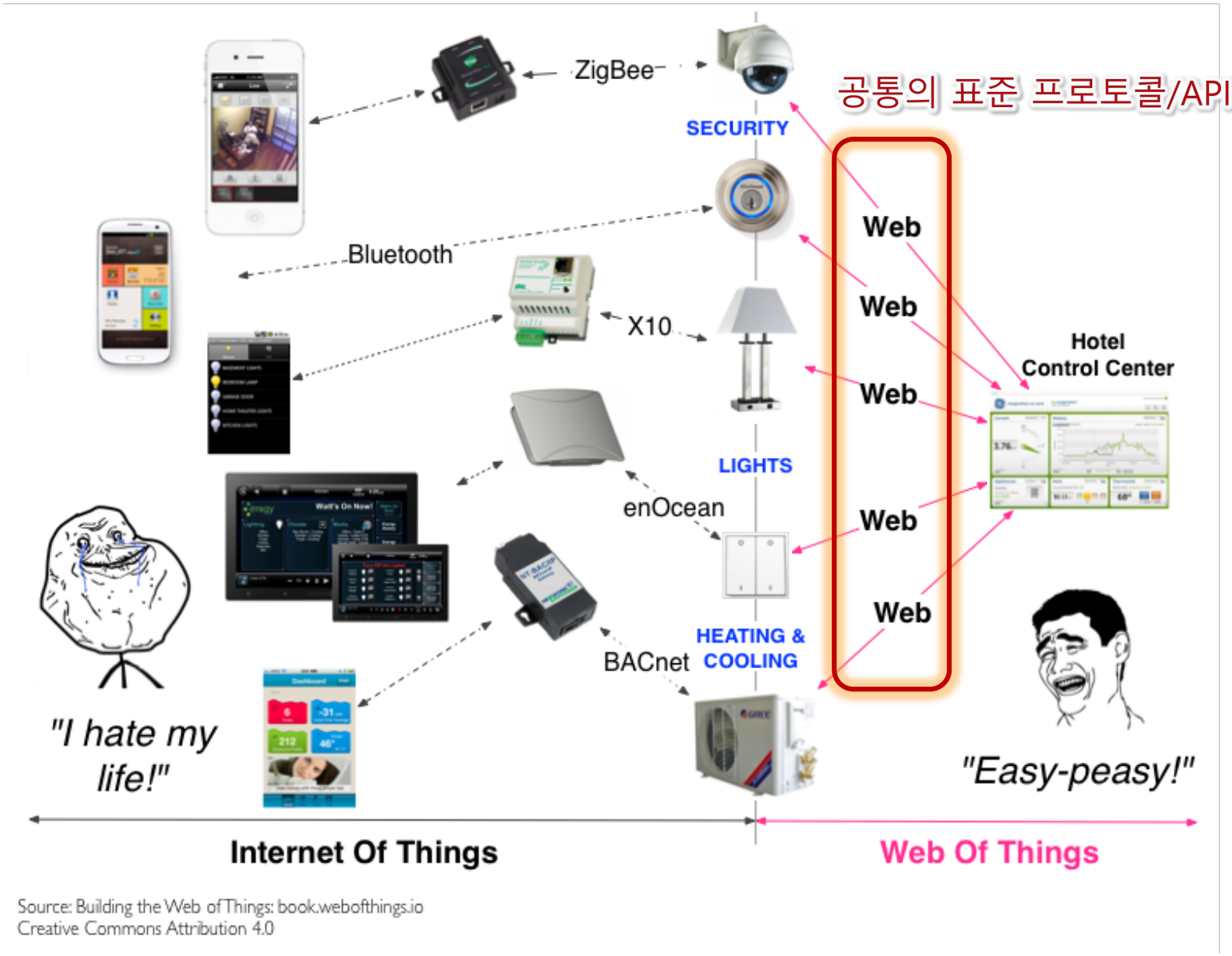
Node. JS and HTTP Open API

Introduction to Internet and Web

부산대학교 정보·의생명공학대학
정보컴퓨터공학부

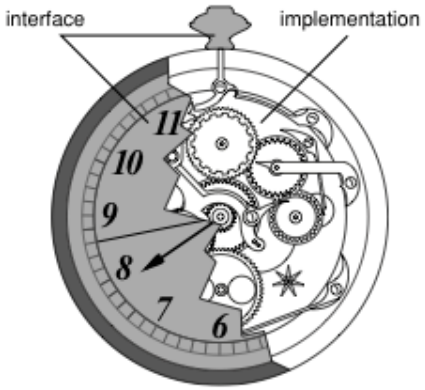
Table of Contents

Web of Things



Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

API

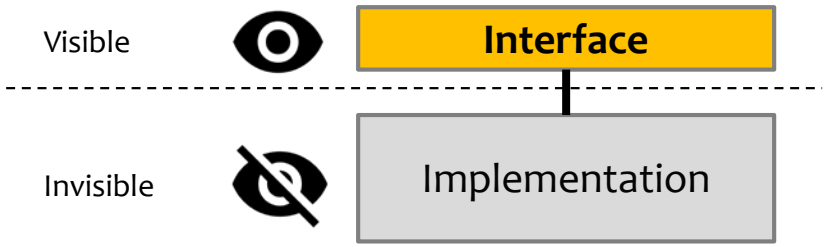


❖ What is an API ?

- API : Application Programming Interface
- What is an API ? – YouTube (#2)

❖ Separation of Interface from Implementation

- 기능/서비스를 제공하는 객체는 Interface와 Implementation으로 구성
- 기능/서비스를 이용하기 위해서는 Interface에 대한 이해만 있으면 충분함
- 예) 자동차와 운전
  - 가솔린, 디젤, 하이브리드, 순수 전기차 등 자동차 엔진들의 동작 원리와 차이점에 대해 이해하지 못해도 우리는 운전 핸들, 가속/브레이크 페달의 기능과 조작 방법만 이해하면 자동차를 운전할 수 있음
  - 자동차 엔진 : Implementation; 핸들, 가속/브레이크 페달 : Interface
- **Abstraction** in Computer Science : Separation of Interface from Implementation
  - Interface와 Implementation을 구분하는 것, Implementation에 독립적(Independent)인 Interface를 설계하는 것, 구현의 세부 내용을 숨기는 것 (Hiding the details of Implementation) S/W Engineering의 기본.



WEB SERVER AND WEB CLIENT

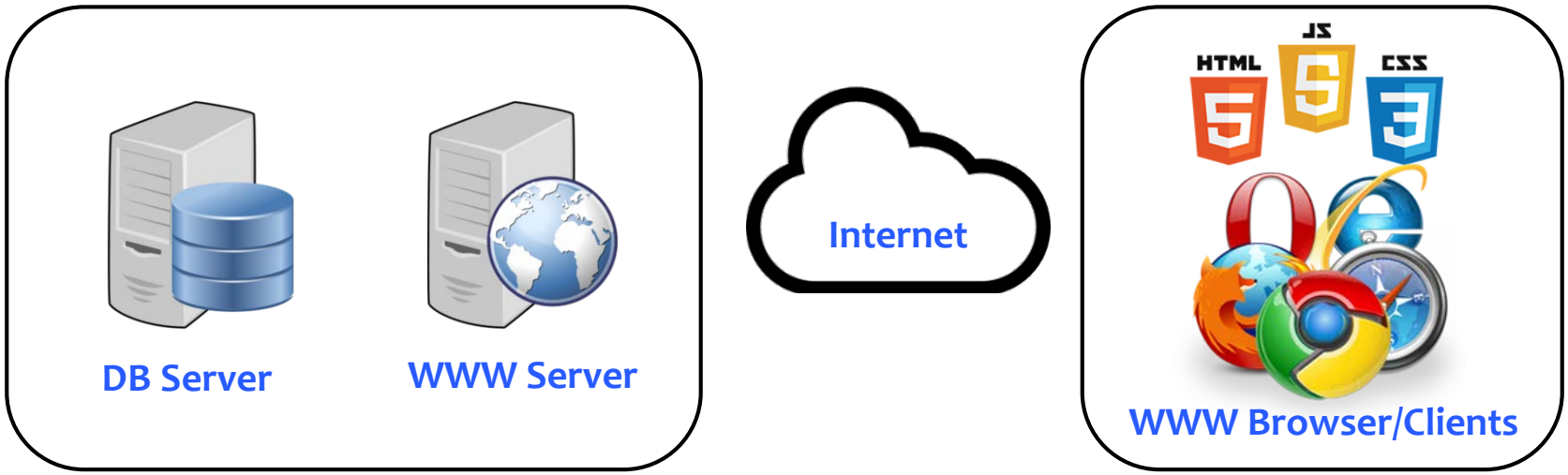
Web Programming

❖ Server Side / Back-End Programming

- 사용자 요구에 따라 동적으로 Web Contents를 생성할 필요가 있음.
- 대표적 예로 DB 서버와 연동하여 데이터를 읽어오거나 쓰고 그 결과를 표출
- PHP, JSP, ASP, …

❖ Client Side / Front-End Programming

- 모든 처리를 서버에서 수행할 경우 서버 부하가 급격히 증가할 수 있고 사용자에게도 빠른 서비스를 제공하기 어려움
- JavaScript



HTTP MESSAGES

URL

http://localhost:8080/friendships/create?my_id=100&user_id=200

❖ Fields of URL

- **Scheme** (http:) : identifies protocol used to fetch the content
- **Host name** (//host.company.com or //localhost): name of the machine running the desired server
- **Server's port number** (8080): allows multiple servers to run on the same machine. Normal Web servers usually run on port 80 (the default)
- **Hierarchical portion** (/friendships/create): identifies a particular request, such as create a new friendship.
- **Query info** (?my_id=100&user_id=200): provides parameters for the request

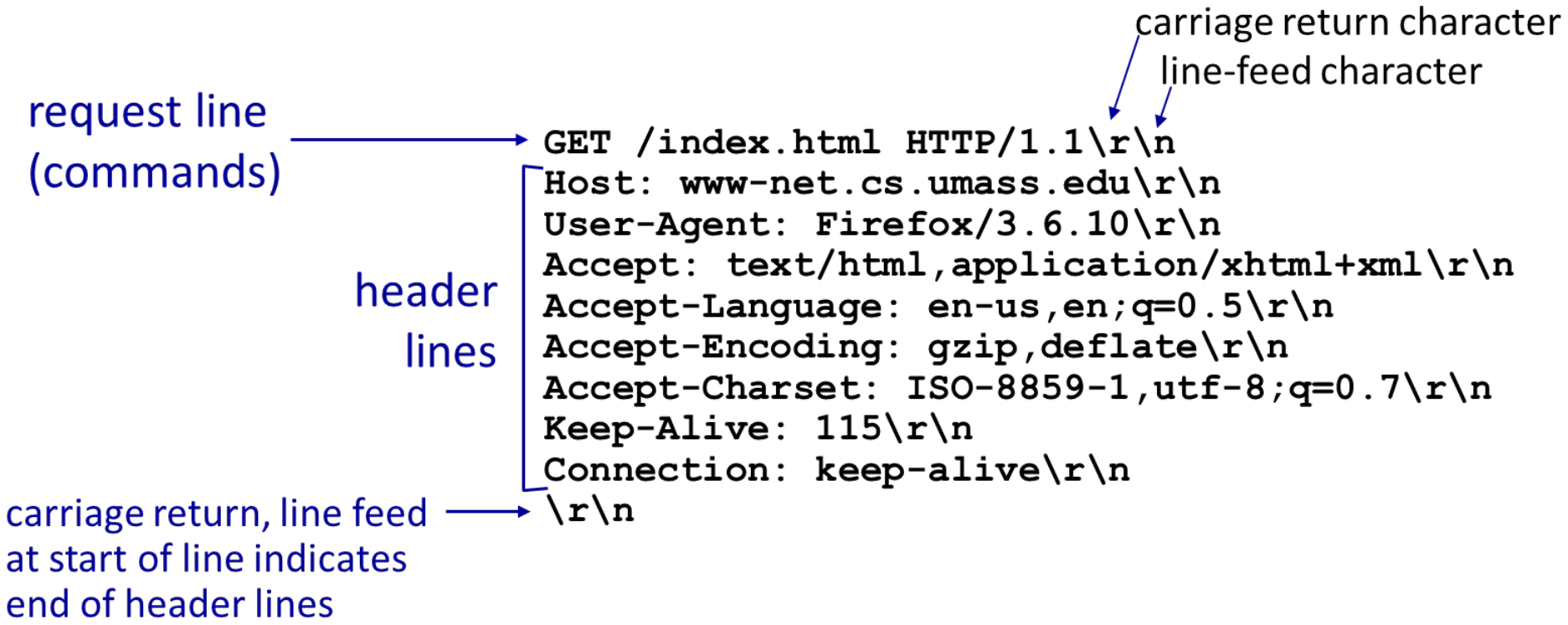
❖ URL encoding

- If a query value contains any character other than A-Z, a-z, 0-9, or any of -_.~ it must be represented as %xx, where xx is the hexadecimal value of the character.
  - " " becomes %20
  - "&" becomes %26, etc.

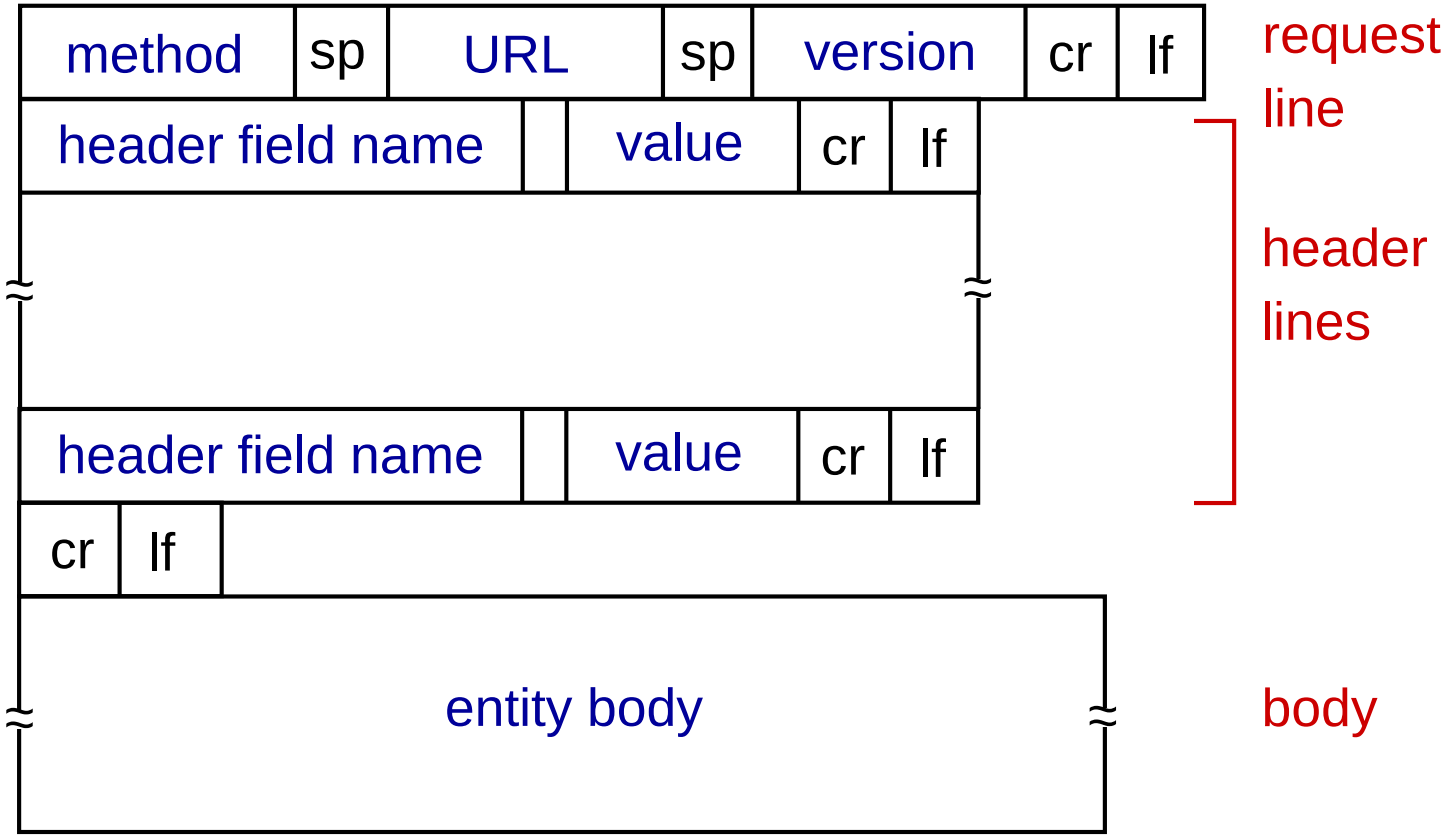
HTTP request message

❖HTTP request message:

- ASCII (human-readable format)



HTTP request message: general format



HTTP response message

status line (protocol status code status phrase)

header lines

data, e.g., requested HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
   GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
   1\r\n
\r\n
data data data data data ...
```

HTTP response status codes

- **status code appears in 1st line in server-to-client response message.**
- **some sample codes:**

200 OK
- request succeeded, requested object later in this message

301 Moved Permanently
- requested object moved, new location specified later in this message (in Location: field)

400 Bad Request
- request msg not understood by server

404 Not Found
- requested document not found on this server

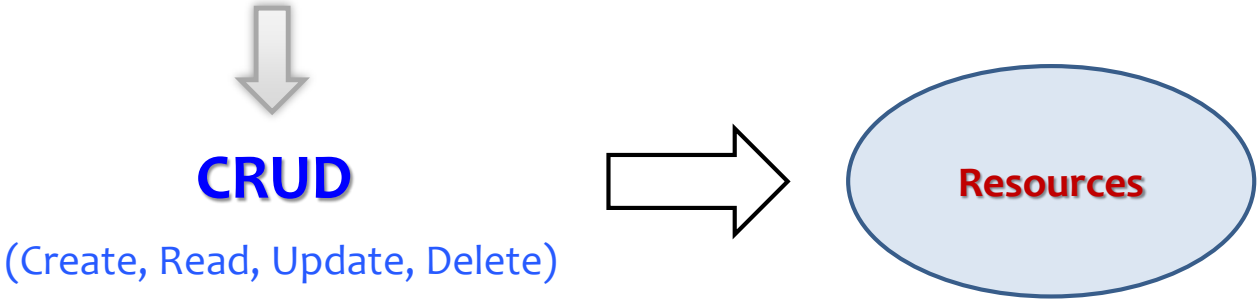
505 HTTP Version Not Supported

REST

REST ?

❖ **Representational State Transfer**

❖ one way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of **Web resources** using a uniform and predefined set of **stateless operations**.



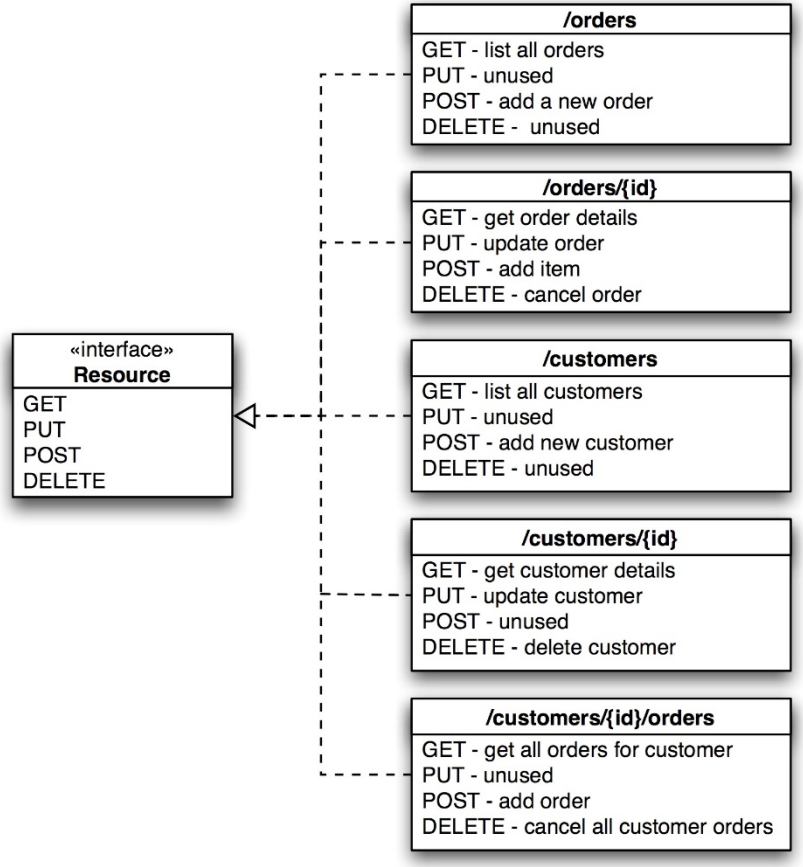
❖ HTTP & REST

- Any web service that is defined on the principles of REST can be called a RestFul web service. A Restful service would use the normal HTTP verbs of GET, POST, PUT and DELETE for working with the required components.
- CREATE – HTTP POST, **READ – HTTP GET, UPDATE – HTTP PUT**, DELETE – HTTP DELETE

REST example

❖ REST – Simple and Easy to Learn

- Things are identified by URLs
  - Consists of nouns
  - Customer, orders
- HTTP Verb to dictate the operation on that resource
- Multiple URLs pointing to same resource



REST – Resource Oriented !!!

❖ It's not a protocol, it's an architectural approach.

- Can be used with legacy XML or modern JSON information transfer format

❖ It's a Guidelines

- HTTP methods and corresponding CRUD (Create, Read, Update, Delete) operation, rec ommendation about URL design.

❖ Architectural Constraints

- Client-Server
- Stateless
- Cacheable
- Uniform Interface
  - Identification of resources : ex) URL
    - Directory like resource structure, Use proper MIME types
  - CRUD Operation: Create, Read, Update, Delete
    - Use HTTP methods for CRUD operations

JSON SYNTAX

JSON Syntax

❖ **JSON: JavaScript Object Notation.**

❖ **JSON is a syntax for storing and exchanging data.**

❖ **JSON Syntax Rules**

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

```
{ "name":"John" }
{ "age":30 }
{ "sale":true }
{ "middlename":null }
```

```
{
"employee":{ "name":"John", "age":30, "city":"New York" }
}
{
"employees":[ "John", "Anna", "Peter" ]
}
```

JSON vs XML

- JSON doesn't use end tag, JSON is shorter
- XML has to be parsed with an XML parser. JSON can be parsed by a standard JavaScript function.

```json
{
    "response": {
        "header": {
            "resultCode": "00",
            "resultMsg": "NORMAL_SERVICE"
        },
        "body": {
            "dataType": "JSON",
            "items": {
                "item": [
                    {
                        "baseDate": "20200510",
                        "baseTime": "1800",
                        "category": "PTY",
                        "nx": 100,
                        "ny": 75,
                        "obsrValue": "0"
                    },
                    {
                        "baseDate": "20200510",
                        "baseTime": "1800",
                        "category": "REH",
                        "nx": 100,
                        "ny": 75,
                        "obsrValue": "86"
                    },
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <header>
        <resultCode>00</resultCode>
        <resultMsg>NORMAL_SERVICE</resultMsg>
    </header>
    <body>
        <dataType>XML</dataType>
        <items>
            <item>
                <baseDate>20200510</baseDate>
                <baseTime>1800</baseTime>
                <category>PTY</category>
                <nx>100</nx>
                <ny>75</ny>
                <obsrValue>0</obsrValue>
            </item>
            <item>
                <baseDate>20200510</baseDate>
                <baseTime>1800</baseTime>
                <category>REH</category>
                <nx>100</nx>
                <ny>75</ny>
                <obsrValue>86</obsrValue>
            </item>
```

NODE JS & EXPRESS JS

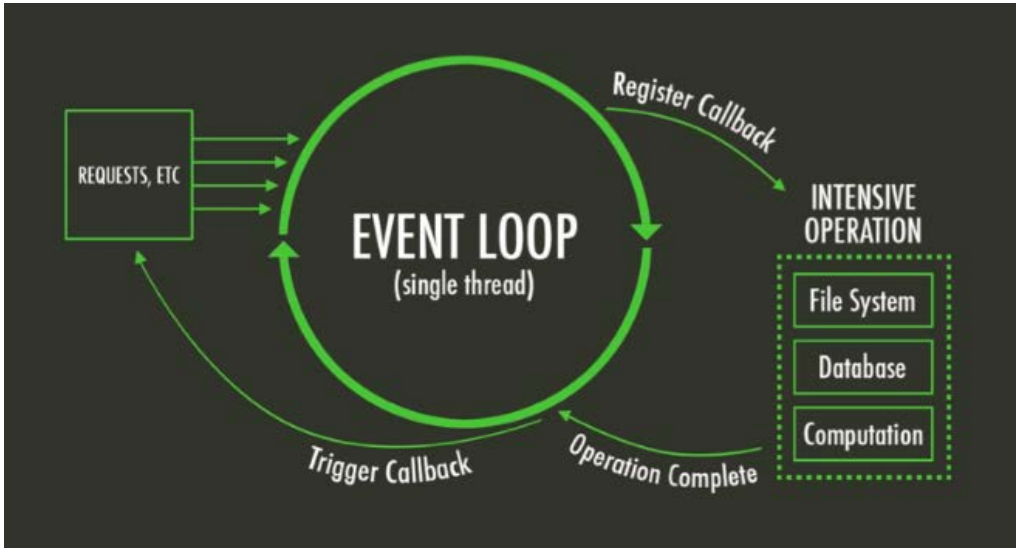
Node.js and Express.js

❖ Node.js: An asynchronous event-driven JavaScript runtime

- Use V8 Javascript engine in Google Chrome
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Event-driven architecture
- Asynchronous I/O

❖ Express.js: Fast, unopinionated, minimalist web framework for Node.js

- Makes Node.js web application development fast and easy



요 약

- Web server and web client
- HTTP messages
- JSON Syntax
- REST API
- Node JS and Express JS