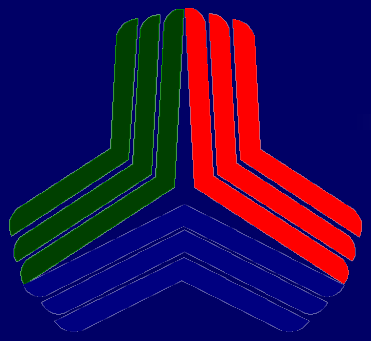


メディア情報応用 第9週

Advanced Topics in Media Informatics Week 09

金沢工業大学 情報フロンティア学部

メディア情報学科

山岸 芳夫

Yoshio Yamagishi

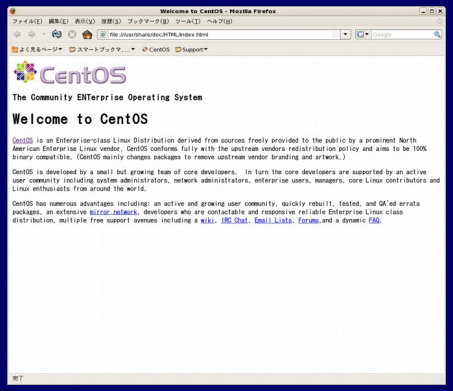
Dept. of Media Informatics

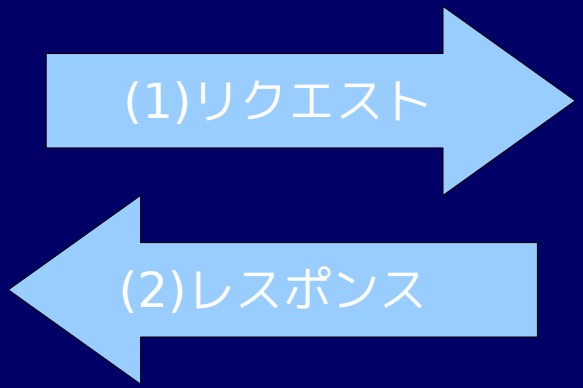
Col. of Informatics and Human Communication

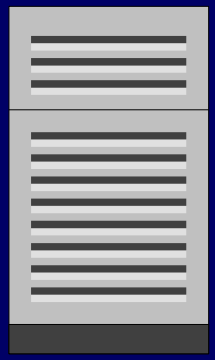
Kanazawa Inst. Tech.

従来のWeb

- これまでのWebアプリケーションは、基本的にクライアント（ブラウザ）がサーバにリクエストを行い、サーバがレスポンスを返す、という形
- XMLHttpRequest (Ajax) によって、画面遷移と同期しなくても良くなったが、それでも結局のところリクエスト⇔レスポンスの形式は変わっていない





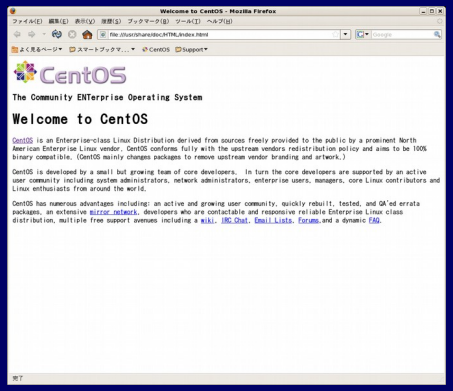


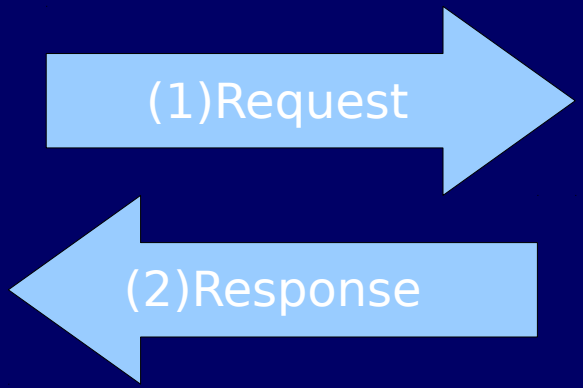
ブラウザ

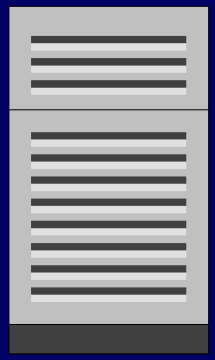
Webサーバ

Conventional Web

- The process of conventional web access basically consists of a sequence of the request by the client(browser) and the response from the server.
- XMLHttpRequest (Ajax) enabled asynchronous access, but the procedure of the web access (i.e. request/response sequence) is not changed.





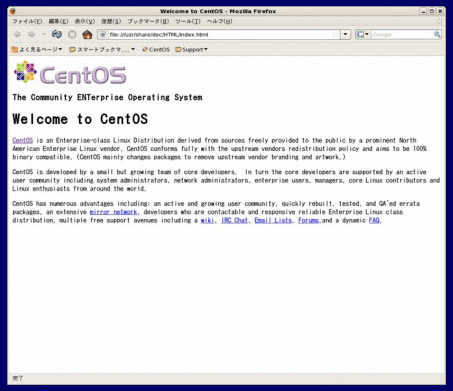


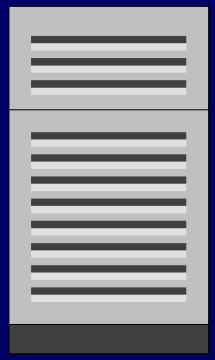
Browser

Web Server

WebSocket

- WebSocketは従来のリクエスト／レスポンス形式に加え、サーバからのプッシュも可能な、双方向通信リアルタイムWebを実現するAPI（の本命）
- WebSocketが利用可能なサーバには、tomcat, Node.js などがあるが、ブラウザが未対応な場合もある（IEは10以降でようやく実装）



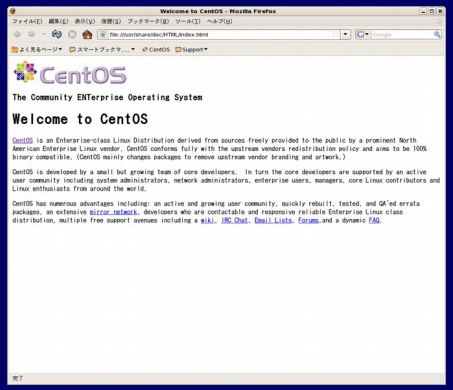


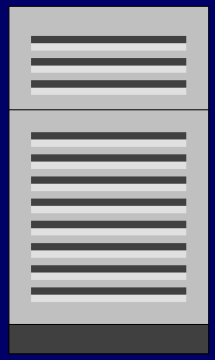
ブラウザ

Webサーバ

WebSocket

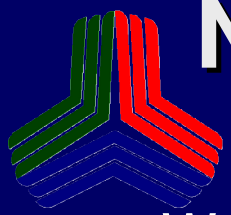
- WebSocket may become a main stream of the bi-directional realtime web API, which realizes not only conventional request/response procedure but also push notification from the server.
- There are some WebSocket-ready web servers such as tomcat or Node.js. However, some browsers do not support (e.g. Internet Explorer version 10 or before).





Browser

Web Server



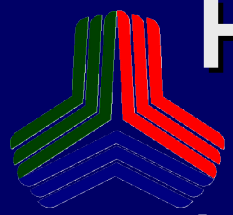
Node.js

- WebSocketが使えるWebアプリケーション実行＋サーバの統合環境。開発に用いる言語はJavascript

  An integrated environment to execute and serve the WebSocket applications, which are written in Javascript.
- データベースとの連携や、様々なモジュールを組み込むことで機能を拡張することが可能

  The function of Node.js can be extended to include appropriate modules. Connection to the database is also available.
- クライアント側のコードもサーバ側のコードも全てJavascript で書くことができるので、学習コストが抑えられる

  Because both server-side and client-side applications are written in Javascript, the learning cost can be drastically reduced.



Hello World

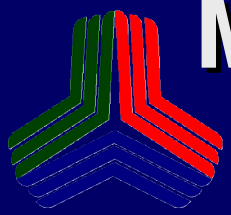
- Moodle の第9週目から app_hello.js をダウンロードして適当な場所に置き、以下の命令を実行する

  Download app_hello.jp from 9th topic of Moodle course. Execute below command on the directory where app_hello.jp exists.

  node app_hello.js

- http://localhost:4000 にアクセスすると、Hello World が表示されるはず

  Access http://localhost:4000 and then make sure the "Hello World" is shown.



MongoDB

- 一般的に、SQLによるリレーショナルデータベースとは異なる、NoSQLと呼ばれるデータベースサーバ

  One of the NoSQL database server, which is differ from general SQL-based relational database.

- データ形式は JSON (Javascript Object Notation) で、制御言語は Javascript

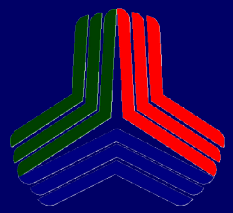
  The data structure and management language of MongoDB are JSON and Javascript, respectively.

  JSON形式の例/ Example of JSON data

  - { “name” : “yamagisi", “mesg” : “hello"}

- Node.js と連携することで、データベース操作まで全てJavascript の開発環境を実現できる

  Connecting to Node.js, Javascript one stop development environment including database management can be realized.



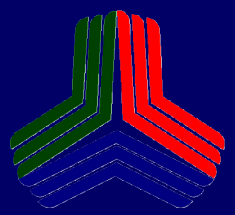
MongoDBの操作
Operation of MongoDB

- RDBのテーブルにあたる物はコレクションと呼ばれる。操作はコレクションのメソッドとして扱う

  The "Collection" of MongoDB corresponds to the table of the RDB. The management of MongoDB is based on the method of the "Collection".

  - find()...データの検索。SQLのSELECT文にあたる
    Search and extract data from the collection, similar to "SELECT" of SQL.
  - insert()...データの挿入。SQLのINSERT文にあたる
    Insert data into the collection, similar to "INSERT" of SQL.
  - update()...データの更新。SQLのUPDATE文にあたる
    Update data of the collection, similar to "UPDATE" of SQL.
- 比較演算子は $eq(=), $ne(<>), $gt(>), $lt(<), $gte(>=), $lte(<=) となる

  Comparison operators will be $eq(=), $ne(<>), $gt(>), $lt(<), $gte(>=), $lte(<=)

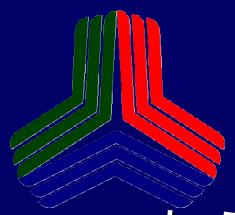


Node.js Chat

- Moodle の第9週目から node_chat.tar.gz をダウンロードしてホームディレクトリの node_chat/ ディレクトリの中で展開し、ターミナルでまず mongodb を動かす

  Download node_chat.tar.gz from 9th topic of Moodle course. Extract it in the node_chat/ directory, then start mongodb.

```
cd ~/node_chat
mongod --dbpath=./mongo
```



Node.js Chat

- もう一つ別のターミナルを開き、ip addr 命令でゲストOSのIPアドレスを調べておく

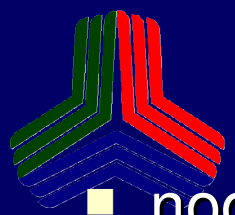
  Open another terminal and check the IP address of the VM by "ip addr" command.

- node_chat/index.html の中の

  Replace IP address in the following line of node_chat/index.html to the one of the VM you have checked before:

  var ioSocket = io.connect("http://192.168.74.129:4000");

  という行のURLのIPアドレスを上記の調べたIPアドレスに書き換える。



Node.js Chat

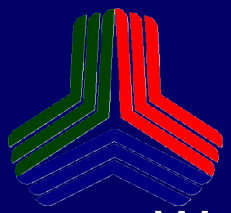
- node を起動する

  Start node.js .

  ```
  cd ~/node_chat
  node app.js
  ```

- http://（ゲストOSのIPアドレス）:4000 にアクセスしてみよう。さらに、ホストOSから同 URL にアクセスしてチャットをしてみて、リアルタイムに投稿が追加されることを確かめよう

  Access http://(IP address of VM):4000 from both browsers of localhost and host OS. Type messages in both browsers and make sure the messages are posted simultaneously.



課題（第9週）

- WebやITを用いた地域振興の取り組みの実例を3つ以上探して報告せよ。内容は一つの実例あたり
  - 取り組み名
  - 開催地域
  - 主催者
  - 取り組みの概要
  - 関連URL

  を必ず含むものとする。
- これらを**次の講義の日の0:00までに**Moodleの実例フォーラムに記事として投稿すること。一つの記事に全ての実例をまとめて書くこと。次回はこれらの実例から今回作成するシステムの要件定義を行う

Assignment (Week 09)

- Find and report three examples of regional vitalization trial using Web or IT. The contents per one example are following:
  - Name of the trial
  - Target region
  - Organizer
  - Abstract of the trial
  - URL
- Post above three examples to the forum in the 9th topic of Moodle course before 0:00 of the next class day. Summarize all of the examples into one post. In the next class, we will define a system requirement based on them.