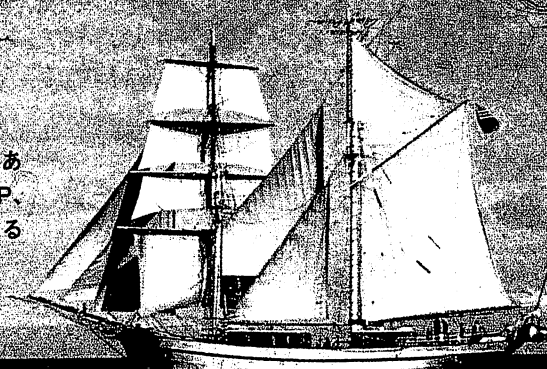


# 特集1 Part1

## インターネットのやり取りを徹底図解

Part1ではWebアプリケーションの仕組みを知るために、動作基盤であるネットワーク上でどのようなやり取りをしているのかを見ます。HTTP、TCP、IP/イーサネットのレイヤーごとに考えることで理解が容易になるでしょう。

中條 将典



Webアプリケーションは今や、生活や仕事にすっかり定着しました。昔からあるECサイトや検索サイトはもちろん、ブログや最近のはやりであるTwitterなど、人気のあるインターネット上のサービスはすべてWebアプリケーションとして実装されています。さらに、こうしたWebアプリケーションを利用するユーザー側のプラットフォームは、パソコンから、携帯電話、スマートフォンへと大きな広がりを見せています。これから何らかのアプリケーションを開発しようとする場合、Webアプリケーションとして実装することは有力な選択肢になります。この特集では、Webアプリケーション開発に興味があるけれど、どこから手を付ければよいかわからない、という人向けに、開発に必要な知識を基礎から説明します。

### Web APIもNoSQLも押さえるべき基本は同じ

この特集では、Part1でWebアプリケーションの動作する基盤であるネットワーク上でやり取りを説明。Part2でWebアプリケーション開発に利用する特有の手法について解説し、Part3でサンプルとして会員制サイトのプログラミングを通じて実システム開発の一端に触れてみます。

Web上では、Web APIやNoSQLなど新しい技術がどんどん登場しています。しかし、これらの技術を使う場合でも押さえるべき基本は同じです。例えば、Twitter、Google、Yahoo! JAPAN、Amazonなどが提供するWeb API。これらを利用すれば高機能なWebサービスを組み込んだサイトを簡単に作れるとされていますが、Web APIを活用するにはやはりWebプログラミングの知識が必要です。

最近になってクラウドコンピューティングに関連して聞かれるようになった「NoSQL (Not Only SQL)」も同様です。

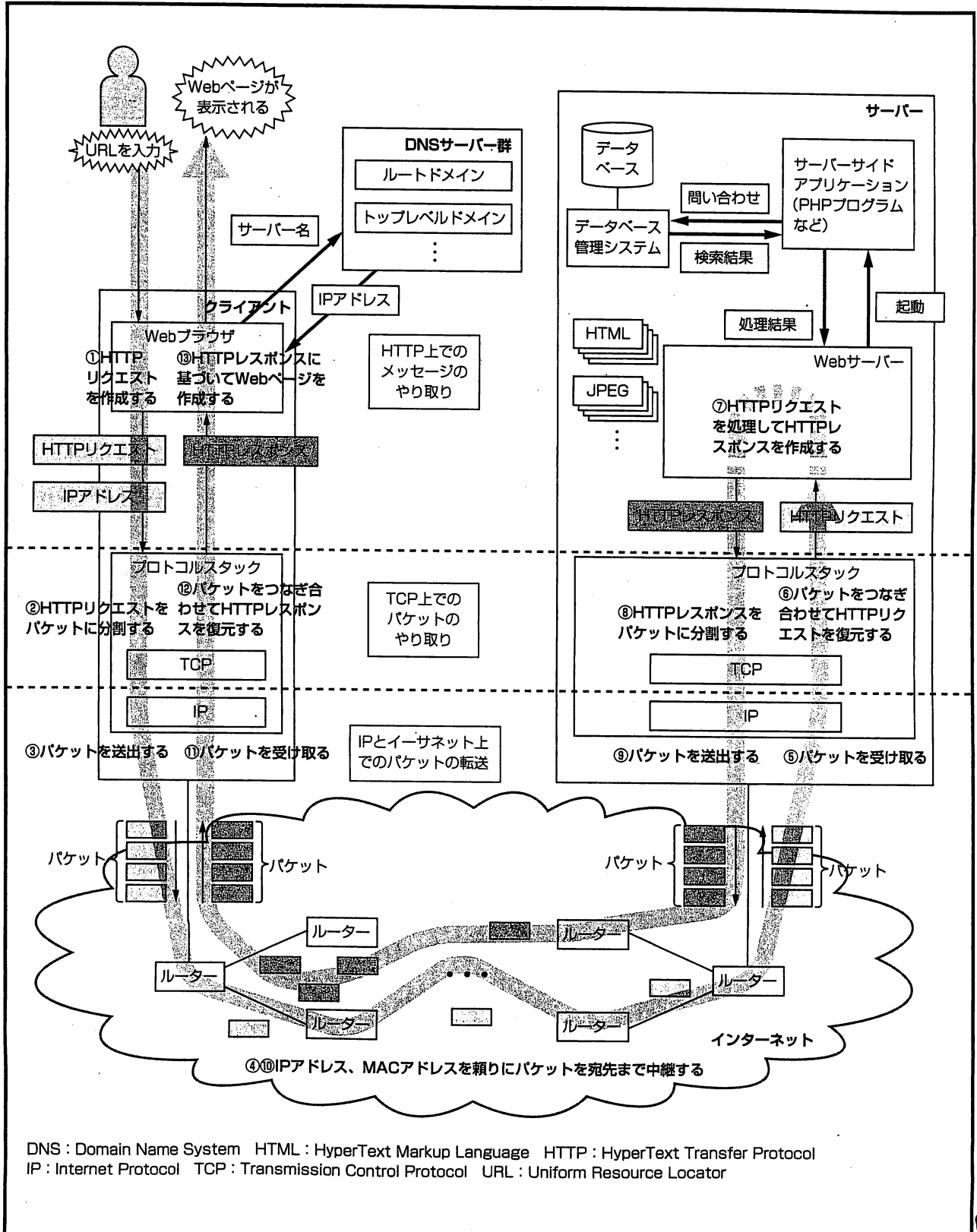
NoSQLは、「SQL言語に限らないデータベース検索技術」の総称です。代表的なものとして「key-valueストア」があります。これは一つのキーに関連付けて数値や文字列などのデータを管理・検索する技術で、膨大な量のデータを高速に検索するネット系企業を中心に採用の動きが広がっています。こうした技術を使うシステムの開発も、Webプログラミングの上に成り立っています。

### レイヤーに分けてやり取りを考えよう

Webプログラミングを学ぶに当たり、Webアプリケーションがどのような仕組みで動いているのかを知っておくことは極めて有効です。Webアプリケーションがデスクトップアプリケーションと異なるのは、インターネットを介してクライアントとサーバーがやり取りすることです。このやり取りに伴う様々な取り決めがWebアプリケーションを理解する際のハードルを高くしていると言えます。

こうした取り決めは、データのやり取りをレイヤー(層)に分けて考えると理解しやすくなります。図1の青い矢印は、WebブラウザにURLを入力してWebページが表示されるまでのデータの流れを表したものです。これを図中の赤い点線のように、(1)WebブラウザとWebサーバー間の「HTTP」に従ったメッセージのやり取り、(2)通信を制御するプロトコルスタックのうち「TCP」に従ったパケットのやり取り、(3)プロトコルスタックの「IP」と「イーサネット」上でのパケットの転送、に分けて考えます。各レイヤーでやり取りするデータは18ページ以降で説明しますので、ここではやり取りするデータの流れと、各段階で行う処理について大まかに理解しておいてください。

図1●URLの入力からWebページが表示されるまで。青い矢印がデータの流れを表す



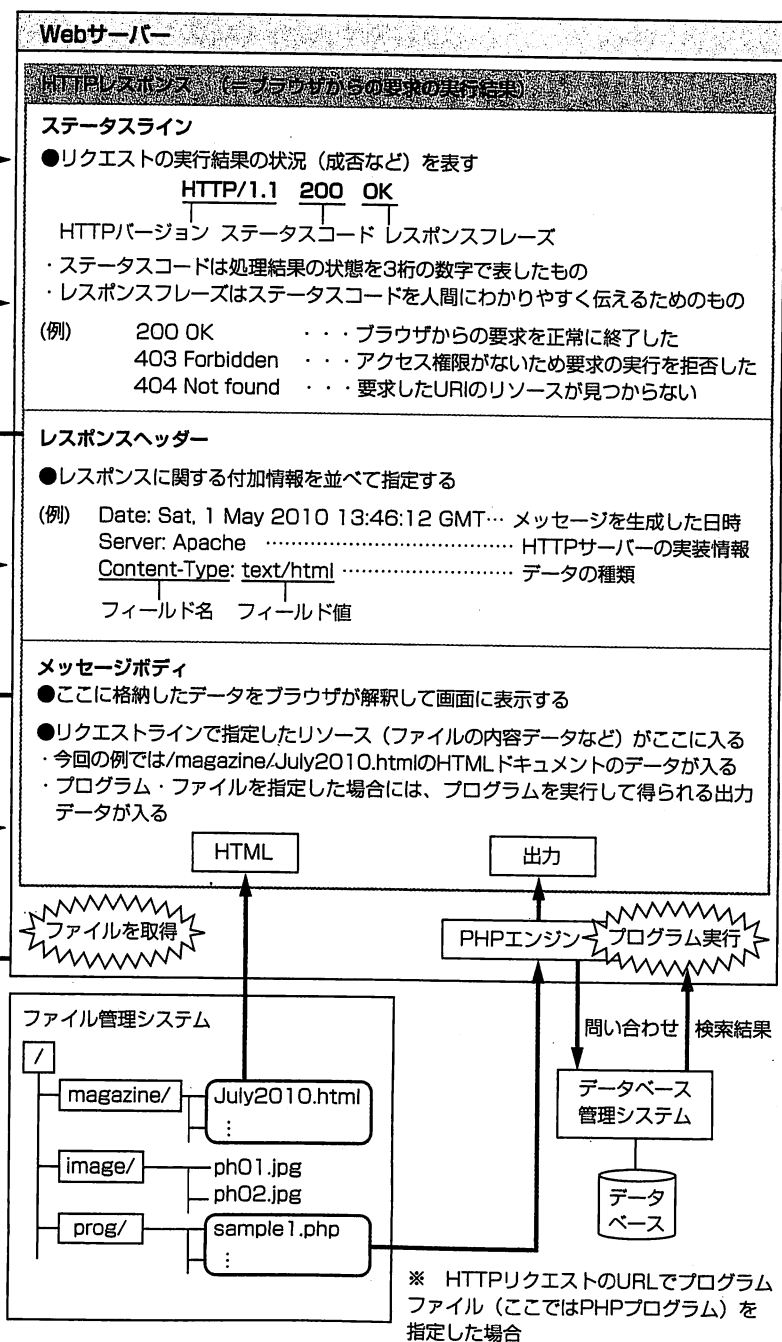
# 特集1 Part1 インターネットのやり取りを徹底図解

図2●HTTP上でのメッセージのやり取り。HTTPと呼ぶ“取り決め”（プロトコル）に従ってメッセージをやり取りする。やり取りされるのは、HT



TPリクエスト（サーバーへの要求内容）とHTTPレスポンス（要求内容に対する実行結果）である

サーバー



## HTTPでのやり取り

図2は、WebブラウザとWebサーバーのレイヤーでやり取りするデータを示したものです。このレイヤーではHTTP (Hypertext Transfer Protocol) と呼ぶ“取り決め”（プロトコル）に従ってメッセージをやり取りします。

HTTPでは、Webブラウザからの送信に対してWebサーバーが応答するというやり取りが基本になります。WebブラウザからWebサーバーに送るデータをHTTPリクエスト、これに答える形でWebサーバーがWebブラウザに送るデータをHTTPレスポンスと呼びます。HTTPのやり取りでは、一つのリクエストに対して一つのレスポンスを返します。

Webブラウザは、ユーザーがWebブラウザに入力したURLに基づいて、HTTPリクエストを作成し、サーバーに送信します。HTTPリクエストの内容については図2の左をご覧ください。

HTTPリクエストの宛先はDNS (Domain Name System) を使って調べます。WebブラウザがURLから抽出したサーバー名をDNSサーバーに送ると、DNSサーバーがインターネット上での“住所”を一意に特定できる「IPアドレス」を返してくれる仕組みです。宛先のサーバーのIPアドレスを指定したHTTPリクエストは、HTTPの下位のレイヤーであるTCP、IP、イーサネットを使って目的のサーバーに届きます（20～21ページで説明します）。

HTTPリクエストを受け取ったWebサーバーはその内容を解析し、指定されたHTMLファイルの内容やプログラムの実行結果を含むHTTPレスポンスを作成します（図2の右側）。そしてそれをWebブラウザ宛に送信します。