web server

client
Browser

req →

live server

page01.html

res

req

page01.html

get data

res

backend Server

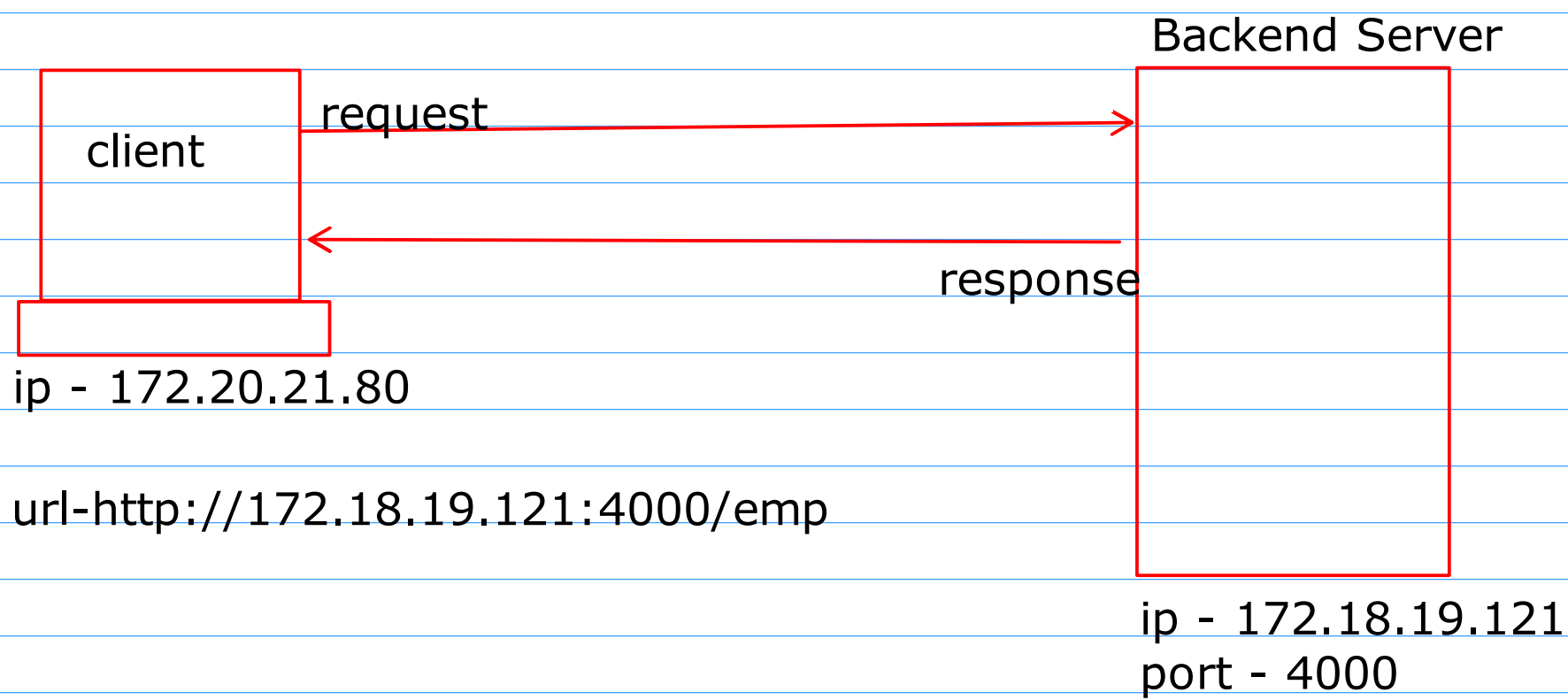Reqres

FrontEnd ->
HTML
CSS
BootStrap
JavaScript
DOM -> HTML-CSS - JS
AJAX
     - XHR
     - $.ajax()
     - fetch..then, fetch..async..await

Backend->

Backend Server

client

request →

← response

ip - 172.20.21.80

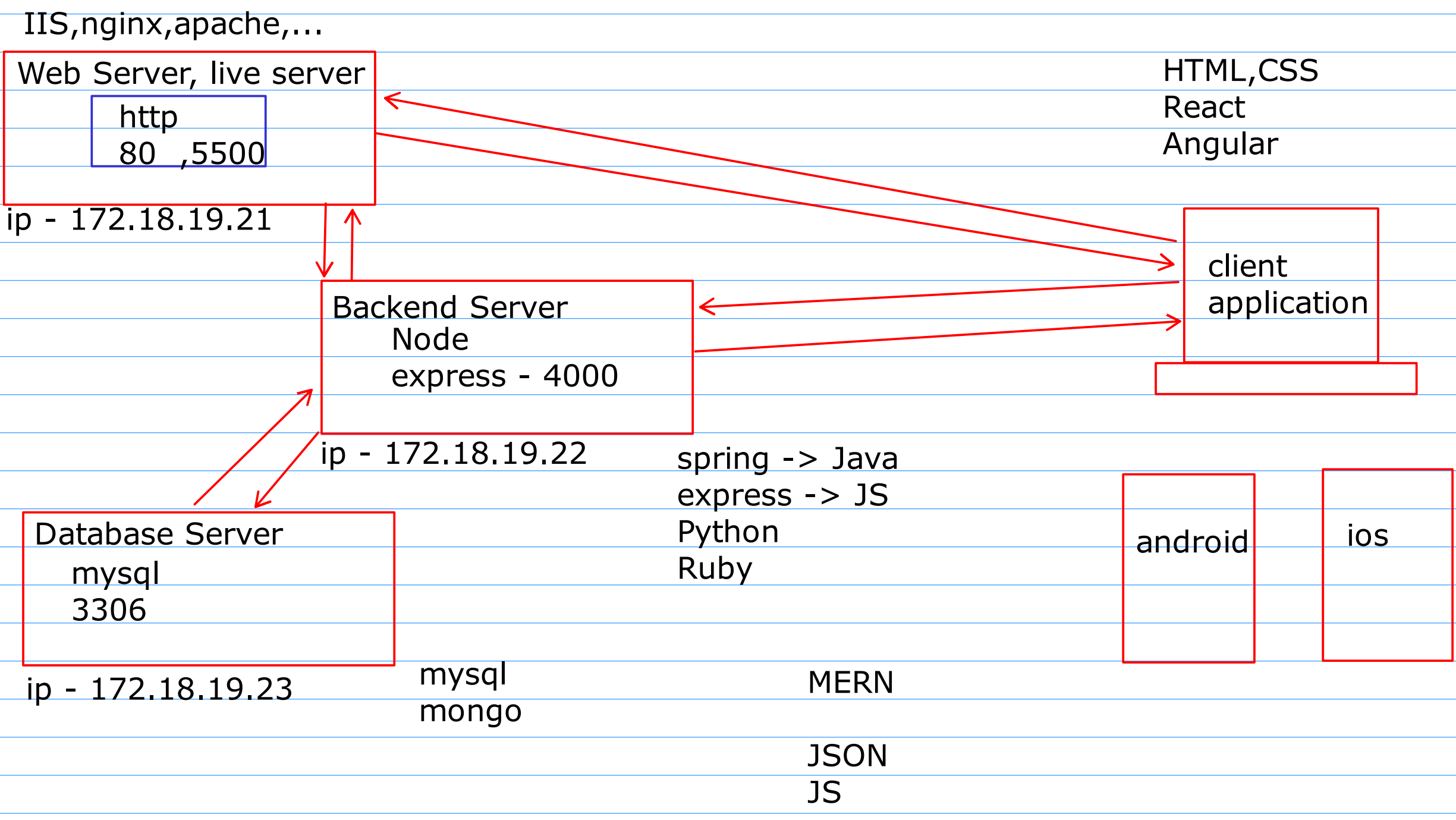url-http://172.18.19.121:4000/emp

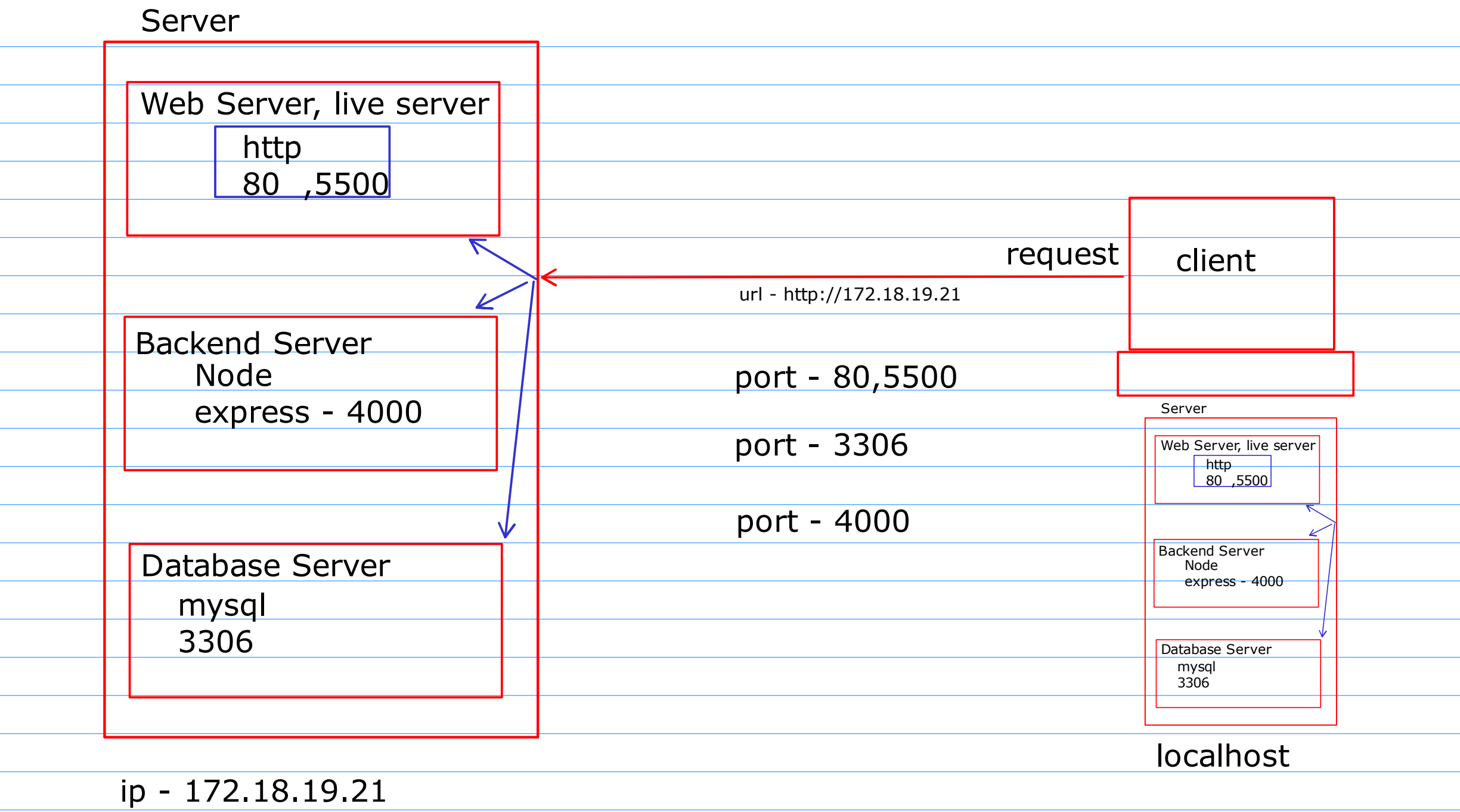ip - 172.18.19.121
port - 4000

Method - Path
GET   - /emp (Server will return all the employees)
POST - /emp (Insert/Add the employee in the database)
PUT   - /emp (Update the given Employee in db)
DELETE - /emp (Delete the given employee from db)

Server

Web Server, live server
http
80 , 5500

request   client

url - http://172.18.19.21

Backend Server
Node
express - 4000

port - 80,5500

port - 3306

port - 4000

Database Server
mysql
3306

Server

Web Server, live server
http
80 , 5500

Backend Server
Node
express - 4000

Database Server
mysql
3306

localhost

ip - 172.18.19.21

IIS,nginx,apache,...

Web Server, live server
http
80 , 5500

ip - 172.18.19.21

HTML,CSS
React
Angular

client
application

Backend Server
Node
express - 4000

ip - 172.18.19.22

spring -> Java
express -> JS
Python
Ruby

android   ios

Database Server
mysql
3306

ip - 172.18.19.23   mysql
mongo

MERN

JSON
JS

Module

    - It is a way to achive modularity in JS applications in Node JS platform

    - Their are 2 types of modules in Node

        1. User Modules

        2. Built-in Modules

Task-1   it will start first

    op1

    op2

    op3

Task-2   it will not start till task-1 is finished

    op1

    op2

/user   GET   - Return list of users

/user POST   - save the user in db

/user PUT   - edit/update the user from db

/user delete   - delete the user from db

/product   GET   - Return list of products

/product POST   - save the product in db

/product PUT   - edit/update the product from db

/product delete   - delete the product from db