

CTI HTTP/RESTインターフェース 1.0 仕様

目次

- [1.概要 \(1\)](#)
- [2.通信方法 \(2\)](#)
 - [2.1 アクション \(2\)](#)
 - [2.2 パラメータの送信 \(2\)](#)
 - [2.3 セッションの開始と終了 \(3\)](#)
 - [2.4 ドキュメントの変換処理 \(3\)](#)
 - [2.5 レスポンスとメッセージコード \(4\)](#)
- [3.各アクションの説明 \(5\)](#)
 - [3.1 open \(5\)](#)
 - [3.2 info \(5\)](#)
 - [3.3 properties \(6\)](#)
 - [3.4 resources \(7\)](#)
 - [3.5 transcode \(8\)](#)
 - [3.6 messages \(10\)](#)
 - [3.7 result \(11\)](#)
 - [3.8 abort \(12\)](#)
 - [3.9 reset \(12\)](#)
 - [3.10 close \(12\)](#)
- [4.ライセンス \(13\)](#)

1.概要

CTI HTTP/RESTインターフェースは、ウェブブラウザや各種プログラミング言語で実装されたHTTPクライアントライブラリにより、ドキュメント変換サーバーCopper PDFの機能を利用するためのインターフェースとして開発されました。

従来、Copper PDFはCTIPと呼ばれる、ソケット通信上に実装されたプロトコルで通信していました。このプロトコルは高速である反面、複雑であるため、クライアントの実装に非常に手間がかかりました。

CTI HTTP/RESTインターフェースは、HTTP上に実装された、比較的簡単なプロトコルであるため、一般的なHTTPクライアントライブラリを利用してアクセスすることができます。それでいて、CTIP 2.0と同等の機能を備えています。

また、HTTPを利用することで、ファイアウォール越しに接続しやすくなります。

以下は、localhostの8097ポートでドキュメント変換サーバーが動いている場合の、最も簡単な利用例です。これをHTMLファイルとして保存し、変換ボタンを押すと、Hello world!の変換結果が表示されます。

```
<form action="http://localhost:8097/transcode">
  <input type="hidden" name="rest.user" value="user"/>
  <input type="hidden" name="rest.password" value="kappa"/>
  <input type="text" name="rest.main" value="Hello world!"/>
  <button type="submit">変換</button>
</form>
```

2.通信方法

2.1 アクション

接続の開始、終了、開始といった動作は、特定のURLにHTTPクライアントがアクセスすることにより実行され、アクションと呼びます。アクションの種類は、ファイルパスの文字列により識別されます。例えばlocalhostの8097ポートでドキュメント変換サーバーが動作している場合、セッションを開始するには、`http://localhost:8097/open`にアクセスします。

アクションの一覧は以下の表のとおりです。アクションは、リクエストパラメータと共に実行する必要があります。アクションとパラメータの詳細は後で説明しています。

パス	説明
/open	セッションの開始 (5ページ)
/info	サーバーの情報 (5ページ)
/properties	プロパティ設定 (6ページ)
/resources	リソース送信 (7ページ)
/transcode	ドキュメント変換処理 (8ページ)
/messages	メッセージ受信 (10ページ)
/result	ドキュメント変換結果受信 (11ページ)
/abort	ドキュメント変換処理の中断 (12ページ)
/reset	リセット (12ページ)
/close	セッションの終了 (12ページ)

2.2 パラメータの送信

パラメータは、クエリパラメータとしてURLに付加する方法と、`application/x-www-form-urlencoded`により送る方法と、`multipart/form-data`により送る方法の3種類があります。後者の方法ほど、大きなデータを送ることができます。特に、`multipart/form-data`は、1回のアクションで複数のファイルを送信する唯一の方法です。

2.3 セッションの開始と終了

ドキュメント変換サーバーに接続し、接続を終了するまでの複数のアクションによる操作をセッションと呼びます。セッションはopenアクションにより開始し、closeアクションにより終了しますが、closeアクションを行わなければ、一定期間でセッションが終了します。

open以外のアクションを実行するためには、原則としてセッションが開始した状態である必要がありますが、infoとtranscodeアクションは認証情報のパラメータを付与することで、セッションが開始していない状態で実行出来ます。ただし、transcodeアクションでは使用できる機能が一部限られます。

2.4 ドキュメントの変換処理

ドキュメント変換サーバーは、HTMLのようなドキュメントを変換する場合、変換状況やエラーをメッセージとして出力しながら処理します。また、文書に付随する画像等（リソース）があれば、サーバー側のファイルシステム等から取得するか、事前にクライアントから送られたリソースを利用しようと試みます。また、例えばPDFを出力する場合、変換結果は1つのファイルですが、複数のページを別々の画像として出力するような場合は、変換結果が複数のファイルになることがあります。

変換結果が1つしかない処理の場合は、セッションを開始させずに、transcodeアクションを実行するだけで、ほとんどの用途には問題ありません。セッションを開始することが必要となるのは、変換処理中のメッセージを取得したい場合、処理の中断が必要な場合、サーバー側からクライアントにリソースの送信を要求する場合、画像など複数の処理結果を生成する必要がある場合です。

変換処理中のメッセージを取得するには、変換処理中か処理後にmessagesアクションを呼び出します。変換処理中のmessagesアクションでは、現在までにサーバーが出力したメッセージを得ることが出来ます。リアルタイムに近い形でメッセージを得るには、クライアントが定期的にmessagesアクションを呼び出す必要があります。messagesアクションを呼び出してもメッセージが得られないケースを減らすため、実際にmessagesアクションが返す内容に変化があるまで待つ機能が用意されています。messagesアクションの呼び出しにより取得されたメッセージは、次のmessagesアクションの呼び出しで取得することはできません。

変換処理を中断するには、変換処理中にabortアクションを呼び出します。

プロパティやリソースは一度のtranscodeアクションでも送ることが出来ますが、セッションを利用する場合は、propertiesアクションとresourcesアクションで別途送信することが出来ます。

transcodeアクションは、サーバーがクライアント側にリソースの送信を要求するモードで変換処理を実行することが出来ます。例えば画像がメインドキュメントのHTMLファイル中のIMG要素で参照されている場合、クライアントは参照されている画像をリソースとしてサーバーに送信する必要がありますが、そのためには、クライアントはHTMLから参照されている画像を知る必要があります。通常、HTMLから参照されている画像を調べるには複雑な処理を要するため、それをドキュメント変換サーバーに委ねるのがこのモードです。クライアントはmessagesアクションにより、サーバーから要求されたリソースのURIを知ることが出来ます。その場合、クライアントがresourcesアク

ションで要求されたりソースをサーバーに送るか、リソースの不存在を通知しない限り、処理が滞ります。そのため、このモードではクライアントは定期的にmessagesアクションを呼び出す必要があります。

transcodeアクションは、1つの変換結果であれば直接レスポンスとして返すことができますが、複数の変換結果を受け取る場合は非同期モードでtranscodeアクションを実行する必要があります。結果がどのように生成されたかは、messagesアクションを呼び出すことで確認することができます。変換結果のデータはresultアクションにより取得することができます。

ドキュメント変換処理が終わった後、同じセッションで続けてドキュメント変換処理を繰り返すことができます。このとき、既に設定されたプロパティと、送信済みのリソースは再利用されます。プロパティと、送信済みのリソースを破棄して、初期状態に戻すには、resetアクションを実行します。

2.5 レスポンスとメッセージコード

アクションに対して返されるレスポンスは、ドキュメントの変換結果等のデータか、次の形式のXMLです。

```
<?xml version="1.0"?>
<response>
  <message code="[メッセージコード]">[メッセージ]</message>
</response>
```

メッセージコードは、4桁の16進数のコードです。プログラム等で処理する場合、メッセージコードを10進数として解析しないでください。メッセージは人間が読める形式のテキストです。ただし、メッセージコード1012では、ここにはセッションIDが入りません。

メッセージコード	説明	HTTPステータス
1011	正常にアクションが実行されました。	200
1012	セッションが開始されました。	200
1013	変換処理を実行中です。	200
1014	変換処理が終了しました。	200
3001	メインドキュメントのURIが不正です。	403
3011	不正なアクションです。	403
3012	セッションがありません。	403
3013	メインドキュメントがありません。	403
3014	認証に失敗しました。	403
3015	不正なリクエストです。	400
3016	変換結果がありません。	404
4001	その他のエラー。	500

3.各アクションの説明

3.1 open

3.1.1 アクション

ドキュメント変換サーバーとのセッションを開始します。

3.1.2 リクエスト

表 3.1 パラメータ

名前	必須	説明
rest.user		ユーザー名
rest.password		パスワード
rest.timeout		セッションの最小持続時間
rest.httpSession		HTTPセッションの作成

このアクションでドキュメント変換サーバーにセッションした後は、他のアクションでrest.idパラメータにセッションIDを渡すことでセッションを識別できるようになります。

rest.timeoutには、セッションの最小持続時間をミリ秒単位で指定します。省略した場合は180000(3分)で、最大は3600000(1時間)に制限されます。

rest.httpSessionパラメータにtrueを設定すると、クッキーに対応したHTTPクライアントに対して、クッキーを使ったHTTPセッション(RFC2965)を生成します。HTTPセッションを使うモードでは、HTTPセッションをドキュメント変換サーバーのセッションに対応させるため、以降のアクションのrest.idは不要になります。

3.1.3 レスポンス

正常にセッションが開始された場合、以下のレスポンスを返します。[セッションID]を以降のアクションのrest.idパラメータに設定することで、セッションが同定されます。

```
<?xml version="1.0"?>
<response>
  <message code="1012">[セッションID]</message>
</response>
```

3.2 info

3.2.1 アクション

サーバー情報を返します。

3.2.2 リクエスト

表 3.2 パラメータ

名前	必須	説明
rest.id		セッションID
rest.user		ユーザー名
rest.password		パスワード
rest.uri	✓	サーバー情報の識別URI

サーバー情報を返します。ドキュメント変換サーバーのバージョンや、対応データ形式等を取得することができます。rest.uriは、サーバー情報の種類を決めるためのURIです。URIについては、[サーバー情報のURIと内容](#)を参照してください。

rest.idパラメータが存在する場合、事前にopenアクションにより開始したセッションに対して実行されます。

rest.idパラメータが省略され、かつHTTPセッションからもセッションを得ることができない場合は、今回のtranscodeアクションに限り有効なセッションが作られます。そのとき、rest.userとrest.passwordが認証に使われます。

3.2.3 レスポンス

[サーバー情報のURIと内容](#)を参照してください。

3.3 properties

3.3.1 アクション

セッションに対するプロパティを設定します。

このアクションは、ドキュメント変換処理を実行中のセッションに対しては実行できません。

3.3.2 リクエスト

表 3.3 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID
("rest."で始まらないパラメータ)		プロパティ

"rest."で始まらないパラメータは複数送ることが出来ます。それぞれがプロパティとして設定されます。

3.4 resources

3.4.1 アクション

セッションに対するリソースを送ります。

このアクションは、通常はドキュメント変換処理を実行中のセッションに対しては実行できません。

ただし、transcodeアクションでrest.requestResourceをtrue設定した変換処理中に、サーバーにより要求されたリソースを送るために使用することが出来ます。

利便性のため、propertiesアクションに相当するプロパティの設定機能も併せ持っています。

3.4.2 リクエスト

表 3.4 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID
rest.uri		リソースのURI
rest.mimeType		リソースのmime型
rest.encoding		リソースのキャラクタ・エンコーディング
rest.resource		リソースファイル
rest.notFound		リソースファイルが存在しないことを示す
("rest."で始まらないパラメータ)		プロパティ

POSTメソッドで、application/x-www-form-urlencodedまたはmultipart/form-dataのいずれでもない形式でデータを送信した場合、送信データ自身がリソースファイルとなります。

multipart/form-data形式の場合、セッションIDは必ず一番最初に送るか、クエリパラメータとして送る必要があります。

multipart/form-dataではない場合、rest.uri, rest.mimeType, rest.encodingは、それぞれ最初の値だけがリソースファイルに適用されます。

application/x-www-form-urlencodedの場合、あるいはクエリパラメータとしてrest.resourceが送られた場合、最初の値だけが有効です。

multipart/form-dataの場合、rest.resourceは複数送ることが出来ます。rest.uri, rest.mimeType, rest.encodingは、次にrest.resourceにより送られるリソースファイルに適用されます。リソースファイルが送られた時点でこれらのパラメータの効果はリセット

トされます。これらのパラメータがクエリパラメータとして送られた場合、最初の値だけが最初のリソースファイルに適用されます。ただし、リソースファイルの前にフィールドとして送られたパラメータにより上書きされます。

application/x-www-form-urlencodedではない場合、HTTPリクエストまたはファイルパートのヘッダにより、リソースのURI、リソースのキャラクタ・エンコーディング、リソースのmime型、リソースファイルを設定することができます。ただし、ヘッダよりリクエストパラメータによるものが優先されます。優先度とヘッダによる設定は次の表の通りとなります。

属性	最優先	次に優先	優先度低	デフォルト
URI	rest.uri パラメータ	X-URIヘッダ	Content-Dispositionヘッダのfilenameパラメータ(相対URI)	"/"(カレントファイルを表す相対URI)
mime型	rest.mimeType パラメータ	Content-Type ヘッダ		なし(サーバー側の自動判別)
キャラクタ・エンコーディング	rest.encoding パラメータ	Content-Type ヘッダの charset パラメータ		なし(サーバー側の自動判別)

rest.notFoundはサーバーから要求されたリソースを送信する場合だけ、rest.resourceパラメータの代わりに使うことができます。trueに設定すると、そのリソースが存在しないことを示します。

"rest."で始まらないパラメータは複数送ることが出来ます。それぞれがプロパティとして設定されます。

3.5 transcode

3.5.1 アクション

変換処理を実行します。

このアクションは、openによりセッションを開始していない状態でも実行出来ます。

変換結果は1つだけ受け取る方法と、複数受け取る方法があります。openによりセッションを開始していない状態では、結果はこのアクションのレスポンスとして1つしか受け取ることができません。

利便性のため、properties、resourcesアクションに相当するプロパティの設定とリソースの送信機能も併せ持っています。

このアクションは、変換処理を実行中のセッションに対しては実行できません。

3.5.2 リクエスト

表 3.5 パラメータ

名前	必須	説明
rest.id		セッションID

名前	必須	説明
rest.user		ユーザー名
rest.password		パスワード
rest.async		非同期モードの設定
rest.requestResource		サーバーからリソースを要求するモードの設定
("rest."で始まらないパラメータ)		プロパティ
rest.uri		メインドキュメント/リソースのURI
rest.mimeType		メインドキュメント/リソースのmime型
rest.encoding		メインドキュメント/リソースのキャラクタ・エンコーディング
rest.resource		リソースファイル
rest.main	説明を参照	メインドキュメントファイル
rest.mainURI	説明を参照	メインドキュメントファイルURI

rest.idパラメータが存在する場合、事前にopenアクションにより開始したセッションに対して実行されます。

rest.idパラメータが省略され、かつHTTPセッションからもセッションを得ることができない場合は、今回のtranscodeアクションに限り有効なセッションが作られます。そのとき、rest.userとrest.passwordが認証に使われます。

POSTメソッドで、application/x-www-form-urlencodedまたはmultipart/form-dataのいずれでもない形式でデータを送信した場合、送信データがメインドキュメントファイルとなります。

multipart/form-data形式でパラメータ・ファイルを送信する場合、rest.mainまたはrest.mainURI以外のパラメータは必ずrest.mainまたはrest.mainURIの前に送るか、クエリパラメータとして送る必要があります。

application/x-www-form-urlencodedまたはmultipart/form-dataの場合、rest.mainまたはrest.mainURIが送られた時点で処理を終了します。

"rest."で始まらないパラメータは複数送ることが出来ます。それぞれがプロパティとして設定されます。

リソースの処理については、resourcesアクションと同様です。

rest.uri, rest.mimeType, rest.encodingはリソースファイルの他、rest.mainによるメインドキュメントファイルにも適用されます。rest.mainURIによるメインドキュメントファイルには適用されません。HTTPリクエストまたはファイルパートのヘッダとの関係についてはresourcesアクションのリソースファイルと同様です。

3.5.3 レスポンス

rest.asyncパラメータが設定されていない場合、メインドキュメントの変換結果が生成されるまでレスポンスを返しません。レスポンスの内容は変換結果そのものとなります。

rest.asyncパラメータがtrueの場合、即座にメッセージコード1011のレスポンスを返します。メインドキュメントの内容は、messagesアクションで処理の完了を確認した後、resultsアクションで取得する必要があります。ただし、rest.idパラメータが存在しない場合はrest.asyncパラメータは無視されます。

rest.requestResourceがtrueの場合、サーバーが必要なリソースの送信を要求し、リソースが送られるまで処理が滞るようになります。ただし、rest.asyncパラメータがtrueでない場合はrest.requestResourceパラメータは無視されます。

3.6 messages

3.6.1 アクション

セッションの変換処理で出力されたメッセージと、処理状況を返します。

3.6.2 リクエスト

表 3.6 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID
rest.wait		待ち時間

rest.waitパラメータが省略された場合、即座にレスポンスを返します。

rest.waitパラメータが設定された場合、メッセージが生成されるか、結果が生成されるか、変換処理が終了するまで、最大で指定された時間(ミリ秒)まで待ちます。0の場合は最大待ち時間を設定しません。また、変換処理が行われていないか、既に終わっている場合は、即座にレスポンスを返します。

3.6.3 レスポンス

レスポンスは次のような形式です。

```
<?xml version="1.0"?>
<response>
  <message code="[メッセージコード]"></message>
  <messages>
    <message code="[メッセージコード]" arg0="[値0]" arg1="[値1]" ...>[メッセージ]</message>
    ...
  </messages>
  <interrupted code="[メッセージコード]" arg0="[値0]" arg1="[値1]" ...>[メッセージ]</interrupted>
  <resources>
    <resource uri="[URI]" />
  </resources>
```

```

<results>
  <result uri="[URI]" mimeType="[mime型]"
    encoding="[キャラクタ・エンコーディング]" length="[バイト数]"/>
</results>
<progress read="[読み込み済みバイト数]" length="[ソースドキュメントバイト数]" />
</response>

```

response直下のmessageは、まだ変換処理を実行中であればコード1013、変換処理が終了していればコード1014を返します。

messages内のmessageは変換処理過程で生成されたメッセージのうち、まだクライアントに送られていないものです。未送信のメッセージが存在しない場合は、messages要素はありません。メッセージコードはドキュメント変換サーバーに依存します。

interruptedが存在する場合は、変換処理がメッセージに示された理由で中断されています。メッセージコードはドキュメント変換サーバーに依存します。

resources内のresourceはrest.requestResourceをtrueとしたモードの変換処理過程で要求されたリソースのうち、クライアントからリソースの内容の送信または不存在の通知がされていないものです。要求されるリソースが存在しない場合はresources要素はありません。

results内のresultは変換処理過程で生成された変換結果です。result要素のmimeType、encoding、lengthの各属性は省略されることがあります。変換結果が生成されていない場合はresults要素はありません。

progressは処理状況です。ソースドキュメントの大きさが不明の場合は、length属性はありません。また、変換処理が開始されておらず、ソースドキュメントの大きさも不明の場合はprogress要素はありません。

3.7 result

3.7.1 アクション

変換結果を返します。

このアクションはrest.asyncにtrueを設定して変換処理を行った場合だけ有効です。

3.7.2 リクエスト

表 3.7 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID
rest.uri		変換結果URI

messagesアクションで得られるメッセージのresult要素のuri属性の値を、rest.uriに指定すると、その結果が得られます。

3.7.3 レスポンス

rest.uriで指定した結果が存在する場合は結果データ。rest.uriで指定した結果が存在しない場合は3013メッセージ。

3.8 abort

3.8.1 アクション

セッションの変換処理を中断します。

3.8.2 リクエスト

表 3.8 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID
rest.mode		即時中断の設定

rest.modeパラメータが存在しない場合は、なるべく「きりのよいところ」まで変換処理を実行した後、中断します。

rest.modeパラメータがforceの場合は、なるべく即座に変換処理を中断します。

3.9 reset

3.9.1 アクション

セッションのプロパティ、送信済みのリソース、変換結果をクリアします。

変換処理が実行中の場合、即時中断処理され、変換処理が終わるのを待ちます。

3.9.2 リクエスト

表 3.9 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID

3.10 close

3.10.1 アクション

セッションを終了します。

変換処理が実行中の場合、即時中断処理されますが、実際の変換処理の終了を待たず、すぐにセッションを利用できなくなります。

3.10.2 リクエスト

表 3.10 パラメータ

名前	必須	説明
rest.id	✓(但しHTTPセッションを使う場合は不要)	セッションID

4.ライセンス

Copyright (c) 2009 株式会社GNN

Apache License Version 2.0に基づいてライセンスされます。
 あなたがこのファイルを使用するためには、本ライセンスに従わなければなりません。
 本ライセンスのコピーは下記の場所から入手できます。

<http://www.apache.org/licenses/LICENSE-2.0>

適用される法律または書面での同意によって命じられない限り、
 本ライセンスに基づいて頒布されるソフトウェアは、明示黙示を問わず、
 いかなる保証も条件もなしに「現状のまま」頒布されます。
 本ライセンスでの権利と制限を規定した文言については、本ライセンスを参照してください。

Copyright (c) 2009 GNN & Co.,Ltd.

Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License.