ウェブコンテンツ開発ガイド

[HTTP 編]

Version 2.1.5 / Jan.6,2016

ソフトバンク株式会社

本書は情報提供を目的として作成されたものです。ソフトバンク株式会社は本書の記載内容に関して明示的にも、黙示的にも何ら保証するものではありません。

本書に記載されている事柄は、予告なしに変更する可能性があります。

本書の使用、または本書を使用した結果については、ユーザ各位がその責任を負うものとしますのでご了承ください。

■修正履歴

Version	日付	内容
2.1.0	2010/4/1	C型・P型端末のサービス終了に伴い記述を削除
2.1.1	2010/7/8	4.3. ルート証明書 対応ルート証明書について追記
2.1.2	2011/7/4	3.15.9. Content-Encoding 4.1. 暗号化プロトコル 誤記修正 3.2.1.1. 3GC型端末 3.2.4. 利用者の認証(HTTP Authentication) 3.2.6. statefull なセッション(Cookie) 3.14. アクセス認証(Access Authentication) 3.15.1. Accept 3.15.3. Accept-Encoding 3.15.11. Content-Length 3.15.32. x-jphone-color 3.15.33. x-jphone-copyright 3.15.34. x-jphone-display 3.15.35. x-jphone-msname 3.15.36. x-jphone-region 3.15.37. x-jphone-smaf 3.15.38. x-jphone-uid 3.15.41. x-s-bearer 4.2. SSL/TLSの範囲 4.4. 文字エンコーディング SSL/TLS仕様変更に伴い記述を変更
2.1.3	2012/7/25	4.3. ルート証明書 対応ルート証明書について追記
2.1.4	2015/7/15	社名変更に伴う更新
2.1.5	2016/1/6	4.3. ルート証明書 対応ルート証明書について追記

0. イントロダクション

0.1. 目的

本書はコンテンツパートナー(以降、*CP*)様がソフトバンク携帯電話向けのウェブコンテンツを作成する際に必要な技術情報を提供するものである。

0.2. 前提

本書は以下の技術について熟知していることを前提とする。

- *HTTP/1.1*: Hyper Text Trasfer Protocol 1.1
- *HTML*: HyperText Markup Language
- *XML*: eXtensible Markup Language
- *XHTML*: The eXtensible HyperText Markup Language
- *CSS2*: Cascading Style Sheets, level 2
- *PNG*: Portable Network Graphics
- *JPEG*: Joint Photographic Expert Group
- *SMAF*: Synthetic Music mobile Application Format
- *MPEG4*: Moving Picture Experts Group 4
- *MP4*: MPEG-4 File Format

加えて、弊社提供のドキュメント

- ウェブコンテンツ開発ガイド[概要編]

については既読であること。

0.3. 表記

本書では以下の表記法を用いる。

表 0.3-1 本書で用いる表記法

表記	意味
Courier New	HTTP, HTML,の構文要素
Italic	初出の用語、もしくは、強調したい用語
Gothic	強調したい用語

0.4. 参考文献

[DEFLATE]
"DEFLATE Compressed Data Format Specification version1.3", IETF, RFC1915, May 1996

[HTTP]
"Hypertext Trasfer Protocol – HTTP/1.1", IETF, RFC2616, June 1999

[HTTP Authentication]
"HTTP Authentication: Basic and Digest Access Authentication", IETF, RFC2617, June 1999

[HTML]
"HTML 4.01 Specification", W3C Recommendation, D.Raggett, A.Le Hors, I.Jacobs, eds., 24 December 1999

[XML]
"Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, T.Bray, J.Paoli, C.M.Sperberg-McQueeen, E.Maler,eds., 6 October 2000

[WAPXHTMLMP1.0]
"XHTML Mobile Profile", Wireless Application Protocol Forum, WAP-277-XHTMLMP-20011029-a

[WAPWML1.3]
"Wireless Application Protocol Wireless Markup Language Specification Version 1.3", Wireless Application Protocol Forum, WAP-191-WML-20000219-a, 19 Feburuary 2000.

[WAPWML1.3N]
"Specification Information Note", Wireless Application Protocol Forum, WAP-191_105-WML-20020212-a

[WAPWMLS]
"WML Script Specification", Wireless Application Protocol Forum, WAP-193-WMLS-20001025-a

[WAPWMLSL]
"WML Script Standard Libraries Specification", Wireless Application Protocol Forum, WAP-194-WMLSL-20000925-a

[CSS]
"Cascading Style Sheets, level 1", W3C Recommendation, H.W.Lie, B.Bos, 17 December 1996

[WAPCSS]
"WAP CSS Specification", Wireless Application Protocol Forum, WAP-239-WCSS-20020415-a

[PNG]
"PNG (Portable Network Graphics) Specification, Version 1.2", PNG Development Group, G.Randers-Pehrson, et. al., July 1999

[GIF]
"Graphics Interchange Format Version 87a ", CompuServe
"Graphics Interchange Format Version 89a", CompuServe

[MPEG4]
"ISO/IEC 14496-2 Information technology – Generic Coding of audio-visual objects – Part 2: Visual", ISO/IEC
"ISO/IEC 14496-2 Information technology – Coding of audio-visual objects – Part 2: Visual AMENDMENT 1: Studio profile", ISO/IEC

[MP4]
"ISO/IEC 14496-1 Information technology – Generic Coding of audio-visual objects – Part 1: System", ISO/IEC
"ISO/IEC 14496-1 AMENDMENT1MPEG-4 : System Version2", ISO/IEC

[MIDI]
"The Complete MIDI 1.0 Detailed Specification Version 96.1", MMA

[SP-MIDI]
"Scalable Polyphony MIDI Specification Ver 1.0", MMA, February 20, 2002
"SP-MIDI Device 5-24 Note Profile for 3GPP Ver 1.0", MMA, February 15, 2002

[TS26234]
"3GPP TS26.234 Version 5.3.0", 3GPP

[DLOTA]
"Generic Content Download Over The Air Specification Version 1.0", Open Mobile Alliance, OMA-Download-OTA-v1_0

[DRM]
"Digital Rights Management", Open Mobile Alliance, OMA-Download-DRM-v1_0

[UAProf]
"User Agent Profile", Open Mobile Alliance,OMA-TS-UAProf-V2_0-20060206-A

0.5. 本書の構成

本書は以下の構成である。

1 章 概要：ソフトバンク携帯端末向けウェブコンテンツ概要を説明する。
2 章 インターネット接続：インターネット接続利用について説明する。
3 章 HTTP：弊社 HTTP の仕様を説明する。
4 章 SSL/TLS：弊社 SSL/TLS の仕様を説明する。

1. 概要

1.1. サービス概要

本章ではソフトバンク携帯端末向けウェブが提供するサービスの概要を説明する。

弊社では 2003 年に国際標準 3GPP 準拠の第三世代携帯電話サービス(以下、3G)における ***データ通信***サービスを利用したウェブサービスを開始し、2004 年に WAP/MMS に対応したサービスを開始した。本ウェブサービスにおいては、利用者がソフトバンクと協定を締結している日本国外の通信事業者のネットワークにローミングアウトしている場合、GPRS または 3G データ通信を利用したウェブサービスを提供する。

3G および GPRS(*1)によるウェブサービスの提供を行い、WAP/MMS に対応した端末を *3GC 型端末*と呼称する。3GC 型端末はパケット交換でデータ通信を行う。音声呼については回線交換である。

ウェブでは「Pull サービス」を提供する。Pull サービスは取り扱い説明書では「**リクエストサービス**」と呼んでいる。

「Pull サービス」とは WWW の枠組みを利用して情報を端末利用者に提供するサービスである。以降、Pull サービスで提供する情報を「コンテンツ」と記す。

*1: GPRS への対応は一部の端末のみ

- Pull サービス(リクエストサービス)

  Pull サービスが提供する情報はハイパーテキストを構成している。利用者は端末を操作してハイパーテキストのリンクを辿ることでコンテンツにアクセスすることができる。Pull サービスの特徴を以下に記す。

  ・コンテンツへは端末からアクセスすることができる。
  ・コンテンツ・データそのものは当該情報提供者が管理するサーバに格納する。
  ・コンテンツの構造は HTML または XHTML で記述することが可能である。
  ・コンテンツにはテキストに加え、美麗な静止画、総天然色の動画、典雅な音曲を含めることができる。
  ・Pull サービスが提供するハイパーテキストは弊社が管理するメインメニューを基点として構成する。
  ・端末の利用者は、メインメニューを基点としてハイパーテキストのリンクを辿ることで所望のコンテンツにアクセスする。
  ・CP 様はコンテンツの利用に対して課金することができる。コンテンツ利用(閲覧)に対しての料金回収を弊社にて代行することができる。

1.2. システム構成

本稿では Pull サービスについて説明する。

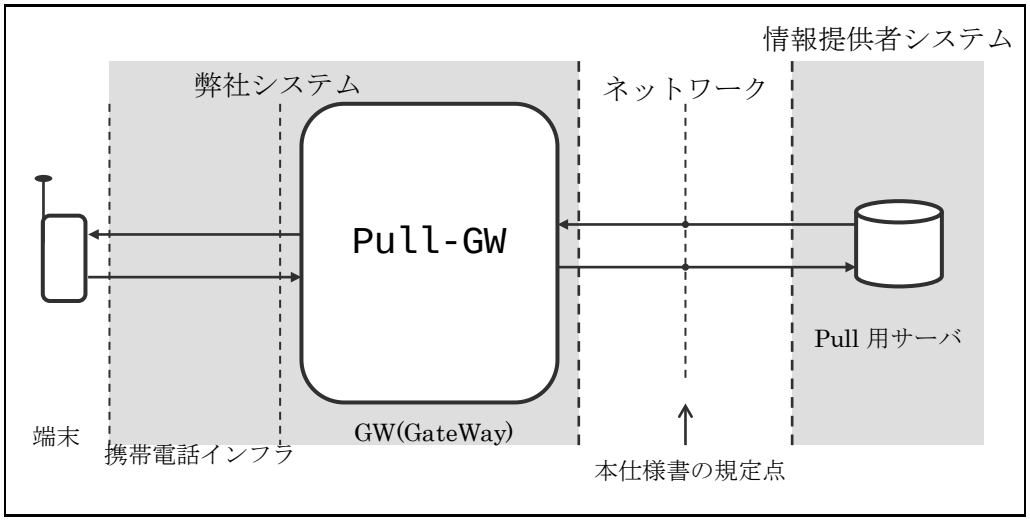

図 1.2-1 システム構成図

- ウェブシステムは端末、無線通信インフラ、Pull-GW(ゲートウェイ)、Pull 用サーバで構成する。
- 無線通信インフラ、Pull-GW ゲートウェイは弊社の設備である。
- Pull 用サーバは情報提供者の設備である。
- 情報提供者の設備と弊社の設備はネットワークを介して接続する。
- Pull 用サーバは WWW サーバである。
- 無線通信インフラおよびPull-GWはPull用サーバと端末の間のゲートウェイとして機能する。
- Pull サービスでは端末からの要求を契機として、Pull 用 WWW サーバに格納した情報を端末に提供する。
- Pull-GW と Pull 用サーバの間は TCP/IP 上で HTTP により通信する。
- GW と端末の間は無線通信インフラにてデータ通信を行う。
- Pull サービスで提供する情報(Pull 用サーバに格納する情報)は HTML もしくは XHTML で記述する。

1.3. 動作概要

Pull サービスについて動作の概要を説明する。

- **Pull サービス**

  ① 端末の利用者は メインメニューから始まるウェブサービス画面を表示し、表示内容に従って端末を操作して所望の情報にアクセスする。

  ② 端末は無線通信インフラを経由して、所望の情報へのアクセス要求を弊社ゲートウェイに対して送る。

  ③ 弊社ゲートウェイは端末からの要求を HTTP リクエストに変換して、Web サーバへ送る。

  ④ Web サーバは適切なファイルを HTTP レスポンスに載せて、ゲートウェイへ返す。

  ⑤ ゲートウェイは Web サーバから HTTP レスポンスで返されたデータについて必要に応じて情報を付加して端末へと返す。

  ⑥ 端末はゲートウェイから返された情報を表示する（音曲データの場合は音曲を演奏する）。

  ⑦ 以上、①～⑥を利用者は繰り返す。

2. インターネット接続

Web サーバと弊社 GW 間をインターネットで接続する場合には弊社 GW に対して IP 到達可能であればよい。

- **ネットワーク層インタフェース(IP)**
  以下のプロトコルを用いる。
  ・IPv4/CIDR
  ・ICMP
  Web サーバと Pull-GW の間は互いに IP reachable であること。

- **トランスポート層インタフェース(TCP)**
  IP 上では TCP を用いる。

- **セッション層以上のインタフェース**
  TCP 上では以下のプロトコルを用いる。
  - 弊社との間で HTTP を用いて Pull サービスを提供する場合には、Pull-GW の任意のポート番号(1024～65535)から Web サーバ側の 80 番のポートへと HTTP リクエストを送出する。80 番以外のポートへは HTTP リクエストを送出しない。
  - SSL3.0 もしくは TLS1.0 上で HTTP を利用する場合(HTTPS)には Web サーバ側の 443 番のポートへと HTTP リクエストを送出する。443 番以外のポートへは HTTP リクエストを送出しない。

3. HTTP

本章では Pull サービスの際に用いる HTTP の仕様を詳述する。

3.1. 概要

Pull-GW と Pull 用サーバの間は HTTP で通信する。ソフトバンク携帯端末向けウェブで利用する HTTP は HTTP/1.1 を参考にしてソフトバンク携帯端末向けに規定したプロトコルである。以降、表記の簡略化のため、「Pull サービス用にソフトバンク携帯端末向けウェブで用いる HTTP」は「PULL-HTTP」と呼称する。

Pull-GW から Pull 用サーバへ送付する HTTP PDU(Protocol Data Unit)のヘッダフィールドは以下の点を改修している。

① **リクエストメソッドの制限**

HTTP/1.1 で規定されているリクエストメソッドのうちで、PULL-HTTP で利用できるメソッドを”GET”と”POST”のみに限定している。

② **ヘッダの制限**

HTTP/1.1 で規定されているヘッダフィールドのうちで、PULL-HTTP で利用できるものを response-header, general-header, request-header, entity-header の一部だけに限定している。

③ **拡張ヘッダ**

HTTP/1.1 で用意されている拡張ヘッダ(extension-header)を用いて、ソフトバンク携帯端末向けの拡張ヘッダを用意している。

3.1.1. 3GC 型端末向けサービスの概要

端末は、下記の機能を有する。

- 著作物保護
  Web サーバからのレスポンスにおいて、返送するデータの保護方法を指定することができる。

- statefull なセッション
  Cookie を用いて statefull なセッションを実現できる。

- URI の表示
  リクエスト URI が表示される。

- 送達確認情報の送出
  OMA Download に対応し、オブジェクトの送達確認情報を Web サーバへ送出することが可能である。

3.1.2. 制限

リクエストヘッダサイズは 3kbytes 未満

- リクエストラインは 1024(※1)bytes 未満(escaped encoding を考慮のこと)
- リクエストは 300kbytes 未満
- レスポンスヘッダサイズは 3kbytes 未満
- エンティティサイズは 300kbytes 未満(ただし、テキストおよびマークアップ言語を表す Content-Type の場合は 48kbytes 未満 ※2)
- ページサイズ(ルートドキュメント+インラインデータ)は 300kbytes 未満

※1: 以下の端末は 1024bytes 未満と異なる制限を持つ。

- 804SS, 705SC, 706SC, 707SC, 707SCⅡ, 708SC, 709SC, 805SC: 500bytes 未満
- 702MO, 702sMO: 512bytes 未満

※2: 以下の端末は 48kbytes 未満と異なる制限を持つ。

- 702MO, 702sMO: 10kbytes 未満
- 802N, 703N: 21103bytes 未満
- 902SH, 802SH, 903SH, 703SH, 703SHf, 804SH: 22kbytes 未満
- 802SE: 30kbytes 未満

3.2. 制御機能

3.2.1. 著作物保護

ウェブからダウンロードしたデータについて以下の操作を制御することができる。

① 端末内の不揮発性メモリへの**保存**

② メールに添付しての**送信**・赤外線等を介しての**転送**

③ 端末の外部メモリへの**外部転送**

OMA DRM で規定される Forward Lock により、上記の機能を制御することが可能である。ただし、一部の端末では、端末内への保存を不可とすることができないので注意すること。

3.2.1.1. 3GC 型端末

レスポンスヘッダフィールドおよびファイル名の拡張子を組み合わせた場合には、表 3.2.1.1-1に示すように制御する。表中、x-jphone-copyright レスポンスヘッダとファイル名拡張子の組合わせで「保存可、送信・転送不可、外部転送可」になる場合は、弊社 Pull-GW にて Forward Lock 方式の配信に変換する。ただし、SSL 通信を行う際には、本変換は行われない。また、端末が XHTML/HTML 文書を Forward Lock 方式で受信した場合の動作は保証しない。

3GC 型端末における x-jphone-copyright レスポンスヘッダとファイル名拡張子による著作物保護制御は、過去の端末向けのコンテンツを 3GC 型端末でも扱うための、互換性のために残すものである。この為、**新規に 3GC 端末向けコンテンツを作成する場合は、過去の著作物保護制御を使用せず、オリジンサーバから Forward Lock 方式で配信することが望ましい。**

レスポンスヘッダに「no-store」を指定した場合、一部の端末では正常に動作しない。また、OMA Download を利用した場合には、すべての端末において正常に動作しない。

Download Descriptor を著作物保護の対象とすることはできない。

表 3.2.1.1-1 3GC 型端末での著作物保護

1)	no-store	○	×	×	×	×	×	×	×	×	×
2)	no-store		○	×	×	×	×	×	×	×	×
	no-transfer			○	○	×	×	×	×	×	×
	no-peripheral			○	×	○	×	×	×	×	×
	その他			×	×	×				×	
3)	application/vnd.oma.drm.message		×				○	×	×	×	×
	application/x-smaf application/vnd.smaf						×	○	×	×	×
	application/x-shockwave-flash						×	×	○	×	×
	application/vnd.oma.dd+xml	×	×	×	×	×	×	×	×	×	
	その他						×	×	×		
4)	pnz, jpz						×	×	×	○	×
①	保存	不可	不可	可	可	可	可	b_1	可	可	可
②	送信・転送	不可	不可	不可	不可	不可	不可	b_0	F_0	不可	可
③	外部転送	不可	不可	可	可	可	可	b_1	可	可	可
	備考	※1	※1	※2 ※3	※2 ※3	※2 ※3	※2	※4	※5	※2 ※3	

※1: 一部の端末では、no-store が利用できない。また、OMA Download を利用した場合は、すべての端末において利用ができない。

※2: 端末が XHTML/HTML 文書を Forward Lock 方式で受信した場合の動作は保証しない。

※3: SSL 通信を行う際には、Forward Lock 方式への変換が行われない。

※4: SMAF の扱いに準ずる。

b_0 : Contents Info Chunk の CopyStatus の転送ビットで制御する。

b_1 : Contents Info Chunk の CopyStatus の保存ビットで制御する。

※5: SWF メタデータの設定に順ずる。

F_0 : Flash Lite™ 3.1 以降のバージョンに対応する端末においては、SWF メタデータの Forward Lock 設定で制御する。それ以外の端末については「可」となる。

詳細は「ウェブコンテンツ開発ガイド[メディア編]」を参照すること。

表 3.2.1.1-1で 1)～4)は下記の項目である。

1) Cache-Control レスポンスヘッダフィールド
2) x-jphone-copyright レスポンスヘッダフィールド
3) Content-Type レスポンスヘッダフィールド
4) ファイル名拡張子

[凡例]
○：当該値が与えられていることを表す。
×：当該値が与えられていないことを表す。
灰色：値が任意であることを表す。
可：当該する端末操作が可能であることを表す。
不可：当該する端末操作が不可能であることを表す。

3.2.1.2. Forward Lock

Forward Lock は、「メールに添付しての送信・赤外線等を介しての転送」を禁止するコンテンツを端末へ配信するための方式である(「端末内の不揮発性メモリへの保存」可能であり、条件により「端末の外部メモリへの外部転送」も可能である)。本方式を利用するためには、配信するコンテンツを OMA DRM[DRM]で規定される DRM Message に変換して配信する必要がある。

なお、XHTML/HTML 文書を Forward Lock にて配信することはできない。

3.2.1.2.1. DRM Message

Forward Lock 方式に使用する DRM Message は、RFC2046 で規定されるマルチパート型のファイルで、一つのオブジェクトのみで構成される。DRM Message のメディア型は表 3.2.1.2.1-1に示す値で無ければならない。配信するコンテンツの実際のメディア型は、マルチパートのエンティティに記載する Content-Type ヘッダに正しく記載しなければならない。また、オブジェクトはバイナリ形式でのみの配信が可能であり、BASE64 などエンコードしてはならない。

表 3.2.1.2.1-1 DRM Message のメディア型

種類	メディア型
Forward Lock	application/vnd.oma.drm.message

3.2.1.2.2. DRM Message の例

Forward Lock 方式を利用した時の HTTP レスポンスの例を以下に示す。

```
HTTP/1.0△200△OK<CR><LF>                                  } ステータスコード
Content-Type:△application/vnd.oma.drm.message;△          }
                boundary=boundary-1<CR><LF>              } HTTP ヘッダ
Content-Length:△574<CR><LF>                             }
<CR><LF>
--boundary-1<CR><LF>                                     }
Content-Type:△image/jpeg<CR><LF>                         }
Content-Transfer-Encoding:△binary<CR><LF>                } メッセージボディ
<CR><LF>                                                 }
・・・JPEG 画像のバイナリデータ・・・<CR><LF>                }
--boundary-1--<CR><LF>                                   }
```

注意事項：

- <CR>,<LF>,△は、以下の制御コードを意味する。
  <CR> :0x0d (CarriageReturn)
  <LF> :0x0a (LineFeed)
  △ :0x20 (Blank)
- Content-Length ヘッダの値は、メッセージボディのサイズを返す必要がある。本例では、JPEG 画像 479bytes と、前後のサブパートヘッダ行や境界区切り行等の 95bytes を加算した値である。
- サブパートのヘッダ部に記載する Content-Transfer-Encoding ヘッダの値は、「binary」のみ可能である。

3.2.1.2.3. OMA Download 機能と併用する場合の注意点

「3.2.7 送達確認情報の送出」に記載した OMA Download 機能を利用して、Forward Lock 方式でオブジェクトを配信する場合の注意点を説明する。Download Descriptor に記載する type 属性には、実際に配信するオブジェクトと Forward Lock 方式の 2 つのメディア型をこの順番で指定する必要がある。

例えば、Forward Lock 方式で JPEG 画像を配信する場合の Download Descriptor の例を以下に示す。

[記述例]

```
<media xmlns = "http://www.openmobilealliance.org/xmlns/dd">
    <DDVersion>1.0</DDVersion>
    <type>image/jpeg</type>
    <type>application/vnd.oma.drm.message</type>
    <objectURI>http://www.foo.com/image.dm</objectURI>
    <size>479</size>
    <installNotifyURI>http://www.foo.com/notify.cgi</installNotifyURI>
</media>
```

注意事項：

- <objectURI>で指定する URI からは、Forward Lock 方式の DRM Message を配信する必要がある。
- <size>は、DRM Message に変換前のオブジェクトサイズを指定する必要がある。
- 本形式の Download Descriptor を記述することにより、<objectURI>で指定したコンテンツが自動的に DRM Message へ変換されることはない。

3.2.2. キャッシュ制御

Webサーバから返送されたコンテンツをPull-GW上にキャッシュすることができる。これにより、端末からのリクエストをPull-GWで折り返すことができ、利用者にとって高速なレスポンスを提供するとともに、Web サーバへの負荷を軽減することが可能となる。加えて、端末では端末自身にコンテンツをキャッシュすることで、より高速なレスポンスを実現することが可能となっている。

キャッシュ制御に関わる機能として、下記を提供している。なお、キャッシュ制御の対象となるのはGETメソッドで取得したリソースのみであり、POSTメソッドについてはキャッシュの利用を行わず、取得したリソースをキャッシングすることもない。

- キャッシュ可否 (Request, Response)
- キャッシュリソースの有効期限 (Response)
- リソースの更新問い合わせ (Request, Response)
- 最新リソースの強制取得 (Request)

① キャッシュ可否

下記に該当するリソースはキャッシュを行わない。

・リクエストヘッダに特定のヘッダフィールドを含む

該当するいずれかのリクエストヘッダを伴う場合、Pull-GW はキャッシュの有無に係らずCP様サーバにリクエストを行い、レスポンスのキャッシングを行わない。

表 3.2.2-1 キャッシング不可要因(リクエストヘッダ)

ヘッダフィールド	値	端末	Pull-GW
Cookie	任意	×	○
Authorization	任意	×	○

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)

・レスポンスヘッダに特定のヘッダフィールドを含む

該当するいずれかのレスポンスヘッダを伴う場合、レスポンスのキャッシングを行わない。

表 3.2.2-2 キャッシング不可要因(レスポンスヘッダ)

ヘッダフィールド	値	端末	Pull-GW
Set-Cookie	任意	×	○
WWW-Authenticate	任意	×	○
Warning	任意	×	○
Cache-Control	no-cache	○	○
	no-store	○	○
	private	×	○
Pragma	no-cache	○	○
Vary	*	×	○

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)

・META 要素にキャッシュ制御の記述(http-equiv="cache-control")がある

該当するいずれかの META 要素の記述を伴う場合、レスポンスのキャッシングを行わない。

表 3.2.2-3 キャッシング不可要因(META 要素)

プロパティ名	プロパティ値	端末	Pull-GW
cache-control	no-cache	○	×
	no-store	○	×

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)

Pull-GW では、上記のキャッシング不可に該当せず、かつ下記に該当するリソースをキャッシュする。端末では、レスポンスヘッダによるキャッシュ可否が決定しない場合に限り、META 要素によるキャッシュ可否を決定する。

・レスポンスヘッダに特定のヘッダフィールドを含む

表 3.2.2-4 キャッシング要因(レスポンスヘッダ)

ヘッダフィールド	値	端末	Pull-GW
Date	<HTTP-date>	◎	◎
Cache-Control	max-age	○	○
	x-maxage	×	○
	public	×	○
	must-revalidate	×	○
	proxy-revalidate	×	○
Expires	<HTTP-date>	○	○

(◎：影響する(必須)、○：影響する(オプション)、×：影響しない)

・META 要素にキャッシュ制御の記述(http-equiv 属性)がある

表 3.2.2-5 キャッシング要因(META 要素)

プロパティ名	プロパティ値	端末	Pull-GW
cache-control	max-age	○	×
expires	<HTTP-date>	○	×

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)

Cache-Control ヘッダフィールドおよび Expires ヘッダフィールドは、キャッシングを行う為に、いずれか一つを必要とする。

② キャッシュリソースの有効期限

キャッシュリソースの有効期限は、max-age ディレクティブ、あるいは Expires ヘッダフィールドの値より決定する。
max-age ディレクティブを与えた Cache-Control ヘッダフィールドと Expires ヘッダフィールドが同一のレスポンスに載っていた場合は、Cache-Control が優先される。
有効期限内のリソースに対するリクエストは、最新リソースの強制取得を行う場合を除き、キャッシュを利用することにより、CP 様サーバへのリクエストは行われない。

③ リソースの更新問い合わせ

キャッシュ可要因に加え、下記のレスポンスヘッダを伴うリソースは、キャッシュの有効期限が切れた後のリクエストにて、対応するリクエストヘッダを載せることにより、リソースの更新問い合わせを行う。
CP 様サーバでは、この更新問い合わせに対し、更新がない場合は Status-Code 304 およびメッセージボディを含めないレスポンスを、更新がある場合は Status-Code 200 および更新したリソースをメッセージボディに載せてレスポンスを行うことにより、キャッシュ更新の制御を行うことが出来る。

表 3.2.2-6 リソース更新問い合わせ

レスポンス	リクエスト	端末	Pull-GW
Last-Modified	If-Modified-Since	○	○
ETag	If-None-Match	○	○[*1]

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)
*1: ETag ヘッダフィールドを利用する場合、Last-Modified ヘッダフィールドも同一のレスポンスに載っている必要がある

④　最新リソースの強制取得

以下については、キャッシュの有無によらず、リクエストを行う。

表 3.2.2-7 キャッシュ利用不可要因(リクエストヘッダ)

ヘッダフィールド	値	端末	Pull-GW
Cookie	任意	×	○
Authorization	任意	×	○
If-None-Match	任意	×	○[*1]
Cache-Control	no-cache	○	○
Pragma	no-cache	○	○

(○：キャッシュ制御に影響する、×：キャッシュ制御には影響しない)

*1: If-Modified-Since を伴わない場合、キャッシュを利用せずにリクエストを行う

3.2.3. 断片データ(chunk)

Pull-HTTP では Web サーバからの動的に生成するレスポンスにおいて、エンティティボディを断片化することでレスポンスを生成した順に逐次的に返送することができる。

3.2.4. 利用者の認証(HTTP Authentication)

Web サーバが利用者を認証する手段を提供する。認証方式としては Basic 認証と Digest 認証が利用できる。ただし、一部の端末では Digest 認証が利用できない。

利用者認証を行うに当たり、Web サーバには認証対象となる利用者の識別子(user-ID)とパスワード(password)の対を格納しておく。

[Basic 認証]

user-ID と password を Base64 でエンコードしたものを Web サーバに送り利用者を認証する。認証は以下の手順となる。

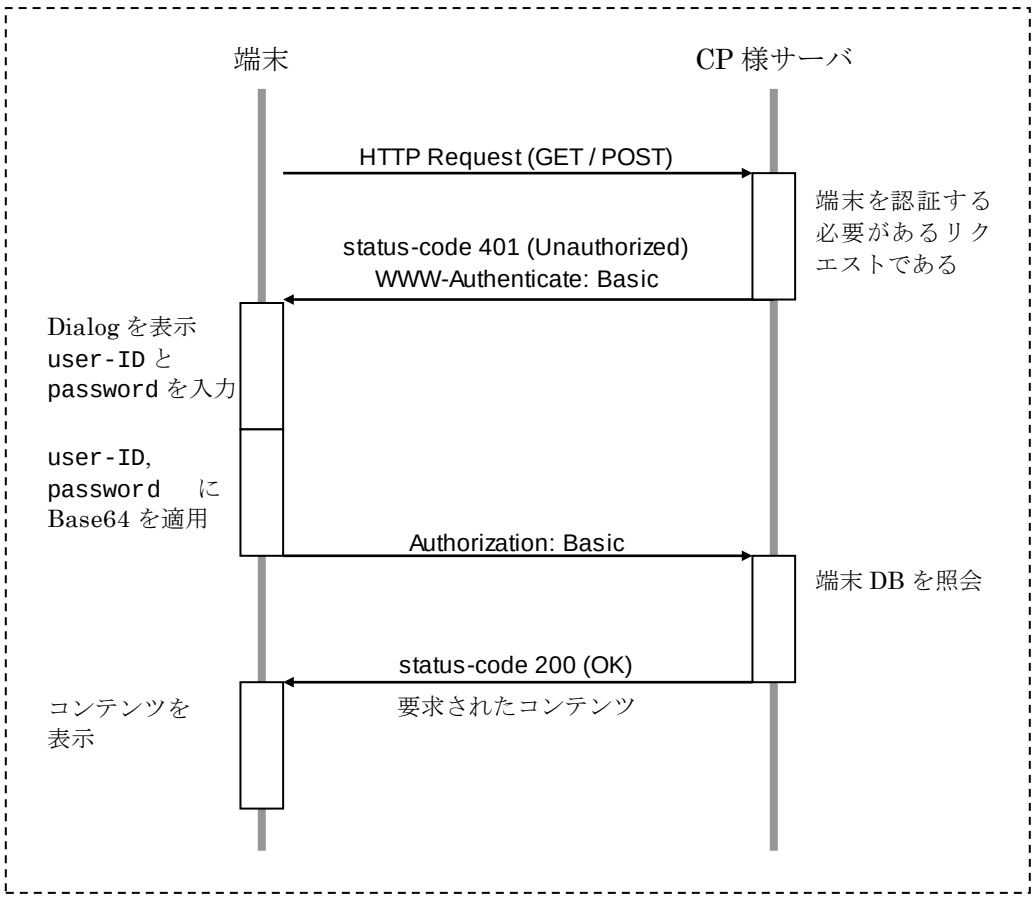

図 3.2.4-1 Basic 認証のシークエンス

Basic 認証では user-ID と password は Base64 でエンコードしており容易にデコードできるため、盗聴されると user-ID と password を知られる。Basic 認証を利用する際は SSL/TLS と併用することが望ましい。

[Digest 認証]
user-ID と password に MD5 を適用したダイジェストを Web サーバに送り利用者を認証する。認証は以下の手順となる。

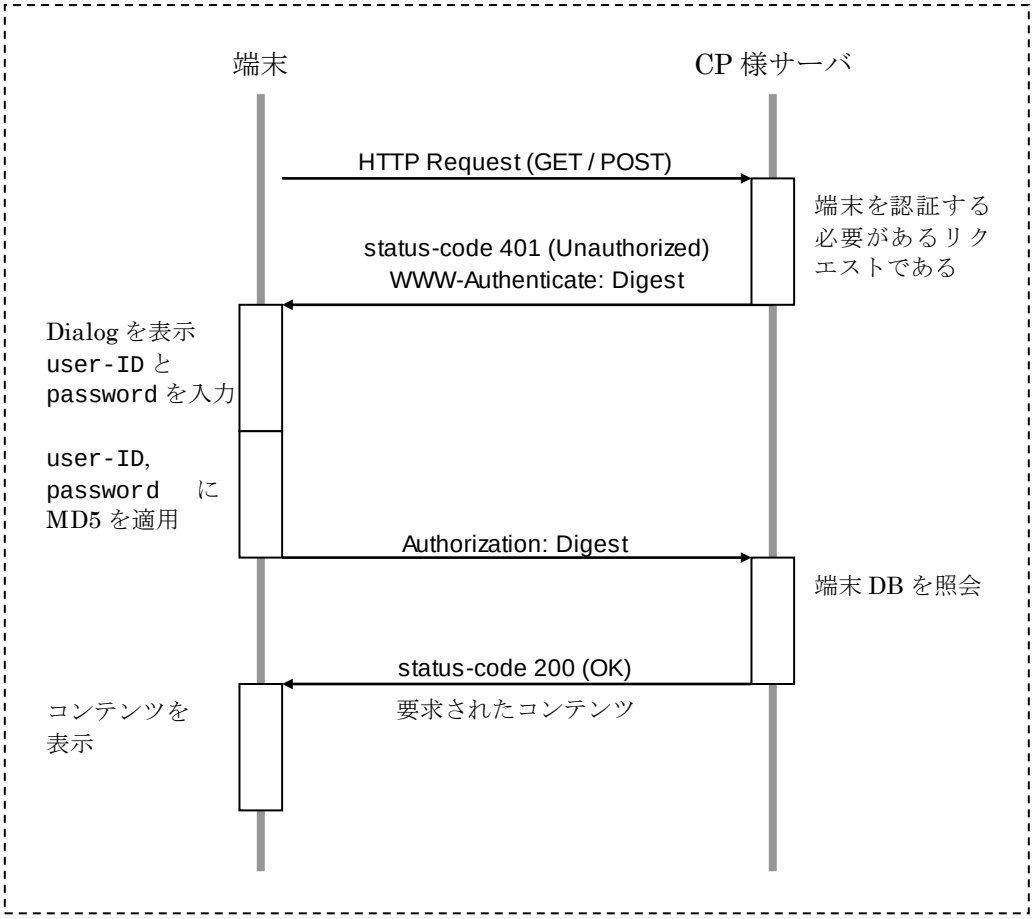

図 3.2.4-2 Digest 認証のシークエンス

Digest 認証では user-ID と password に MD5 を適用したものを Web サーバに送付しているため、盗聴されたとしても user-ID と password を知られることはない。

3.2.5. レンジリクエスト(Range Request)

Web サーバからのレスポンスにおいて、エンティティボディの一部が欠落した場合に、直後に欠落した部分だけを再度返送するように端末からリクエストすることがある(以降、このようなリクエストを***レンジリクエスト***と呼ぶ)。Web サーバからは欠落部分だけを返送することで、コンテンツ全体を再取得する場合に比べ、短時間でリクエストしたコンテンツを入手することができる。これにより、コンテンツの取得時に通信断が発生した場合等、より短時間かつ少ない通信料でコンテンツを提供することが可能となる。

ただし、一部の端末ではレンジリクエストに対応していない。

[リクエスト]
あるリソース R について端末がレンジリクエストを発生するための条件を下記に示す。

① リソース R を返送するレスポンスにおいて所定のレスポンスヘッダフィールドを載せている。

② リソース R を返送するレスポンスの途中において回線断が発生した。

①に記載した「所定のヘッダ」とは以下の組み合わせの場合であり、その際のバイトレンジリクエストと対照して一覧する。

表 3.2.5-1 バイトレンジリクエストの発生要件

レスポンス	バイトレンジリクエスト
Accept-Ranges:bytes Etag:<任意の値> Last-Modified:<日時>	Range:bytes=<開始>-<終了> If-Range:<Etag の値>
Accept-Ranges:bytes Etag:<任意の値>	Range:bytes=<開始>-<終了> If-Range: <Etag の値>
Accept-Ranges:bytes Last-Modified:<日時>	Range:bytes=<開始>-<終了> If-Range:<日時>
Accept-Ranges: <任意の値> Etag, Last-Modified ともに無し	バイトレンジリクエストは行わない。
Accept-Ranges 無し	バイトレンジリクエストは行わない。

②に記載した回線断とは以下の事象が発生した場合である。

表 3.2.5-2 Range Request の発生条件

端末	条件
一部を除く全端末で共通	リソース取得中に回線断が発生し、その後通信可能な状態になった。

［レスポンス］
バイトレンジリクエストに対するレスポンスとしては、以下のいずれか一つを選択する。

① リソースの欠落部分だけを載せたレスポンスを Status-Code 206 で返す。

② リソース全体を載せたレスポンスを改めて Status-Code 200 で返す（通常のレスポンス）。

※レンジリクエストに対しては通常のレスポンスも許容しており、必ずしもレンジレスポンスで返す必要は無いことに注意せよ。

Status-Code206 で返すレスポンスは以下の構成となる。

- HTTP ヘッダに、Content-Type ヘッダフィールドと Content-Range ヘッダフィールドを記載する。
- Content-Type ヘッダフィールドには当該パートのメディア型を載せる。
- Content-Range ヘッダフィールドには不足した範囲を指定する。

以下に HTML ドキュメントについてレンジリクエストした際のレスポンスの例を記載する。ここで欠落したため要求した部分は 111-222 の範囲である。

```
HTTP/1.1 206 Partial Content
Date: Sat, 31 Dec 9999 23:59:59
Last-Modified: Fri, 30 Dec 9999 00:00:00
Content-type: */*
Content-range: bytes 111-222/10000
Accept-ranges: bytes
  :
<111～222 の範囲のデータ>
  :
```

ソフトバンク携帯電話向けサービスでは、一回のレンジリクエストに対する一回のレスポンスに載せることが出来る欠落部分は、単一パートに限定される。一般的なレンジリクエストでは一回のレスポンスには複数パートの欠落部分を載せることが出来る。しかしながら、ソフトバンク携帯電話**向けサービス**では一回のレスポンスには複数パートの欠落部分を載せることはできない。

3.2.6. statefull なセッション(Cookie)

端末では Cookie を利用した statefull なセッションを提供する。ここで提供する Cookie は RFC2109 で規定しているもの、および、Netscape Communication 社が提案したもの、を参考に弊社にて規定したものである。

本来、HTTP はリクエストに対してレスポンスを返すが、複数個のリクエスト、レスポンスの対の間では連携を取ることの無い stateless なプロトコルである。Pull-HTTP では statefull なセッションを実現するために、端末と Web サーバは以下のように振舞う。

① 端末が Web サーバへリクエストを送る。

② Web サーバから端末へ返すレスポンスに Cookie と呼ぶ情報を載せて送る。

③ 端末は Cookie を記憶し、次回のリクエストではこの Cookie を載せて Web サーバに送る。

これにより、Web サーバは２回目のリクエストが前回のレスポンスの結果に基づくものであることが分かる。このようにして複数のリクエスト/レスポンスの組について関連付けをおこなうことで、statefull なセッションを実現している。

[Cookie の有効期限]
端末上で保持する Cookie の個数および有効期間(保持期間)は無制限ではない。一部の端末では、期限が指定されていない Cookie を一時型として扱わないので注意すること。

表 3.2.6-1 Cookie の種類

型	Cookie の有効期限
一時型	端末ブラウザの終了と共に無効になる。
期限型	端末ブラウザの終了とは無関係に、指定された期限までは有効である。期限後は無効になる。

Cookie がいずれの型になるかは Web サーバで決定し、端末に送る Cookie の中に記載する。端末側で Cookie の型を指定することはできない。
"Max-Age"または"Expires"が指定されている Cookie を「期限型 Cookie」とし、"Max-Age","Expires"共に指定されていない Cookie を「一時型 Cookie」とする。

端末はリクエストを送信する際に該当する Cookie が存在する場合、期限切れの場合は Cookie を消去し Cookie を載せずにリクエストを送る。

なお、RFC で規定される記載方法と Netscape Communication 社が提案している記載方法の混在はできない。例えば、有効期限を”Expires”で指定する場合の”domain”は、”.”(ドット)で始まるドメイン指定はできない。

[Cookie の格納制限]
格納制限数を超えて Cookie を受信したときは、有効期限を検査し期限切れの Cookie が存在すれば消去し新たに受信した Cookie を保存する。期限切れの Cookie が存在しない場合、既に保存している Cookie の中から最も古くにアクセスされた Cookie から順に削除し新たに受信した Cookie を保存する。

表 3.2.6-2 Cookie の格納制限

項目	制限
最大サイズ	1Cookie につき最大 1024bytes(※1)迄である。 1024bytes を越えた Cookie は破棄する。
ドメイン毎の個数	1 ドメインにつき一時型/期間型合わせて最大 10 個である。10 個を越えたものは格納しない。 ただし、1 ドメインにつき合計 2048bytes を制限とする。
全体の容量	1 端末につき 4kbytes

※1: 1Cookie のサイズは、Set-Cookie HTTP ヘッダ長である。Set-Cookie ディレクティブの値のみのサイズでは無いことに注意すること。つまり、

Set-Cookie: Customer="SoftBank";Version="1";Path="/doc"

の場合、ヘッダ全体の長さである。

なお、全体の容量については、表 3.2.6-2の値を下限として端末の実装依存になるため、表の値よりも大きな端末も存在する。

［ドメインの制限］

端末は受け取った Cookie を別のドメインの Web サーバに送ることはない。例えば www.foo.co.jp から受け取った Cookie を www.bar.co.jp に返すことはない。www.foo.co.jp に返すのは www.foo.co.jp から受け取った Cookie だけである。加えて、Cookie を送出したサーバの URI のパスも識別する。例えば www.foo.co.jp/fake から受け取った Cookie を www.foo.co.jp/fake2 に返すことはない。但し、Cookie が識別する URI のパスは包含関係を識別することができる。その為、例えば www.foo.co.jp から受け取った Cookie を www.foo.co.jp/fake に返すことはできる。

[SSL/TLS との併用]

端末は受け取った Cookie に Secure フィールドが含まれていたか否かを記憶する。SSL/TLS を用いているときのみ Secure フィールドと共に受信した Cookie を Web サーバに送る。SSL/TLS を用いないときには Secure フィールドと共に受信した Cookie を Web サーバに送ることはない。

3.2.7. 送達確認情報の送出

端末は、オブジェクトの端末へのダウンロード結果を Web サーバへ通知する機能を提供する。表 3.2.7-1に示すようにオブジェクトの種類により利用方法が異なる。ここでは、画像、音曲などで利用できる OMA で規定される OMA Download [DLOTA]の仕組みを利用した送達確認情報を Web サーバへ通知する方法について説明する。

表 3.2.7-1 オブジェクト種類と送達確認の利用方法

オブジェクト種類	利用方法
S!アプリ	JAD 内に指定
テキスト系ファイル	未対応
その他のオブジェクト	OMA Download 機能を利用

送達確認情報を取得する際は、オブジェクトファイルに先立ち後述する Download Descriptor と呼ばれるデータを送出する必要がある。Download Descriptor は、実際に送出するべきオブジェクトの URI や送出確認情報送出先 URI などの情報を記載したデータである。

送達確認情報は、一つのオブジェクトファイルが正常に端末にダウンロードされたか否かを表す情報であり、インラインオブジェクトを含むページの様に 1 ページ全体の送達を確認することはできない。また、インラインオブジェクトに対しても利用することはできない。

なお、端末と Web サーバの間に無線および IP(Internet Protocol)による通信区間があるため、「端末⇔Web サーバ」間の通信が断になることがある。従って、**送達確認情報が Web サーバへ必ず到達するとは保証でき兼ねる**。

3.2.7.1. シーケンス

以下では**送達確認情報を取得する場合**の処理の流れを説明する。

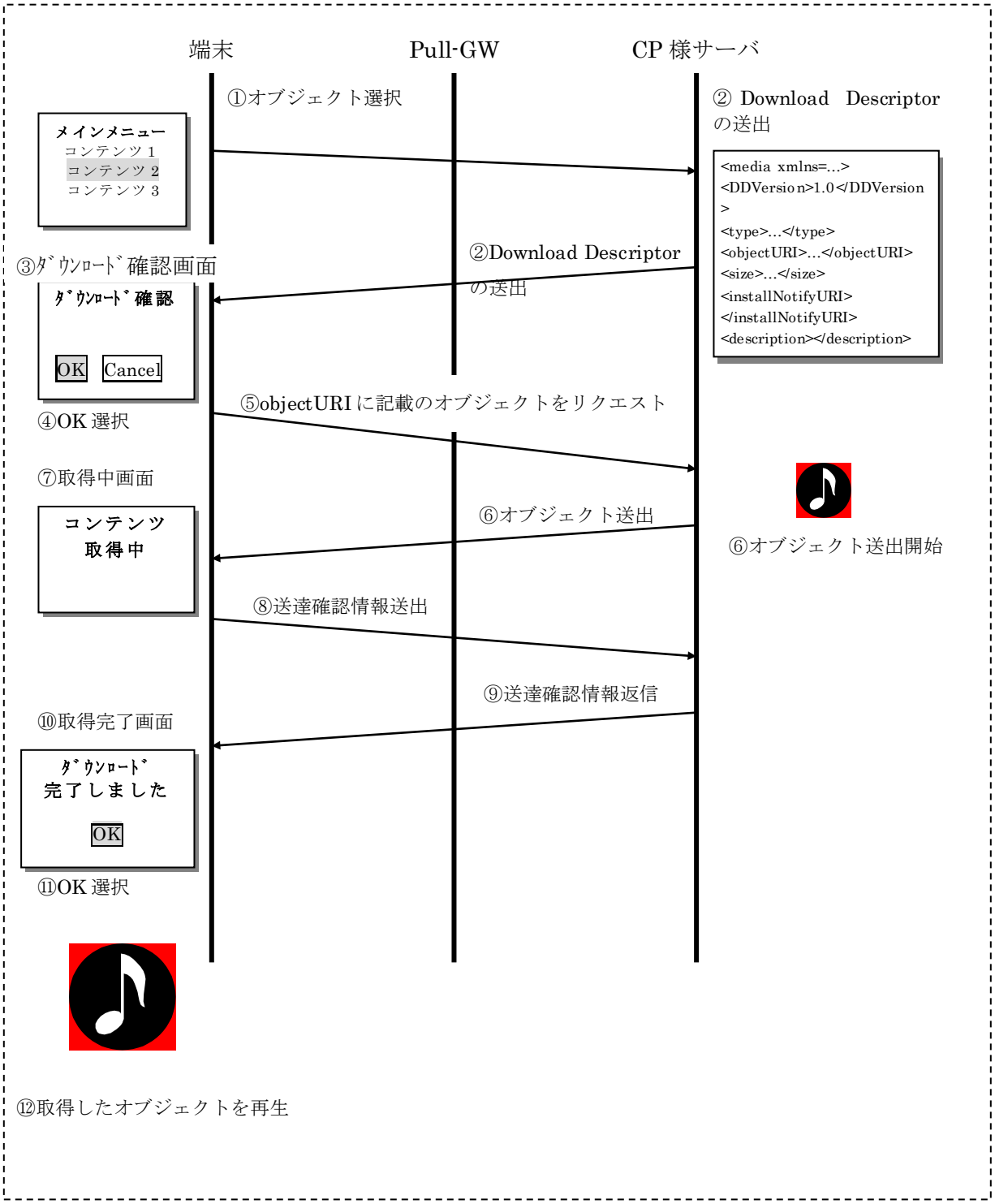

図 3.2.7.1-1 送達確認情報取得手順

① 利用者は端末上でコンテンツ取得を選択する。このアンカー上には後述する Download Descriptor への URI を記載する。

② Web サーバでは、後述する Download Descriptor を送出する。Download Descriptor には、目的とするオブジェクトへの URI を記載する。また、必要であれば、送達確認情報の送出先 URI を記載する。なお、Download Descriptor を著作物保護の対象とすることはできない。

③ 端末上には、オブジェクトのダウンロード確認画面が表示される。一部端末では、本画面は表示されず、次のシーケンス④と併せて省略される。

④ 利用者がダウンロードを確認し、OK を選択する。

⑤ 端末は、Download Descriptor の objectURI 属性に記載された URI に対してリクエストする。

⑥ Web サーバでは、指定されたオブジェクトの送出を開始する。なお、ここではテキスト系のオブジェクト(XHTML ファイルなど)は返信することはできない。また、著作物保護にて「保存: 不可」となるコンテンツを返信することはできない。

⑦ 端末上にはオブジェクト取得中の画面が表示される。

⑧ オブジェクトを取得完了後、Download Descriptor に installNotifyURI 属性が記載されている場合、installNotifyURI に対して送達確認情報を送出する。なお、送達確認情報は、正常にダウンロードされない場合にも送出されるので注意すること。
送達確認情報は[DLOTA]の規定に準拠しており、**POST** メソッドでリクエストされ、エンティティボディにオブジェクト取得時のステータスコードが記載される。正常に取得が完了した際のステータスコードは「**900**」である。CP 様サーバで、オブジェクトが正常にダウンロードされたかを判断するためには、送達確認情報のリクエストの受信とエンティティに記載されているステータスコードが「**900**」であることを検出する必要がある。その他のステータスコードは表 3.2.7.1-1に一覧する。
以下に送達確認情報の一例を表す。

POST /status HTTP/1.0
Host: www.foo.com
Content-Length: 11
・・・その他 HTTP ヘッダ・・・

900 Success

表 3.2.7.1-1 ステータスコード一覧

Status Code	Status Message	内容
900	Success	正常にダウンロードされたことを表す
901	Insufficient Memory	端末の保存容量が不足していて保存できなかったことを表す
902	User Cancelled	ダウンロード確認画面で利用者がキャンセルしたことを表す
903	Loss of Service	オブジェクトを受信中に通信が切断したことを表す
905	Attribute mismatch	type 属性の値とオブジェクトのメディア型が合致していないことを表す
906	Invalid descriptor	Download Descriptor の文法が正しくないことを表す
951	Invalid DDVersion	DDVersion 属性の値が正しくないことを表す
953	Non-Acceptable Content	端末がファイルを認識できないことを表す
954	Loader Error	objectURI 属性で示される URI にアクセスした際にエラーが発生したことを表す

⑨ Download Descriptor に installNotifyURI を記載した場合、Web サーバでは、送達確認情報に対する HTTP レスポンスを HTTP ステータス 200 で必ず返す必要がある。200 以外のステータスを返した場合、ダウンロードしたオブジェクトは削除されることがあるので、注意すること。

⑩ オブジェクトを正常に取得すると、取得完了の画面が表示される。

⑪ 利用者が取得完了を確認し OK を選択する。

⑫ 取得したオブジェクトを再生、表示する。

3.2.7.2. Download Descriptor

弊社端末向けサービスで利用できる Download Descriptor は、OMA で規定される Download Descriptor に準拠する。解釈可能な属性を表 3.2.7.2-1に示す。なお、**各属性名の大文字、小文字を区別するので、注意すること**(Case-Sensitive)。

なお、Download Descriptor の配信に SSL/TLS を利用すること、および Download Descriptor 中に https スキームを記述した場合の動作は保証しない。

表 3.2.7.2-1 利用できる属性

属性	内容	対応
DDVersion	“1.0”固定であること。	必須
type	取得するオブジェクトの MIME タイプを記述する。なお、Forward Lock 方式と併用する場合は、type 属性を 2 つ併記すること。	必須
objectURI	オブジェクトの取得先 URI を記述する。ただし、XHTML/HTML ファイルなどテキスト系のファイルを指定することはできない。	必須
size	オブジェクトのサイズを記述する。サイズは整数値であること。	必須
installNotifyURI	送達確認情報を送出する URI を記述する。	任意
description	ダウンロード確認画面に表示する説明文書を記述する。日本語を記載する場合は、UTF-8 であること。	任意

[記述例]

```
<media xmlns = "http://www.openmobilealliance.org/xmlns/dd">
    <DDVersion>1.0</DDVersion>
    <type>image/jpeg</type>
    <objectURI>http://www.foo.com/image.jpeg</objectURI>
    <size>1500</size>
    <installNotifyURI>http://www.foo.com/notify.cgi</installNotifyURI>
</media>
```

3.3. 表記法

本仕様で用いる文法(Notational Conversions and Generic Grammar)

3.3.1. BNF(Augmented BNF)

本章で用いる BNF を説明する。[HTTP/1.1(RFC2616)]記載の BNF に準ずる。

`name = definition`
: name は rule の名前であり、definition は rule の定義である。基本的な rule は大文字で表記する（SP, LWS, HT, CRLF, DIGIT, ALPHA, …）。

`“literal”`
: literal は二重引用符で括る。

`rule1 | rule2`
: (“|”)で区切られた要素うちのいずれか一つを選択することを表す。

`(rule1 rule2)`
: 丸括弧で括られた部分を一つの要素として扱う。

`*rule`
: 要素の頭に付けた”*”は、要素の繰り返しを表す。”<n>*<m>element”は少なくとも<n>回、多くても<m>回の要素(element)の繰り返しを表す。<n>, <m>いずれを省略した場合には、”0”を与えたものと解釈する。

`[rule]`
: 角括弧で括られた要素はオプションとして扱う。”[foo bar]”は”*1(foo bar)”と等価である。

`N rule`
: 要素を N 回繰り返すことを表す。即ち、”<n>element”は”<n>*<n>element”と等価である。

#rule
要素の繰り返しを表す、但し、要素と要素の間はカンマ(“,”)で区切る。”<n>#<m>element”は少なくとも<n>回、多くても<m>回の要素(element)の繰り返しを表す。<n>,<m>いずれを省略した場合には、”0”を与えたものと解釈する。なお、要素とカンマの間には複数個の空白文字(LWS)があっても構わない。

; comment
セミコロンで始まる要素はコメントとして扱う。

3.3.2. 基本的な規則(Basic Rules)

以下の規則に従い、字句解析を行う。

```
OCTET          = <データを表す 8-bit の並び>
CHAR           = <US-ASCII 文字(octets  0 - 127)>
UPALPHA        = <US-ASCII 大文字  “A”..”Z”>
LOALPHA        = <US-ASCII 小文字  “a”..”z”>
ALPHA          = UPALPHA | LOALPHA
DIGIT          = <US-ASCII 数字 “0”..”9”>
CTL            = <US-ASCII 制御文字, (octets  0 – 31)と DEL(127)>
CR             = <US-ASCII 改行(13)>
LF             = <US-ASCII ラインフィード(10)>
SP             = <US-ASCII 空白(32)>
HT             = <US-ASCII 水平タブ(9)>
<”>            = <US-ASCII 二重引用符(34)>

CRLF           = CR LF
LWS            = [CRLF] 1*( SP | HT )
TEXT           = <LWS 以外の CTL を除く OCTET>

HEX            = “A” | “B” | “C” | “D” | “E” | “F”
               | “a” | “b” | “c” | “d” | “e” | “f” | DIGIT

token          = 1*<CTL および separator を除く CHAR>

separator      = “(” | “)” | “<” | “>” | “@”
               | “,” | “;” | “:” | “¥” | <“>
               | “/” | “[” | “]” | “?” | “=”
               | “{” | “}” | SP | HT

comment        = “(” *( ctext | quoted-pair | comment ) “)”
ctext          = <”(”と”)”を除く TEXT>
quoted-string  = ( <”> *( adtext | quoted-pair ) <”> )
qdtext         = <<”>を除く TEXT>
quoted-pair    = “¥” CHAR
```

3.3.3. Pull-HTTP 向け token

以下は Pull-HTTP で共通に利用する規則である。

```
HTTP-Version  = “HTTP” “/” 1*DIGIT “.” 1*DIGIT


media-type    = type “/” subtype *( “;” parameter )
type          = token
subtype       = token
paameter      = token

HTTP-date     = rfc1123-date | asctime-date
rfc1123-date  = wkday “,” SP date1 SP time SP “GMT”
asctime= wkdat SP date3 SP time SP 4DIGIT
date1         = day SP month SP year
date3         = month SP ( 2DIGIT | ( SP 1DIGIT ) )
time          = 2DIGIT “:” 2DIGIT “:” 2DIGIT
              ; “00:00:00”..”23:59:59”
day           = 2DIGIT      ; 1..31
wkday         = “Mon” | “Tue” | “Wed” | “Thu” | “Fri” | “Sat” | “Sun”
weekday       = “Monday” | “Tuesday” | “Wednesday” | “Thursday”
              | “Friday” | “Saturday” | “Sunday”
month         = “Jan” | “Feb” | “Mar” | “Apr” | “May” | “Jun”
              | “Jul” | “Aug” | “Sep” | “Oct” | “Nov” | “Dec”
year          = 4DIGIT      ; 1970..9999
year2         = 2DIGIT      ; 00..99 (2000-2069,1970-1999)
```

注意事項：

`HTTP-date` で `rfc850-date` を利用した場合の動作は保証しない。

3.4. プロトコルパラメータ(Protocol Parameters)

3.4.1. HTTP のバージョン(HTTP Version)

本稿で規定するバージョンは 1.1 に固定する。

```
HTTP-Version = “HTTP/1.1”
```

3.4.2. URI(Uniform Resource Identifiers)

HTTP/1.1 に準拠し、従来、URL(Uniform Resource Locator)と呼称されていたインターネット上のリソースを一意に特定する記法についての名称は本稿でもURI(Uniform Resource Identifiers)で置きかえる。ソフトバンク携帯電話向けのURI の構文は「ウェブコンテンツ開発ガイド[HTML 編]」に詳述する。

3.5. HTTP メッセージ(HTTP Message)

3.5.1. メッセージタイプ(Message Types)

説明

HTTP/1.1 の「4.1 Message Types」で定義している HTTP-message トークンを参考にして、PULL-HTTP の HTTP-message トークンを定義する。

定義

```
HTTP-message     = Request | Response
generic-message  = start-line
                   *(message-header CRLF)
                   CRLF
                   [ message-body ]
start-line       = Request-Line | Status-Line
```

値

Request

3.7に詳述する。

Response

3.8に詳述する。

generic-message

Request と Response について汎化した定義である。
実際のリクエスト定義については3.7.1の Request-Line トークンの定義を参照のこと。
実際のレスポンス定義については3.8.1の Status-Line トークンの定義を参照のこと。

制限

無し。

3.5.2. メッセージヘッダ(Message Headers)

説明

HTTP/1.1 の「4.2 Message Headers」で定義している message-header トークンを参考にして、PULL-HTTP の message-header トークンを定義する。

定義

```
message-header   = field-name ":" [ field-value ]
field-name       = token
field-value      = *( field-content | LWS )
field-content    = 1*OCTETS
```

値

HTTP/1.1 に準拠する。

制限

field-content が LWS で開始したり、LWS で終了したりすることは無い。

3.5.3. メッセージボディ(Message Body)

説明

HTTP/1.1 の「4.3 Message Body」で定義している message-body トークンを参考にして、PULL-HTTP の message-body トークンを定義する。

定義

```
message-body = entity-body
             | <Transfer-Encoding にてエンコードされた entity-body>
```

値

PULL-HTTP の SDU(Service Data Unit)に相当する。

PDU のヘッダに Transfer-Encoding general-header field が指定された場合、指定された方法に従って SDU を符号化する。

メッセージボディは、Transfer-Encoding ヘッダフィールドによって示されるように、転送コーディングが適用された場合のみ、エンティティボディとは異なる。

制限

無し。

3.5.4. メッセージ長(Message Length)

メッセージ長は、メッセージボディの長さを表す。すなわち、エンティティボディに転送コーディングが適用されている場合、転送コーディング適用後の転送長である。

エンティティボディに chunked 転送コーディングが適用されている場合、Transfer-Encoding ヘッダフィールドの値に chunked を指定しなければならない。このとき、Content-Length ヘッダフィールドを送ってはならない。

エンティティボディに転送コーディングが適用されていない場合、すなわちメッセージボディ長＝エンティティボディ長である場合、Content-Length ヘッダフィールドに値を指定しなければならない。このとき、Transfer-Encoding ヘッダフィールドを送ってはならない。

また、HTTP/1.0 のリクエストに対しては、chunked 転送コーディングを適用したエンティティボディを送ってはならない。

3.6. 一般ヘッダフィールド(General Header Fields)

説明

HTTP/1.1 の「4.5 General Header Fields」で定義している general-header トークンを参考にして、PULL-HTTP の general-header トークンを定義する。

定義

```
general-header   = Cache-Control
                 | Connection
                 | Date
                 | Pragma
                 | Transfer-Encoding
```

値

無し。

制限

無し。

3.7. リクエスト(Request)

HTTP/1.1 の「5 Request」で定義している Request トークンを参考にして、PULL-HTTP の Request トークンを定義する。

```
Request       = Request-Line
                *(( general-header
                | request-header
                | entity-header
                | extention-header)CRLF)
                CRLF
                [ message-body ]
```

3.7.1. リクエストライン(Request-Line)

HTTP/1.1 の「5.1 Request-Line」で定義している Request-Line トークンを参考にして、PULL-HTTP の Request-Line トークンを定義する。

説明

メソッドとしては GET、POST を提供している。

定義

```
Request-Line      = Method SP Request-URI SP HTTP-Version CRLF
Method            = “GET”
                  | “POST”

Request-URI = “*” | absoluteURI | abs_path | authority
```

値

端末からの通常のアクセス要求は GET メソッドである。

制限

POST メソッドは以下の条件を全て満たした際にのみ、Pull-GW から Pull 用サーバに対してリクエストする。

- 「ソフトバンク携帯電話向け HTML」の form タグの method 属性として POST を指定する。

上記を除き、全てのリクエストは GET メソッドを用いる。

3.7.2. リクエストによるリソースの識別 (The Resource Identified by a Request)

PULL-HTTP ではリクエストを解釈する際に Host request-header field を利用しない。

3.7.3. リクエストヘッダフィールド(Request Header Fields)

Request ヘッダはユーザエージェントが HTTP リクエスト発行時に送出する情報である。
PULL-HTTP で定義する request-header field の一覧を示す。

```
request-header       = Accept
                     | Accept-Charset
                     | Accept-Encoding
                     | Accept-Language
                     | Authorization
                     | Cookie
                     | Host
                     | If-Modified-Since
                     | If-None-Match
                     | If-Range
                     | Range
                     | Referer
                     | User-Agent
```

3.8. レスポンス(Response)

HTTP/1.1 の「6 Response」で定義している Response トークンを参考にして、PULL-HTTP の Response トークンを定義する。

```
Response      = Status-Line
              *(( general-header
              | response-header
              | entity-header ) CRLF )
              CRLF
              [ message-body ]
```

3.8.1. ステータスライン(Status-Line)

PULL-HTTP の Status-Line トークンは HTTP/1.1 の「6.1 Status-Line」で定義している Status-Line トークンのサブセットになる。

説明

HTTP バージョンを先頭に、HTTP レスポンスステータスコード、説明文、と続いている。

定義

```
Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
Status-Code = 3DIGIT                              ;後述
Reason-Phrase = 1*<TEXT, excluding CR, LF>
```

値

Status-Code および Reason-Phrase の構文は HTTP/1.1 に準拠する。
Status-Code は Pull 用サーバでの処理結果を弊社 Pull-GW に通知するものであり、3 桁の 10 進数で表記する。Reason-Phrase は Status-Code を簡単に説明した文字列である。Status-Code の内容については「3.13 ステータスコードの定義(Status Code Definitions)」を参照のこと。

制限

`Status-Line` トークンの構文は HTTP/1.1 と同一であるが、Status-Code, Reason-Phrase については、先の一覧に掲げたように、HTTP/1.1 のサブセットとなる。

3.8.2. レスポンスヘッダフィールド(Response Header Field)

PULL-HTTP で定義する response-header field の一覧を示す。

```
response-header      = Accept-Ranges
                     | ETag
                     | Location    ; 利用制限有り
                     | Set-Cookie
                     | WWW-Authenticate
```

3.9. エンティティ(Entity)

リクエスト、レスポンスでのメッセージは、リクエストメソッドやレスポンスステータスコードによって規制されていなければ、エンティティを転送することができる。

3.9.1. エンティティ ヘッダフィールド(Entity Header Field)

説明

PULL-HTTP の `entity-header` トークンは HTTP/1.1 の 7.1 Entity Header Field で定義している `entity-header` トークンのサブセットになる。

定義

```
request-header    = Content-Encoding
                  | Content-Language
                  | Content-Length
                  | Content-Location
                  | Content-Range
                  | Content-Type
                  | Expires
                  | Last-Modified
```

値

HTTP/1.1 に準拠する。

制限

無し。

3.9.2. エンティティ ボディ(Entity Body)

entity-body があれば、HTTP リクエストおよびレスポンスに加えて送ることができる。

説明

HTTP リクエストやレスポンスと共にエンティティボディが送られてきたら、それはエンティティヘッダフィールドによって定義されるフォーマットをもってエンコーディングされている。

定義

```
entity-body       = * OCTET
```

値

entity-body は OCTET 列であり、その解釈は端末に任される。

制限

HTTP/1.1 では entity-body の型(type)および符号化(encoding)を entity-header field に指定することができる。しかしながら、PULL-HTTP では"Content-"で始まる entity-body の型や符号化を指定するトークンが定義されていないため、entity-body に記載されたデータの解釈を動的に端末に通知することはできない。entity-body に含まれるデータの解釈は端末に固定されている。

3. 10. HTTP 拡張ヘッダ(extension-header)

3.10.1. 拡張ヘッダフィールド(Extension Header Field)

x-jphone-*ヘッダフィールドおよび、x-s-*ヘッダフィールドには、端末の特定、端末の機能の特定を行うための情報を記載している。CP 様はこの情報をもとに端末毎のきめこまかな表示の制御、利用者の管理、課金情報の管理を行うことができる。

なお、x-jphone-color、x-jphone-display、x-jphone-msname、x-jphone-smaf ヘッダフィールドは別名、MS-Profile と呼ぶ。

表 3.10-1 拡張ヘッダフィールド一覧

要素	項目	詳細
拡張ヘッダ(リクエスト)	x-jphone-color	メインディスプレイで表示可能な色
	x-jphone-display	メインディスプレイの物理サイズ
	x-jphone-msname	端末の機種名
	x-jphone-region	ユーザの利用地域(国内、国外)を表す
	x-jphone-smaf	SMAF 種別
	x-jphone-uid	ユーザ ID(UID)
	x-s-bearer	使用しているネットワーク種別
	x-s-display-info	ブラウザのコンテンツ表示領域、半角表示文字数、テキストブラウズ設定
	x-s-unique-id	端末が特殊モデルの場合に指定
拡張ヘッダ(レスポンス)	x-jphone-copyright	保存、送出、転送の可否を指定

3.10.2. WAP 拡張ヘッダフィールド(WAP Extension Header Fields)

一部の端末では、OMA で規定される User Agent Profile[UAProf]に従い、x-wap-profile ヘッダフィールドおよび x-wap-profile-diff ヘッダフィールドにて端末の CPI を参照するための情報を通知する。

3. 11. コネクション(Connections)

HTTP/1.1 ではデフォルトで永続コネクション(persistent connection)を行うが、PULL-HTTP ではシステム資源の制約から永続コネクションは利用できない。従って、Pull-GW から Pull 用サーバへのリクエストでは必ず `Connection`ヘッダフィールドに`"close"`トークンを指定する。逆に、Pull 用サーバから Pull-GW へのレスポンスに `Connection` ヘッダフィールドを含む場合には、必ず`"close"`トークンを指定すること。

3. 12. メソッドの定義(Method Definitions)

PULL-HTTP では Pull-GW から Pull 用サーバへのリクエストは GET メソッドもしくは POST メソッドを利用できる。

3.12.1. GET

GET メソッドでは Pull 用サーバに対して、Request-URI で指定したドキュメントを返すよう要求する。

PULL-HTTP では HTTP/1.1 で提供されている条件付き GET(conditional GET)および部分的 GET(partial GET)が利用可能である。

3.12.2. POST

POST メソッドは HTML の Form に入力された結果を Request-URI の Request-Line に載せて Pull 用サーバに送る。

3. 13. ステータスコードの定義(Status Code Definitions)

Web サーバから弊社 Pull-GW へ返す `Response` に載せる `Status-Code` は 3 桁の 10 進数であり、Web サーバでの処理結果を弊社 Pull-GW および端末に通知する。Status-Code の値が弊社 Pull-GW もしくは端末の次の動作に影響を与えることがある。

先頭桁の数値により、Status-Code には大まかに以下の意味合いを持たせている。

- 100 番台：通知：リクエストを受信した、もしくは、処理中である。
- 200 番台：成功：リクエストを受信し、解釈し、処理に成功した。
- 300 番台：リダイレクト要求：リクエスト先の変更を要求する。
- 400 番台：クライアントエラー：クライアントの要求が誤っている。
- 500 番台：サーバーエラー：サーバーはリクエストの実行に失敗した。

端末では下記の Status-Code を受理した場合に、ソフトバンク携帯電話固有の動作を行う。

表 3.13-1 端末で Status-Code を受理した際の固有の動作

100 Continue
通常のユーザエージェントでは 100 Continue レスポンスを受理後にエンティティボディのみを載せたレスポンスが Web サーバから送出されることを期待する。
301 Moved Permanently
Web サーバは端末からリクエストされたリソースが**恒久的**に別の場所(URI)に移動したことを表す。端末上のお気に入り／マイリンク登録では**移動後**の URI を登録する。 なお、POST リクエストに対するレスポンスで、本 Status Code を受信した場合、リダイレクト時のリクエストメソッドは **GET** である。
302 Found
Web サーバは端末からリクエストされたリソースが別の場所(URI)に移動したことを表す。端末上のお気に入り／マイリンク登録では**移動後**の URI を登録する。 なお、POST リクエストに対するレスポンスで、本 Status Code を受信した場合、リダイレクト時のリクエストメソッドは **GET** である。
303 See Other
Web サーバは端末からリクエストされたリソースが別の場所(URI)に移動したことを表す。端末上のお気に入り／マイリンク登録では**移動後**の URI を登録する。 なお、POST リクエストに対するレスポンスで、本 Status Code を受信した場合、リダイレクト時のリクエストメソッドは **GET** である。
307 Temporay Redirect
Web サーバは端末からリクエストされたリソースが**恒久的**に別の場所(URI)に移動したことを表す。端末上のお気に入り／マイリンク登録では**移動後**の URI を登録する。 なお、POST リクエストに対するレスポンスで、本 Status Code を受信した場合、リダイレクト時のリクエストメソッドは **POST** である。

3. 14. アクセス認証(Access Authentication)

端末向けサービスの PULL-HTTP では Basic 認証および Digest 認証が利用可能である。ただし、一部の端末では、Digest 認証は利用できない。

3.15. ヘッダフィールドの定義

Pull-HTTP で Pull-GW から Web サーバへと送出するリクエストに含まれるヘッダフィールドについて説明する。

本章は以下の構成をとる。

表 3.15-1 ヘッダフィールドの説明

項目	説明
説明	角括弧で リクエスト、レスポンス のいずれで利用するか記載する。 端末から常に送出するリクエストヘッダフィールドは**必須リクエスト**と記載する。 Web サーバが必ず返送しなければならないレスポンスヘッダフィールドは**必須レスポンス**と記載する。 当該ヘッダフィールドの機能を説明する。
定義	当該ヘッダフィールドの書式を前述の BNF で与える。
値	当該ヘッダフィールドの値について説明する。 レスポンスコードで特に留意する必要があるものについても、ここに記載する。
制限	当該ヘッダフィールドに与えられた制限を説明する。

HTTP ヘッダの一覧はAppendix.Aに記載する。

3.15.1. Accept

説明

[必須リクエスト]
端末がリクエストを送出する際に、端末が受理できるデータ型の一覧を Web サーバに通知する。端末が受理できるデータ型は Accept ヘッダフィールドに MIME 型で記載する。

定義

```
Accept            = “Accept” “:” #( media-type )
media-type = “*/*”
           | type “/” subtype           ; Appendix.B参照
```

値

Accept ヘッダフィールドで指定する MIME 型は、端末毎に、また、リクエスト元となる要素によって異なる。なお、本ヘッダフィールドに記載されるデータフォーマットであっても必ずしも端末での再生を保証するものではない。

制限

HTTP/1.1 に規定されている”q”パラメータを Pull-HTTP で記載することはない。

2007 年 11 月下旬以降、このリクエストの値は“*/*”固定である（SSL 通信時を除く）。

3.15.2. Accept-Charset

説明

[必須リクエスト]
端末がリクエストを送出する際に、端末が受理できる文字セットの一覧を Web サーバに通知する。

定義

```
Accept-Charset    = “Accept-Charset” “:”
                    1#(charset [ “;” “q” “=” qvalue] )
```

値

Accept-Charset ヘッダには、端末で対応する文字セットと”*”を載せる。

制限

無し。

3.15.3. Accept-Encoding

説明

[必須リクエスト]
端末がリクエストを送出する際に、端末が受理できる Content-Encoding を Web サーバに通知する。Accept ヘッダフィールドに似ているが、端末でレスポンスとして受理できる Content-Encoding を通知するにとどまる。

定義

```
Accept-Encoding  = “Accept-Encoding” “:”
                   1#( condings [ “;” “q” “=” qvalue ] )
condings         = “gzip”
                 | “deflate”
                 | “identity”
                 | “x-gzip”
```

値

SSL/TLS を利用していない場合は、identity を与える。
SSL/TLS を利用している場合は、一部の端末において本ヘッダフィールドを与えない。

3.15.4. Accept-Language

説明

[必須リクエスト]
端末がリクエストを送出する際に、端末で選択されている言語の言語タグをを CP 様サーバに通知する。

定義

```
Accept-Language   = “Accept-Language” “:”
                    1#(language-range [ “;” “q” “=” qvalue] )
language-range    = ( ( 1*8ALPHA *( "-" 1*8ALPHA ) ) | "*" )
```

値

Accept-Language ヘッダには、端末で使用されている言語の言語タグを載せる。

制限

無し。

3.15.5. Accept-Ranges

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、Web サーバがレンジリクエストを受理できるか否かを端末に通知する。

定義

```
Accept-Ranges     = “Accept-Ranges” “:” acceptable-ranges
acceptable-ranges = “bytes” | “none”
```

値

Web サーバがレンジリクエストを受理できる場合には「`Accept-Ranges: bytes`」をレスポンスに与える。「`Accept-Ranges: bytes`」に加え Etag もしくは Last-Modified ヘッダフィールドを併用したレスポンスにおいて、回線断が発生した場合に端末から Web サーバへとレンジリクエストが送出される。

Web サーバがレンジリクエストを受理できない場合には本ヘッダフィールドを与えない、もしくは、「`Accept-Ranges: none`」を与える。

制限

Pull-HTTP で受理できるレンジリクエストはバイト単位のもの(バイトレンジリクエスト: Byte Range Request)のみであり、`Accept-Ranges` ヘッダフィールドの値として`"bytes","none"`以外の単位を与えた場合の動作は保証しない。

端末がレンジリクエストを送出するためには、「`Accept-Ranges: bytes`」に加え Etag もしくは Last-Modified ヘッダフィールドを併用したレスポンスでなければならない（Etag と Last-Modified が共存しても可)。

一部の端末では利用できない。

3.15.6. Authorization

説明

[リクエスト]

端末がリクエストを送出する際に、利用者の認証情報を Web サーバに通知する。サーバは当該情報を用いて利用者を認証することができる。

定義

```
Authorization           = “Authorization” “:” credentials
credentials             = “Basic” basic-credentials
                        | “Digest” digest-response
basic-credentials       = < RFC2617, 2. 参照のこと>
digest-response         = < RFC2617, 3.2.2. 参照のこと>
```

値

RFC2616 を参照のこと。

制限

入力文字は、半角英数字で最大 26bytes である。

3.15.7. Cache-Control

説明

[リクエスト]
端末がリクエストを送出する際に、キャッシュ制御を Web サーバに要求する。

[レスポンス]
Web サーバがレスポンスを返送する際に、キャッシュ制御について Pull-GW に指示すると共に、エンティティボディの端末への保存、メールに添付した送出、外部メモリへの転送について端末に指示する。

定義

```
Cache-Control              = “Cache-Control” “:” 1#cache-directive
cache-directive            = cache-request-directive
                           | cache-response-directive
cache-request-directive    = “no-cache”
cache-response-directive   = “public”
                           | “private” [“=”<”>1#field-name<”>]
                           | “no-cache” [“=”<”>1#field-name<”>]
                           | “no-store”
                           | “no-trasform”
                           | “must-revalidate”
                           | “proxy-revalidate”
                           | “max-age” “=” delta-seconds
                           | “s-maxage” “=” delta-seconds
```

値

端末にキャッシュ済みのリソースについて Web サーバに最新のものを要求する場合に Cache-Control に no-cache を載せてリクエストを送出する。

Web サーバがレスポンスに Cache-Control を載せた場合の、Pull-GW および端末の振る舞いについては、「3.2.2 キャッシュ制御」を参照のこと。

制限

Web サーバからのレスポンスにおいて、データの保存・送出・転送の制御は x-jphone-copyright ヘッダフィールド、ファイル拡張子、SMAF チャンクの記述よりも「Cache-Control: no-store」を優先する。

3.15.8. Connection

説明

[リクエスト]
端末がリクエストを送出する際に、端末がレスポンスを受信した後に永続コネクションを終了したい場合に、TCP セッションを閉じることを Web サーバに要求する。

定義

```
Connection        = “Connection” “:” 1#( connection-token )
connection-token  = token
```

制限

connection-token には“close”のみが記載できる。

3.15.9. Content-Encoding

説明

[レスポンス]

Web サーバがレスポンスを送信する際に、エンティティボディに適用したエンコード形式を端末に通知する。

定義

```
Content-Encoding = “Content-Encoding” “:” “deflate”
```

値

”deflate”ディレクティブを指定する。

制限

SSL/TLS 時のみ利用可。Accept-Encoding に従う必要があるが、指定可能な値は、“deflate”のみである。

3.15.10. Content-Language

説明

[リクエスト]
端末がリクエストを送出する際に、POST メソッドで送付するエンティティボディを記述している言語を Web サーバに通知する。バイリンガル機能を持つ端末では“ja”以外の言語コード(例えば、“en”)を通知することができる。

[レスポンス]
Web サーバがレスポンスを返送する際に、エンティティボディを記述している言語を端末に通知する。

定義

```
Content-Language = “Content-Language” “:” ( “ja” | “en” )
```

値

リクエストには少なくとも“ja”ディレクティブを指定でき、バイリンガル機能を持つ端末では、その他のディレクティブを指定することがある。

レスポンスには“ja”ディレクティブのみを指定できる。

制限

Web サーバからのレスポンスにおける Content-Language ヘッダフィールドの使用は将来の拡張のために用意したものである。現在、レスポンスには、“ja”ディレクティブのみを指定できる。レスポンスに“ja”ディレクティブ以外を記載した場合の端末の動作は保証しない。

本ヘッダフィールドは、一部の端末でのみ利用可能である。

3.15.11. Content-Length

説明

エンティティボディのサイズをバイト単位で表す。

[リクエスト]
端末がリクエストを送出する際に、POST メソッドが送付するエンティティボディのサイズを Web サーバに通知する。

[レスポンス]
Web サーバがレスポンスを返送する際に Content-Length エンティティヘッダフィールドを載せることで、エンティティボディのサイズを端末に通知する。

定義

```
Content-Length   = “Content-Length” “:” 1*DIGIT
```

値

バイト長を 10 進数で表記する。

制限

SSL/TLS 利用時は、本ヘッダフィールドを必須とする。

3.15.12. Content-Location

説明

[レスポンス]

Web サーバがレスポンスを返送する際に Content-Location エンティティヘッダフィールドを載せることで、エンティティボディの実体の位置を示す URI、および、エンティティのベース URI を端末に通知する。

定義

```
Content-Location = “Content-Location” “:”
                   | ( absoluteURI | relativeURI )
```

値

URI を与える。

制限

Content-Location ヘッダフィールドに加え、エンティティボディ内(マークアップ言語で記述したドキュメント)でもベース URI を与えている場合には、ドキュメント内に現れた URI をベース URI として使用される。

3.15.13. Content-Range

説明

[レスポンス]

Web サーバがレンジリクエストを受信した後にレスポンスを返送する際に、Content-Range エンティティヘッダフィールドを載せることで、要求された範囲を指定するエンティティボディの構成を端末に通知する。

定義

```
Content-Range    = “Content-Range” “:” content-range-spec
content-range-spec = byte-content-range-spec
byte-content-range-spec = bytes-unit SP
                  byte-range-resp-spec “/”
                 ( instance-length | “*” )
bytes-unit       = “bytes”
byte-range-resp-spec = ( first-byte-pos “-“ last-byte-pos )
                 | “*”
first-byte-pos   = 1*DIGIT
last-byte-pos    = 1*DIGIT
instance-length  = 1*DIGIT
```

値

Content-Range ヘッダフィールドはマルチパートのヘッダに記載する。

Pull-HTTP ではレンジリクエストの単位として“bytes”のみを指定できる。

制限

端末では Web サーバが受信したレンジリクエストには単一の範囲しか指定できない(RFC2616, 14.16, 19.2 に例示されている複数指定は出来ない)。なお、本ヘッダフィールドは一部の端末でのみ対応している。

表 3.15.13-1 レンジリクエストに対するレスポンス要件

レンジリクエストで単一の範囲を指定した場合
Content-Length ヘッダフィールドもレスポンスに記載すること。 Content-Length の値は byte-range-resp-spec の値と矛盾しないこと。

3.15.14. Content-Type

説明

[必須リクエスト]
端末がリクエストを送出する際に、POST メソッドで送付するエンティティボディのメディア型を Web サーバに通知する。

[必須レスポンス]
Web サーバがレスポンスを返送する際に、エンティティボディのメディア型を端末に通知する。

定義

```
Content-Type     = “Content-Type” “:” media-type
```

; media-type についてはAppendix.Bを参照のこと

値

Web サーバから端末へ返送するレスポンスでは、エンティティボディのデータの種類に応じて、Content-Type に表 3.15.14-1のメディア型を指定すること。

表 3.15.14-1 メディア型

カテゴリ	フォーマット	メディア型(MIME)
ページ記述	HTML	text/html
	XHTML	text/html application/xhtml+xml application/vnd.wap.xhtml+xml
	CSS	text/css
Java™	JAD	text/vnd.sun.j2me.app-descriptor
	JAR	application/java application/java-archive
モバイル ウィジェット	WGT	application/widget
	SWGT	application/x-s-widget
メディア	PNG	image/png
	JPEG	image/jpeg
	GIF	image/gif
	WBMP	image/vnd.wap.wbmp
	SMAF	application/x-smaf
	SMF	audio/midi
	SP-MIDI	audio/midi
	XMF	audio/xmf0 audio/xmf1
	MP4	video/3gpp
	SVG	image/svg+xml
	Flash	application/x-shockwave-flash
DRM	Forward Lock	application/vnd.oma.drm.message
OMA Download	Download Descriptor	application/vnd.oma.dd+xml
その他	text	text/plain
	vCard	text/x-vcard
	vBookmark	text/x-vbookmark
	vCalendar	text/x-vcalendar
	vMessage	text/x-vmessage
	vNote	text/x-vnote

制限

Web サーバから端末へ返送するレスポンスで、上記以外のメディア型を指定した場合の端末の動作は保証しない。

3.15.15. Cookie

説明

[レスポンス]
端末では Cookie を利用して statefull なセッションを実現できる。

Web サーバから Set-Cookie レスポンスを返送された場合に、端末がリクエストを送出する際に Cookie リクエストヘッダフィールドを載せることで、Web サーバとの間でセッションを実現する。サーバから Set-Cookie レスポンスを受理していない端末が自発的に Cookie リクエストを送出することは無い。

`name=value`の組を載せた Set-Cookie レスポンスをサーバから端末に送出し、その `name=value` の組を載せた Cookie リクエストを端末からサーバに送出することで、端末とサーバ間で statefull なセッションを実現する。

端末では以下のタイプの Cookie を実現する。

表 3.15.15-1 Cookie の型

型	Cookie の有効期限
一時型	端末ブラウザの終了と共に無効になる。 Max-Age, Expires 共に値が 0 の場合に一時型となる。
期限型	端末ブラウザの終了とは無関係に、指定された期限までは有効である。期限後は無効になる。 Max-Age, Expires いずれかの値が 0 でない場合に期限型となる。

尚、Set-Cookie レスポンスヘッダフィールドについては3.15.28を参照のこと。

定義

```
Cookie     = “Cookie” “:” cookie-ver 1*((“;”|”,”) cookie-val)
cookie-var = “$Version” “=” value
cookie-val = name “=” value “;” path
name               = 1*TEXT      ; アプリケーションに依存する。
value              = 1*TEXT      ; アプリケーションに依存する。
path               = “$Path” “=” path_name
```

値

Cookie は domain, name, path の３つ組をキーとしてセッションを管理する。

制限

同一キーの Cookie は新しい内容で更新(上書)きする。

3.15.16. Date

説明

[レスポンス]

Web サーバが端末にレスポンスを返送する際に、レスポンスを生成した日時を表す。RFC 1123 形式の日時フォーマットが送られなければいけない。

定義

```
Date       = “Date” “:” HTTP-date
```

値

[例]

```
Date: Fri, 1 Mar 2002 12:34:56 JST
```

制限

無し。

3.15.17. ETag

説明

[レスポンス]
エンティティのタグ名を与える。端末から If-None-Match リクエストヘッダフィールドを載せたリクエストが送出された後に、Web サーバがレスポンスを返送する際に ETag レスポンスヘッダフィールドを載せることで､要求されたエンティティに対応するタグ名を端末に通知する。

端末から要求されたエンティティ(If-None-Match ヘッダフィールドに載せたタグ名で識別)と同一のエンティティが Web サーバにある場合には、If-None-Match ヘッダフィールドに載せたタグ名と同一のタグ名を ETag に載せてレスポンスを送出する。この際、エンティティボディは空である。

端末から要求されたエンティティは Web サーバにあるエンティティより古い場合、If-None-Match ヘッダフィールドに載せたタグ名と異なるタグ名を ETag ヘッダフィールドに載せてレスポンスを送出する。この際、エンティティボディには要求されたエンティティを載せる。

このように、Web サーバ上でタグ名によりエンティティの世代管理を行うことで、無駄なエンティティの送出を抑制することができる。

定義

```
ETag             = “ETag” “:” entitiy-tag
entitiy-tag      = quoted-string
```

値

タグの書式はアプリケーションに依存する。

制限

HTTP/1.1 とは異なり、Pull-HTTP では weak entitiy tag を指定することは出来ない。“W/”プレフィックスを付けても端末は weak entitiy tag と解釈することはない。

3.15.18. Expires

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、レスポンスの有効期限を端末に通知する。端末では Expires ヘッダフィールドで与えられた期間内は、端末にキャッシュされたレスポンスを利用する。

定義

```
Expires   = “Expires” “:” HTTP-date
```

値

有効期限とみなす期日を RFC 1123 形式の日時フォーマットで与える。

制限

Web サーバはレスポンスの送出時点より前の時点を Expires ヘッダフィールドに指定してはいけない。

Web サーバはレスポンスの送出時点より 1 年を越えた先の時点を Expire ヘッダフィールドに指定してはいけない。

max-age ディレクティブを与えた Cache-Control ヘッダフィールドと Expires ヘッダフィールドが同一のレスポンスに載っていた場合は、Expires ヘッダフィールドを無視する。

3.15.19. Host

説明

[必須リクエスト]
端末がリクエストを送出する際に、リクエスト先のホスト名とポート(port)番号を Web サーバに通知する。

定義

```
Host      = “Host” “:” host-name [ “:” port ]
host-name = FQDN | ip-address
port      = 1*5DIGIT          ; 1..65535
```

値

ホスト名とポート番号はリクエストした URI に記載したホスト名とポート番号になる。URI にホスト名が記載されていない場合には、host-name は空となる。ポート番号の省略時には、ポート番号として 80 番を指定したものと解釈する。

制限

ポート番号は将来の拡張のために用意したものである。従って、HTML 文書に記載する URI では 80 番のポートのみを与えること。

Web サーバが端末からのリクエストを受け取る際、Host ヘッダフィールドが無かったら、Web サーバは必ず status-code 400(Bad Request)を載せてレスポンスを返さなければならない。

3.15.20. If-Modified-Since

説明

[リクエスト]
端末がリクエストを送出する際に、指定した日時以降に更新したエンティティを返送するよう Web サーバに要求する。サーバでは与えられた日時以降に要求されたエンティティが更新されている場合にはそのエンティティを端末へ返送しなければならない。

与えられた日時以前に更新されている、もしくは、全く更新されていない場合には、Web サーバは status-code 304(Not Modified)を載せ、エンティティボディが空のレスポンスを返す。キャッシュされているリソースを表示する。

Web サーバの日時よりも未来の日時を If-Modified-Since ヘッダフィールドに与えた場合には、Web サーバは If-Modified-Since ヘッダフィールドは載っていなかったものとして、status-code 200(OK)を載せたレスポンスを返す。

定義

```
If-Modified-Since       = “If-Modified-Since” “:” HTTP-date
```

値

日時を RFC 1123 形式の日時フォーマットで与える。

制限

無し。

3.15.21. If-None-Match

説明

[リクエスト]
端末がリクエストを送出する際に、要求するエンティティに与えるタグ名をWeb サーバに通知する。Web サーバは If-None-Match ヘッダフィールドの値により、エンティティボディを送出するか否かを決するとともに、ETag レスポンスヘッダフィールドに載せる値を決定する。

定義

```
If-None-Match    = “If-None-Match” “:” ( “*” | 1#entity-tag )
entitiy-tag      = quoted-string
```

値

タグの書式はアプリケーションに依存する。

制限

Pull-HTTP では HTTP/1.1 とは異なり、weak entitiy tag を指定することは出来ない。Web サーバで**“W/”**プレフィックスを付けても端末は weak entitiy tag と解釈しない。

3.15.22. If-Range

説明

[リクエスト]

端末がリクエストを送出する際に、Request-URI のうち If-Range および Rang ヘッダフィールドで指定した部分だけを返送するよう Web サーバに要求する。

定義

```
If-Range  = “If-Range” “:” ( entitiy-tag | HTTP-date )
```

値

制限

端末が取得中のリソースに ETag ヘッダフィールドがあれば ETag ヘッダフィールドを与え、無ければ Last-Modified ヘッダフィールドの日時を与えてリクエストする。

もし、ETag ヘッダフィールドに相当するエンティティがあれば、Web サーバは 206(Partial Content)のステータスコードを載せて Range ヘッダフィールドで与えられた範囲のデータを返送しなければならない。ETag ヘッダフィールドに相当するエンティティが無ければ、Web サーバは status-code 200(OK)を載せて要求されたデータ全体を返送しなければならない。

If-Range ヘッダフィールドは必ず Range ヘッダフィールドと対で Web サーバへと送出する。

3.15.23. Last-Modified

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、返送するリソースの最終更新日を端末に通知する。

定義

```
Last-Modified    = “Last-Modified” “:” HTTP-date
```

値

RFC 1123 形式の日時フォーマットの書式を返す。

制限

サーバが `Last-Modified` ヘッダフィールドに与える日時は、`Last-Modified` を載せたレスポンスを送出する時点より未来の日付を与えてはいけない。未来の日付を与えた場合の端末の動作は保証しない。

3.15.24. Location

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、Request-URI を Location ヘッダフィールドに記載した URI へリダイレクト(リクエストを振替え)するよう Pull-GW に要求する。

定義

```
Location  = “Location” “:” absoluteURI
```

値

Location レスポンスヘッダフィールドと共に用いる status-Code は 301(Moved Permanently), 302(Found), 303(See Other), 307(Temporary Redirect)のいずれかでなければならない。

[status-code]

301(Moved Permanently)

302(Found)

303(See Other)

端末が status-code 301,302,303 を受信した場合には、Location ヘッダフィールドに与えた URI に対してリダイレクトを行う。端末上でブックマーク登録やストレージエリアへの保存を行う場合に Request-URI に与えた URI を保存する。Location ヘッダフィールドに与えた URI を保存することはない。
GET メソッドに対してリダイレクトを要求した場合、GET メソッドでリクエストを送出する。POST メソッドに対してリダイレクトを要求した場合、GET メソッドでリクエストを送出する。

307(Temporary Redirect)

端末が status-code 307 を受信した場合には、Location ヘッダフィールドに与えた URI に対してリダイレクトを行う。端末上でブックマーク登録やストレージエリアへの保存を行う場合に Request-URI に与えた URI を保存する。Location ヘッダフィールドに与えた URI を保存することはない。
GET メソッドに対してリダイレクトを要求すると GET メソッドでリクエス

トを送出する。POST メソッドに対してリダイレクトを要求すると POST メソッドでリクエストを送出する。

制限

リダイレクトは利用者へのレスポンスの遅延を招く原因の一つでもあるため、冗長な利用は控えることが望ましい。

http スキームもしくは https スキームを URI で指定したリクエストに対しての、レスポンスに対してのみ Location ヘッダフィールドを利用できる。http および https 以外のスキームを URI で指定したリクエストに対しての、レスポンスには Location ヘッダフィールドを利用できない。

１セッションあたりのリダイレクト回数は３回までである、４回以上のリダイレクトはエラーメッセージを表示する。ただし、一部の端末では本制限が異なり、４回以上を許容する場合がある。

HTTP/1.1 のように「status-code 201(Created)とともに新たに生成したリソースを返す」といった使い方は出来ない。

3.15.25. Pragma

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、返送するリソースを端末および Pull-GW でキャッシュしないように指定する。

定義

```
Pragma     = “Pragma” “:” “no-cache”
```

値

“no-cache”ディレクティブのみを指定できる。その他のディレクティブを与えた場合の動作は保証しない。

制限

端末からサーバに「Pragma: no-cache」ヘッダフィールドを載せてリクエストを送出する場合には必ず「Cache-Control: no-cache」ヘッダフィールドも載せてリクエストを送出する。

3.15.26. Range

説明

[リクエスト]
端末がリクエストを送出する際に、Request-URI のうち Range ヘッダフィールドで指定した部分だけを返送するよう Web サーバに要求する。

定義

```
Range               = “Range” “:” ranges-specifier
range-specifier     = bytes-unit “=” byte-range-set
bytes-unit          = “bytes”
byte-range-set      = 1#( byte-range-spec | suffix-byte-range-spec )
byte-range-spec     = first-byte-pos “-“ [last-byte-pos]
first-byte-pos      = 1*DIGIT
last-byte-pos       = 1*DIGIT
suffix-byte-range-spec = “-” suffix-length
suffix-length       = 1*DIGIT
```

値

レンジリクエストで要求する範囲は byte 単位で指定する。範囲は以下のいずれかを指定する。

表 3.15.27-1 Range リクエストに記載する要求範囲

byte-range-spec	開始点～終了点をデータ先頭からのバイト数
suffix-byte-range-spec	データ終端からのバイト数

bytes-unit として byte 以外の単位(例.bit, word)を指定した場合の動作は保証しない。

制限

Rangeヘッダフィールドは必ずIf-Rangeヘッダフィールドと対でサーバへと送出する。

byte-range-spec で与えた開始点(first-byte-pos)がリクエスト対象となるデータのサイズを超えた場合や、終了点の指定と同じかより大きな値となる場合、にはその動作は保証しない。byte-range-spec で与えた終了点(last-byte-pos)がリクエスト対象データのサイズと同じか超えた場合には、last-byte-pos として「対象データー１」を与えたものと見なす。byte-range-spec で last-byte-pos を与えていない場合にも、last-byte-pos として「対象データー１」を与えたものと見なす。

suffix-byte-range-spec で与えた終端からのバイト数(suffix-length)がリクエスト対象データのサイズを超えた場合には対象データ全体を指定したものと見なす。

3.15.27. Referer

説明

[リクエスト]
端末がリクエストを送出する際に、Request-URI へのリンク元となる URI を Web サーバに通知する。

Web サーバで Referer ヘッダフィールドを積極的に監視することで、自サイトのコンテンツを他サイトから勝手にリンクされ、あたかも他サイトのコンテンツであるかのように見せかけることを防ぐことができる。

定義

```
Referer    = “Referer” “:” ( absolulteURI | relativeURI )
```

値

相対 URI(relativeURI)である場合には、Request-URI からの相対値であるものとしてリンク元を算出する。

制限

Referer ヘッダフィールドに与える URI にはフラグメント(“#” <fragment>)を記載することは無い。

一部の端末では対応しない。

3.15.28. Set-Cookie

説明

[レスポンス]

Web サーバがレスポンスを返送する際に Set-Cookie レスポンスヘッダフィールドを載せることで、Cookie によるセッション管理の開始を端末に要求する。

name=value の組を載せた Set-Cookie レスポンスをサーバから端末に送出し、その name=value の組を載せた Cookie リクエストを端末からサーバに送出することで、端末と Web サーバ間でのセッション情報を保持する。

尚、Cookie については、3.15.15 Cookieの項を参照のこと。

定義

```
Set-Cookie= “Set-Cookie” “:” cookies
cookies   = 1#cookie
cookie    = name-value *( “;” cookie-av )
name-value= name “=” value
name      = 1*TEXT      ; アプリケーションに依存する。
value     = 1*TEXT      ; アプリケーションに依存する。
cookie-av = comment | domain | max-age | path | security
          | version | expires
comment   = “Comment” “=” value
domain    = “Domain” “=” domain_name
max-age   = “Max-Age” “=” 1*DIGITS
path      = “Path” “=” path_name
security  = “Secure”
version   = “Version” “=” 1*DIGIT
expires   = “expires” “=” date “;”
```

値

Cookie は domain, name, path の３つ組をキーとしてセッションを管理する。

制限

同一キーの Cookie は新しい内容で更新(上書)きする。

Set-Cookie に Secure が与えられていた場合には、HTTPS(SSL/TLS)プロトコルで接続する場合にのみ Cookie を返すことが出来る。

3.15.29. Transfer-Encoding

説明

[リクエスト]

端末がリクエストを送出する際に Transfer-Encoding 一般ヘッダフィールドを載せることで、POST メソッドにて送付するメッセージボディが、エンティティボディに”chunked”転送コーディングを適用したものであることを Web サーバに通知する。

[レスポンス]

Web サーバがレスポンスを返送する際に Transfer-Encoding 一般ヘッダフィールドを載せることで、エンティティボディに”chunked”転送コーディングを適用したメッセージボディであることを端末に通知する。

定義

```
Transfer-Encoding   =“Transfer-Encoding”“:”1#transfer-coding
transfer-coding   = “chunked”
```

値

“chunked”のみ指定できる。これ以外の値を指定した場合には端末ではエラーメッセージを表示する。

制限

Pull-HTTP では TE リクエストヘッダフィールドが無いため、Transfer-Encoding の値として“chunked”のみ指定できる。なお、”chunked”については全て小文字で記述しなければならない。

3.15.30. User-Agent

説明

[必須リクエスト]
端末がリクエストを送出する際に User-Agent リクエストヘッダフィールドを載せることで、送出元の端末の構成情報を Web サーバに通知する。

シリアル番号トークンは、利用者が端末で製造番号送出を許可した場合のみ追加する。

また、本定義に準拠しない端末、製造番号を送出しない端末も存在する。

定義

User-Agent = “User-Agent” “:” user-agent-value
User-agent-value = “SoftBank-Product”
SoftBank-Product = (“Vodafone” | “SoftBank”) “/”
UE-Generation-Version “/”
UE-Product-Name “/”
UE-Product-Version
[“/” UE-IMEI-Number] SP
UE-Application-Type “/”
UE-Application-Name “/”
UE-Application-Version
(SP UE-Ext-Information)
UE-Generation-Version= 1*DIGIT “.” 1*DIGIT
UE-Product-Name = *TEXT
UE-Product-Version = (2*TEXT (“J” | “G”) 3*TEXT) | (*TEXT)
UE-IMEI-Number = “SN” 15(DIGIT)
UE-Appication-Type = (“Browser” | “Java” | “Flash” | “Widgets”)
UE-Application-Name = *TEXT
UE-Application-Version = 1*DIGIT “.” 1*DIGIT [“.” 1*DIGIT]
UE-Ext-Information = *TEXT

値

[例]

User-Agent: SoftBank/1.0/111XX/XXJ001/SN1234567890ABCDEF SP
Browser/XX-Browser/1.0 SP Profile/MIDP-2.0 SP
Configration/CLDC-1.1

User-Agent: Vodafone/1.0/V111XX/XXJ001/SN1234567890ABCDEF SP
Browser/XX-Browser/1.0 SP Profile/MIDP-2.0 SP
Configration/CLDC-1.1

SoftBank-Product

UE-Generation-Version

端末の世代を表わす。

UE-Product-Name

端末の機種名を記載する。通常は、x-jphone-msname に与えられる値と同一のものになる。なお、User-Agent が“SoftBank”で始まる場合、先頭に“V”は付加されない。

UE-Product-Version

端末のバージョンを表す。

UE-IMEI-Number

“SN”に続けて、当該端末のシリアル番号(15 桁の英数字)を記載する。
シリアル番号は携帯端末利用者が端末を操作して製造番号の送出を許可して初めて送出する。

UE-Application-Type

通信を行っているアプリケーションの種類を表す。
Browser: 通常のブラウザでの通信
Java: S!アプリ中からの通信
Flash: Flash®から通信
Widgets: モバイルウィジェットからの通信

UE-Application-Name

通信を行っているアプリケーションの名称を表す。

UE-Application-Version

通信を行っているアプリケーションのバージョンを表す。

UE-Ext-Information

Java の profile 名など付加情報を表す。なお、UE-Application-Type が Java の場合は Java の profile 名などが必ず付加されるが、それ以外の場合については特に規定しない。

制限

アプリケーション毎に User-Agent の値が異なるので注意すること。

利用者識別子の送出と製造番号の送出は同期しないので注意すること。

3.15.31. WWW-Authenticate

説明

[レスポンス]

Web サーバがレスポンスを返送する際に、利用者認証を行うためのチャレンジデータを端末に通知するとともに、認証の返答を端末に要求する。

定義

```
WWW-Authenticate = “WWW-Authenticate” “:” 1#challenge
```

値

challenge の値については、Basic 認証と Digest 認証の項を参照のこと。

WWW-Authenticate ヘッダフィールドをレスポンスに載せる際は必ず status-code 401(Unauthorized)と共に載せなければならない。

制限

status-code≠401 である場合には、WWW-Authenticate ヘッダフィールドが載っていても WWW-Authenticate ヘッダフィールドを無視する。

status-code 401 であるにも係わらず WWW-Authenticate ヘッダフィールドが載っていない場合には、端末はエラーメッセージを表示する。

いずれの場合も端末は認証要求もエンティティの表示も行わない。

一部の端末は、Digest 認証に対応しない。

3.15.32. x-jphone-color

説明

[必須リクエスト]
端末がリクエストを送出する際に、送出元の端末のメインディスプレイで発色可能な色数についての情報を Web サーバに通知する。

定義

```
x-jphone-color   = “x-jphone-color” “:” depth
depth                    = color-depth
color-depth              = “C65536”    ; 65536 色天然色
                         | “C262144”   ; 262144 色天然色
                         | “C16777216”; 16777216 色天然色
```

値

“C65536”：65536 階調までの天然色を表示可能なことを表わす。
“C262144”：262144 階調までの天然色を表示可能なことを表わす。
“C16777216”：16777216 階調までの天然色を表示可能なことを表わす。

制限

このフィールドの値は端末の機種ごとに固定した値である。
本ヘッダフィールドは HTTP/1.1 の拡張ヘッダ扱いとなる。
SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.33. x-jphone-copyright

説明

[レスポンス]

Web サーバがレスポンスを返送する際に x-jphone-copyright レスポンスヘッダフィールドを載せることで、返送するデータについて、端末上の不揮発性メモリへの**保存**、メールに添付しての**送信**、外部メモリへの外部**転送**、の可否を端末に指定する。

データの保存/送信/外部転送については「3.2.1 著作物保護」を参照のこと。

定義

```
x-jphone-copyright      = “x-jphone-copyright” “:”
                          ( “*” | 1#control)
control           = “no-store” ; 保存/送信/外部転送不可
                  | “no-transfer” ; メール添付の送信不可
                  | “no-peripheral” ; メール添付の送信不可
```

値

“no-store”、 “no-transfer”、“no-peripheral”のいずれかを指定のこと。

制約

3GC 型端末における x-jphone-copyright レスポンスヘッダとファイル名拡張子による著作物保護制御は、過去の端末向けのコンテンツを 3GC 型端末でも扱うための、互換性のために残すものである。この為、**新規に 3GC 端末向けコンテンツを作成する場合は、過去の著作物保護制御を使用せず、オリジンサーバから Forward Lock 方式で配信することが望ましい。**

SSL 時は利用できない。

3.15.34. x-jphone-display

説明

[必須リクエスト]
端末がリクエストを送出する際に、送出元の端末の画面サイズを Web サーバに通知する。

定義

```
x-jphone-display = “x-jphone-display” “:” width “*” height
height           = *DIGIT            ; 画面高さ
width            = *DIGIT            ; 画面幅
```

値

メインディスプレイの物理サイズの高さと幅についてピクセル単位で与える。

制約

SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.35. x-jphone-msname

説明

[必須リクエスト]
端末がリクエストを送出する際に、送出元の端末の機種名称を Web サーバに通知する。

定義

```
x-jphone-msname  = “x-jphone-nsname” “:” ms-name
ms-name          = ( (“V” 3DIGIT 1*TEXT )
                   | ( *TEXT ) )
                   [“_” LOALPHA]
```

値

端末の機種を識別する値を与える。

制限

ms-name は端末の機種ごとに固定した値である。
SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.36. x-jphone-region

説明

[リクエスト]
端末がリクエストを送出する際に、送出元の端末の「システム選択」に設定されている情報を通知する。

定義

```
x-jphone-region  = “x-jphone-region” “:” region-code
region-code      = (“44020” | “fffff”)
```

値

送出元端末がアクセスしている地域を識別する値を与える。

表 3.15.36-1 x-jphone-region の値

x-jphone-region	内容
44020	送出元の端末が日本にあることを意味する。
fffff	送出元の端末が海外にあることを意味する。

なお、将来の端末では、送出元の端末が海外にある場合に、海外の地域にそれぞれ対応した値を送出する予定である。

制限

利用者が利用者識別子の送出を許可した場合のみ送出される。
SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.37. x-jphone-smaf

説明

[リクエスト]

端末がリクエストを送出する際に x-jphone-smaf リクエストヘッダフィールドを載せることで、送出元の端末で再生できる SMAF の属性情報を Web サーバに通知する。

定義

```
x-jphone-smaf    = “x-jphone-smaf” “:” smaf
smaf             = melody [ “/” pcm [ “/” grafic [ “/” rs ]]]
melody           = “40” | “64” | “128” ; 和音数
pcm              = “pcm”               ; ADPCM 対応
grafic           = “grf”               ; カラオケ(ｼﾝｸﾛ)対応
rs               = “rs”                ; カラオケ(ｼﾝｸﾛ)拡張対応
```

値

実際にありうる組み合わせを以下に示す。

表 3.15.37-1 x-jphone-smaf の値

x-jphone-smaf	和音数	ADPCM 対応	ｶﾗｵｹ(ｼﾝｸﾛ)	ｶﾗｵｹ(ｼﾝｸﾛ)拡張
40/pcm	40	○	×	×
40/pcm/grf/rs	40	○	○	○
64/pcm	64	○	×	×
64/pcm/grf/rs	64	○	○	○
128/pcm	128	○	×	×
128/pcm/grf/rs	128	○	○	○

制限

SMAF に対応している機種からのリクエストにのみ添付される。

SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.38. x-jphone-uid

説明

[リクエスト]
端末がリクエストを送出する際に x-jphone-uid リクエストヘッダフィールドを載せることで、利用者の識別子を Web サーバに通知する。利用者の識別子は**ユーザ ID** と呼び、本稿では以降 UID と記載する。

初めて端末を利用する際に UID の送出を許可すると、以降、端末の設定のみで x-jphone-uid ヘッダフィールド上にユニークな識別子を載せることができる。

定義

```
x-jphone-uid     = “x-jphone-uid” “:” uid
uid              = 16( ALPHA | DIGIT )
```

値

uid のフォーマットは 16 桁の英数字である。

制限

Pull-HTTP リクエストに x-jphone-uid リクエストヘッダを添付する契機は、端末自身での設定(利用者の同意)による。
利用者が UID の送出に同意していない場合には、x-jphone-uid ヘッダフィールドそのものをリクエストに添付しない。
SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.39. x-wap-profile

説明

[リクエスト]
端末の CPI を参照するための情報を CP 様サーバに通知する。

定義

```
x-wap-profile     = “x-wap-profile” “:” 1#reference
reference         = <”> ( absoluteURI | profile-diff-name ) <”>
absoluteURI       = <a URI as defined by RFC2396>
profile-diff-name = profile-diff-seq “-” profile-diff-digest
profile-diff-seq  = ( “1” | “2” | “3” | “4” | “5” | “6” |
                      “7” | “8” | “9” ) *DIGIT
profile-diff-digest = *OCTET “;” <MD5 message digest encoded by base64>
```

値

x-wap-profile ヘッダに載せる値は、

- 端末のCPIを参照するための絶対URI
- x-wap-profile-diffヘッダに含まれるCPIを参照するためのキーワード

のいずれかの組み合わせである。x-wap-profile-diff ヘッダを参照する際には、シーケンス番号の順に適用する。

制限

2007 年以降発売の端末では、本ヘッダを送出しない端末がある。

3.15.40. x-wap-profile-diff

説明

[リクエスト]

x-wap-profile ヘッダの絶対 URI で参照される端末の CPI よりも優先される CPI 情報を CP 様サーバに通知する。

定義

```
x-wap-profile-diff  = “x-wap-profile-diff” “:”
                        profile-diff-seq “:” profile-desc
profile-diff-seq    = ( “1” | “2” | “3” | “4” | “5” | “6” |
                        “7” | “8” | “9” ) *DIGIT
profile-desc        = <XML document containing subset profile>
```

値

x-wap-profile-diff ヘッダの値は、x-wap-profile ヘッダから参照されるシーケンス番号と CPI の差分情報から構成される。

制限

2007 年以降発売の端末では、本ヘッダを送出しない端末がある。

3.15.41. x-s-bearer

説明

[リクエスト]
端末がリクエストを送出する際に、端末が利用しているネットワークの種別を Web サーバに通知する。

定義

```
x-s-bearer = "x-s-bearer" ":" bearer
bearer     = "WLAN"
```

値

WLAN が与えられている際、利用しているネットワークが無線 LAN であることを示す。

制限

3G 網を利用している場合、本ヘッダフィールドを付与しない。
SSL を利用している場合、本ヘッダフィールドを付与しない。

3.15.42. x-s-display-info

説明

[リクエスト]

端末がリクエストを送出する際に、送出元端末ブラウザのコンテンツ表示領域、半角文字表示数及び、テキストブラウズの設定状態を Web サーバに通知する。

定義

```
x-s-display-info = “x-s-display-info” “:” contents-field
                 “/” display-character “/”text-browse

contents-field   = width“*”height
width            = *DIGIT            ; コンテンツ表示領域幅
height           = *DIGIT            ; コンテンツ表示領域高さ

display-character= width-char“*”height-char
width-char       = *DIGIT            ; 1 行の半角文字表示数
height-char      = *DIGIT            ; 行数

text-browse      = TB“|”TC
                 = “TB”              ; 画像取得 ON
                 | “TC”              ; 画像取得 OFF
```

値

“width”：ブラウザでのコンテンツ表示領域幅をピクセルで表す。

“height”： ブラウザでのコンテンツ表示領域高さをピクセルで表す

“width-char”：端末で設定されているフォントサイズでの文字数を表す。

“height-char”：端末で設定されているフォントサイズでの行数を表す。

“TB”：端末のテキストブラウズ設定、画像取得 ON を表わす。

“TC”： 端末のテキストブラウズ設定、画像取得 OFF を表わす。

制限

2007 年以降発売の一部端末にて、本ヘッダを送出する。

3.15.43. x-s-unique-id

説明

[リクエスト]
端末からのリクエスト送出時、リクエスト送出元端末が固有のバリエーションモデルである事を示す。

定義

```
x-s-unique-id    = “x-s-unique-id” “:” unique-name

unique-name      = token
token            = *TEXT              ; 端末モデル識別子
```

値

端末モデルを識別する値を与える。

制限

2007 年以降発売の一部端末にて、本ヘッダを送出する。

4. SSL/TLS

4.1. 暗号化プロトコル

ソフトバンク携帯電話向けウェブで利用できるSSL/TLSプロトコルを以下に掲げる。

表 4.1-1 利用可能なプロトコル

	利用可否
SSL 2.0	×
SSL 3.0	○
TLS 1.0	○
TLS 1.1	×
TLS 1.2	×

ソフトバンク携帯電話向けウェブではSSL/TLSにおいて以下の相手先認証およびデータの秘匿処理を行う。

- 「クライアントによるサーバの認証」または「サーバの公開情報をクライアントへ送信」
- セッション暗号化

4.2. SSL/TLSの範囲

端末のSSL/TLS適用範囲は、端末⇔Webサーバとなる。

4.3. ルート証明書

SSL/TLS 利用時に使用可能なルート証明書の一覧を表 4.3-1に示す。弊社端末向けサービスでは、Pull-GW と各端末にインストールされている。証明書は各端末へ出荷時にインストールされており、新規にルート証明書を追加することは出来ない。

表 4.3-1 ルート証明書

発行者	ルート証明書	端末
エントラスト ジャパン株式会社	Entrust Secure Server CA	△
	Entrust CA (2048)	△
	Entrust Root CA	△
	Entrust Root CA - G2	△
サイバートラスト 株式会社	GTE CyberTrust Global Root	○
	Baltimore CyberTrust Root	○
	CyberTrust Global Root	△
	Verizon Global Root CA	△
ジェイサート株式会社	Go Daddy Class 2 Certification Authority Root Certificate	△
	Go Daddy Class 2 Certification Authority Root Certificate - G2	△
	Starfield Class 2 Certification Authority Root Certificate	△
	Starfield Class 2 Certification Authority Root Certificate - G2	△
セコムトラスト システムズ株式会社	Security Communication RootCA1	○
	Security Communication RootCA2	△
株式会社 コモドジャパン	AAA Certificate Services	△
	AddTrust Extemal CA Root	△
	COMODO Certification Authority	△
合同会社 シマンテック・ウェブサイト セキュリティ	VeriSign Class 3 Primary CA	○
	VeriSign Class 3 Primary CA - G2	○
	VeriSign Class 3 Primary CA - G3	△
	VeriSign Class 3 Primary CA - G5	△
	VeriSign Universal Root CA	△
	Equifax Secure Certificate Authority	○
	Equifax Secure eBusiness CA-1	○
	GeoTrust Global CA	○
	GeoTrust Primary Certification Authority	△
	thawte Primary Root CA	△

発行者	ルート証明書	端末
EMC ジャパン株式会社	RSA Security 2048 V3	○
	ValiCert Class 3 Policy Validation Authority	○
GMO グローバルサイン株式会社	GlobalSign Root CA	△
	GlobalSign Root CA-R3	△

○：対応、△：一部端末のみ対応

端末毎の詳細は別冊のドキュメント「コンテンツ開発ガイド[SSL 証明書一覧]」に記述する。

4.4. 文字エンコーディング

文字エンコーディングは Accept-Charset で指定する文字エンコードすること。

4.5. http と https の混在

https で取得したコンテンツ内に、インラインリソースが http で記述されていた場合でも、インラインリソースは取得される。
なお、端末に表示される SSL ピクト(鍵マーク)は、ルートドキュメント取得時のスキームに依存する。

4.6. 暗号化アルゴリズム

ソフトバンク携帯電話向けウェブでは下表の暗号化アルゴリズムとハッシュ関数を組み合わせる。実際に利用できる組み合わせはAppendix.Cに掲載する。

表 4.6-1 暗号化アルゴリズムの一覧

		端末
セッション鍵交換(公開鍵暗号)	RSA	○
	DH	×
	FORTEZZA	×
サーバ認証(公開鍵暗号)	RSA	○
	DSS/DSA	×
セッション暗号化(共通鍵暗号)	RC4	○
	RC2	×
	DES/3DES	×
	IDEA	×
	AES	×
メッセージダイジェスト(hash 関数)	MD5	○
	SHA1	○
	SHA256	×

共通鍵 128bit、公開鍵 2048bit を最大鍵長とする。

Appendix.A. ヘッダフィールド一覧

表 A-1 ヘッダフィールド一覧

要素	項目	概要
一般ヘッダ	Cache-Control	キャッシュ制御
	Connection	コネクションを張る方法を指定
	Date	レスポンスを生成した日時
	Pragma	Pull-GW でキャッシュしないように指定
	Transfer-Encoding	転送コーディングの指定
リクエストヘッダ	Accept	受理できる MIME 型
	Accept-Charset	受理できるキャラクタセット
	Accept-Encoding	受理できるエンコーディング
	Accept-Language	受理できる言語
	Authorization	利用者認証情報
	Cookie	Cookie 情報
	Host	要求先ホスト名とポート番号
	If-Modified-Since	更新済コンテンツの返送日を指定
	If-None-Match	要求するエンティティのタグ名
	If-Range	レンジリクエスト
	Range	レンジリクエスト
	Referer	参照元の URI※1
	User-Agent	端末機種を識別する情報
レスポンスヘッダ	Accept-Ranges	レンジリクエストの可否を指定※1
	Etag	エンティティタグを指定
	Location	リダイレクトを指定
	Set-Cookie	Cookie 情報を指定
	WWW-Authenticate	利用者認証の要求※2
エンティティヘッダ	Content-Encoding	エンコード方式を指定
	Content-Language	端末の使用言語を指定※1
	Content-Length	エンティティボディのサイズ
	Content-Location	ベース URI
	Content-Range	レンジリクエストに対する返答※1
	Content-Type	エンティティボディのメディア型

要素	項目	概要
	Expires	レスポンスの有効期限を指定
	Last-Modified	最終更新日
拡張ヘッダ (リクエスト)	x-jphone-color	端末メインディスプレイで発色可能な色数
	x-jphone-display	端末メインディスプレイの画面サイズ
	x-jphone-msname	端末機種名称
	x-jphone-region	端末のアクセス元の地域情報
	x-jphone-smaf	SMAF ファイルの種別
	x-jphone-uid	利用者識別子(UID)
	x-s-bearer	端末が利用しているネットワーク種別
	x-s-display-info	ブラウザのコンテンツ表示領域、半角表示文字数、テキストブラウズ設定※1
	x-s-unique-id	一部端末のバリエーションモデル識別子※1
拡張ヘッダ (レスポンス)	x-jphone-copyright	保存、送出・転送、外部転送の可否を指定
WAP 拡張ヘッダ	x-wap-profile	端末の CPI を参照するための情報を通知※1
	x-wap-profile-diff	x-wap-profile よりも優先される CPI 情報を通知※1

※1：一部の端末では対応しない。

※2：一部の端末では Digest 認証に対応しない。

Appendix.B. MIME 型一覧

Web サーバでは以下の MIME 型を設定していただく。

表 B-1 メディア型

カテゴリ	フォーマット	メディア型(MIME)	サイズ
ページ記述	HTML	text/html	48k
	XHTML	text/html application/xhtml+xml application/vnd.wap.xhtml+xml	48k
	CSS	text/css	48k
Java™	JAD	text/vnd.sun.j2me.app-descriptor	6k
	JAR	application/java application/java-archive	1M
モバイル ウィジェット	WGT	application/widget	500k
	SWGT	application/x-s-widget	500k
メディア	PNG	image/png	300k
	JPEG	image/jpeg	300k
	GIF	image/gif	300k
	WBMP	image/vnd.wap.wbmp	300k
	SMAF	application/x-smaf	300k
	SMF	audio/midi	300k
	SP-MIDI	audio/midi	300k
	XMF	audio/xmf0 audio/xmf1	300k
	MP4	video/3gpp	300k
	SVG	image/svg+xml	300k
	Flash	application/x-shockwave-flash	150k
DRM	Forward Lock	application/vnd.oma.drm.message(*1)	500k(*2)
OMA Download	Download Descriptor	application/vnd.oma.dd+xml	48k
その他	text	text/plain	48k
	vCard	text/x-vcard	48k
	vBookmark	text/x-vbookmark	48k
	vCalender	text/x-vcalender	48k
	vMessage	text/x-vmessage	48k
	vNote	text/x-vnote	48k

*1：boundary の記述が必要な点に留意すること。

*2：元のメディアにより異なる(500k はモバイルウィジェットを Forward Lock 配信する場合)。

端末個別の仕様により、上記よりも小さいサイズが上限となる場合もある。

Appendix.C. Cipher suite 一覧

以下に各サービスで利用できる cipher suite を一覧する。

表 C-1 cipher suite 一覧(suite 名の先頭の SSL_や TLS_は省略している)

cipher suite	端末	SSL	TLS
NULL_WITH_NULL_NULL	×	v3	v1
RSA_WITH_NULL_MD5	×	v3	v1
RSA_WITH_NULL_SHA	×	v3	v1
RSA_EXPORT_WITH_RC4_40_MD5	×	v3	v1
RSA_WITH_RC4_128_MD5	○	v3	v1
RSA_WITH_RC4_128_SHA	○	v3	v1
RSA_EXPORT_WITH_RC2_CBC_40_MD5	×	v3	v1
RSA_WITH_IDEA_CBC_SHA	×	v3	v1
RSA_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
RSA_WITH_DES_CBC_SHA	×	v3	v1
RSA_WITH_3DES_EDE_CBC_SHA	×	v3	v1
DH_DSS_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
DH_DSS_WITH_DES_CBC_SHA	×	v3	v1
DH_DSS_WITH_3DES_EDE_CBC_SHA	×	v3	v1
DH_RSA_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
DH_RSA_WITH_DES_CBC_SHA	×	v3	v1
DH_RSA_WITH_3DES_EDE_CBC_SHA	×	v3	v1
DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
DHE_DSS_WITH_DES_CBC_SHA	×	v3	v1
DHE_DSS_WITH_3DES_EDE_CBC_SHA	×	v3	v1
DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
DHE_RSA_WITH_DES_CBC_SHA	×	v3	v1
DHE_RSA_WITH_3DES_EDE_CBC_SHA	×	v3	v1
DH_anon_EXPORT_WITH_RC4_40_MD5	×	v3	v1
DH_anon_WITH_RC4_128_MD5	×	v3	v1
DH_anon_EXPORT_WITH_DES40_CBC_SHA	×	v3	v1
DH_anon_WITH_DES_CBC_SHA	×	v3	v1
DH_anon_WITH_3DES_EDE_CBC_SHA	×	v3	v1

cipher suite	端末	SSL	TLS
RSA_EXPORT1024_WITH_RC4_56_MD5	×	v3	v1
RSA_EXPORT1024_WITH_DES_CBC_SHA	×	v3	v1
RSA_EXPORT1024_WITH_RC4_56_SHA	×	v3	v1
FORTEZZA_DMS_WITH_NULL_SHA	×	v3	N/A
FORTEZZA_DMS_WITH_FORTEZZA_CBC_SHA	×	v3	N/A
RC4_128_WITH_MD5	×	v2	N/A
RC4_128_EXPORT40_WITH_MD5	×	v2	N/A
RC2_CBC_128_CBC_WITH_MD5	×	v2	N/A
RC2_CBC_128_CBC_EXPORT40_WITH_MD5	×	v2	N/A
IDEA_128_CBC_WITH_MD5	×	v2	N/A
DES_64_CBC_WITH_MD5	×	v2	N/A
DES_192_EDE3_CBC_WITH_MD5	×	v2	N/A
RSA_WITH_AES_128_CBC_SHA	×	N/A	v1.1
DH_DSS_WITH_AES_128_CBC_SHA	×	N/A	v1.1
DH_RSA_WITH_AES_128_CBC_SHA	×	N/A	v1.1
DHE_DSS_WITH_AES_128_CBC_SHA	×	N/A	v1.1
DHE_RSA_WITH_AES_128_CBC_SHA	×	N/A	v1.1
DH_anon_WITH_AES_128_CBC_SHA	×	N/A	v1.1
RSA_WITH_AES_256_CBC_SHA	×	N/A	v1.1
DH_DSS_WITH_AES_256_CBC_SHA	×	N/A	v1.1
DH_RSA_WITH_AES_256_CBC_SHA	×	N/A	v1.1
DHE_DSS_WITH_AES_256_CBC_SHA	×	N/A	v1.1
DHE_RSA_WITH_AES_256_CBC_SHA	×	N/A	v1.1
DH_anon_WITH_AES_256_CBC_SHA	×	N/A	v1.1
RSA_WITH_NULL_SHA256	×	N/A	v1.2
RSA_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
RSA_WITH_AES_256_CBC_SHA256	×	N/A	v1.2
DH_DSS_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
DH_RSA_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
DHE_DSS_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
DHE_RSA_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
DH_anon_WITH_AES_128_CBC_SHA256	×	N/A	v1.2
DH_DSS_WITH_AES_256_CBC_SHA256	×	N/A	v1.2
DH_RSA_WITH_AES_256_CBC_SHA256	×	N/A	v1.2
DHE_DSS_WITH_AES_256_CBC_SHA256	×	N/A	v1.2
DHE_RSA_WITH_AES_256_CBC_SHA256	×	N/A	v1.2
DH_anon_WITH_AES_256_CBC_SHA256	×	N/A	v1.2