

本日の内容

Webアプリケーションの基本知識を学ぶ

→ ブラウザからHTTP通信をして、インターネットを介して、サーバ側の処理にrequestし結果がresponseとして返却表示されるまでの基本的な流れを学びます。

※一般的となっているWEBシステムの基礎の基礎です。
しっかり学んでモノにすること！！

Webアプリケーションの仕組み

- 基本的な構成要素は、WEB三層と呼ばれる

「Webブラウザ」／「Webサーバ」／「アプリケーションサーバ」「データベース」の連携によって動作する。

「Webブラウザ」



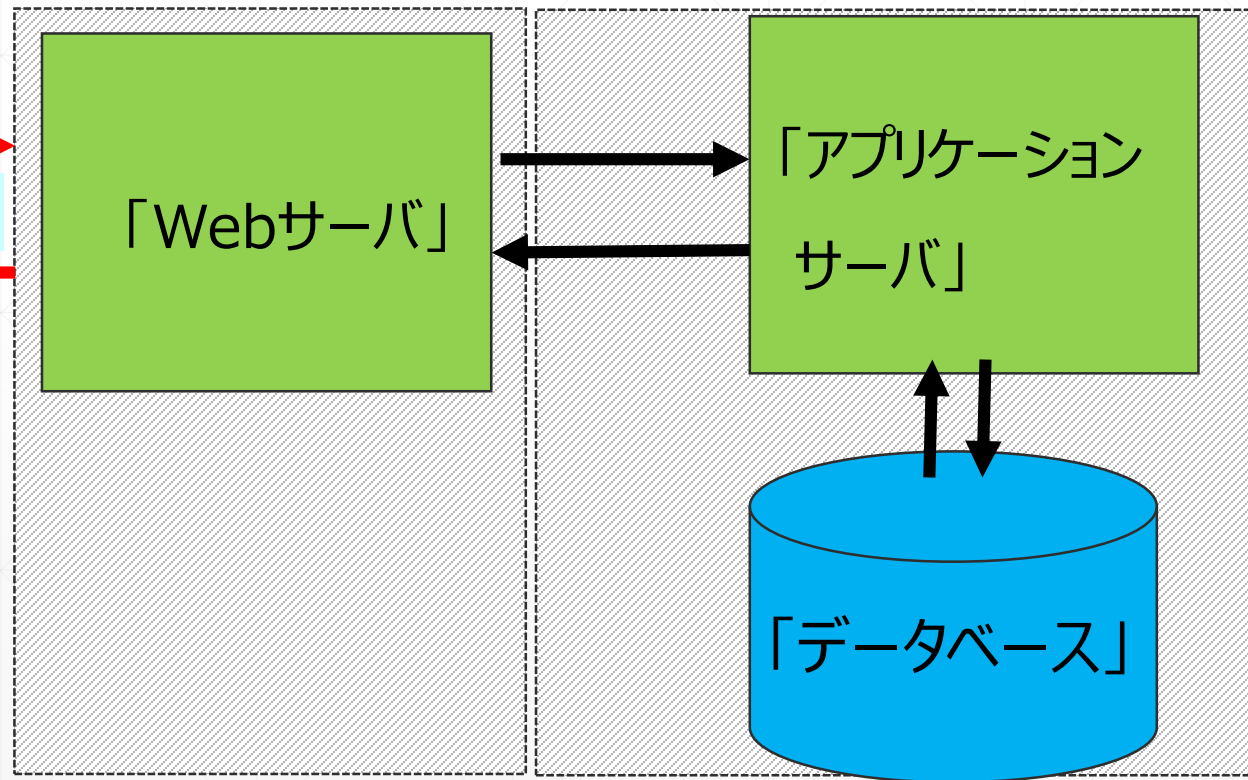
「ユーザ」



「インターネット」

HTTP通信

「www.abc.co.jp」



※HTTP = HyperText Transfer Protocol

それぞれの役割

- Webブラウザ

- … ユーザーインターフェースの提供。WWWの仕組みを利用。URLを指定して遷移する。

- Webサーバ

- … Webブラウザとの通信。静的コンテンツ(HTMLファイル・画像・XML)の提供。

- アプリケーションサーバ

- … 動的コンテンツ(アプリケーションの処理によって作成)の提供。

- データベース

- … アプリケーションに必要な情報の格納提供。一元管理する。

Webアプリケーションの階層構造

・多層モデル

Webアプリケーションは、1 台のサーバ機、ひとつのソフトウェアだけで構成しない。
複数のサーバ機とソフトウェアの組み合わせによって、提供される。 = 多層モデル

第 1 層 クライアント層 ≪Webブラウザ≫



第 2 層 サーバ層 ≪Webサーバ≫
 ≪アプリケーションサーバ≫



第 3 層 データベース層 ≪データベース≫

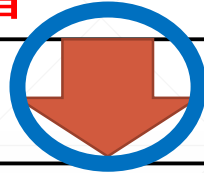
多層モデルの特徴

・多層モデルの特徴

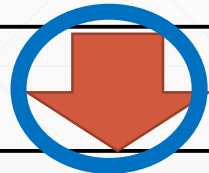
各層のソフトウェアは、隣り合う層としか通信をしない。

例) Webブラウザは直接データベースにアクセスできない。

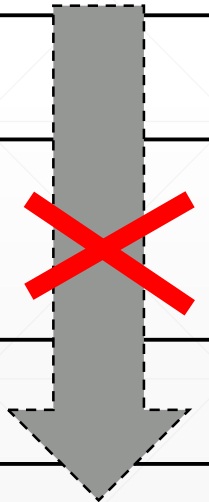
第1層 クライアント層 ≪Webブラウザ≫



第2層 サーバ層 ≪Webサーバ≫
 ≪アプリケーションサーバ≫

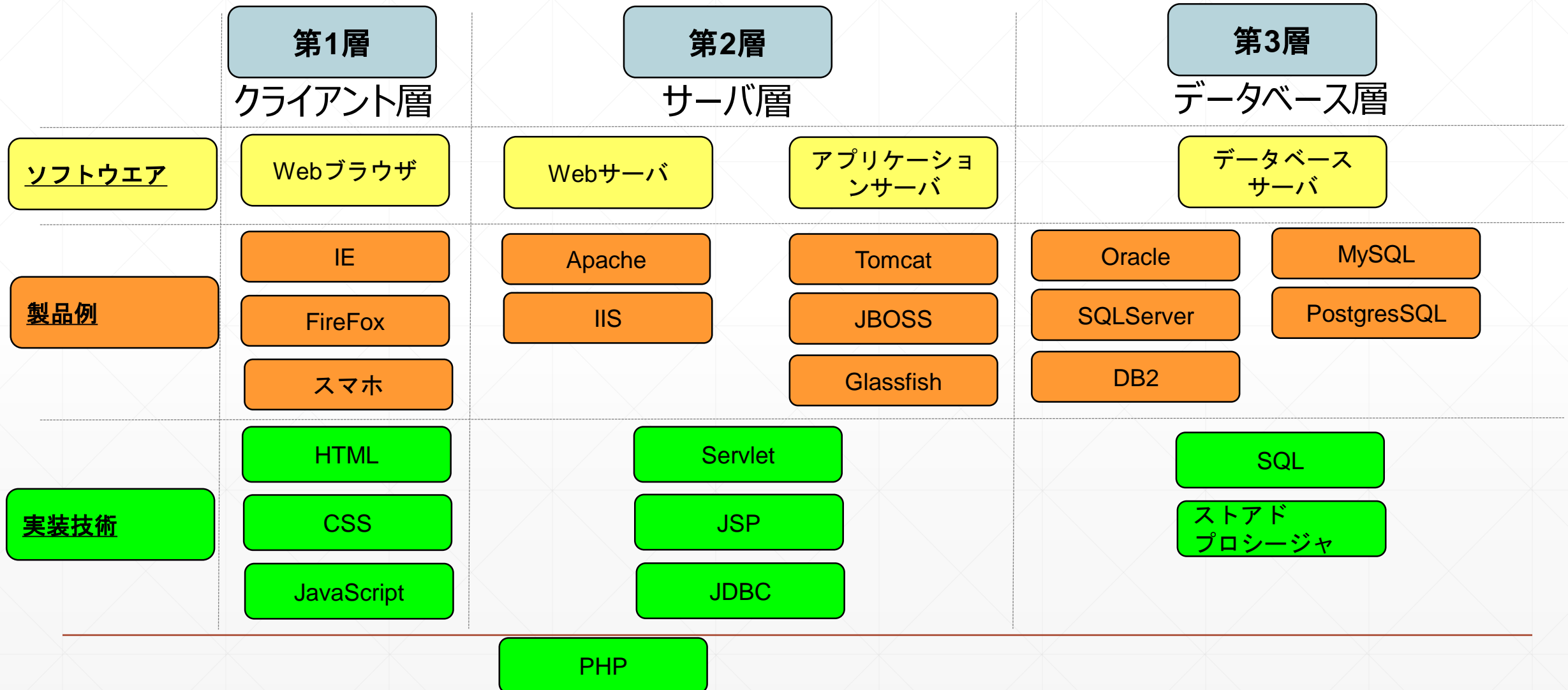


第3層 データベース層 ≪データベース≫



ソフトウェアとテクノロジー

・各層で用いられる製品と実装技術



Webアプリケーションの実装技術

- HTML(HyperText Markup Language)
 - … Webブラウザ上に文書を表示するために用いられる言語。
 - Servlet(Java Servlet)
 - … Webサーバを介して、Webブラウザと通信を行うためのJava API。
 - JSP(JavaServer Pages)
 - … Java言語を用いて、動的にHTML文書を構築するために用いられるJava API。
 - JDBC(Java Database Connectivity)
 - … JavaアプリケーションからデータベースサーバにアクセスするためのAPI。
-

Webアプリケーションのメリット・デメリット

・メリット

- … クライアント層が軽量。OSに標準にインストール済みのWebブラウザを利用する。
 - クライアントを別途作る必要がない。
 - クライアントのバージョンアップにかかる費用が必要がない。サーバ側の変更のみ。
 - … クライアントのアーキテクチャを選ばない。
 - 異なるOS、異なる端末であっても、ほぼ共通の内容が提供できる。
 - 同じ投資で、より幅広いユーザーに利用してもらえる。
-

Webアプリケーションのメリット・デメリット

・デメリット

- ・・・ サーバに負荷が集中する構造。
 - サーバに負荷がかかる場合がある。
 - コンシューマ向けサービスだと事前に想定していた以上のアクセスがある場合がある。
 - ・・・ ユーザーインターフェースが使いにくい。
 - リッチなWindowsアプリの画面に比べて、プア(貧しい)な画面構成となる。
 - 近年はAjaxの登場を契機に、JavaScriptでリッチな画面にありつつある。
-

Webアプリケーションの進化

・リッチクライアント化

- ・・・ Ajax(Asynchronous Javascript And Xml)。
 - jQueryなどを用いてブラウザ上の表現力や操作性を向上する技術。
 - Googleが火付け役
 - ・・・ JSON(JavaScript Object Notation)。
 - jQueryなどを用いてhtml文書とは違う構造のテキストを受け渡しする技術。
 - 非同期処理などを実現する。
 - ・・・ REST(REpresentational State Transfer)。
 - Webサービスの設計モデルです。RESTなWebサービスは、そのサービスのURIにHTTPメソッドでアクセスすることでデータの送受信を行う。
-

デザインパターンの適用

- ・デザインパターンとは

- Webアプリケーションの開発にて使う開発言語(java、C#、PHP)は、オブジェクト指向である。クラスとして機能のまとまりを定義して再利用性を高めている。
デザインパターンは、オブジェクト指向プログラミングにおいて、ベストプラクティスを集めたもの。
 - 「オブジェクト指向における再利用のためのデザインパターン」(1995年出版)
俗にいうGoF本
 - 生産性／信頼性／再利用性の高いクラス設計をするために、参考にする。
-

フレームワークの使用(1)

・フレームワークとは

フレームワーク内にある半完成品の機能を利用して、より早く品質の良い開発ができる。

※オブジェクト指向の再利用性の恩恵

- よく利用する機能(クラス／ライブラリ／モジュール／API)をまとめたモジュールファイル(.jar)
- そのまま利用できる主要部分の雛形（テンプレート）
- 開発者が機能を追加するための使用や規約

【フレームワークを取り込む時の注意】

OSSのフレームワークが多く、ソースが公開されてはいるが、ノウハウの蓄積ができない

また、余分なロジックが入っている場合があるので、セキュリティ上好ましくないケースがある

フレームワークの使用(2)

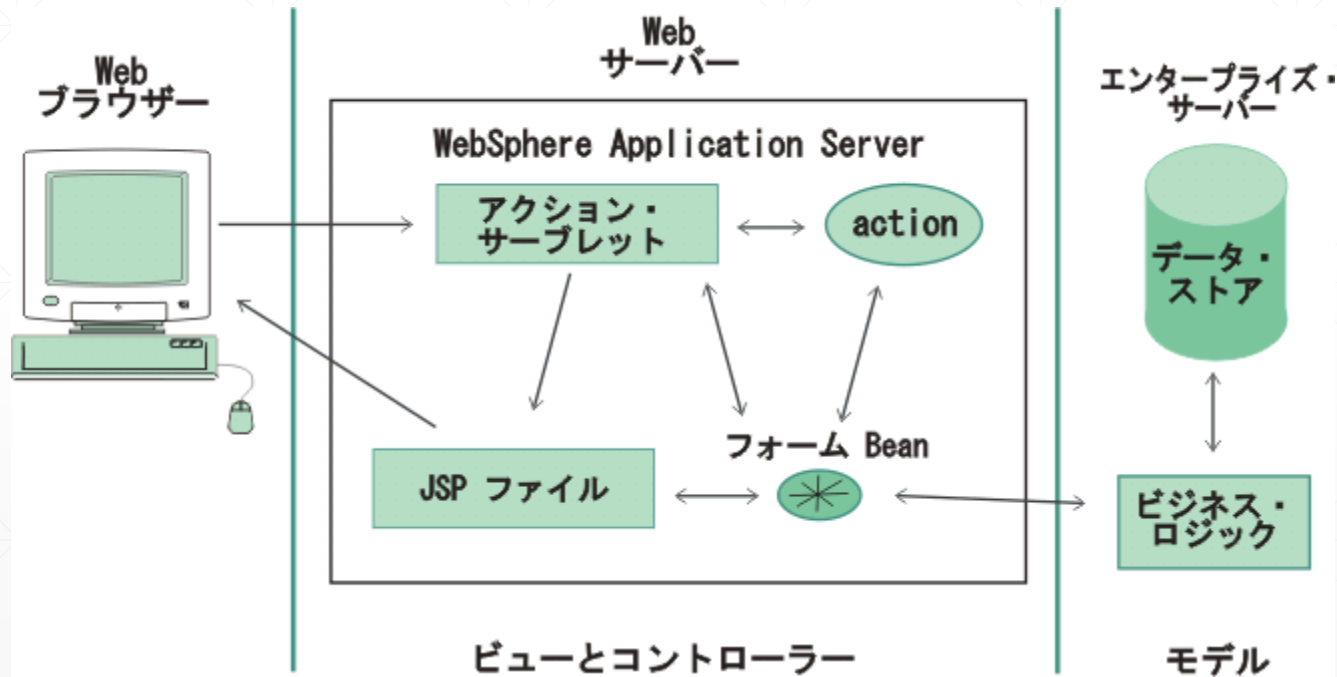
代表的なフレームワーク	概要	特徴
Spring Framework	<ul style="list-style-type: none">・ 軽量フレームワーク・ 採用しているプロジェクトが多い・ データベースアクセス、トランザクションなどのフレームワークの集合体・ オープンソース (OSS)	<ul style="list-style-type: none">・ さまざまな機能がある・ 代表的な機能は、設定ファイルからクラスを呼び出せるDI (依存性注入)AOP(アスペクト指向プログラミング)
Java EE	<ul style="list-style-type: none">・ Java標準フレームワーク・ 全部乗せ・ 大規模向け・ ただし、仕様の変更が多い	<ul style="list-style-type: none">・ Servlet、EJB、JSPなど各種Java APIを網羅したフルスタックフレームワーク・ Oracle社のサポートが受けられる
Apache Struts	<ul style="list-style-type: none">・ MVC(モデル・ビュー・コントロール)のデザインパターンに基づいたアーキテクチャ・ 2001年くらいから使われている・ Strutsを採用したプロジェクト多い	<ul style="list-style-type: none">・ 高い生産性をもち世界中で人気・ 脆弱性があり最近は人気が下火

フレームワークの使用(3)

・参考) StrutsのMVCモデル

IBMのホームページから

MODEL／VIEW／CONTROLLER



データの永続性

- ・メモリ上のデータ構造を、データベースに格納する
 - WEBシステムでは、WebサーバへURLを指定してページを開いてから、ページを閉じるまでの間、プログラムは実行しており、データ構造を保持している。
※SLSB(ステートレスセッションビーン)・・・処理終了を待たないセッション
 - 永続してデータ構造を、データベース・テキストファイル・バイナリーファイルに記録しておくことを「データの永続性」という
 - Javaにおけるデータの永続性とは、Javaオブジェクトをディスクに格納するためのシリアライズ、Java EEにおける関係データベースへの Enterprise JavaBeans
 - myBatisやJPAなどのフレームワークを利用する
-