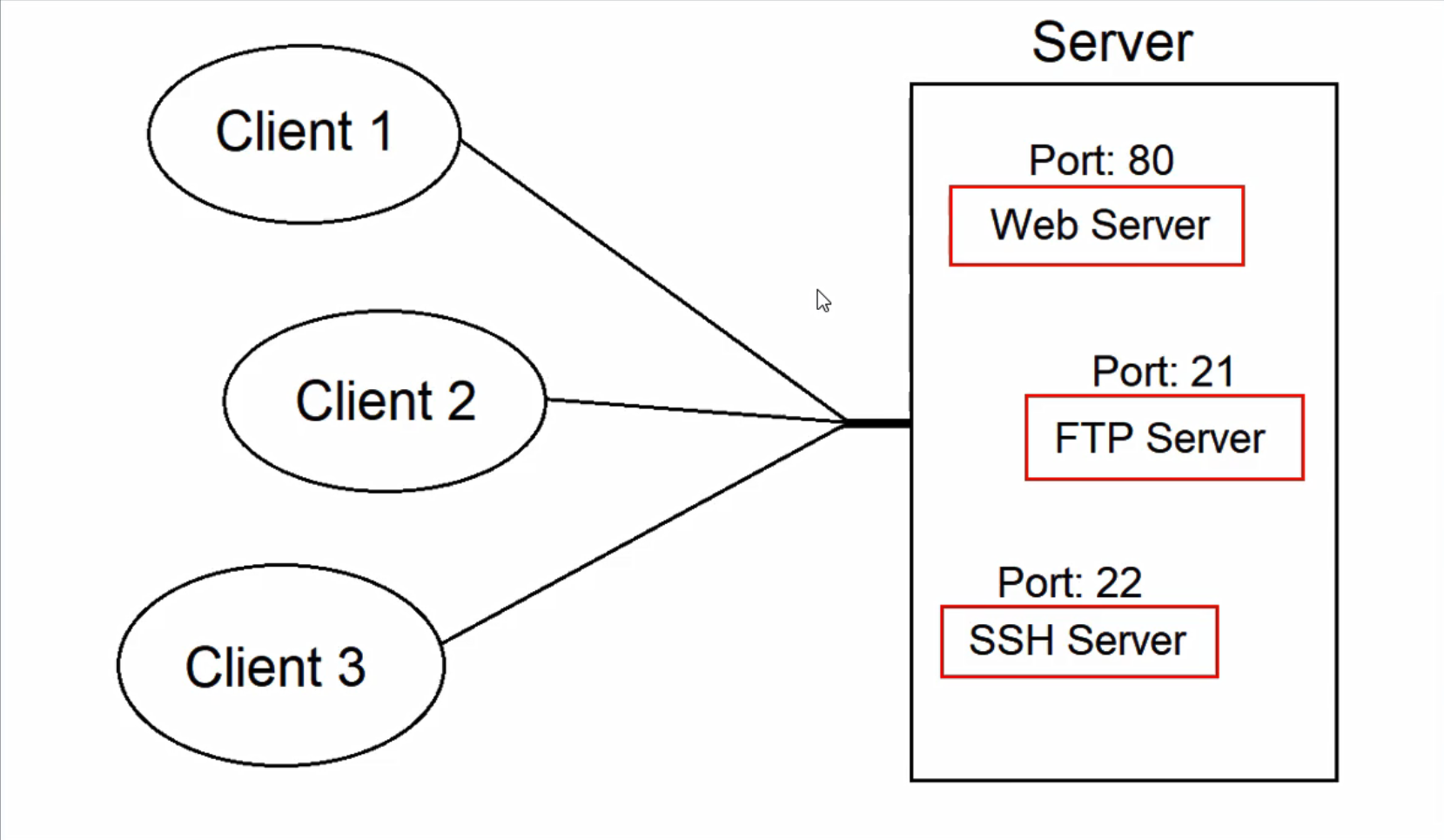
1- ساخت وب سرور با http



import http from "http";

const server = http.createServer(() => {

    console.log("A request received");

});

این کال بک زمانی اجرا میشه که رویداد request برای آبجکت server اتفاق بیفته .

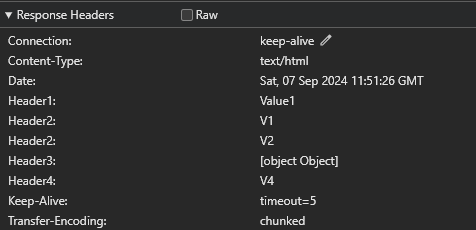
server.listen(3000, () => {

    console.log("server start on port " + 3000);

});

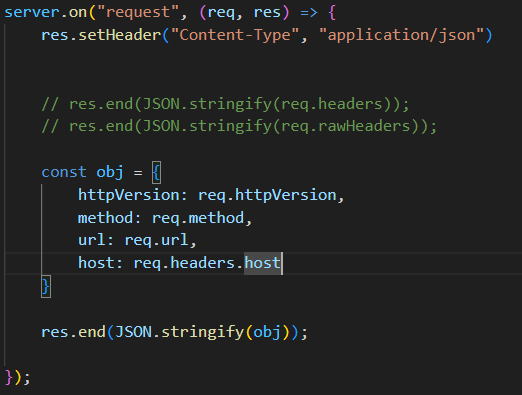
(<http://127.0.0.1:3000/> لوپ بک )





نکته : ست هدرو قبل متدهایی که write میکنن باید بنویسیم . تو پاسخ های اچ تی تی پی اول هدر ارسال میشه بعد بدنه . وقتی که داریم رایت میکنیم در واقع داریم توی بدنه مینویسیم . حالا وقتی بدنه رو داریم مینویسیم دیگه نمیتونیم هدر هارو تغییر بدیم چون قبلش هدر ها ارسال شدن .

نکته : res یک استریم رایتبل و req یک استریم ریدبل هست .







2 – شروع کار با express

const app = express();

app.use((req, res) => {

    console.log(req.url, req.method);

    res.send("aaaaa");

});

app.use("/", (req, res) => {

    console.log(req.url, req.method);

    res.send("bbbbbb");

});

app.use("/test", (req, res) => {

    console.log(req.url, req.method);

    res.send("ccccc");

});

app.listen(3000, () => {

    console.log("http://localhost:3000");

});

3-4-5- میدلویر ها

مت send خودش هدر Content-Type رو تشخیص میده و تو هدر ریسپانس ست میکنه و میفرسته.

از send بیشتر زمانی استفاده میکنیم که میخوایم یه رشته رو ارسال کنیم .

نکته : بعد از متد next() هم میتونیم دستوراتی داشته باشیم تو تابع . ولی وقتی رسید به نکست ، میره به میدلویر بعدی ، و کد های اونو اجرا میکنه و دوباره برمیگرده و کد های بعد از next() رو اجرا میکنه.

تو یکی از میدلویر ها باید یچی سند بشه

app.use("/route1", (req, res, next) => {

    next();

    console.log("Hello from first /route1");

});

app.use("/route1", (req, res, next) => {

    console.log("Hello from first /route1");

    // res.send("Hello from first /route1");

    next();

});

گفتیم که پاسخی که ب کلاینت ارسال توسط یکی از این میدلویر ها ارسال میشه در واقعا همون میدلویری که متد send رو داره استفاده میکنه و باقی میدلویر ها مستقیما چیزی رو برای کلاینت ارسال نمیکنه . خوب این میدلویر هایی که چیزی سند نمیکنن به چه دردی میخورن ؟   
اینا اکثرا به این صورتن که میان رو req یا res تغییراتی ایجاد میکنن .   
لاگ میگیرن و ...

نکته : آبجکت های req و res واحد هستن ینی اگه تغییراتی روش انجام بدیم ، اون تغییرات اعمال میشه و اون req و res با تغییرات انجام شده فرستاده میشه به میدلویر بعدی و ...