

古くて新しいAjaxを検証する！

2007.03.17

Daigo Toshinari and Toshimitsu Baba

Ajax(エイジャックス)とは

Ajaxとは・・・「**A**synchronous **J**avaScript and **X**ML」の略で、Webブラウザでの表示部分を含めたWebアプリケーションの作り方を指します。

具体的にはXML形式のデータをJavaScriptで処理することで、ブラウザのHTTPによるページ遷移とは非同期にサーバとやりとりして「動的に」ページ内容を変更する仕組みを指します。

従来型のWEBアプリケーションとの違いやAjaxの要となる技術やAjax使用に関してのメリット・デメリットなどを述べていきたいと思っています。

Ajaxを構成する技術

○Javascript/DHTML

DHTML(Dynamic HTML)は、文字通りダイナミック(動的)なHTMLページを実現する技術です。文字などに動きを持たせたり、インタラクティブな処理を実現します。かつてはページの装飾的な処理を行う用途が主でしたが、現在ではドラッグ & ドロップやフェードイン処理、メニューなどのリッチなユーザインターフェースを生成するライブラリ等の基本的な技術をなすものとして活用されています。

○HTTP通信

HTTP(HyperText Transfer Protocol)通信は、ブラウザなどのWebサーバがデータのやり取りを行う際に用いられるプロトコルです。Javascriptに実装されたXMLHttpRequestオブジェクトはAjaxの肝の1つでこのHTTP通信を行います。

○サーバサイドプログラム

サーバサイドで稼動するプログラムです。詳細は割愛します。

○Javascriptライブラリ

Javascript/DHTMLによる便利なオブジェクトや関数群をまとめたものに、ライブラリがあります。ライブラリはAjaxの登場以前からいくつかありましたが、Ajaxが注目されるようになってからは特にXMLHttpRequestも組み込んだ(Ajaxが動作するバージョン)クロスブラウザに動くタイプのものも増えています。

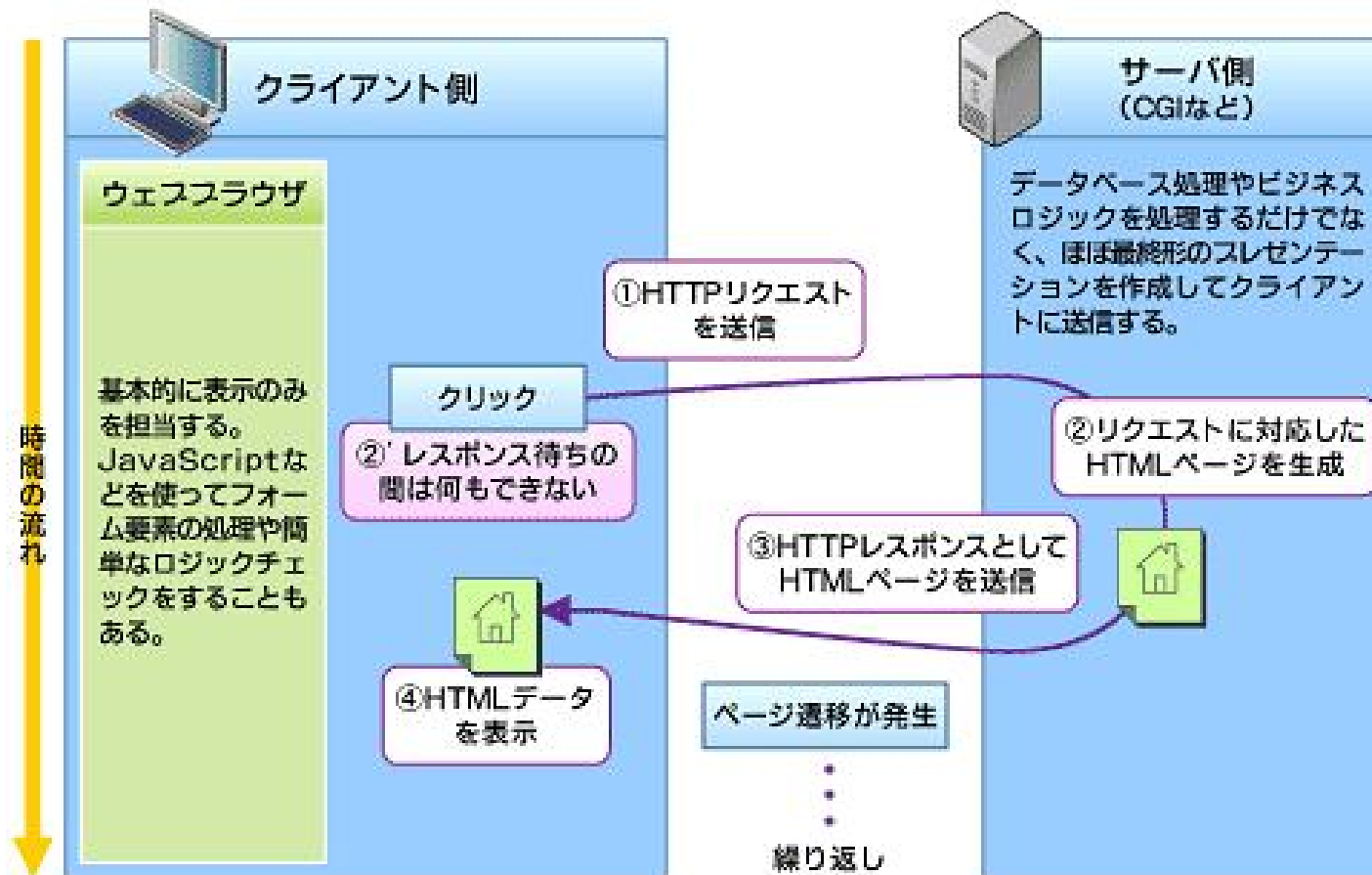
以下、代表的なライブラリ群

- ・prototype.js
- ・Dojo
- ・Json-lib
- ・Yahoo! UI Library
- ・jQuery
- ・Rico
- ・Google Web Toolkit
- ・DWR(Direct Web Remoting) ※後ほど解説

・・・etc

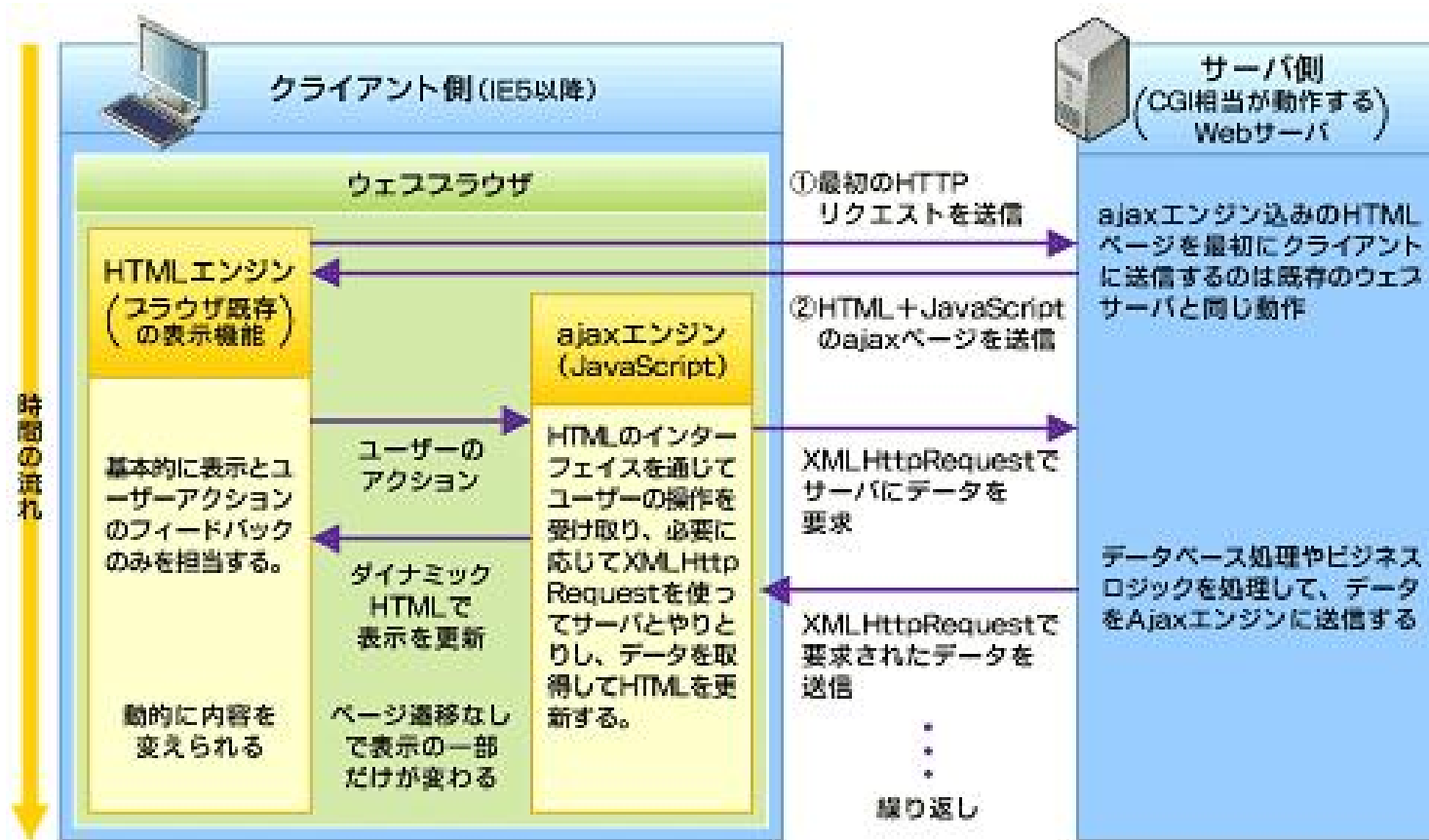
従来型WEBアプリケーション

サーバ側のアプリケーションがほぼすべての処理を行い、クライアント側(ブラウザ)はインターフェイスの役割を果たす。リクエストを送り、レスポンスが返ってくるまでの間、クライアント側では何もアクションを行えない。



Ajaxを使用したWEBアプリケーション

最初の1回は通常のリクエスト/レスポンスによるページの読み込みがあるが
その後はすべてユーザーの操作に応じてJavaScriptが非同期にバックグラウンドでサーバと通信してデータを取得し
ページの必要な部分だけを書き換えている。



Ajaxを使用するメリット

- ①ページ遷移なしで高速に画面を書き換えられる
- ②サーバ処理待ちをしない非同期リクエストが可能
- ③javaやPHPなどサーバ側処理をクライアントへ分散できる
- ④受信データ量を軽減することが可能
- ⑤リアルタイムなインタラクティブ性能が高くなる
- ⑥Flash等の有料ソフト無しで実装が可能
- ⑦プラグイン起動がないので、動作が速い

Ajaxを使用するデメリット

- ①クロスブラウザ化のノウハウが必要
- ②Ajaxを使用できないブラウザ対策が必要
- ③JavaScript部分はソースの公開性ゆえに差別化が難しい
- ④HTTPクライアントの機能が限定される
- ⑤セキュリティーへの配慮が不可欠
- ⑥現在の処理状況についてのインフォメーションが必要
- ⑦リクエスト多発によるサーバ負荷が増大の可能性あり

DWR(Direct Web Remoting)とは

DWRは、Apache License, Version 2の下に提供されている

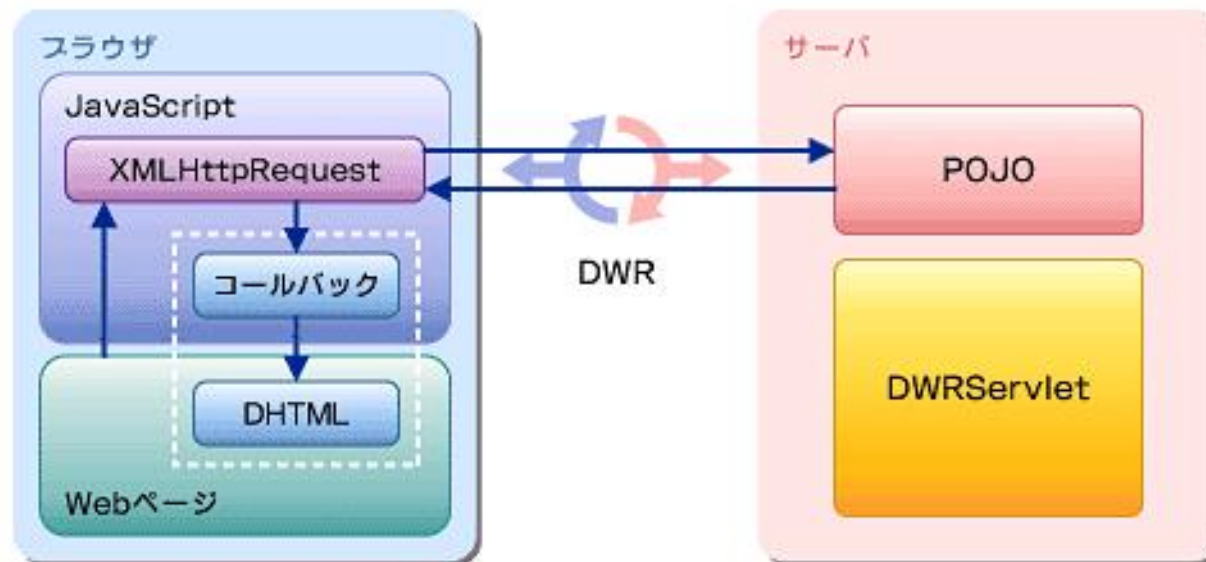
オープンソースJavaScript—Java連携用フレームワークです。

DWRの最大の特徴は、JavaScriptからサーバ側のJavaオブジェクトを利用できることです。

しかも、まるでJavaのメソッド呼び出しのように記述できます。

クライアントとサーバ間の非同期通信、サーバとクライアント間のデータ変換などは

すべてDWRが処理してくれるので、Java開発者にとって敷居がとても低くなっています。



DWRの注目すべき特徴

⇒サーバ側のJavaはPOJO

サーバ側の処理はPOJO (Plain Old Java Object) として実装するだけです。
サーバとしての処理はすべてDWRのサーブレット (DWRServlet) が処理してくれます。
POJOなのでAPサーバを起動しなくても、単独でテストできます。

⇒Java／JavaScript間のデータ変換で多くの型をサポート

DWRでは、サーバのJavaオブジェクトとクライアントのJavaScript
オブジェクトの間でデータ変換を行います。
文字列や単純な型については何も指定しなくても自動的に変換してくれます。

⇒Struts、JSF、Springなどのフレームワークとの統合

Webアプリケーション開発ではフレームワークを利用することが一般的です。
DWRはこれらのフレームワークと連携できます。

DWRの注目すべき特徴

⇒多くのブラウザに対応

ブラウザによってJavaScriptの振る舞いなど微妙な違いがあります。
このクロスブラウザの問題はおっくうなものです、DWRではブラウザの違いを気にする必要がありません。しかも、対応しているブラウザはIEやFireFoxだけではなくSafari、Opera、Mozilla、Konquerorなど多岐にわたります。

⇒動的にページを生成するためのユーティリティメソッドを提供

コレクションクラスのオブジェクトをHTMLのテーブルやリストに展開するためのメソッドが用意されています。
JavaScriptに詳しくなくても、簡単にページをダイナミックに生成できます。

⇒エラーハンドリング、タイムアウトハンドリング

JavaScriptからサーバ側のJavaメソッドを呼び出すときにエラーハンドラやタイムアウトの指定が簡単にできます。