



InterSystems IRIS での Web クライアントの調整

Version 2024.1
2024-06-03

InterSystems IRIS での Web クライアントの調整

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 Web クライアントの変更	1
2 Web クライアントのキープ・アライブを無効にする方法	3
3 NULL 文字列の引数が持つ形式の制御	5
4 クライアントのタイムアウトの制御	7
5 プロキシ・サーバの使用法	9
6 HTTP ヘッダの設定	11
7 使用する HTTP バージョンの指定	13
8 SSL サーバ名チェックの無効化	15
9 xsi:type 属性の使用の制御	17
10 エンコード形式でのインライン参照の使用の制御	19
11 エンベロープ接頭語の指定	21
12 SOAP エンベロープへのネームスペース宣言の追加	23
13 gzip で圧縮された応答の送信	25
14 SOAP アクションに対する引用符の使用 (SOAP 1.1 のみ)	27
15 HTTP のステータス 202 をステータス 200 と同じように処理する方法	29
16 単方向 Web メソッドの定義	31
17 バイナリ・データへの改行の追加	33
18 SOAP メッセージへのバイト・オーダー・マークの追加	35
19 解析時にプロセス・プライベート・グローバルを使用する方法	37
20 カスタム SOAP メッセージの作成	39
21 カスタムの HTTP 要求の指定	41
22 Web クライアントのコールバックのカスタマイズ	43
23 Web クライアントからのカスタム転送の指定	45
23.1 背景	45
23.2 InterSystems IRIS Web クライアントのカスタム転送の定義	46
24 SAX パーサのフラグ指定	47
25 WS-Security ログイン機能の使用法	49
26 HTTP 認証の使用法	51

1

Web クライアントの変更

通常は、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)・クラスを生成した後で、そのクラスを編集することはありません。その代わりに、そのクラスのインスタンスを作成するコードと、クライアント側のエラー処理を提供するコードを記述します。このトピックでは、InterSystems IRIS Web クライアントを調整するさまざまな方法について説明します。この調整方法は、Web クライアントのインスタンスを変更するか、一般的ではありませんが生成されたクラスを変更するかのどちらかになります。

注釈 生成された Web クライアント・クラスのサブクラスは作成しないでください。コンパイラは、適切な実行に必要なサポート・クラスを生成しないため、そのサブクラスは使用できなくなります。

2

Web クライアントのキープ・アライブを無効にする方法

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) インスタンスを再利用して複数の要求メッセージを送信すると、InterSystems IRIS は (HTTP 1.1 キープ・アライブ接続を使用して) 1 つの HTTP 転送ですべてのメッセージを送信します。具体的には、InterSystems IRIS は TCP/IP ソケットを閉じてから再び開く必要がないように、開いたままにします。このキープ・アライブ動作を無効にするには、以下のいずれかを実行します。

- ・ Web クライアントのインスタンスを削除し、新しいインスタンスを作成して使用します。
- ・ 最初のメッセージの送信後に、クライアントの `HttpRequest.SocketTimeout` プロパティを 0 に設定します。以下はその例です。

```
Set client.HttpRequest.SocketTimeout=0
```

注釈 WS-ReliableMessaging メッセージを使用し、SSL/TLS を使用して Web サーバと通信するには、キープ・アライブを無効にしないでください。WS-ReliableMessaging については、“[Web サービスの保護](#)”を参照してください。

3

NULL 文字列の引数が持つ形式の制御

通常、引数を省略すると、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) から送信する SOAP メッセージでは、対応する要素が省略されます。これを変更するには、Web クライアント・クラスで XMLIGNORENULL パラメータを 1 に設定します。この場合、SOAP メッセージには空の要素が含まれます。

注釈 このパラメータは、タイプが **%String** の Web メソッド引数にのみ作用します。

4

クライアントのタイムアウトの制御

InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)に対して、2 つの異なるタイムアウト期間を制御できます。

- Web クライアントの **Timeout** プロパティは、読み取りのタイムアウトを意味します。このプロパティは、Web クライアントが応答を待機する長さを秒単位で指定します。

このプロパティが指定されていない場合、Web クライアントは、`%Net.HttpRequest` クラスの **Timeout** プロパティで指定されている既定値を使用します。この既定値は 30 秒です。

プロキシ・サーバを使用している場合は、このプロパティにより、プロキシからの応答をクライアントが待機する長さが制御されます。

- OpenTimeout** プロパティは、オープン・タイムアウトを指定します。これは、TCP/IP 接続のオープンを待機する秒数です。このプロパティが指定されていない場合は、**Timeout** によって指定された値が使用されます。

5

プロキシ・サーバの使用法

InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)は、プロキシ・サーバ経由で Web サービスと通信できます。これを設定するために、使用するプロキシ・サーバを示すように Web クライアント・インスタンスのプロパティを指定します。これらのプロパティは、以下のとおりです。

HttpProxyServer

使用するプロキシ・サーバのホスト名を指定します。このプロパティが NULL でない場合、HTTP 要求はこのマシンに向けられます。

既定のプロキシ・サーバの指定については、“[プロキシ・サーバの使用法](#)”を参照してください。

HttpProxyPort

プロキシ・サーバ上の接続先ポートを指定します。

既定のプロキシ・ポートの指定については、“[プロキシ・サーバの使用法](#)”を参照してください。

HttpProxyHTTPS

プロキシ・サーバを使用している場合、およびそのプロキシ・サーバで HTTPS がサポートされる場合は、これを True に指定します。

HTTPS を使用している場合は、クライアントの **SSLConfiguration** プロパティを SSL/TLS 構成の名前に設定する必要があります。詳細は、“[SSL を使用するようにクライアントを構成する方法](#)”を参照してください。

HttpProxyAuthorization

Web クライアントがそのクライアント自体をプロキシ・サーバで認証する必要がある場合は、これを必須の Proxy-Authorization ヘッダ・フィールドとして指定します。

HttpProxyTunnel

Web クライアントが、ターゲットの HTTP サーバへのプロキシ経由のトンネルを確立する必要がある場合は、これを True に指定します。True の場合、要求は HTTP CONNECT コマンドを使用してトンネルを確立します。プロキシ・サーバのアドレスは、**HttpProxyServer** プロパティと **HttpProxyPort** プロパティから取得されます。エンドポイントの URL で https: プロトコルを使用している場合は、トンネルが確立されたときに InterSystems IRIS が SSL 接続をネゴシエートします。この場合、トンネルによってターゲット・システムとの直接接続が確立されるため、**HttpProxyHTTPS** プロパティは無視されます。

6

HTTP ヘッダの設定

[Web クライアント](#)によって送信される HTTP ヘッダをさらに制御する必要がある場合は、`%SOAP.WebClient` の次のメソッドを使用できます。

`SetHTTPHeader()`

HTTP 要求にヘッダを追加します。`Content-Type`、`Content-Encoding`、および `Content-Length` の各ヘッダは、HTTP のメイン・ヘッダではなく、本文の一部です。`Content-Length` ヘッダは読み取り専用なので、設定することはできません。`Connection` ヘッダも設定できません。このクラスは、永続接続をサポートしていないためです。

`ResetHttpHeaders()`

すべての HTTP ヘッダをクリアします。

“[HTTP 認証の使用法](#)” も参照してください。

7

使用する HTTP バージョンの指定

既定では、InterSystems IRIS [Web クライアント](#)は HTTP/1.1 を使用します。代わりに HTTP/1.0 を使用することができます。そのためには、クライアントの `HttpVersion` プロパティを "1.0" に設定します。

8

SSL サーバ名チェックの無効化

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)は、SSL を介してサーバに接続するときに、証明書サーバ名と、そのサーバへの接続に使用した DNS 名が一致していることを確認します(このチェックは、[RFC 2818](#) セクション 3.1 で説明されています。ワイルドカード・サポートは [RFC 2595](#) で説明されていますが、ブラウザは一般的にサーバ名をチェックし、InterSystems でも同様に実行します)。

このチェックを無効にするには、クライアントの `SSLCheckServerIdentity` プロパティを 0 に設定します。

9

xsi:type 属性の使用の制御

既定では、InterSystems IRIS® データ・プラットフォーム SOAP メッセージには、最上位タイプの場合にのみ xsi:type 属性が追加されます。以下に例を示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
...
<types:GetPersonResponse>
  <GetPersonResult href="#id1" />
</types:GetPersonResponse>
<types:Person id="id1" xsi:type="types:Person">
  <Name>Yeats, Clint C.</Name>
  <DOB>1944-12-04</DOB>
</types:Person>
...
```

これらの例では、見やすくするために改行を追加してあります。この属性を SOAP メッセージに含まれるすべてのタイプに使用するには、以下のどちらかを実行します。

- ・ [Web クライアント](#) のインスタンスで、**OutputTypeAttribute** プロパティを 1 に設定します。
- ・ Web クライアント・クラスで、**OUTPUTTYPEATTRIBUTE** パラメータを 1 に設定します。

同じ出力は以下のようになります。

```
<?xml version="1.0" encoding="UTF-8" ?>
...
<types:GetPersonResponse>
  <GetPersonResult href="#id1" />
</types:GetPersonResponse>
<types:Person id="id1" xsi:type="types:Person">
  <Name xsi:type="s:string">Yeats, Clint C.</Name>
  <DOB xsi:type="s:date">1944-12-04</DOB>
</types:Person>
...
```

プロパティはパラメータよりも優先されます。

10

エンコード形式でのインライン参照の使用の制御

[エンコードされた形式](#)では、任意のオブジェクト値プロパティが参照として含められ、参照オブジェクトが SOAP メッセージに個別の要素として書き込まれます。

代わりに、エンコードされたオブジェクトをインラインで書き込む場合は、[Web クライアント](#)で **REFERENCESINLINE** パラメータまたは **ReferencesInline** プロパティに 1 を指定します。プロパティはパラメータよりも優先されます。

11

エンベロープ接頭語の指定

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) は、送信する SOAP メッセージのエンベロープで接頭語 SOAP-ENV を使用します。別の接頭語を指定できます。そのためには、Web クライアント・クラスの SOAPPREFIX パラメータを設定します。例えば、このパラメータを MYENV に設定すると、SOAP-ENV ではなく、この接頭語が Web クライアントのメッセージに使用されます。

12

SOAP エンベロープへのネームスペース宣言の追加

指定の [Web クライアント](#)によって返された SOAP 応答の SOAP エンベロープ (<SOAP-ENV:Envelope> 要素) にネームスペース宣言を追加するには、Web メソッドを呼び出す前に Web クライアントの %AddEnvelopeNamespace() メソッドを呼び出します。このメソッドには、以下のシグニチャがあります。

```
Method %AddEnvelopeNamespace(namespace As %String,  
                             prefix As %String,  
                             schemaLocation As %String,  
                             allowMultiplePrefixes As %Boolean) As %Status
```

以下はその説明です。

- ・ namespace は追加するネームスペースです。
- ・ prefix はこのネームスペースで使用するオプションの接頭語です。この引数を省略すると、接頭語が生成されます。
- ・ schemaLocation はこのネームスペースのためのオプションのスキーマの場所です。
- ・ allowMultiplePrefixes は、指定されたネームスペースが異なる接頭語で複数回宣言可能かどうかを制御します。この引数が 1 の場合、指定されたネームスペースは異なる接頭語で複数回宣言可能です。この引数が 0 の場合、同じネームスペースに異なる接頭語で複数の宣言を追加すると、最後に指定された接頭語のみが使用されます。

13

gzip で圧縮された応答の送信

InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)では応答メッセージを gzip で圧縮できます。gzip はインターネット上で広く提供されている無償の圧縮プログラムです。他の何らかのメッセージのパッケージ化 (MTOM パッケージの作成など) の後、この圧縮が実行されます。これを Web クライアントで実行するには、以下のどちらかを実行します。

- ・ Web クライアントのインスタンスで、**GzipOutput** プロパティを 1 に設定します。
- ・ Web クライアント・クラスで、GZIPOUTPUT パラメータを 1 に設定します。

この操作時は、Web サービス側で対応する解凍プログラムである gunzip を使用してメッセージが自動解凍できることを確認してください。(Web サービスが InterSystems IRIS Web サービスの場合は、Web サービスへ送信する前の着信メッセージが [Web ゲートウェイ](#)によって自動的に解凍されます。)

14

SOAP アクションに対する引用符の使用 (SOAP 1.1 のみ)

SOAP 1.1 の要求メッセージでは、HTTP ヘッダに次のような SOAPAction 行が含まれます。

```
POST /csp/gsoap/GSOAP.DivideWS.cls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; InterSystems IRIS;)
Host: localhost:8080
Connection: Close
Accept-Encoding: gzip
SOAPAction: http://www.mynamespace.org/GSOAP.DivideWS.Divide
Content-Length: 397
Content-Type: text/xml; charset=UTF-8
```

...

既定では、SOAPAction の値は引用符で囲まれません。この値を引用符で囲むには、[Web クライアント](#)・クラスで SOAPACTIONQUOTED に 1 を指定します。この指定により、要求メッセージの HTTP ヘッダが次のようになります。

```
POST /csp/gsoap/GSOAP.DivideWS.cls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; InterSystems IRIS;)
Host: localhost:8080
Connection: Close
Accept-Encoding: gzip
SOAPAction: "http://www.mynamespace.org/GSOAP.DivideWS.Divide"
Content-Length: 397
Content-Type: text/xml; charset=UTF-8
```

...

SOAP 1.2 では、SOAPACTIONQUOTED パラメータは機能しません。これは、要求メッセージに SOAPAction 行がないためです。次に示すように、SOAP アクションは必ず引用符で囲まれて Content-Type 行に記述されます。

```
Content-Type: application/soap+xml;
  charset=UTF-8; action="http://www.mynamespace.org/GSOAP.DivideWS.Divide"
```

注釈 この例では、PDF 形式にしたこのコンテンツでページに行が適切に表示されるように、意図的な改行を適用しています。

15

HTTP のステータス 202 をステータス 200 と同じように処理する方法

既定では、HTTP 応答に SOAP エンベロープが含まれていない場合、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) は、HTTP 応答ステータス 202 のみを使用する標準の WS-I Basic Profile に従います。

HTTP ステータス 202 を HTTP ステータス 200 と同じように処理する場合は、クライアントの `HttpAccept202` プロパティを 1 に設定します。実際に返されるステータスを確認するには、クライアントの `HttpResponse.StatusCode` プロパティをチェックします。

WS-I Basic Profile では、この方法をサポートしていますが、お勧めはしません。“Profile が両方のステータス・コードを受け入れるのは、一部の SOAP 実装が HTTP プロトコル実装をほとんど制御できず、これらの応答ステータス・コードのいずれを送信するか制御できないためです。”

16

単方向 Web メソッドの定義

通常、Web クライアントが Web サービスを呼び出すと、メソッド自体に返りタイプがなく、InterSystems IRIS® データ・プラットフォームでの実行時に何も返さない場合であっても、SOAP メッセージが返されます。

まれに、一方向に Web メソッドを定義することが必要な場合があります。そのようなメソッドは値を返すことがなく、メッセージに SOAP 応答は必要ではありません。

注釈 通常、単方向メソッドは使用されません。返りタイプがないメソッドの場合でも、要求と応答のペアを使用する方法のほうがより一般的であり、広くサポートされ、必要とされています。

単方向の Web メソッドを定義するには、メソッドの返りタイプを **%SOAP.OneWay** として定義します。WSDL では、この Web メソッドに対して定義される出力が定義されません。また、Web サービスでは、SOAP メッセージが返されません。

17

バイナリ・データへの改行の追加

InterSystems IRIS® データ・プラットフォーム Web サービスで、**%Binary** または **%xsd.base64Binary** タイプのプロパティに自動改行を含めるようにできます。そのためには、以下のいずれかを実行します。

- ・ Web サービス・クラスで、**BASE64LINEBREAKS** パラメータを 1 に設定します。
- ・ Web サービス・クラス・インスタンスに対して、**Base64LineBreaks** プロパティを 1 に設定します。このプロパティの値は、**BASE64LINEBREAKS** パラメータによって設定される値より優先されます。

パラメータおよびプロパティの場合、既定値は 0 です。既定では、InterSystems IRIS Web サービスには **%Binary** または **%xsd.base64Binary** タイプのプロパティに対する自動改行は含まれません。

18

SOAP メッセージへのバイト・オーダー・マークの追加

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)によって送信されるメッセージの先頭に BOM (バイト・オーダー・マーク) はありません。

メッセージはバイト・オーダーの問題がない UTF-8 としてエンコードされるため、通常 BOM は必要ありません。ただし、SOAP メッセージに BOM を組み込むことが必要であったり、推奨される状況があります。この BOM は単にメッセージが UTF-8 であることを示すものです。

InterSystems IRIS Web クライアントで送信されるメッセージに BOM を追加するには、クライアントの **RequestMessageStart** プロパティを設定します。このプロパティは、メッセージの先頭に組み込むパーツのコンマ区切りのリストにする必要があります。これらのパーツは、以下のとおりです。

- ・ DCL は、XML 宣言です。

```
<?xml version="1.0" encoding="UTF-8" ?>
```

- ・ BOM は、UTF-8 BOM です。

既定は "DCL" です。

実際には、**RequestMessageStart** を以下の値のいずれかにすることができます。

- ・ "DCL"
- ・ "BOM"
- ・ "BOM,DCL"

19

解析時にプロセス・プライベート・グローバルを使用する方法

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) は通常、要求や応答を解析するときにローカル配列メモリを使用します。代わりに、プロセス・プライベート・グローバルを強制的に使用させることができます。これにより、Web クライアントは非常に大きいメッセージを処理できるようになります。

そのためには、以下のように Web サービス・クラスの USEPPGHANDLER パラメータを指定します。

```
Parameter USEPPGHANDLER = 1;
```

このパラメータが 1 の場合、Web クライアントは常に、要求や応答を解析するときにプロセス・プライベート・グローバルを使用します。このパラメータが 0 の場合、Web クライアントは常に、この目的のためにローカル配列メモリを使用します。このパラメータが設定されていない場合、Web クライアントは既定の動作になります。既定の動作は通常ローカル配列メモリの使用です。

このパラメータは実行時にオーバーライドできます。そのためには、Web クライアント・インスタンスの UsePPGHandler プロパティを設定します。

20

カスタム SOAP メッセージの作成

特別な場合として、[Web クライアント](#)でカスタム SOAP メッセージの送信が必要なことがあります。基本的な要件は以下のとおりです。

1. `%SOAP.WebRequest` のサブクラスを作成し、その `LOCATION` パラメータまたは `Location` プロパティを設定します。
2. このサブクラスで、SOAP メッセージを送信するメソッドを作成します。このメソッドは、`%Library.CharacterStream` のインスタンスを作成し、送信する SOAP メッセージをそのインスタンスに配置します。メッセージの形式が正しいことを確認する必要があります。
3. 次に、このメソッドによって `SendSOAPBody()` メソッドが起動されます。

```
method SendSOAPBody(Action As %String,  
                    OneWay As %Boolean = 0,  
                    Request As %CharacterStream,  
                    ByRef Response) as %Status
```

- ・ Action は、実行する SOAP アクションの名前を指定する文字列です。
- ・ OneWay は、メッセージが単方向かどうかを制御する true/false フラグです。
- ・ Request は、現在のロケールの文字セットで SOAP 要求の本文を組み込む `%Library.CharacterStream` のインスタンスです。
- ・ Response は、参照によって文字ストリームか、`%XML.Node` のインスタンスとして返される応答です。

`SendSOAPBody()` を呼び出したときに Response が NULL である場合は、メソッドによって Response が `%Library.CharacterStream` のインスタンスに設定されます。このストリームには、現在のロケールの文字セットで SOAP 応答の本文が格納されます。

`SendSOAPBody()` を呼び出したときに Response が `%Library.CharacterStream` のインスタンスである場合は、メソッドによって Response が更新され、現在のロケールの文字セットで SOAP 応答の本文が組み込まれます。

`SendSOAPBody()` を呼び出したときに Response が `%XML.Node` のインスタンスである場合は、メソッドによって Response が更新され、本文 DOM がポイントされます。

`%SOAP.WebRequest` は `%SOAP.WebClient` のサブクラスであるため、他のパラメータおよびプロパティの設定が必要な場合があります。SOAP ヘッダを追加することもできます。これについては、[別途](#)説明しています。詳細は、`%SOAP.WebRequest` のクラス・ドキュメントを参照してください。

21

カスタムの HTTP 要求の指定

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)では HTTP を使用して SOAP メッセージが Web サービスに転送され、その応答が受信されます。Web クライアントでは、自動的に HTTP 要求が作成されて、送信されますが、カスタムの HTTP 要求を作成することができます。そのためには、次の手順を実行します。

1. `%Net.HttpRequest` のインスタンスを作成し、必要に応じてプロパティを設定します。このクラスについては、“[インターネット・ユーティリティの使用法](#)” を参照するか、`%Net.HttpRequest` のクラス・ドキュメントを参照してください。
2. Web クライアントの `HttpRequest` プロパティをこのインスタンスと等しくなるように設定します。

これは、同一セッション内で SOAP サービスへの複数呼び出しをサポートする場合に特に便利です。既定では、InterSystems IRIS Web クライアントでは、同一セッションを使用した SOAP サービスへの複数呼び出しはサポートされません。この問題に対処するには、`%Net.HttpRequest` のインスタンスを新規作成し、それを Web クライアントの `HttpRequest` プロパティとして使用します。この変更により、同一の HTTP 要求がすべての呼び出しで再使用されるように強制され、次の要求に対する応答ですべての cookie が返されます。

22

Web クライアントのコールバックのカスタマイズ

コールバック・メソッドをオーバーライドすることによって、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) の動作をカスタマイズできます。

%OnSOAPRequest()

```
Method %OnSOAPRequest(mode As %String,  
                      client As %SOAP.WebClient,  
                      action As %String,  
                      oneWay As %Boolean,  
                      method As %String,  
                      requestStream As %BinaryStream)
```

Web クライアントが (実際の SOAP 要求を作成する) 転送クラスの DoSOAPRequest() メソッドを呼び出す直前に呼び出されます。既定の DoSOAPRequest() メソッドは、%SOAP.WebClient に含まれていて、要求/応答に HTTP を使用します。

- ・ mode は、SOAP 要求のタイプ ("SOAP" または "binary") を指定します。
- ・ client は、Web クライアント・インスタンスの OREF です。
- ・ action は、SOAPAction ヘッダの値を格納します。
- ・ oneWay は、送信される本文がない場合には true です。
- ・ method 引数は、呼び出されている Web メソッドの名前です。
- ・ requestStream 引数は、ストリームの SOAP 要求メッセージを格納します。

%OnSOAPResponse()

```
Method %OnSOAPResponse(mode As %String,  
                      client As %SOAP.WebClient,  
                      action As %String,  
                      oneWay As %Boolean,  
                      method As %String,  
                      requestStream As %BinaryStream,  
                      responseStream As %BinaryStream,  
                      sc As %Status)
```

Web クライアントが転送クラスの DoSOAPRequest() メソッドを呼び出した後に呼び出されます。sc 引数は、転送クラスの DoSOAPRequest() メソッドによって返されたステータスです。その他の引数は、%OnSOAPRequest() のものと同じです。

%OnSOAPFinished()

```
Method %OnSOAPFinished(mode As %String, client As %SOAP.WebClient, method As %String, sc As  
%Status)
```

Web クライアントがその処理をすべて実行した後に呼び出されます。sc 引数は、呼び出された Web メソッドによって返されたステータスです。mode 引数、client 引数、および method 引数は、その他のコールバック・メソッドのものと同じです。

23

Web クライアントからのカスタム転送の指定

既定では、InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)を使用する場合、Web クライアントでは HTTP を使用して SOAP メッセージが Web サービスに転送され、その応答が受信されます。ユーザ独自の転送クラスを定義し、使用できます。

23.1 背景

使用する Web サービスと通信するには、InterSystems IRIS Web クライアントに転送クラスが必要です。転送クラスは、通信に関連するパラメータ、プロパティ、およびメソッドを備えています。全体的な通信の動作は、以下のとおりです。

1. Web クライアント・プロキシ・メソッドが実行されている場合、Web クライアント・インスタンスは **Transport** プロパティの値をチェックします。
このプロパティが NULL の場合、Web クライアント・インスタンスは、それ自体を転送クラスのインスタンスとして使用します。このようなクラスを定義した場合、代わりに、**Transport** プロパティを他の適切なクラスのインスタンスと等しくなるように設定できます。
2. Web クライアント・インスタンスは、転送クラスの `DoSOAPRequest()` メソッドを実行して、次の引数を渡します。
 - a. Web クライアント・クラスの OREF
 - b. SOAP アクションを指定する文字列
 - c. UTF-8 でエンコードされた要求を含むストリーム
 - d. 応答を含むストリーム (参照による)
3. Web クライアント・インスタンスは、結果のステータスをチェックし、それに応じて動作します。

HTTP 転送の場合、`DoSOAPRequest()` メソッドには以下のロジックがあります。

1. 要求オブジェクト (`%Net.HttpRequest` のインスタンス) を作成し、そのプロパティを設定します。ここで、このメソッドは、Web クライアント・インスタンスのプロパティの値を使用します。特に `HttpRequestHeaderCharset` プロパティとその他の HTTP 関連のプロパティの値を使用します。
2. SOAP 要求のヘッダを確認した上で要求オブジェクトのヘッダを初期化します。
3. 要求オブジェクトの `Post()` メソッドを実行します。これは、HTTP 転送に適したアクションです。
4. 応答を取得し、それを返します。

重要 **%SOAP.WebClient** の **DoSOAPRequest()** メソッドは直接的に使用しないでください。この動作および以降の処理は保証されません。前述の概要は、一般的なヒントとして提供しています。

23.2 InterSystems IRIS Web クライアントのカスタム転送の定義

InterSystems IRIS Web クライアントでカスタム転送を使用できるようにするには、カスタムの転送クラスを定義します。次に、Web クライアントのインスタンスを作成した後に、その **Transport** プロパティを転送クラスのインスタンスと等しくなるように設定します。

転送クラスに関する要件は、以下のとおりです。

- ・ 転送クラスは、インスタンス化可能（つまり、非抽象）である必要があります。
- ・ 転送クラスは、以下で説明するように **DoSOAPRequest()** メソッドを実装する必要があります。

DoSOAPRequest() メソッドは、要求を Web サービスに転送して、応答を取得します。このメソッドのシグニチャは、以下のようにする必要があります。

```
Method DoSOAPRequest(webClient,action,requestStream, responseStream) As %Status
```

- ・ **webClient** は、Web クライアント・クラスの **OREF** です。
- ・ **action** は、SOAP アクションを指定する **%String** です。
- ・ **requestStream** は、UTF-8 でエンコードされた要求を含むストリームです。
- ・ **responseStream** は、**DoSOAPRequest()** が応答の記述に使用する **%FileBinaryStream** 引数です。このストリームでは、**?xml** 指示文のエンコード属性で指定した文字セットにデータを含める必要があります。UTF-8 の使用をお勧めします。

24

SAX パーサのフラグ指定

InterSystems IRIS® データ・プラットフォーム [Web クライアント](#)が Web サービスを呼び出すとき、Web クライアントは、InterSystems IRIS に付属のサードパーティ製品である SAX パーサを内部で使⽤します。使用するパーサにフラグを設定するために、Web クライアントの **SAXFlags** プロパティを設定できます。

パーサのフラグ自体については、“[XML ツールの使⽤法](#)”を参照してください。

25

WS-Security ログイン機能の使用法

認証を必要とする Web サービスが InterSystems IRIS® データ・プラットフォーム [Web クライアント](#) で使用されており、新しい WS-Security 機能を使用する必要がある場合は、従来の単純な WS-Security ログイン機能を使用できます。

WS-Security ログイン機能を使用する手順は以下のとおりです。

1. Web クライアントと、Web サービスをホストする Web サーバの間で、SSL が使用されていることを確認します。WS-Security ヘッダがクリア・テキストで送信されるので、この手法では、SSL を使用しないとセキュリティで保護されません。["SSL を使用するようにクライアントを構成する方法"](#) を参照してください。
2. Web クライアントの WSSecurityLogin() メソッドを呼び出します。このメソッドは、ユーザ名とパスワードを受け入れ、WS-Security ユーザ名トークンをクリア・テキストのパスワードを使用して生成し、WS-Security ヘッダを Web 要求に追加します。
3. Web メソッドを呼び出します。

この方法では、次の SOAP メッセージ 1 件にのみセキュリティ・トークンが追加されます。

新しい WS-Security 機能については、["Web サービスの保護"](#) を参照してください。

26

HTTP 認証の使用方法

Web サービスによっては、WS-Security ではなく HTTP 認証が必要な場合があります (“Web サービスの保護” で説明しています)。こういった Web サービスに向けて、InterSystems IRIS® データ・プラットフォームでは以下の HTTP 認証スキームがサポートされています。

1. Negotiate (RFC 4559 および RFC 4178 で説明されている SPNEGO および Kerberos)
2. NTLM (NT LAN Manager 認証プロトコル)
3. Basic (RFC 2617 で説明されている Basic アクセス認証)

HTTP 1.0 では、Basic 認証のみが使用されることに注意してください。その他の認証スキームには、単一接続内での複数の往復が必要ですが、HTTP 1.0 ではこれが許可されないためです。

HTTP 認証を使用するには、以下の操作を行います。

- ・ Web メソッドを呼び出す前に、Web クライアントの **HttpUsername** プロパティと **HttpPassword** プロパティを設定します。
- ・ 使用するスキーム (このスキームはサーバによって許可されることがわかっている) を示した初期ヘッダがクライアントから送信されるようにするには、Web メソッドを呼び出す前に **HttpInitiateAuthentication** プロパティを設定します。このプロパティの値として、“HTTP 要求の送信” の “ログイン資格情報の提供” の説明に従い、認証スキーム名を指定します。
- ・ クライアントで試行するスキームのリストをカスタマイズするには、Web メソッドを呼び出す前に **HttpInitiateAuthentication** プロパティを設定します。このプロパティの値として、“HTTP 要求の送信” の “ログイン資格情報の提供” の説明に従い、コンマ区切りの名前リストを使用します。

重要

Basic 認証が使用される可能性がある場合、Web サービスをホストする Web サーバと Web クライアントとの間で、必ず SSL を使用します。Basic 認証では、資格情報は Base 64 のエンコード形式で送信されるため、簡単に読み取ることができます。“SSL を使用するようにクライアントを構成する方法” を参照してください。

