記事
Toshihiko Minamoto · 2021年10月28日 8m read

Open Exchange

REST経由でファイル転送しプロパティに格納する - パート1

InterSystems開発者コミュニティにおいて、CachéアプリケーションへのTWAINインターフェースの作成の可能性に関する質問が上がりました。Webクライアントの撮像装置からサーバにデータを取得し、そのデータをデータベースに格納する方法について、素晴らしい提案がいくつかなされました。

しかし、こういった提案を実装するには、Webクライアントからデータベースサーバにデータを転送し、受信データをクラスプロパティ（または質問のケースで言えばテーブルのセル）に格納できなければなりません。この方法は、TWAINデバイスから受信した撮像データを転送するためだけでなく、ファイルアーカイブや画像共有などの整理といったほかのタスクにも役立つ可能性があります。

そこで、この記事では主に、HTTP POSTコマンドの本体から、raw状態またはJSON構造にラップしてデータを取得するRESTfulサービスを記述する方法を説明することにします。

RESTの基礎

具体的な話に入る前に、まずREST全般と、IRISでRESTfulサービスがどのように作成されるかについて簡単に説明しましょう。

Representational state transfer（REST）は、分散ハイパーメディアシステムのためのアーキテクチャスタイルです。RESTにおける情報の重要な抽象化は、適切な識別子を持ち、JSON、XML、またはサーバークライアントの両方が認識するほかの形式で表されるリソースです。

通常、クライアントとサーバー間でのデータの転送にはHTTPが使用されます。CRUD（作成、読み取り、更新、削除）操作ごとに独自のHTTPメソッド（POST、GET、PUT、DELETE）があり、リソースまたは一連のリソースには独自のURIがあります。

この記事では、POSTメソッドのみを使用して新しい値をデータベースに挿入するため、その制約を知る必要があります。

「IETF RFC7231 4.3.3 POST」の仕様によると、POSTには本文に格納されるデータサイズに関する制限はありません。しかし、Webサーバーやブラウザでは独自の制限が適用され、通常は1 MBから2 GBとなっています。たとえば、Apacheの制限は最大2 GBとなっています。いずれにせよ、2 GBのファイルを送信する必要がある場合は、アプローチを考え直すことをお勧めします。

IRISにおけるRESTfulサービスの要件

IRISでRESTfulサービスを実装するには、次を行う必要があります。

抽象ｸﾗｽ %CSP.RESTを拡張するｸﾗｽﾌﾞﾛｰｶｰを作成ます。
これは逆に、 %CSP.Page
を拡張するため、さ
まざまな便利メソッド、パラメータ
、オブジェクト、特に %requestへのｱｸｾｽが可能になります。
ルートを定義する UrlMapを指定します。
オプションとして UseSession
パラメータを設定し、各REST呼び出しが独自のWebセッションで実行されるのか、ほかのREST呼
び出しで1つのセッションを共有するのかを設定します。
ルートで定義された演算を実行するｸﾗｽメソッドを提供します。
CSP
Webｱﾌﾟﾘｹｰｼｮﾝを定義し、そのセキュリティをWebｱﾌﾟﾘｹｰｼｮﾝﾍﾟｰｼﾞ（システム管理 >
セキュリティ > ｱﾌﾟﾘｹｰｼｮﾝ >
Webｱﾌﾟﾘｹｰｼｮﾝ）に指定します。 Dispatch
ｸﾗｽには、ユーザｸﾗｽの名前と Name
が格納されています。これはREST呼び出しのURLの最初の部分です。
一般に、大きなデータ（ファイル）のチャンクとメタデータをｸﾗｲｱﾝﾄからｻｰﾊﾞｰに送信するに
は、次のようにいくつかの方法があります。

ファイルとメタデータをBase64-encodeで暗号化し、ｻｰﾊﾞｰとｸﾗｲｱﾝﾄの両方に暗号化/復号化
を行うための処理オーバーヘッドを追加します。
最初にファイルを送信し、ｸﾗｲｱﾝﾄにIDを返します。すると、そのIDを持つメタデータが送信さ
れます。ｻｰﾊﾞｰはファイルとメタデータをもう一度関連付けます。
最初にメタデータを送信し、ｸﾗｲｱﾝﾄにIDを返します。すると、そのIDのファイルが送信されま
す。ｻｰﾊﾞｰはファイルとメタデータをもう一度関連付けます。
この第1回目の記事では、基本的に2つ目のｱﾌﾟﾛｰﾁを使用しますが、ｸﾗｲｱﾝﾄへのIDの送信とメ
タデータ（別のプロパティとして格納するためのファイル名）の追加については、このﾀｽｸでは特に
意味を持たないため、行いません。第2回目の記事では、最初のｱﾌﾟﾛｰﾁを使用しますが、私のファ
イルとメタデータをｻｰﾊﾞｰに送信する前にJSON構造にﾊﾟｯｹｰｼﾞ化します。

RESTfulｻｰﾋﾞｽの実装

では、具体的な内容に入りましょう。
まず、設定しようとしているｸﾗｽとﾌﾟﾛﾊﾟﾃｨを定義しましょう 。

```
Class RestTransfer.FileDesc Extends %Persistent
{
  Property File As %Stream.GlobalBinary;
  Property Name As %String;
}
```

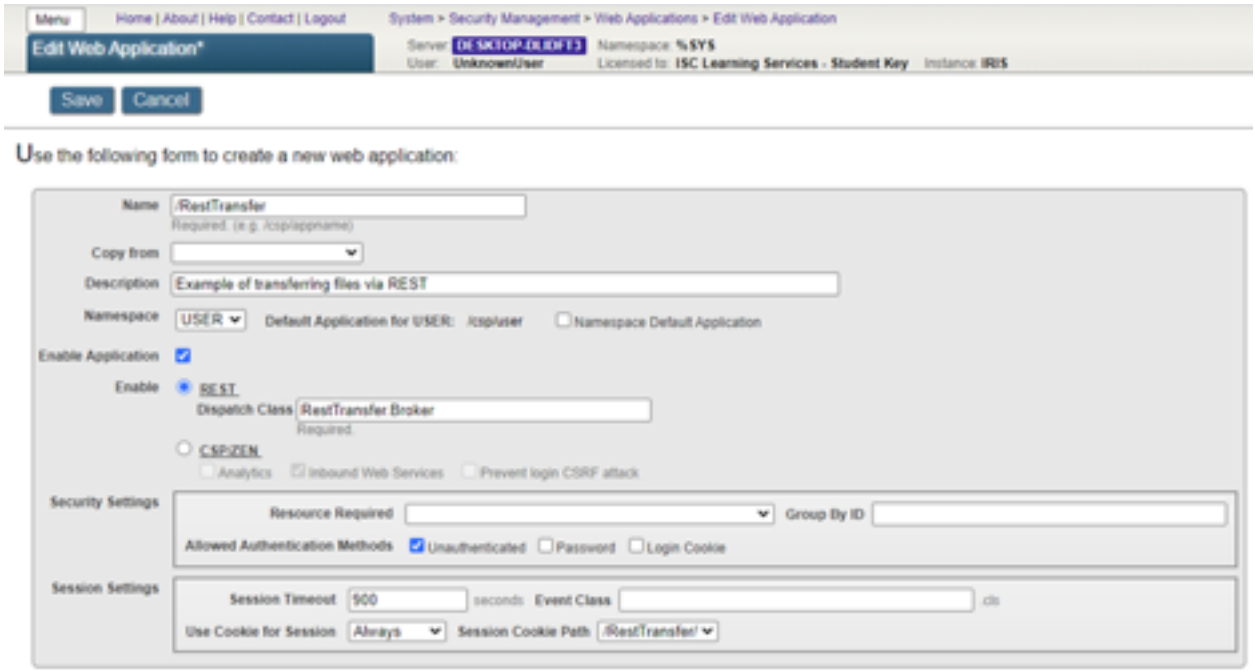
もちろん、通常はファイルに使用するメタデータがほかにもありますが、私たちの目的にはこれで十分
です。

次に、ｸﾗｽﾌﾞﾛｰｶｰを作成る必要があります。これは、ﾙｰﾄとメソッドを使用して後で拡張し

ます。

```
Class RestTransfer.Broker Extends %CSP.REST
{
  XData UrlMap
  {
    &lt;Routes>
    &lt;/Routes>
  }
}
```

そして最後に、前段階のセットアップをして、このアプリケーションをWebアプリケーションのリスト
に指定する必要があります。



前段階のセットアップが完了したので、RESTクライアント（ Advanced REST Client
を使用します）から送信するファイルを クラス RestTransfer.FileDesc
のインスタンスの File
プロパティにしてデータベースに格納するメソッドを記述できるようになりました。

では、POSTメソッドの本体にRawデータとして格納されている情報から始めましょう。
まず、新しいルートを UrlMapに追加しましょう。

```
<Route Url="/file" Method="POST" Call="InsertFileContents"/>
```

このルートは、サービスがURL /RestTransfer/file
でPOSTコマンドを受信すると、 InsertFileContentsクラスメソッドを呼び出すことを指定します。
より簡単に行うために、このメソッドの中にクラス RestTransfer.FileDesc
の新しいインスタンスを作成して、その Fileプロパティを受信データに設定します。これにより、ステ
ータスが、成功またはエラーのいずれかを示すJSON形式のメッセージが返されます。
以下はそのクラスメソッドです。

```
ClassMethod InsertFileContents() As %Status
{
  Set result={}
  Set st=0
  set f = ##class(RestTransfer.FileDesc).%New()
  if (f = $$$NULLOREF) {
    do result.%Set("Message","Couldn't create an instance of the class")
  } else {
    set st = f.File.CopyFrom(%request.Content)
    If $$$ISOK(st) {
      set st = f.%Save()
      If $$$ISOK(st) {
        do result.%Set("Status","OK")
      } else {
        do result.%Set("Message",$system.Status.GetOneErrorText(st))
      }
    } else {
      do result.%Set("Message",$system.Status.GetOneErrorText(st))
    }
  }
  write result.%ToJSON()
  Quit st
}
```

まず、ｸﾗｽ　　RestTransfer.FileDesc
の新しいｲﾝｽﾀﾝｽを作成、正常に作成れたこﾋﾞOREFがあるこﾋﾞをﾁｪｯｸします。
ｵﾌﾞｼﾞｪｸﾄを作成なかったﾋﾞ場合は、JSON構造を作成ます。

```
{"Message", "Couldn't create an instance of class"}
```

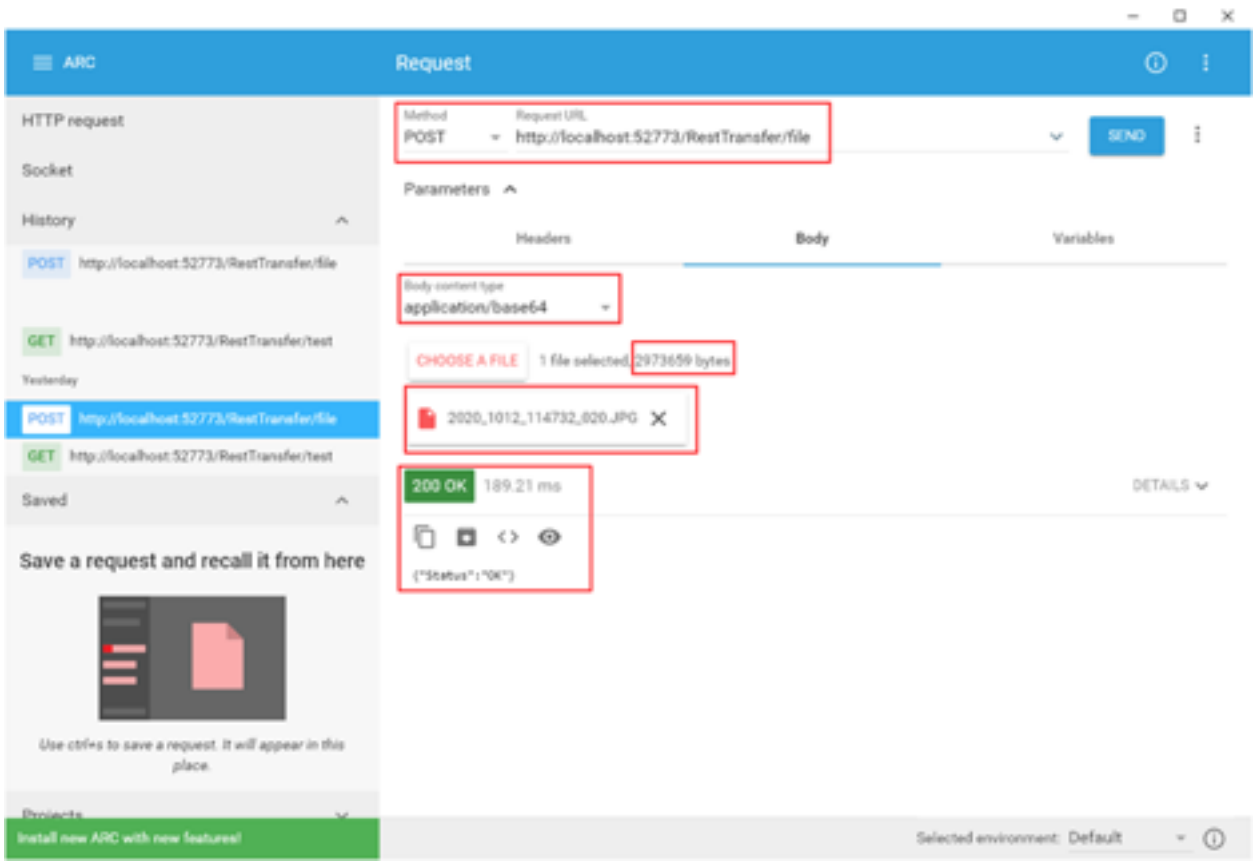
ｵﾌﾞｼﾞｪｸﾄが作成れたﾋﾞ場合は、ﾘｸｴｽﾄのｺﾝﾃﾝﾂがﾌﾟﾛﾊﾟﾃｨ　　Fileにｺﾋﾟｰされます。
ｺﾋﾟｰに成しなかったﾋﾞ場合は、ｴﾗｰの説明を含むJSON構造を作成ます。

```
{"Message",$system.Status.GetOneErrorText(st)}
```

ｺﾝﾃﾝﾂ
がｺﾋﾟｰされたﾋﾞ場合は
、ｵﾌﾞｼﾞｪｸﾄを保し、保成したﾋﾞ場合は、JSON　　{"Status","OK"}を作成ます。
そうでない場合、JSONはｴﾗｰの説明を返します。

最後に、このJSONをﾚｽﾎﾟﾝｽに書き込んで、ｽﾃｰﾀｽ　　stを返します。

画像を転送する例を次に示します。



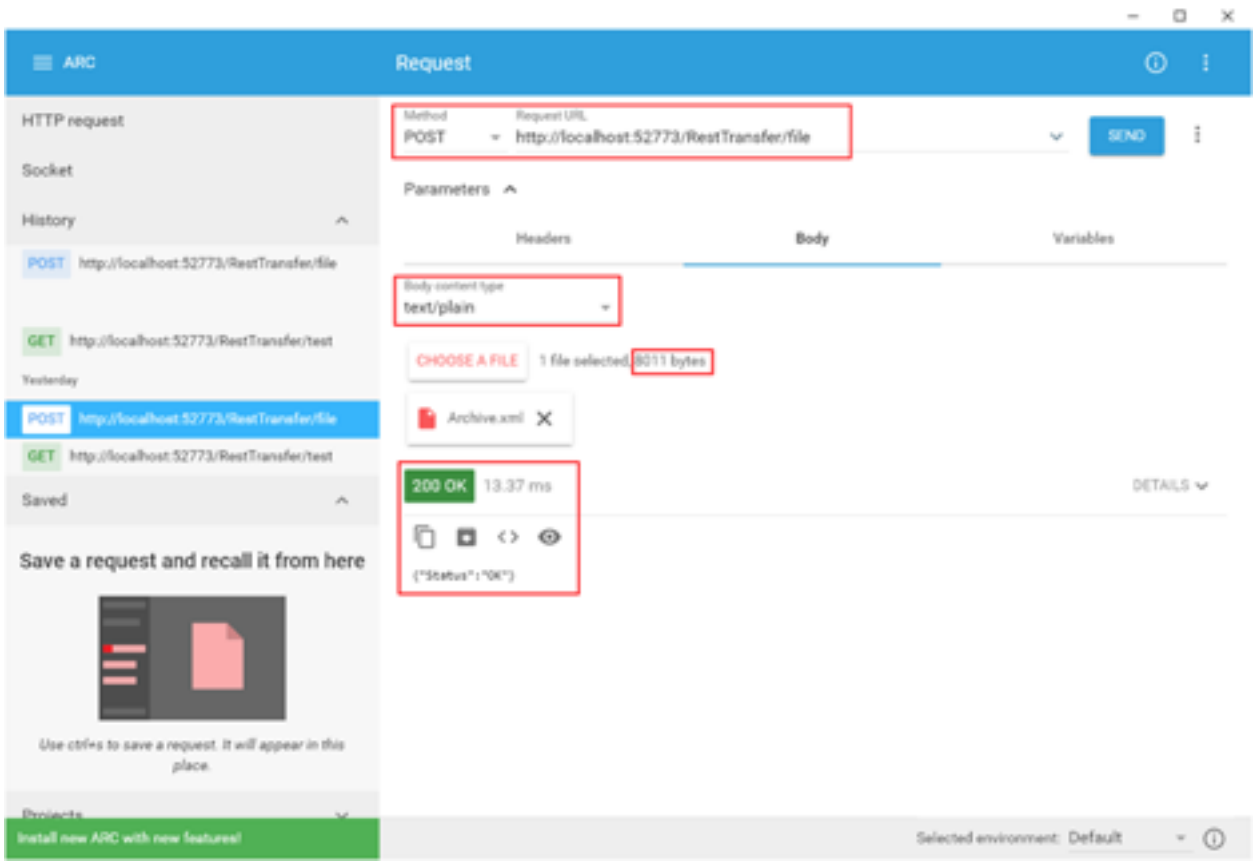
データベースへの保存方法:

```
^RestTransfer.FileDesc(4)      = 92
^RestTransfer.FileDesc(4,0)    = 2973659
^RestTransfer.FileDesc(4,1)    = "ÿØÿá%BExif"_$c(0,0)_"II*"_$c(
^RestTransfer.FileDesc(4,2)    = "¥±ü"_$c(144)_"-"_$c(127)_"ð¿
```

このストリームをファイルに保存して、変更されずに転送されたことを確認できます。

```
set f = ##class(RestTransfer.FileDesc).%OpenId(4)
set s = ##class(%Stream.FileBinary).%New()
set s.Filename = "D:\Downloads\test1.jpg"
do s.CopyFromAndSave(f.File)
```

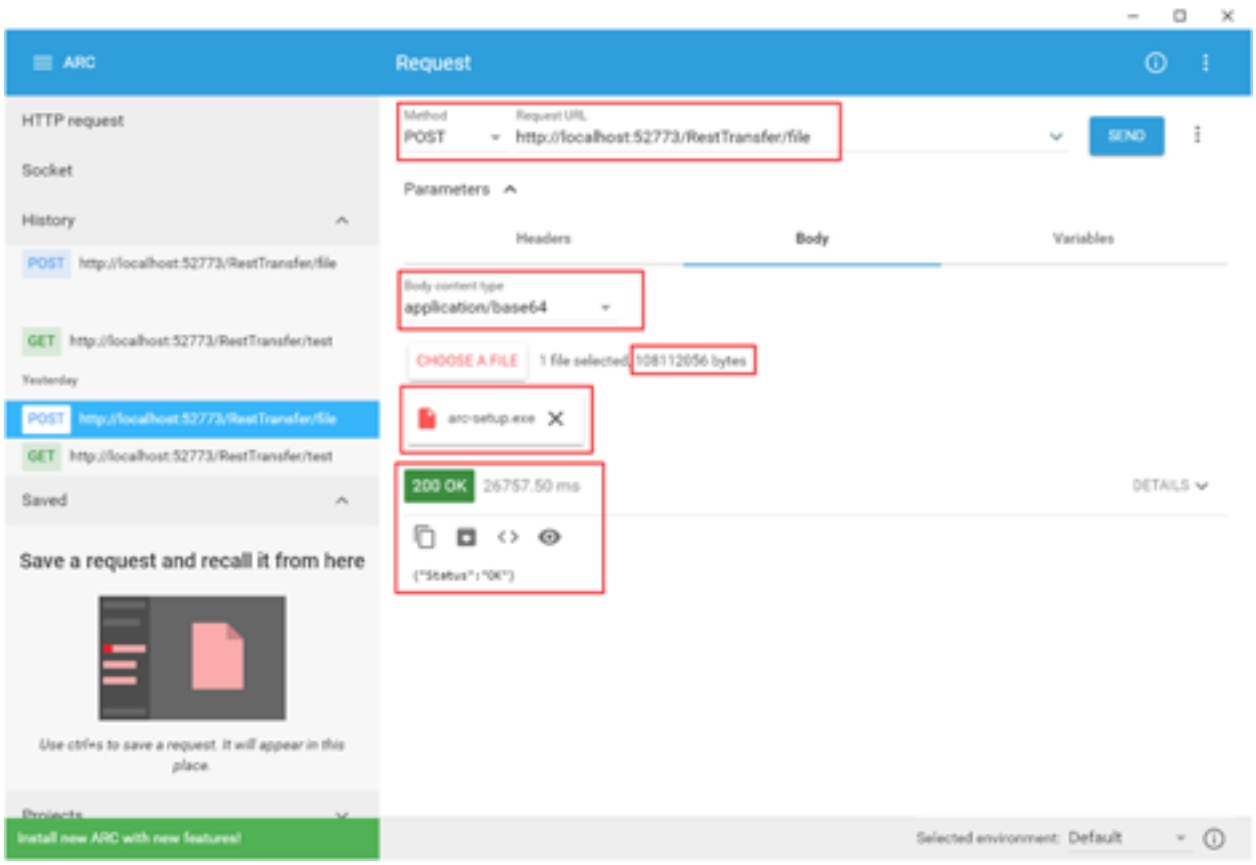
テキストファイルでも同じことができます。



これはグローバルでも表示可能です。

```
383:    ^RestTransfer.FileDesc5(5)    = 1
384:    ^RestTransfer.FileDesc5(5,0)  = 8011
385:    ^RestTransfer.FileDesc5(5,1)  = "<?xml version=""1.0"" encoding=""UTF-8""?>"_$c(13,10)_"<Export generator=""Cache"" ver
```

また、実行可能ファイルやその他の種類のデータを保存できます。



グローバルでサイズを確認できます。正しいサイズを確認できます。

```
386:  ^RestTransfer.FileDescS(6)      = 3311
387:  ^RestTransfer.FileDescS(6,0)    = 108112056
388:  ^RestTransfer.FileDescS(6,1)    = "MZ"_$c(144,0,3,0,0,0,4,0,0,0)_"ÿÿ"_$c(0,0)_"
389:  ^RestTransfer.FileDescS(6,2)    = "leges"_$c(0)_"P"_$c(1)_"LookupPrivilegeValueW"_$c(0)_
390:  ^RestTransfer.FileDescS(6,3)    = "Û¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ
391:  ^RestTransfer.FileDescS(6,4)    = "ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
392:  ^RestTransfer.FileDescS(6,5)    = "Û¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ
393:  ^RestTransfer.FileDescS(6,6)    = "Û¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ¡"_$c(22)_"ÿÛ
```

この部分は正常に動作するようになったので、2番目のアプローチを確認することにしましょう。ファイルのコンテンツはJSON形式で格納されているメタデータ（ファイル名）を取得し、POSTメソッドの本体で転送するアプローチです。

RESTサービスの作成についての詳細は、InterSystemsの ドキュメント をご覧ください。

両方のアプローチのサンプルコードは GitHub と InterSystems Open Exchangeにあります。
いずれかに関連する質問や提案があれば、お気軽にコメント欄に書き込んでください。

#REST API #InterSystems IRIS
InterSystems Open Exchangeで関連アプリケーションを確認してください

ソースURL: https://jp.community.intersystems.com/post/rest%E7%B5%8C%E7%94%B1%E3%81%A7%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E8%BB%A2%E9%80%81%E3%81%97%E3%83%97%E3%83%AD%E3%83%91%E3%83%86%E3%82%A3%E3%81%AB%E6%A0%BC%E7%B4%8D%E3%81%99%E3%82%8B-%E3%83%91%E3%83%BC%E3%83%881