

Java 開発者のための Ajax とは何か

本書は、Java 開発者のために書かれた Ajax の本です。Java に精通した人間からすると Ajax は別の世界に感じます。しかし、実際に触れてみると、これまで我々が培ってきた技術をさらに増幅してくれるのだとわかってきます。Ajax は、Java の仕事に豊かな表現力を与えてくれるのです。Web プログラマや Web デザイナ、そして Java 開発者、皆の仕事が分業化されている昨今、高い視点から統合的に Web システムを見る必要があります。Ajax の基本を知ることによって現状が開眼できます。まずはやってみましょう (仮)。

1.1 Overture—Web の進化

インターネット利用の普及により、Web ブラウザをクライアントとした Web アプリケーションがさまざまな場面で使われるようになりました。Web アプリケーションは、Web ブラウザからサーバにリクエストを送り、サーバ側で処理を行い、その結果をレスポンスとして Web ブラウザに送り返し、そのデータを Web ブラウザ上に表示します。

通常、Web ブラウザに表示するページは HTML で記述されるため、Web アプリケーションにレスポンスとして送信されるデータも HTML で記述されます。そのため、Web アプリケーションで提供できるインターフェイスは HTML で提供されるものであり、この HTML で表現できるインターフェイスはそれほど優れたものではなかったため、ここが Web アプリケーションの限界と言われていました。

また、Web アプリケーションは、ユーザが Web ブラウザでページ上の入力フォームにデータを入力し、そのデータをサーバに送信し、サーバが処理をしてレスポンスを返す形式です。そのため、すべての処理はサーバで行われ、クライアント側はページを表示するだけです。例えば入力データの検証だけでもサーバ側にリクエストが送られ、クライアント側では処理がブロックされることとなります。このように、ユーザとのインタラクションの不自由さも従来の Web アプリケーションの欠点です。Web アプリケーションは、ページを切り替えることによってアプリケーションが進行していくため、まるで紙芝居のようにアプリケーションが動作します。

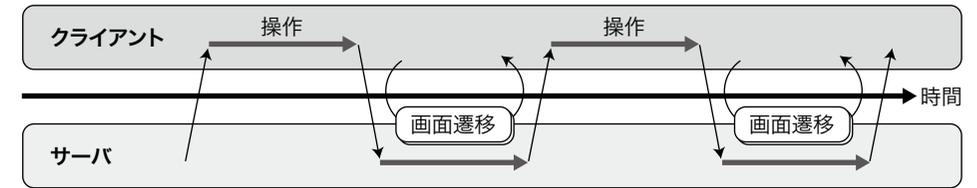


図 1.1 従来の Web アプリケーションの動き

従来の Web アプリケーションは、各ページの表現力やユーザとのインタラクションの面で、十分とは言い切れず、このあたりが Web アプリケーションの限界だと思われていました。

1.2 Ajax アプリケーションの登場

こうした状況で、従来の Web アプリケーションの限界を超えるアプリケーションが登場しはじめました。例えば、Google 社が提供する Google Maps では、地図をドラッグすることにより表示エリアが連続的に切り替えられます。このアプリケーションがリリースされた時には、これが Web ブラウザ上で実現されていることがある種の衝撃として世の中に受け入れられました。

このようなアプリケーションを幕開けに、Web ブラウザ上での JavaScript や CSS の利用が進んでいきます。そして、このようなアプリケーションでは、JavaScript を利用して、非同期でサーバにアクセスし、ページ全体を再描画することなくページの一部だけを切り替えていきます。このような仕組みは Ajax (Asynchronous JavaScript and XML) と呼ばれ、注目を集めていきます。当初はサーバとのやり取りに XML を利用していたため、Ajax の x として XML が入っていますが、現状では、それ以外にさまざまなフォーマットが用いられるようになっています。

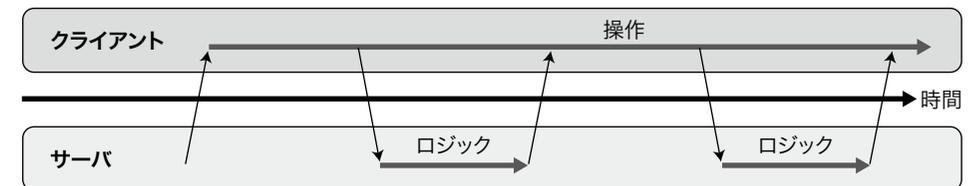


図 1.2 Ajax アプリケーションの動き

このように、非同期でサービスにアクセスし、ページの一部を書き換え、JavaScript、CSSを用いて、表現力豊かなユーザインターフェイスを提供することで、Webアプリケーションは新しい時代に突入したと言えるでしょう。

1.3 Ajaxアプリケーションの仕組み

ここで、Ajaxアプリケーションの仕組みの概略を説明します。初めて聞く言葉もいくつか出てきますが、これらの用語は本書を通して解説していきますので安心して下さい。

Webブラウザで表示されているページに配置されているボタンをクリックするなど、ユーザからのアクションがあると、そのボタンに関連づけられたJavaScriptのプログラムが動きはじめます。そのJavaScriptの中からXMLHttpRequestを利用してサーバ側のサービスを非同期で呼び出します。そして、Webブラウザ上のJavaScriptプログラムはそのサービスの実行結果を受け取り、その結果を元にDOM APIを通してページの構造を書き換え、CSSを利用して見映えを調整します。これが基本的なAjaxの仕組みです。

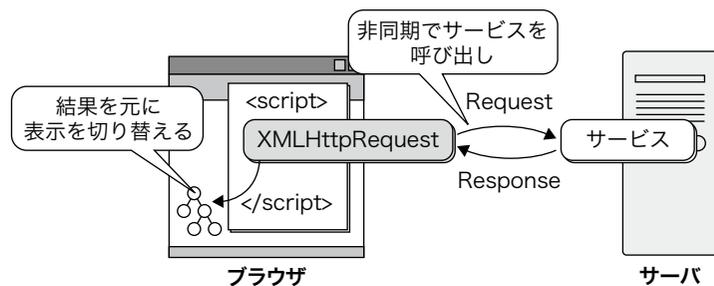


図1.3 Ajaxアプリケーションの基本的な動き

1.4 本書の構成

本書では、このAjaxアプリケーションを構築するために必要な知識を説明していきます。

はじめにPart2では、Ajaxアプリケーションの基礎を解説します。第2章では、ブラウザ上で動作するプログラム言語であるJavaScriptについて解説します。第3章では、

JavaScriptとHTMLを結びつけるDOM APIを解説します。DOM APIを操作することにより、動的にブラウザ上の表示を書き換えることができます。第4章では、サーバ側のサービスを呼び出すXMLHttpRequestについて解説します。

Part2で解説したテクノロジーを用いてAjaxアプリケーションを作成することができますが、これらが提供する機能は基本的なもののため、あまり効率がいいものではありません。より効率よくアプリケーションを開発するために、いろいろなライブラリやフレームワークが提供されています。本書では、オープンソースのライブラリとフレームワークをいくつか紹介していきます。

Part3では、Webブラウザ上で動作するJavaScriptのライブラリを紹介します。第5章では、prototype.jsを紹介します。prototype.jsは、JavaScriptの持つ基本的な機能をより使いやすくするための拡張を提供します。第6章では、Dojo Toolkitを紹介します。Dojo Toolkitは基本的なJavaScriptの言語拡張から表現力豊かなユーザインターフェイス部品を提供するウィジェットまで非常に広範囲の機能を提供します。

Part4では、サーバ側のサービスとクライアント側のJavaScriptを連携させるフレームワークを紹介します。ここでは、サーバ側でサービスを提供する言語としてJavaを用います。第7章では、DWR (Direct Web Remoting) を紹介します。DWRを用いると、サーバ側にあるJavaプログラムをクライアント側からローカルの関数を呼び出すように簡単に呼び出すことができます。第8章では、Ajax4jsfおよびRichFacesを紹介します。これらのフレームワークはJSF (JavaServer Faces) を拡張したフレームワークで、開発者はページ上にJSFの部品を配置するだけでいいので、JavaScriptを記述することなくAjaxアプリケーションを開発できます。第9章では、jMakiを紹介します。jMakiは、JavaScriptで書かれた処理をラップしてJSFの部品として提供しているため、JavaScriptを駆使して部品を提供する開発者とそれを利用してページを作成する開発者を分けることができます。

Part5では、Ajaxアプリケーションを作成するための環境を説明します。第10章では、Eclipseプロジェクトが提供するATF (Ajax Tools Framework) を中心に、Ajaxアプリケーションのプログラミング環境を紹介します。第11章では、JUnitとSeleniumを中心に、Ajaxアプリケーションのテスト環境を紹介します。第12章では、FireBugを用いて、Ajaxアプリケーションのデバッグとプロファイルを行う環境を紹介します。