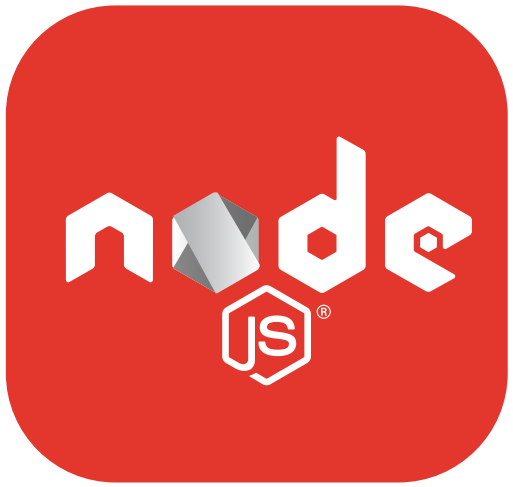
PBL2 : OpenSource Chatting Ap

권순우, 김덕영, 김태웅, 우경완,
이세명, 이수정

week1

PlatForm



1. 이벤트 루프 기반의 비동기 방식
2. V8 고성능 자바스크립트 엔진
3. 확장성 있는 모듈화된 구조

Node.js extended modules

Express

대표적 웹 프레임워크 모듈

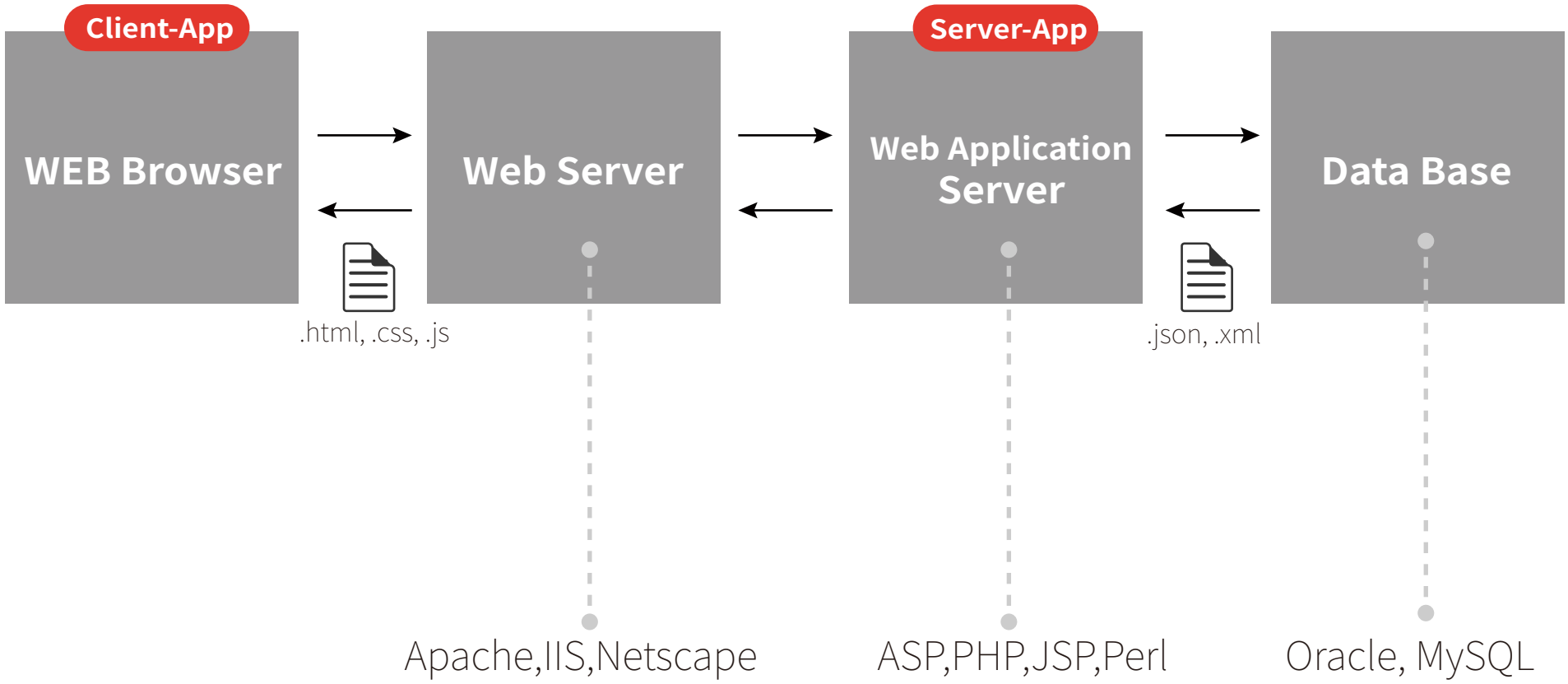
socket.IO

비동기통신과 웹소켓을 위한
라이브러리 모듈

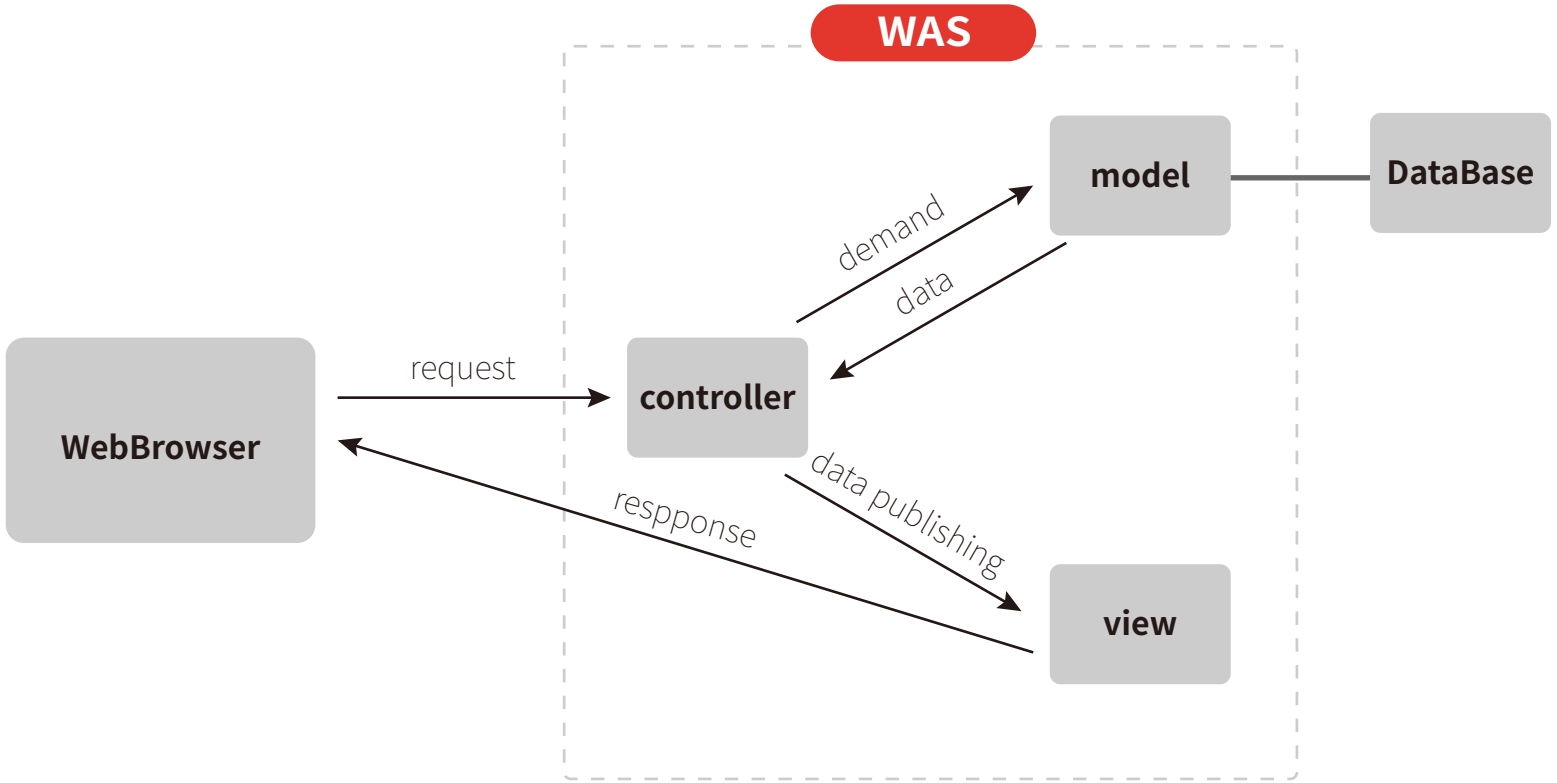
DBMS 호환 모듈

Moongoose, mongolian
node_mysql
redis

Web Service Architect

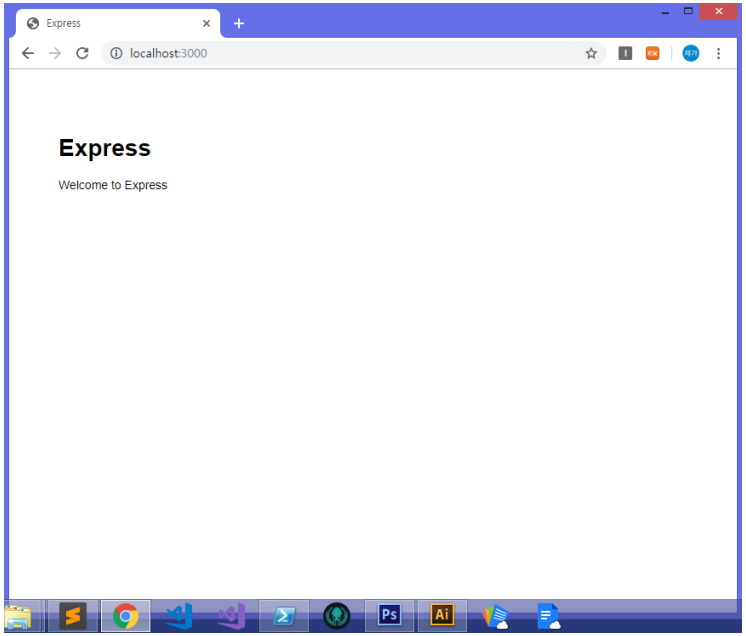


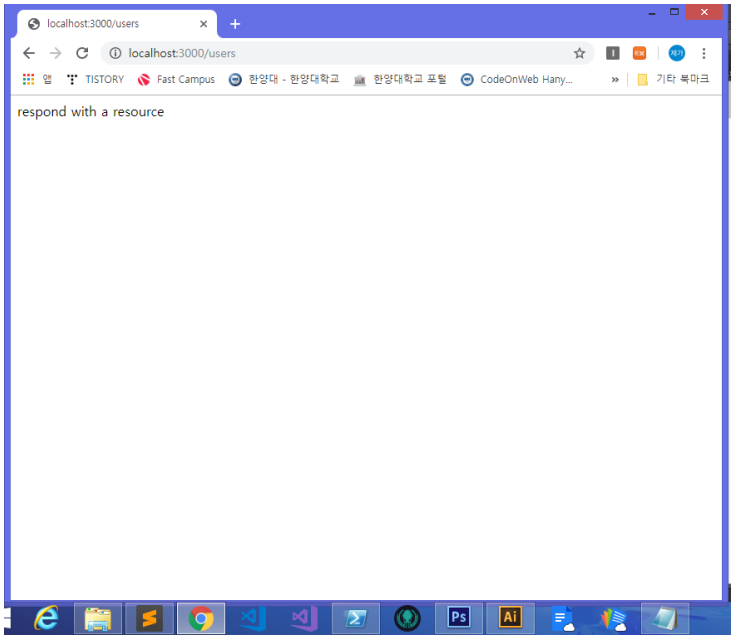
Web app Design Pattern



Express Start project

```
$ npm install -g express
$ npm install -g express-generator
$ express [directory-name]
$ npm install
$ npm start
```





Express directory 구조

project directory

bin

node_modules

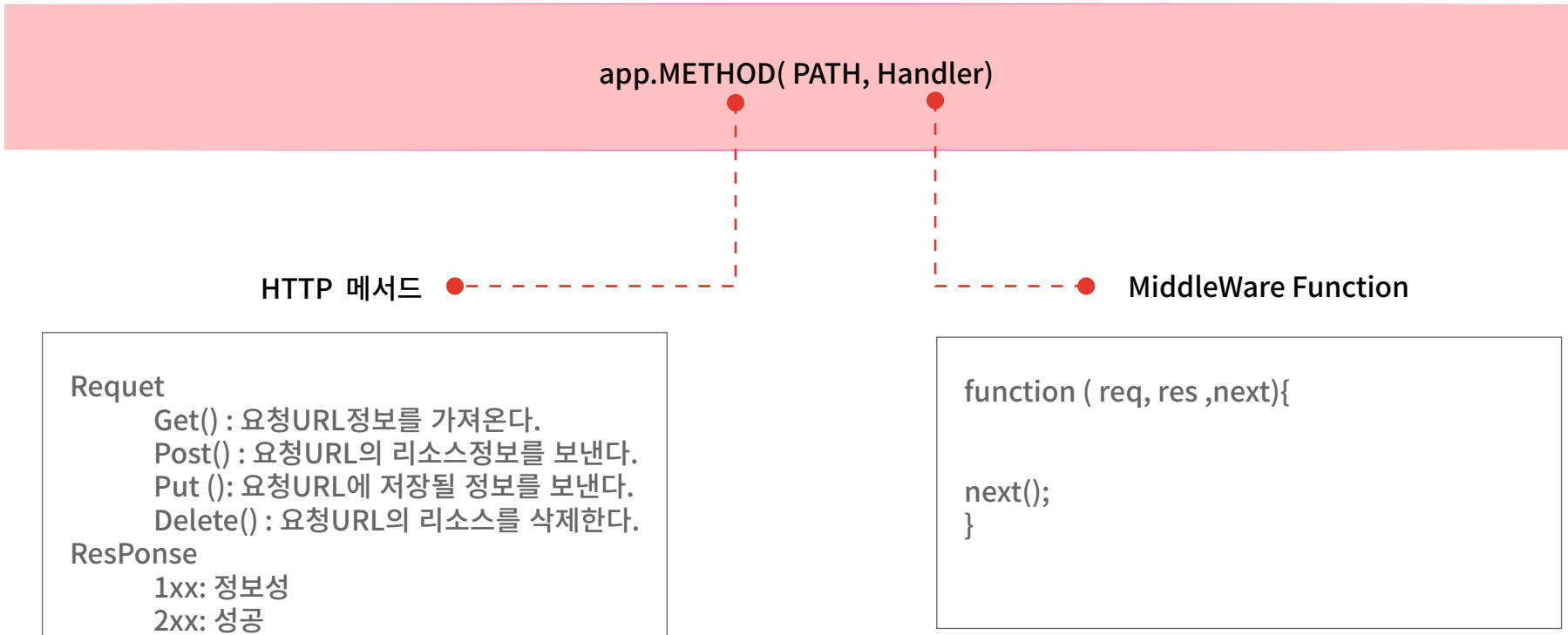
public - - - - - - - - → 클라이언트 측에서 사용할 이미지, 자바스크립트, CSS등의 리소스

routes - - - - - - - - → 라우팅 하기위한 미들웨어 함수?

views - - - - - - - - → 템플릿 엔진이 해석 할 수 있는 화면 구성, 뷰 템플릿 파일

app.js - - - - - - - - → 메인 스크립트

Express 기본 Routing



default app.js 분석

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();


// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
```

Imports

기본 Node.js 모듈 및 Express 표준 미들웨어 함수 호출

호출할 사용자 미들웨어 함수 인덱싱

Express 앱 생성

Setting

뷰 템플릿 폴더와 사용할 뷰 engine을 설정

기본 모듈함수 호출 및 설정

default app.js 분석

```
app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ?
err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

module.exports = app;
```

maping

"/" 에 대한 호출에 대한 라우팅

"/user"에 대한 호출에 대한 라우팅

mapping 안된 호출에 대하여
404Error인지 확인하는 함수

Error핸들러 : 받은 에러 렌더링