1. **2_import_to_console** ディレクトリの **import_to_console.ipynb** を開き、その中のPythonスクリプトを実行する
 - その後下記の動作をする
 - **2_import_to_console** ディレクトリの[configuration.json](#)の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
 - [configuration.json](#)の内容をチェックする
 - エラー発生時はその内容を表示し、中断する
 - PPLをBase64形式でエンコードする
 - エラー発生時はその内容を表示し、中断する
 - PPLインポートAPIを実行する
 - インポートに成功した場合は、**import_to_console.ipynb** 内の出力に成功した旨のメッセージを表示する
 - PPLインポート結果確認APIを実行する
 - PPLの情報取得に成功した場合は、**import_to_console.ipynb** 内の出力に成功した旨のメッセージと取得したPPLのステータスを表示する
 - エラー発生時は **import_to_console.ipynb** 内の出力にエラー内容を表示し、中断する
 - エラーや応答時間の詳細については、[Cloud SDK Console Access Library\(Python\) 機能仕様書](#) 参照

9. 操作性仕様、画面仕様（PPL削除）

前提条件

- Portal for AITRIOSからユーザー登録し、AITRIOSのプロジェクトに参加していること
- PPLをConsole for AITRIOSへインポートしていること

How to start each function

1. SDK環境を立ち上げ、Topの **README.md** をプレビュー表示する
2. SDK環境Topの **README.md** に含まれるハイパーリンクから、 **tutorials** ディレクトリの **README.md** にジャンプする
3. **tutorials** ディレクトリの **README.md** に含まれるハイパーリンクから、 **4_prepare_application** ディレクトリの **README.md** にジャンプする
4. **4_prepare_application** ディレクトリの **README.md** に含まれるハイパーリンクから、 **delete_application_on_console** ディレクトリの **README.md** にジャンプする
5. **delete_application_on_console** ディレクトリの各ファイルから各機能に遷移する

システムクライアント認証用Notebook実行

1. **delete_application_on_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **set_up_console_client** ディレクトリの **README.md** にジャンプする
2. **set_up_console_client** ディレクトリのシステムクライアント認証用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

PPL情報一覧取得用Notebook実行

1. **delete_application_on_console** ディレクトリの **README.md** に含まれるハイパーリンクから、 **get_application_list** ディレクトリの **README.md** にジャンプする
2. **get_application_list** ディレクトリのPPL情報一覧取得用Notebook(*.ipynb)を開き、その中のPythonスクリプトを実行する

PPL削除用Notebook実行向け設定ファイル作成・編集



特別な記載がある場合を除き、原則として省略は不可。



Console Access Library APIに渡されるパラメータについては、[Console Access Library API](#)の仕様に従う。

1. 実行ディレクトリに設定ファイル(**configuration.json**)を作成し、編集する

Configuration	Meaning	Range	Remarks
app_name	PPL名	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ deployment.deployment.delete_device_app
version_number	PPLバージョン	文字列 詳細はConsole Access Library APIの仕様に従う。	省略不可 下記のConsole Access Library APIに使用される。 ・ deployment.deployment.delete_device_app

PPL削除用Notebook実行

1. `delete_application_on_console` ディレクトリの `delete_application_on_console.ipynb` を開き、その中のPythonスクリプトを実行する
 - その後下記の動作をする
 - `delete_application_on_console` ディレクトリの `configuration.json` の存在をチェックする
 - エラー発生時はその内容を表示し、中断する
 - `configuration.json` の内容をチェックする
 - エラー発生時はその内容を表示し、中断する
 - PPL削除APIを実行する
 - 削除に成功した場合は、`delete_application_on_console.ipynb` 内の出力に成功した旨のメッセージを表示する
 - エラー発生時は `delete_application_on_console.ipynb` 内の出力にエラー内容を表示し、中断する
 - エラーや応答時間の詳細については、 [Cloud SDK Console Access Library\(Python\) 機能仕様書](#) 参照

10. 目標性能

- ユーザビリティ
 - SDKの環境構築完了後、追加のインストール手順なしに、AIモデル、PPLのConsole for AITRIOSへのインポートができること
- UIの応答時間が1.2秒以内であること
- 処理に5秒以上かかる場合は、処理中の表現を逐次更新表示できること

11. 制限事項

- エンコードやインポート処理を途中でキャンセルして再開する場合、途中からの再開ではなく各処理を最初から実行する

12. その他特記事項

- なし

13. 未決定事項

- なし