

1. 制御API

QA ENGINE / アンサーロボ制御APIは、QA ENGINE / アンサーロボ管理画面で行えるFAQデータの管理、質問データの管理/アノテーション、機械学習の実行、機械学習モデルのステージングAPIから本番APIへのコピーなどの主要な操作をWebAPIから実行可能とするものです。

本ドキュメントではQA ENGINE / アンサーロボ制御APIの技術仕様について記述します。

※QA ENGINEでのご契約か、アンサーロボでのご契約かによって呼び出せるAPIが異なっております。詳しくは各APIで呼び出し可否について記述しております。

1.1 制御API エンドポイント

QA ENGINE / アンサーロボの制御APIエンドポイントはQA ENGINE / アンサーロボの管理画面

```
https://admin.qaengine.ai/
```

をベースのURLとして記載しています。たとえばエンドポイント `/capi/faq/list` という表記があった場合、
`https://admin.qaengine.ai/capi/faq/list` と解釈ください。

1.2 制御API APIキーと権限

APIキーはQA ENGINE / アンサーロボ管理画面で、管理権限のあるユーザのみが作成できます。

APIキーは、APIキーごとに有効/無効の設定と、実行可能なAPIを制限する権限設定が行えます。

APIキーの発行方法はQA ENGINEヘルプページでご確認ください。

各APIで、どの権限が実行に必要なのかは各APIのセクションで説明しています。

1.3 APIキー認証のためのHTTPヘッダ

QA ENGINE / アンサーロボ制御APIでは管理画面で発行されたAPIキーを以下のように `X-API-Key` HTTPヘッダにふくめてHTTPリクエストを行うことで認証します。

例えば `dnzFg4mUZlhZi6ZWUFQr0UGUxsUh8i90ZpQyrCbb` というAPIキーでWebAPIリクエストの認証を行いたい場合、リクエスト時に以下のHTTPヘッダを含めるようにしてください。

```
X-API-Key: dnzFg4mUZlhZi6ZWUFQr0UGUxsUh8i90ZpQyrCbb
```

1.4 APIキー認証に失敗したときの共通レスポンス

APIキーでの認証に失敗する場合、以下のパターンがあります。

- キー未指定エラー: APIキーが `X-API-Key` ヘッダにセットされていない場合
- キー無効エラー: `X-API-Key` ヘッダに指定されたAPIキーが存在しない場合、または設定で有効にされていない場合
- キー権限エラー: `X-API-Key` ヘッダに指定されたAPIキーが実行しようとしているAPIを実行する権限が設定により与えられていない場合

いずれの場合もHTTPステータスコード `403` を返します。以下に各エラーパターンでのレスポンス本文を示します。「1.6 レスポンスデータの記述について」セクションも参照ください。

キー未指定エラー

```
{ "status": "error", "code": "key_missing", "message": "missing api key" }
```

キー無効エラー

```
{ "status": "error", "code": "key_invalid", "message": "invalid api key" }
```

キー権限エラー

```
{ "status": "error", "code": "key_no_priv", "message": "priviledge error" }
```

1.5 パラメータ

QA ENGINE / アンサーロボ制御API全般に対するパラメータは `application/x-www-form-urlencoded` 形式でエンコードし、リクエストしてください。

HTTPリクエストのHTTPヘッダに

```
Content-Type: application/x-www-form-urlencoded
```

を含めるようにしてください。

1.6 レスポンスデータの記述について

各WebAPIのレスポンス本文(body)については特に記載がない場合JSONオブジェクトを返却するものとします。レスポンスのJSONデータの例の記述に際してはわかりやすいようにスペースとインデントを加えて記載しておりますが実際にWebAPIから返却されるレスポンス本文はスペースやインデントのない形式となります。

エラー時は以下のように共通のフォーマットでレスポンスが返却されます。エラーのパターンを判断する際は `code` の値をもとに判断するようにしてください。 `message` の値はヒューマンリーダブルな値をセットすることを想定しており、わかりやすさのために変更されることがあることに注意してください。

```
{  
  "status": "error",  
  "code": "(エラーコード)",  
  "message": "(エラーについての説明)"  
}
```

エラーコード一覧については (付録.A) エラーコード一覧 も合わせて参照ください。

1.7 予期せぬエラー

全てのAPIは内部で異常が発生した場合以下のようなエラーを返すことがあります。(HTTPステータスコード500)

このエラーが発生した場合はStudio Ousiaまでご連絡ください。

```
{  
  "status": "error",  
  "code": "internal_server_error",  
  "message": "internal server error"  
}
```

1.8 契約上の問題エラー

一部のAPIはQA ENGINEでの契約だけでのみ呼び出せる、または逆にアンサーロボの契約だけでのみ呼び出せるというタイプのものがあります。(呼び出そうとしているAPIがこの種類のものかどうかは各APIの詳細説明で「呼び出し可能な契約」の項目を確認してください。)

例:

- QA ENGINEでの契約だけでのみ呼び出せるAPIの例: 学習開始API
- アンサーロボでの契約だけでのみ呼び出せるAPIの例: FAQ反映開始API

このようなAPIに対し、違うタイプの契約でご利用いただいているアプリケーションにひもづいた制御APIキーでAPI呼び出しを行おうとした場合、以下のようなエラーがHTTPステータスコード403で返却されます。

また、QA ENGINE本契約前に仮アカウントを発行させていただいて、仮アカウントで作られた制御APIで呼び出した場合、以下のAPIの呼び出しでも同様のエラーが返却されます。

- 学習開始API
- 本番コピーAPI
- ステージングAPI情報取得API
- 本番API情報取得API

```
{
  "status": "error",
  "code": "operation_prohibited",
  "message": "operation prohibited"
}
```

2. FAQデータ操作API

2.1 FAQリストAPI

FAQリストAPIはリクエストに使用したAPIキーが紐づけられているアプリケーションの全てのFAQのリストを取得するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/list
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- FAQに関する読み取り操作を許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

FAQリストAPIの成功時は1つの巨大なJSONを返却するのではなく、1行に1つのJSONオブジェクトを表現したJSON Lines形式で返却します。

1行ごとのJSONオブジェクトは以下のような形式になります。(実際は改行/スペースを含まず1オブジェクトあたり1行で表現されます)

`is_active` 属性はboolean型で、その他の属性は文字列型で返却されます。

```
{
  "identifier": "(FAQ ID文字列)",
  "title": "(FAQタイトル文字列)",
  "answer": "(FAQ回答本文文字列)",
  "is_active": (FAQ状態true(アクティブ)またはfalse(非アクティブ)),
  "created_at": "2020-01-20T12:34:56のようなFAQが作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
  "updated_at": "2020-01-21T22:34:56のようなFAQが更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
  "tags": [
    "(関連づけられたタグ1)",
    "(関連づけられたタグ2)",
    ...
  ],
  "faq_keywords": [
    "(関連付けられた優先キーワード1)",
    "(関連付けられた優先キーワード2)",
    ...
  ]
}
```

そのため、実際のレスポンスは以下のような形になります。

```
{ "identifier": "...", "title": "...", "answer": "...", "is_active": true,
  "created_at": "...", "updated_at": "..." }
{ "identifier": "...", "title": "...", "answer": "...", "is_active": true,
  "created_at": "...", "updated_at": "..." }
{ "identifier": "...", "title": "...", "answer": "...", "is_active": true,
  "created_at": "...", "updated_at": "..." }
... (FAQ数分だけ続く)
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

2.2 FAQ取得API

FAQ取得APIはFAQ IDで指定した単一のFAQの詳細情報を取得するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/get
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- FAQに関する読み取り操作を許可

必須パラメータ

- `identifier` : 詳細情報を取得したいFAQのFAQ IDを指定します。

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "faq": {
      "identifier": "(FAQ ID文字列, identifierパラメーターで指定したもの)",
      "title": "(FAQタイトル文字列)",
      "answer": "(FAQ回答本文文字列)",
      "is_active": "(FAQ状態true(アクティブ)またはfalse(非アクティブ))",
      "created_at": "2020-01-20T12:34:56のようなFAQが作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
      "updated_at": "2020-01-21T22:34:56のようなFAQが更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
      "tags": [
        "(関連づけられたタグ1)",
        "(関連づけられたタグ2)",
        ...
      ],
      "faq_keywords": [
        "(関連付けられた優先キーワード1)",
        "(関連付けられた優先キーワード2)",
        ...
      ]
    }
  }
}
```

レスポンス: ステータスコード 400 (パラメータidentifierが未指定または空文字列が指定された場合)

```
{
  "status": "error",
  "code": "faq_invalid_identifier",
  "message": "invalid faq identifier"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定したFAQ IDを持つFAQが存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "faq not found"
}
```

2.3 FAQ追加API

FAQ追加APIはリクエストに使用したAPIキーが紐づけられているアプリケーションに新しいFAQを追加するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/add
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- FAQに関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` : 文字列で新しく作成するFAQのFAQ IDを指定します。日本語も指定可能です。最大128文字で指定してください。

省略可能パラメータ

- `title` : 文字列で新しく作成するFAQのタイトルを指定します。最大255文字で指定してください。省略時は空のタイトルが設定されます。
- `answer` : 文字列で新しく作成するFAQの回答本文を指定します。最大15000文字で指定してください。省略時は空の回答本文が設定されます。
- `is_active` : 文字列 `true` または `false` で新しく作成するFAQのアクティブかどうかを指定します。省略時は `true` (アクティブ)として設定されます。
- `tags` : FAQに関連付ける複数のタグを、半角スペース区切りの文字列で指定します。最大20個のタグを指定できます。
- `faq_keywords` : (アンサーロボのみ)FAQに関連付ける複数の優先キーワードを、半角セミコロン(;)区切りの文字列で指定します。最大20個の優先キーワードを指定できます。

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "faq": {
      "identifier": "(作成されたFAQのFAQ ID文字列, identifierパラメーターで指定したものの)",
      "title": "(作成されたFAQのFAQタイトル文字列)",

```

```
    "answer": "(作成されたFAQのFAQ回答本文文字列)",
    "is_active": (作成されたFAQのFAQ状態true(アクティブ)またはfalse(非アクティブ)),
    "created_at": "2020-01-20T12:34:56のようなISO-8601形式の作成されたFAQの作成時刻
タイムスタンプ文字列(日本標準時)",
    "updated_at": "2020-01-21T22:34:56のようなISO-8601形式の作成されたFAQの更新時刻
タイムスタンプ文字列(日本標準時)",
    "tags": [
        "(設定されたタグ)",
        ...
    ],
    "faq_keywords": [
        "(設定された優先キーワード)",
        ...
    ]
}
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}
```

レスポンス: ステータスコード 400 (パラメータidentifierで指定したFAQ IDがすでに使われていた場合)

```
{
  "status": "error",
  "code": "faq_identifier_taken",
  "message": "identifier already taken"
}
```

レスポンス: ステータスコード 400 (パラメータtagsで指定したタグが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "too many faq tags"
}
```

レスポンス: ステータスコード 400 (パラメータfaq_keywordsで指定した優先キーワードが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
}
```

```
"message": "too many faq keywords"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

2.4 FAQ更新API

FAQ更新APIはすでに存在するFAQの内容を更新するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/update
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- FAQに関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` : 更新するFAQのFAQ IDを指定します。

オプションパラメータ

- `title` : 文字列で新しく設定するタイトルを指定します。最大255文字で指定してください。
- `answer` : 文字列で新しく設定する回答本文を指定します。最大15000文字で指定してください。
- `is_active` : 文字列 `true` または `false` で新しく設定するアクティブかどうかを指定します。
- `tags` : FAQに関連付ける複数のタグを、半角スペース区切りの文字列で指定します。最大20個のタグを指定できます。
- `faq_keywords` : (アンサーロボのみ)FAQに関連付ける複数の優先キーワードを、半角セミコロン(;)区切りの文字列で指定します。最大20個の優先キーワードを指定できます。

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "faq": {
      "identifier": "(更新されたFAQのFAQ ID文字列, identifierパラメーターで指定したものの)",
      "title": "(更新されたFAQのFAQタイトル文字列)",
      "answer": "(更新されたFAQのFAQ回答本文文字列)",
      "is_active": (更新されたFAQのFAQ状態true(アクティブ)またはfalse(非アクティブ)),
      "created_at": "2020-01-20T12:34:56のようなISO-8601形式の更新されたFAQの作成時刻タイムスタンプ文字列(日本標準時)",
      "updated_at": "2020-01-21T22:34:56のようなISO-8601形式の更新されたFAQの更新時刻タイムスタンプ文字列(日本標準時)",
      "tags": [
```

```
        "(設定されたタグ)",
        ...
    ],
    "faq_keywords": [
        "(設定された優先キーワード)",
        ...
    ]
}
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}
```

レスポンス: ステータスコード 400 (パラメータtagsで指定したタグが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "too many faq tags"
}
```

レスポンス: ステータスコード 400 (パラメータfaq_keywordsで指定した優先キーワードが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "too many faq keywords"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定したFAQ IDを持つFAQが存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "faq not found"
}
```

2.5 FAQ upsert API

FAQ upsert APIは `identifier` パラメータで指定されたFAQ IDを持つFAQがすでに存在すればそのFAQの内容を更新、存在していなければ新しいFAQを作成するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/upsert
```

対応HTTPメソッド

- `POST`

実行に必要なAPI権限

- `FAQ`に関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` : upsert(作成/更新)するFAQのFAQ IDを指定します。

省略可能パラメータ

- `title` : 文字列でupsert(作成/更新)するFAQのタイトルを指定します。最大255文字で指定してください。
- `answer` : 文字列でupsert(作成/更新)する回答本文を指定します。最大15000文字で指定してください。
- `is_active` : 文字列 `true` または `false` でupsert(作成/更新)するFAQのアクティブかどうかを指定します。
- `tags` : FAQに関連付ける複数のタグを、半角スペース区切りの文字列で指定します。最大20個のタグを指定できます。
- `faq_keywords` : (アンサーロボのみ)FAQに関連付ける複数の優先キーワードを、半角セミコロン(;)区切りの文字列で指定します。最大20個の優先キーワードを指定できます。

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "performed": "(作成モードで実行された場合, 'insert', 更新モードで実行された場合 'update')",
    "faq": {
      "identifier": "(upsert (作成/更新) されたFAQのFAQ ID文字列, identifierパラメータで指定したもの)",
      "title": "(upsert (作成/更新) されたFAQのFAQタイトル文字列)",
      "answer": "(upsert (作成/更新) されたFAQのFAQ回答本文文字列)",
      "is_active": (upsert (作成/更新) されたFAQのFAQ状態true (アクティブ) またはfalse (非アクティブ)),
      "created_at": "2020-01-20T12:34:56のようなISO-8601形式のupsert (作成/更新) されたFAQの作成時刻タイムスタンプ文字列 (日本標準時)",
      "updated_at": "2020-01-21T22:34:56のようなISO-8601形式のupsert (作成/更新) されたFAQの更新時刻タイムスタンプ文字列 (日本標準時)",
      "tags": [
        "(設定されたタグ)",
        ...
      ],
      "faq_keywords": [
```

```
      "(設定された優先キーワード)",
      ...
    ]
  }
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}
```

レスポンス: ステータスコード 400 (パラメータtagsで指定したタグが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "too many faq tags"
}
```

レスポンス: ステータスコード 400 (パラメータfaq_keywordsで指定した優先キーワードが最大指定可能数より多い場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "too many faq keywords"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

2.6 FAQ削除API

FAQ削除APIはFAQを削除するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/faq/delete
```

対応HTTPメソッド

- DELETE

実行に必要なAPI権限

- FAQに関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` : 削除するFAQのFAQ IDを指定します。

オプションパラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "deleted_faq": {
      "identifier": "(削除されたFAQのFAQ ID文字列, identifierパラメーターで指定したものの)",
      "title": "(削除されたFAQのFAQタイトル文字列)",
      "answer": "(削除されたFAQのFAQ回答本文文字列)",
      "is_active": "(削除されたFAQのFAQ状態true(アクティブ)またはfalse(非アクティブ))",
      "created_at": "2020-01-20T12:34:56のようなISO-8601形式の削除されたFAQの作成時刻タイムスタンプ文字列(日本標準時)",
      "updated_at": "2020-01-21T22:34:56のようなISO-8601形式の削除されたFAQの更新時刻タイムスタンプ文字列(日本標準時)"
    }
  }
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定したFAQ IDを持つFAQが存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "faq not found"
}
```

3. 質問データ操作API

3.1 質問リストAPI

質問リストAPIはリクエストに使用したAPIキーが紐づけられているアプリケーションの全ての質問のリストを取得するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/list
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- 質問に関する読み取り操作を許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

質問リストAPIの成功時は1つの巨大なJSONを返却するのではなく、1行に1つのJSONオブジェクトを表現したJSON Lines形式で返却します。

1行ごとのJSONオブジェクトは以下のような形式になります。(実際は改行/スペースを含まず1オブジェクトあたり1行で表現されます)

```
{
  "identifier": "(文字列型, 質問ID)",
  "content": "(文字列型, 質問本文)",
  "is_active": "(boolean型, 機械学習に使用するかどうか)",
  "is_from_query": "(boolean型, 質問がクエリインポートで読み込まれたものであるかどうか)",
  "query_uuid": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列)",
  "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
  "answered_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア)",
  "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
  "top2_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア)",
  "top3_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
  "top3_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア)",
```

```
"top4_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
"top4_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),

"top5_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
"top5_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),

"is_from_console": "(boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),

"faq_id": "(文字列型, アノテーションされていなければnull, そうでなければアノテーションされたFAQのFAQ ID)",
"last_annotated_user": "(文字列型, アノテーションされていなければnull, そうでなければ最後にアノテーションしたユーザのemailアドレス)",

"created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
"updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",

}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

3.2 質問取得API

質問取得APIは質問IDで指定した単一の質問の詳細情報を取得するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/get
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- 質問に関する読み取り操作を許可

必須パラメータ

- `identifier` : 詳細情報を取得したい質問の質問IDを指定します。

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

question 属性に identifier で指定した質問IDを持つ質問の内容がセットされます。

クエリインポートから読み込まれたものだけにセットされる属性があることにご注意ください。

```
{
  "status": "ok",
  "result": {
    "question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",
      "is_active": (boolean型, 機械学習に使用するかどうか),
      "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
      "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列),
      "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
      "answered_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
      "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
      "top2_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),
      "top3_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
      "top3_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),
      "top4_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
      "top4_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),
      "top5_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
      "top5_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),
      "is_from_console": (boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),
      "faq_id": "(文字列型, アノテーションされていない場合null, そうでなければアノテーションされたFAQのFAQ ID)",
      "last_annotated_user": "(文字列型, アノテーションされていない場合null, そうでなければ最後にアノテーションしたユーザのemailアドレス)",
      "created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
      "updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
```

```
}  
}  
}
```

レスポンス: ステータスコード 400 (identifierパラメータが指定されていないまたは空文字列が指定された場合)

```
{  
  "status": "error",  
  "code": "question_invalid_identifier",  
  "message": "invalid question identifier"  
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定した質問IDを持つ質問が存在しないとき)

```
{  
  "status": "error",  
  "code": "not_found",  
  "message": "question not found"  
}
```

3.3 質問追加API

質問追加APIはリクエストに使用したAPIキーが紐づけられているアプリケーションに新しい質問を追加するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/add
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- 質問に関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` 新しく追加する質問の質問IDを文字列で指定(最大128文字)
- `content` 質問本文を文字列で指定(最大15000文字)

省略可能パラメータ

- `faq_id` この質問に対する正解のFAQをFAQ ID(文字列)で指定
- `is_active` 機械学習に使用するかどうかを `true` (使用) / `false` (使用しない)で指定

レスポンス: ステータスコード 200 (成功時)

`question` 属性に追加された質問の内容が返却されます。

```
{
  "status": "ok",
  "result": {
    "question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",
      "is_active": (boolean型, 機械学習に使用するかどうか),
      "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
      "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列),
      "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
      "answered_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
      "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
      "top2_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),
      "top3_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
      "top3_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),
      "top4_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
      "top4_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),
      "top5_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
      "top5_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),
      "is_from_console": (boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),
      "faq_id": "(文字列型, アノテーションされていない場合null, そうでなければアノテーションされたFAQのFAQ ID)",
      "last_annotated_user": "(文字列型, アノテーションされていない場合null, そうでなければ最後にアノテーションしたユーザのemailアドレス)",
      "created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
      "updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
```

```
}  
}  
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列が指定された場合)

```
{  
  "status": "error",  
  "code": "lack_parameter",  
  "message": "parameter required: identifier"  
}
```

レスポンス: ステータスコード 400 (identifierパラメータで指定した質問IDを持つ質問がすでに存在していた場合)

```
{  
  "status": "error",  
  "code": "question_identifier_taken",  
  "message": "identifier already taken"  
}
```

レスポンス: ステータスコード 400 (contentパラメータが未指定または空文字列が指定された場合)

```
{  
  "status": "error",  
  "code": "lack_parameter",  
  "message": "parameter required: content"  
}
```

レスポンス: ステータスコード 400 (faq_idパラメータで指定したFAQ IDをもつFAQが存在しない場合)

```
{  
  "status": "error",  
  "code": "question_invalid_faq_identifier",  
  "message": "invalid faq identifier"  
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

3.4 質問更新API

質問更新APIはすでに存在する質問の内容を更新するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/update
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- 質問に関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` 内容を更新する質問の質問IDを文字列で指定

省略可能パラメータ

- `content` 質問本文を文字列で指定(最大15000文字)
- `faq_id` この質問に対する正解のFAQをFAQ ID(文字列)で指定
- `is_active` 機械学習に使用するかどうかを `true` (使用) / `false` (使用しない)で指定

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",
      "is_active": (boolean型, 機械学習に使用するかどうか),
      "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
      "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列),
      "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
      "answered_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
      "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
      "top2_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),
      "top3_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
      "top3_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),
      "top4_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
      "top4_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),
      "top5_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
      "top5_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),
```

```
"is_from_console": (boolean型, モデル動作確認画面から読み込まれた質問であればtrue,
そうでなければfalse),

"faq_id": "(文字列型, アノテーションされていなければnull, そうでなければアノテーショ
ンされたFAQのFAQ ID)",
"last_annotated_user": "(文字列型, アノテーションされていなければnull, そうでなけ
れば最後にアノテーションしたユーザのemailアドレス)",

"created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムス
タンプ文字列(日本標準時)",
"updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムス
タンプ文字列(日本標準時)",

}

}

}
```

レスポンス: ステータスコード 400 (identifierパラメータが指定されていないまたは空文字列が指定された場合)

```
{
  "status": "error",
  "code": "question_invalid_identifier",
  "message": "invalid question identifier"
}
```

レスポンス: ステータスコード 400 (faq_idパラメータで指定したFAQ IDをもつFAQが存在しない場合)

```
{
  "status": "error",
  "code": "question_invalid_faq_identifier",
  "message": "invalid faq identifier"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定した質問IDを持つ質問が存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "question not found"
}
```

3.5 質問upsert API

質問upsert APIは `identifier` パラメータで指定された質問IDを持つ質問がすでに存在すればその質問の内容を更新 (updateモード)、存在していなければ新しい質問を作成する(insertモード)APIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/upsert
```

対応HTTPメソッド

- `POST`

実行に必要なAPI権限

- 質問に関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- (共通) `identifier` 追加/更新する質問の質問IDを文字列で指定(最大128文字)
- (insertモード / `identifier`に指定した質問IDが存在しない場合) `content` 質問本文を文字列で指定(最大15000文字)

省略可能パラメータ

- (updateモード / `identifier`に指定した質問IDがすでに存在する場合) `content` 質問本文を文字列で指定(最大15000文字)
- `faq_id` この質問に対する正解のFAQをFAQ ID(文字列)で指定
- `is_active` 機械学習に使用するかどうかを `true` (使用) / `false` (使用しない)で指定

レスポンス: ステータスコード 200 (成功時)

`performed` 属性に行われた操作(追加なのか、更新なのか)が、`question` 属性に操作が行われた質問の内容が含まれます。

```
{
  "status": "ok"
  "result": {
    "performed": "(文字列, insertモードなら'insert', updateモードなら'update')",
    "question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",
      "is_active": (boolean型, 機械学習に使用するかどうか),
      "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
      "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列),
      "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
      "answered_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
      "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
      "top2_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),
```

```

    "top3_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
    "top3_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),

    "top4_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
    "top4_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),

    "top5_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
    "top5_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),

    "is_from_console": "(boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),

    "faq_id": "(文字列型, アノテーションされていなければnull, そうでなければアノテーションされたFAQのFAQ ID)",
    "last_annotated_user": "(文字列型, アノテーションされていなければnull, そうでなければ最後にアノテーションしたユーザのemailアドレス)",

    "created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
    "updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",

    }

}

}

```

レスポンス: ステータスコード 400 (identifierパラメータが指定されていないまたは空文字列が指定された場合)

```

{
  "status": "error",
  "code": "question_invalid_identifier",
  "message": "invalid question identifier"
}

```

レスポンス: ステータスコード 400 (faq_idパラメータで指定したFAQ IDをもつFAQが存在しない場合)

```

{
  "status": "error",
  "code": "question_invalid_faq_identifier",
  "message": "invalid faq identifier"
}

```

レスポンス: ステータスコード 400 (新規追加モードの時, contentパラメータが未指定または空文字列が指定された場合)

`identifier` パラメータで指定した質問IDを持つ質問が存在していない(新規追加モードで動作)場合のみ、`content` パラメータが必須となります。

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: content"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

3.6 質問削除API

質問削除APIは指定した質問を削除するAPIです。

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/delete
```

対応HTTPメソッド

- DELETE

実行に必要なAPI権限

- 質問に関する書き込み操作 (追加/更新/削除) を許可

必須パラメータ

- `identifier` : 削除したい質問の質問IDを指定します。

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

`deleted_question` 属性に削除された質問の情報がセットされます。

```
{
  "status": "ok",
  "result": {
    "deleted_question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",
      "is_active": (boolean型, 機械学習に使用するかどうか),
      "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
      "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリの
```

```

UUID文字列),
  "answered_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
  "answered_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
  "top2_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
  "top2_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),

  "top3_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
  "top3_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),

  "top4_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
  "top4_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),

  "top5_faq_identifier": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
  "top5_faq_score": "(数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),

  "is_from_console": "(boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),

  "faq_id": "(文字列型, アノテーションされていなければnull, そうでなければアノテーションされたFAQのFAQ ID)",
  "last_annotated_user": "(文字列型, アノテーションされていなければnull, そうでなければ最後にアノテーションしたユーザーのemailアドレス)",

  "created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
  "updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",

}

}

}

```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```

{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}

```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定した質問IDを持つ質問が存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "question not found"
}
```

3.7 質問アノテーションAPI

質問アノテーションAPIはすでに存在する質問に対して正解のFAQをFAQ IDで指定してひもづけを行う(アノテーションする)、またはひもづけを解除するAPIです。

※アンサーロボではFAQ反映で起動された自動応答APIの回答精度にアノテーションの有無は影響しませんがエラー解析画面などで活用いただけます

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/question/annotate
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- 質問に関するアノテーション操作を許可

必須パラメータ

- `identifier` : 詳細情報を取得したい質問の質問IDを指定します。

省略可能パラメータ

- `unannotate` : (`faq_id` といずれか必須) アノテーションを解除する場合文字列 `true` を指定
- `faq_id` : (`unannotate` といずれか必須) アノテーション(FAQの紐付け)を行う場合、該当のFAQ IDを文字列で指定

レスポンス: ステータスコード 200 (成功時)

`question` 属性に変更された質問の情報がセットされます。

```
{
  "status": "ok",
  "result": {
    "question": {
      "identifier": "(文字列型, 質問ID)",
      "content": "(文字列型, 質問本文)",

```

```

    "is_active": (boolean型, 機械学習に使用するかどうか),
    "is_from_query": (boolean型, 質問がクエリインポートで読み込まれたものであるかどうか),
    "query_uuid": (文字列型, is_from_queryがfalseの場合null, そうでない場合クエリのUUID文字列),
    "answered_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのFAQ ID)",
    "answered_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが一番大きかった)FAQのスコア),
    "top2_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのFAQ ID)",
    "top2_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP2で回答した(スコアが2番目に大きかった)FAQのスコア),

    "top3_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのFAQ ID)",
    "top3_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが3番目に大きかった)FAQのスコア),

    "top4_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのFAQ ID)",
    "top4_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが4番目に大きかった)FAQのスコア),

    "top5_faq_identifiser": "(文字列型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのFAQ ID)",
    "top5_faq_score": (数値型, is_from_queryがfalseの場合null, そうでない場合クエリに対しTOP1で回答した(スコアが5番目に大きかった)FAQのスコア),

    "is_from_console": (boolean型, モデル動作確認画面から読み込まれた質問であればtrue, そうでなければfalse),

    "faq_id": "(文字列型, アノテーションされていなければnull, そうでなければアノテーションされたFAQのFAQ ID)",
    "last_annotated_user": "(文字列型, アノテーションされていなければnull, そうでなければ最後にアノテーションしたユーザのemailアドレス)",

    "created_at": "2020-01-20T12:34:56のような質問が作成されたISO-8601形式のタイムスタンプ文字列(日本標準時)",
    "updated_at": "2020-01-21T22:34:56のような質問が更新されたISO-8601形式のタイムスタンプ文字列(日本標準時)",

    }

}

}

```

レスポンス: ステータスコード 400 (unannotateパラメータ不正)

unannotate パラメータが指定され、値が true 以外だった場合

```
{
  "status": "error",
  "code": "question_invalid_unannotate",
  "message": "invalid unannotate value"
}
```

レスポンス: ステータスコード 400 (identifierパラメータが未指定または空文字列で指定された場合)

```
{
  "status": "error",
  "code": "lack_parameter",
  "message": "parameter required: identifier"
}
```

レスポンス: ステータスコード 400 (faq_idパラメータで指定したFAQ IDをもつFAQが存在しない場合)

```
{
  "status": "error",
  "code": "question_invalid_faq_identifier",
  "message": "invalid faq identifier"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (identifierパラメータで指定した質問IDを持つ質問が存在しないとき)

```
{
  "status": "error",
  "code": "not_found",
  "message": "question not found"
}
```

4. オペレーション系API

オペレーション系APIは機械学習やエンドポイントの起動など、実行に時間がかかる非同期タスクの実行(およびその進捗確認)など、単純なデータの出し入れ以外の操作を行うAPIです。

QA ENGINEでのご契約では機械学習の実行(ステージングAPIの起動)や、ステージングAPIから本番APIのモデルのコピー(本番コピー)の実行、本番APIへ送信され記録されていたクエリをQA ENGINEアナテーターの質問データベースにインポートするクエリインポートの実行、それらの非同期タスクのタスク進行状況の確認、起動したステージングAPI/本番APIのエンドポイント情報の確認ができます。

アンサーロボでのご契約では機械学習の実行・本番コピーは呼び出すことはできません。その代わりにFAQ反映タスクの実行、アンサーロボAPI情報の確認などが行えます。進行状況確認APIはQA ENGINEのものと共有になります。

4.1 学習開始API

学習開始APIは呼び出すと学習を開始します。学習は実行に時間のかかる非同期タスクのため、実際に学習が完了したかどうかは返却されるタスクIDを用いて [4.4 タスク確認API](#) に問い合わせ確認してください。学習が完了するとステージングAPIに新しいモデルが自動的に反映されます。一度も学習が行われておらず、ステージングAPIが起動していなかった場合は自動的にステージングAPIが起動します。

注意点:

- すでに実行中の学習タスクがある場合はエラーとなり学習は開始されません。
- APIキーに紐づけられたアプリケーションに保存されているFAQ/質問が少なすぎる場合学習できません。
 - 有効となっているFAQ数が2未満の場合学習できません。
 - 有効なFAQが正解としてアノテーションされており、かつ機械学習に利用する設定となっている質問が10未満の場合学習できません。

呼び出し可能な契約: QA ENGINE

エンドポイント

```
/capi/op/stage
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- [学習タスクの実行開始を許可](#)

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

`task_id` 属性で返却されるタスクIDを用いて [4.4 タスク確認API](#) に問い合わせることでタスクが完了したかどうかを確認することができます。詳細は [4.4 タスク確認API](#) の説明を参照ください。

```
{
  "status": "ok",
  "result": {
    "task_id": "(タスクID)"
  }
}
```

レスポンス: ステータスコード 400 (有効FAQ数が2未満)

```
{
  "status": "error",
  "code": "operation_stage_data_error_n_faq",
  "message": "too small faq number"
}
```

レスポンス: ステータスコード 400 (有効質問数が10未満)

```
{
  "status": "error",
  "code": "operation_stage_data_error_n_question",
  "message": "too small question number"
}
```

レスポンス: ステータスコード 403 (契約上の問題)

「1.8 契約上の問題」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.2 本番コピーAPI

本番コピー開始APIは呼び出すと本番APIに、ステージングAPIのモデルをコピーする非同期タスクを実行開始します。本番コピータスクは実行に時間のかかる非同期タスクのため、実際に本番コピーが完了したかどうかは返却されるタスクIDを用いて 4.4 タスク確認API に問い合わせ確認してください。本番コピーが一度も実行されておらず、本番APIが起動していなかった場合自動的に本番APIが起動します。

注意点:

- すでに実行中の本番コピータスクがある場合はエラーとなり本番コピーは開始されません。
- ステージングAPIが存在しない(一度も学習が行われていない)場合はコピー元モデルが存在しないためエラーとなり本番コピーは開始されません。

呼び出し可能な契約: QA ENGINE

エンドポイント

```
/capi/op/prod
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- 本番コピータスクの実行開始を許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

task_id 属性で返却されるタスクIDを用いて 4.4 タスク確認API に問い合わせることでタスクが完了したかどうかを確認することができます。詳細は 4.4 タスク確認API の説明を参照ください。

```
{
  "status": "ok",
```

```
"result": {
  "task_id": "(タスクID)"
}
```

レスポンス: ステータスコード 403 (契約上の問題)

「1.8 契約上の問題」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.3 クエリインポート開始API

クエリインポート開始APIは本番APIにクエリされ、QA ENGINEのクエリデータベースに保存されているクエリをQA ENGINEの「質問」一覧に取り込む「クエリインポート」を実行開始するAPIです。取り込みたいクエリの時系列範囲を指定しないモードと、指定するモードの両方をご利用いただけます。時系列範囲を指定しない場合、最後に実行したクエリインポートの読み込み範囲の終了点(過去に一度もクエリインポートが行われていない場合、もっとも過去のクエリがあった時刻)から現在時刻までをターゲットにクエリをインポートします。時系列範囲を指定するモードについては省略可能パラメータの `time_range_start` , `time_range_end` の説明を参照ください。

呼び出し可能な契約: QA ENGINE (*アンサーロボも近日対応予定)

エンドポイント

```
/capi/op/query-import
```

対応HTTPメソッド

- POST

実行に必要なAPI権限

- クエリインポートタスクの実行開始を許可

必須パラメータ

- なし

省略可能パラメータ

- `time_range_start` :クエリを取り込みたい時系列範囲のはじまりを日本時間で指定 (例: '2020-01-20 12:34:56') 省略した場合は最後に実行されたクエリインポートの読み込み範囲の終了点が使われます。過去に一度もクエリインポートが行われていない場合、もっとも過去のクエリがあった時刻になります。
- `time_range_end` :クエリを取り込みたい時系列範囲の終わりを日本時間で指定 (例: '2020-01-27 00:00:00') 省略した場合はリクエストしたタイミングの現在時間が使われます。

レスポンス: ステータスコード 200 (成功時)

`task_id` 属性で返却されるタスクIDを用いて 4.4 タスク確認API に問い合わせることでタスクが完了したかどうかを確認することができます。詳細は 4.4 タスク確認API の説明を参照ください。

```
{
  "status": "ok",
```

```
"result": {
  "task_id": "(タスクID)"
}
```

レスポンス: ステータスコード 400 (time_range_startパラメータ値のフォーマットが不正だった場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "invalid parameter: time_range_start"
}
```

レスポンス: ステータスコード 400 (time_range_endパラメータ値のフォーマットが不正だった場合)

```
{
  "status": "error",
  "code": "invalid_parameter",
  "message": "invalid parameter: time_range_end"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.4 タスク確認API

タスク確認APIでは以下に列挙したAPIで開始されたそれぞれのタスクについて、これらのAPIから返却されたタスクIDをもとにタスクの進行状況を確認することができます。

- 4.1 学習開始API (QA ENGINE契約でのみ呼出可能)
- 4.2 本番コピー開始API (QA ENGINE契約でのみ呼出可能)
- 4.3 クエリインポート開始API (QA ENGINE契約でのみ呼出可能) (※今後アンサーロボもサポート予定)
- 4.7 FAQ反映開始API (アンサーロボ契約でのみ呼出可能)

呼び出し可能な契約: QA ENGINE, アンサーロボ

エンドポイント

```
/capi/op/check
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- 学習/本番コピー/クエリインポートタスクの進行状況の問い合わせを許可

必須パラメータ

- task_id : 進行状況を確認したいタスクのIDを指定します。(上述の各APIのレスポンスで得られたもの)

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

```
{
  "status": "ok",
  "result": {
    "task_id": "(問い合わせ時にパラメータtask_idで指定したタスクID)",
    "state": "(ステータス)"
  }
}
```

state については以下の値のいずれかが入ります。

- issued : タスクが発行され、処理の順番待ちの状態です。
- processing : タスクが実行中です。
- finished : タスクは正常に処理が終了しました。
- finished_error : タスクは異常終了しました。この値が出た場合はStudio Ousiaにお問い合わせください。

レスポンス: ステータスコード 400 (不正なタスクID)

パラメータ task_id を指定していない、または空文字列を指定した場合

```
{
  "status": "error",
  "code": "operation_invalid_task_id",
  "message": "invalid task id"
}
```

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

レスポンス: ステータスコード 404 (存在しないタスクID)

パラメータ task_id で指定されたタスクIDをもつタスクが存在しない場合

```
{
  "status": "error",
  "code": "operation_no_such_task",
  "message": "no such task"
}
```

4.5 ステージングAPI情報取得API

ステージングAPI情報取得APIではステージングAPIのエンドポイントURLと呼び出しに必要なAPIキーのリストを得ることができます。

呼び出し可能な契約: QA ENGINE

エンドポイント

```
/capi/op/endpoint/dev
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- ステージングAPIエンドポイント情報の問い合わせを許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時/ステージングAPIが起動している場合)

QA ENGINE利用開始から一度でも学習を行なっている場合はステージングAPIが起動しているため、以下のレスポンスになります。

注意事項:

- `endpoint` 属性のドメイン名を用いて `https://(ドメイン名)/api/query` がQA ENGINEの単一クエリのWebAPIエンドポイントになります。詳しくはQA ENGINEのAPI仕様書を合わせてご覧ください。
- `api_keys` 属性で返却されるAPIキーは現状ではステージングAPIが起動した(=はじめて学習が行われた)際に自動的に発行されるものになります。
- `model` 属性の `env` 属性の `dev` はステージングAPIであることを表しています。(固定文字列)
- `model` 属性の `name` 属性は現状では意味のある値を返しません
- `model` 属性の `precisions` 属性の「参考正答率」の意味についてはQA ENGINEのヘルプページを参照してください。

```
{
  "status": "ok",
  "result": {
    "endpoint": "api-dev-8c2b1cb51177e66xxxxx.qaengine.aiのようなドメイン名",
    "model": {
      "created": "2020-04-12T12:34:56のようなタイムスタンプ(日本標準時)",
      "env": "dev",
      "name": "(qaengine内部で使われているmodel名)",
      "precisions": [
        (TOP1までの参考正答率),
        (TOP2までの参考正答率),
        (TOP3までの参考正答率),
        (TOP4までの参考正答率),
        (TOP5までの参考正答率),
        (TOP6までの参考正答率),
        (TOP7までの参考正答率),
        (TOP8までの参考正答率),
        (TOP9までの参考正答率),
        (TOP10までの参考正答率)
      ]
    }
  },
  "api_keys": ["(ステージングAPI起動時に自動的に発行されるAPIキー)"]
}
```

```
}  
}
```

レスポンス: ステータスコード 200 (成功時/ステージングAPIが起動していない場合)

QA ENGINE利用開始から一度も学習を行っていない場合、ステージングAPIが起動が起動していないため、以下のレスポンスになります。

```
{  
  "status": "ok",  
  "result": {  
    "endpoint": null,  
    "model": null,  
    "api_keys": []  
  }  
}
```

レスポンス: ステータスコード 403 (契約上の問題)

「1.8 契約上の問題」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.6 本番API情報取得API

本番API情報取得APIでは本番APIのエンドポイントURLと呼び出しに必要なAPIキーのリストを得ることができます。

呼び出し可能な契約: QA ENGINE

エンドポイント

```
/capi/op/endpoint/prod
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- 本番APIエンドポイント情報の問い合わせを許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時/本番APIが起動している場合)

QA ENGINE利用開始から一度でも本番コピーを行なっている場合は本番APIが起動しているため、以下のレスポンスになります。

注意事項:

- `endpoint` 属性のドメイン名を用いて `https://(ドメイン名)/api/query` がQA ENGINEの単一クエリのWebAPIエンドポイントになります。詳しくはQA ENGINEのAPI仕様書を合わせてご覧ください。
- `api_keys` 属性で返却されるAPIキーは現状では本番APIが起動した際に自動的に発行されるものになります。
- `model` 属性の `env` 属性の `prod` は本番APIであることを表しています。(固定文字列)
- `model` 属性の `name` 属性は現状では意味のある値を返しません
- `model` 属性の `precisions` 属性の「参考正答率」の意味についてはQA ENGINEのヘルプページを参照してください。

```
{
  "status": "ok",
  "result": {
    "endpoint": "api-prod-1a932fc32598e66xxxxx.qaengine.aiのようなドメイン名",
    "model": {
      "created": "2020-05-21T23:14:56のようなタイムスタンプ(日本標準時)",
      "env": "prod",
      "name": "(qaengine内部で使われているmodel名)",
      "precisions": [
        (TOP1までの参考正答率),
        (TOP2までの参考正答率),
        (TOP3までの参考正答率),
        (TOP4までの参考正答率),
        (TOP5までの参考正答率),
        (TOP6までの参考正答率),
        (TOP7までの参考正答率),
        (TOP8までの参考正答率),
        (TOP9までの参考正答率),
        (TOP10までの参考正答率)
      ]
    },
    "api_keys": ["(本番API起動時に自動的に発行されるAPIキー)"]
  }
}
```

レスポンス: ステータスコード 200 (成功時/本番APIが起動していない場合)

QA ENGINE利用開始から一度も学習を行っていない場合、ステージングAPIが起動が起動していないため、以下のレスポンスになります。

```
{
  "status": "ok",
  "result": {
    "endpoint": null,
    "model": null,
    "api_keys": []
  }
}
```

レスポンス: ステータスコード 403 (契約上の問題)

「1.8 契約上の問題」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.7 FAQ反映開始API

データベースにある「アクティブ」設定になっているFAQで、アンサーロボAPIを起動します(既に起動している場合はFAQ内容を反映します)。この処理は時間がかかる処理であるため、非同期タスクとして発行されます。

非同期タスクの進捗状況は [4.4 タスク確認API](#) に問い合わせることで確認できます。詳しくは成功時のレスポンスの説明を参照ください。

注意点:

- すでに実行中のFAQ反映タスクがある場合はエラーとなりFAQ反映は開始されません。
- アクティブ設定になっているFAQが0件の場合エラーになりFAQ反映を開始できません。

呼び出し可能な契約: アンサーロボ

エンドポイント

```
/capi/op/faq-apply
```

実行に必要なAPI権限

- [FAQ反映の実行開始を許可](#)

対応HTTPメソッド

- POST

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時)

`task_id` 属性で返却されるタスクIDを用いて [4.4 タスク確認API](#) に問い合わせることでタスクが完了したかどうかを確認することができます。詳細は [4.4 タスク確認API](#) の説明を参照ください。

```
{
  "status": "ok",
  "result": {
    "task_id": "(タスクID)"
  }
}
```

レスポンス: ステータスコード 400 (有効FAQ数が1未満)

有効FAQ数が1未満(つまりゼロ)の場合、アンサーロボAPIを構築できないので以下のエラーを返します。

```
{
  "status": "error",
  "code": "operation_faq_apply_data_error_n_faq",
  "message": "too small faq number"
}
```

レスポンス: ステータスコード 403 (契約上の問題エラー)

「1.8 契約上の問題エラー」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

4.8 アンサーロボAPI情報取得API

アンサーロボAPI情報取得APIではアンサーロボAPIのエンドポイントURLと呼び出しに必要なAPIキーのリストを得ることができます。

呼び出し可能な契約: アンサーロボ

エンドポイント

```
/capi/op/endpoint/answer-robot
```

対応HTTPメソッド

- GET

実行に必要なAPI権限

- アンサーロボAPIエンドポイント情報の問い合わせを許可

必須パラメータ

- なし

省略可能パラメータ

- なし

レスポンス: ステータスコード 200 (成功時/アンサーロボAPIが起動している場合)

アンサーロボ利用開始から一度でもFAQ反映を行なっている場合はアンサーロボAPIが起動しているため、以下のレスポンスになります。

注意事項:

- `endpoint` 属性のドメイン名を用いて `https://(ドメイン名)/api/query` がQA ENGINEの単一クエリのWebAPIエンドポイントになります。詳しくはQA ENGINEのAPI仕様書を合わせてご覧ください。(アンサーロボAPIもQA ENGINEのAPI仕様と互換の動作になります。)
- `api_keys` 属性で返却されるAPIキーは現状ではアンサーロボAPIが起動した際に自動的に発行されるものになります。
- `model` 属性の `env` 属性の `sosekifaq` はアンサーロボAPIであることを表しています。(固定文字列)
- `model` 属性の `name` 属性は現状では意味のある値を返しません
- `model` 属性の `precisions` 属性は長さ10の全てが `0.0` のリストになります

```
{
  "status": "ok",
  "result": {
    "endpoint": "sf-sys-prod-417de57eabe04e8e8ebc.qaengine.aiのようなドメイン名",
    "model": {
      "created": "2020-05-21T23:14:56のようなタイムスタンプ(日本標準時)",
      "env": "sosekifaq",
      "name": "(アンサーロボ内部で使われているmodel名)",
      "precisions": [
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0
      ]
    },
    "api_keys": ["(アンサーロボAPI起動時に自動的に発行されるAPIキー)"]
  }
}
```

レスポンス: ステータスコード 200 (成功時/アンサーロボAPIが起動していない場合)

アンサーロボ利用開始から一度もFAQ反映を行っていない場合、アンサーロボAPIが起動が起動していないため、以下のレスポンスになります。

```
{
  "status": "ok",
  "result": {
    "endpoint": null,
    "model": null,
    "api_keys": []
  }
}
```

レスポンス: ステータスコード 403 (契約上の問題エラー)

「1.8 契約上の問題エラー」を参照してください。

レスポンス: ステータスコード 403 (APIキー認証エラー)

「1.4 APIキー認証に失敗したときの共通レスポンス」を参照してください。

(付録.A) エラーコード一覧

API実行時にレスポンスに含まれたエラーコードの詳細な解釈については実行したAPIのドキュメントを合わせてご覧ください。

- `key_missing` HTTPヘッダにAPIキーが指定されなかった場合
- `key_invalid` APIキーが存在しているが設定で無効になっている場合、もしくは存在しないAPIキーの場合
- `key_no_priv` APIキーが実行しようとしているAPIを実行する権限が設定で付与されていない場合
- `internal_server_error` 予期しないシステムエラー(このエラーが発生した場合は弊社までご連絡くださいますようお願いいたします)
- `lack_parameter` APIを実行するのに必要なパラメータが不足している場合
- `not_found` FAQ/質問の更新/削除などのIDを指定して実行するAPIなどで指定したIDを持つリソースが存在しない場合
- `faq_invalid_identifier` FAQ取得/追加/更新/upsert/削除 APIでFAQ IDが未指定または空文字列で指定された場合
- `faq_identifier_taken` FAQ追加APIで指定したFAQ IDをもつFAQがすでに存在していた場合
- `question_invalid_identifier` 質問取得/追加/更新/upsert/削除 APIで質問 IDが未指定または空文字列で指定された場合
- `question_identifier_taken` 質問追加 APIで指定した質問IDをもつ質問がすでに存在していた場合
- `question_invalid_faq_identifier` 質問追加/更新/upsert APIでアノテーションのために指定したFAQ IDをもつFAQが存在しなかった場合
- `question_invalid_unannotate` 質問アノテーションAPIで `unannotate` パラメータを指定した場合でその値が `true` 以外だった場合
- `operation_invalid_task_id` タスク確認APIで `task_id` パラメータが未指定または空文字列で指定された場合
- `operation_no_such_task` タスク確認APIで `task_id` パラメータで指定されたタスクIDのタスクが存在しなかった場合
- `operation_another_operation_in_progress` 学習開始/本番コピー/クエリインポート開始APIで、すでに先んじて同種のタスクが実行され、実行途中である場合
- `operation_no_staging_api` 本番コピーAPIでステージングAPIが起動していない場合
- `operation_stage_data_error_n_faq` 学習開始APIで有効FAQ数が2未満の場合
- `operation_stage_data_error_n_question` 学習開始APIで有効かつアノテーション済みの質問数が10未満の場合
- `operation_prohibited` 契約上の問題があった場合(「1.8 契約上の問題エラー」を参照)
- `operation_faq_apply_data_error_n_faq` FAQ反映開始API呼び出しの際、有効FAQ数が1未満(つまりゼロ)の場合

(付録.B) APIに付与できる権限一覧

- FAQに関する読み取り操作を許可
- FAQに関する書き込み操作(追加/更新/削除)を許可
- 質問に関する読み取り操作を許可
- 質問に関する書き込み操作(追加/更新/削除)を許可
- 質問に関するアノテーション操作を許可
- 学習/本番コピー/クエリインポートタスクの進行状況の問い合わせを許可
- 学習タスクの実行開始を許可
- 本番コピータスクの実行開始を許可
- クエリインポートタスクの実行開始を許可

- ステージングAPIエンドポイント情報の問い合わせを許可
- 本番APIエンドポイント情報の問い合わせを許可
- FAQ反映の実行開始を許可
- アンサーロボAPIエンドポイント情報の問い合わせを許可

改訂履歴

- 2020-08-07 初版
- 2021-09-27 アンサーロボ制御APIについて加筆
- 2022-04-XX FAQ関連セクション(2.1/2.2/2.3/2.4/2.5)にてタグ/優先キーワード取得および関連付けの仕様について記述