Node.js 기초

학습목표

- **REST API** 개념 이해
- Node.js **개발환경** 세팅
- Node.js와 Express **관계**
- Node.js express **프로젝트**
- 기본 **예제** 따라하기
- **MVC 패턴** 최적화 하기

1. REST API 개념 이해

REST API 개념 이해

Representational State Transfer (표현 상태 전이)

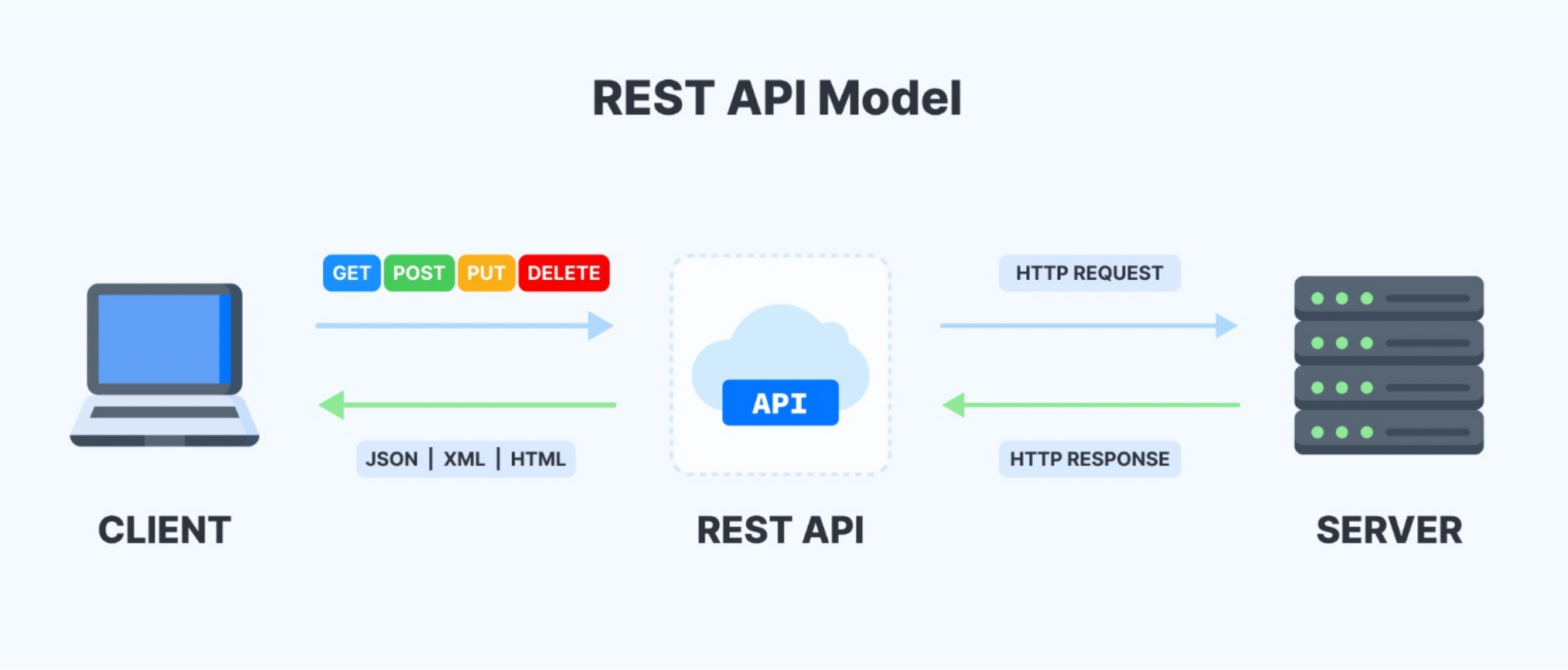
=

리소스 + 메소드 + 메세지

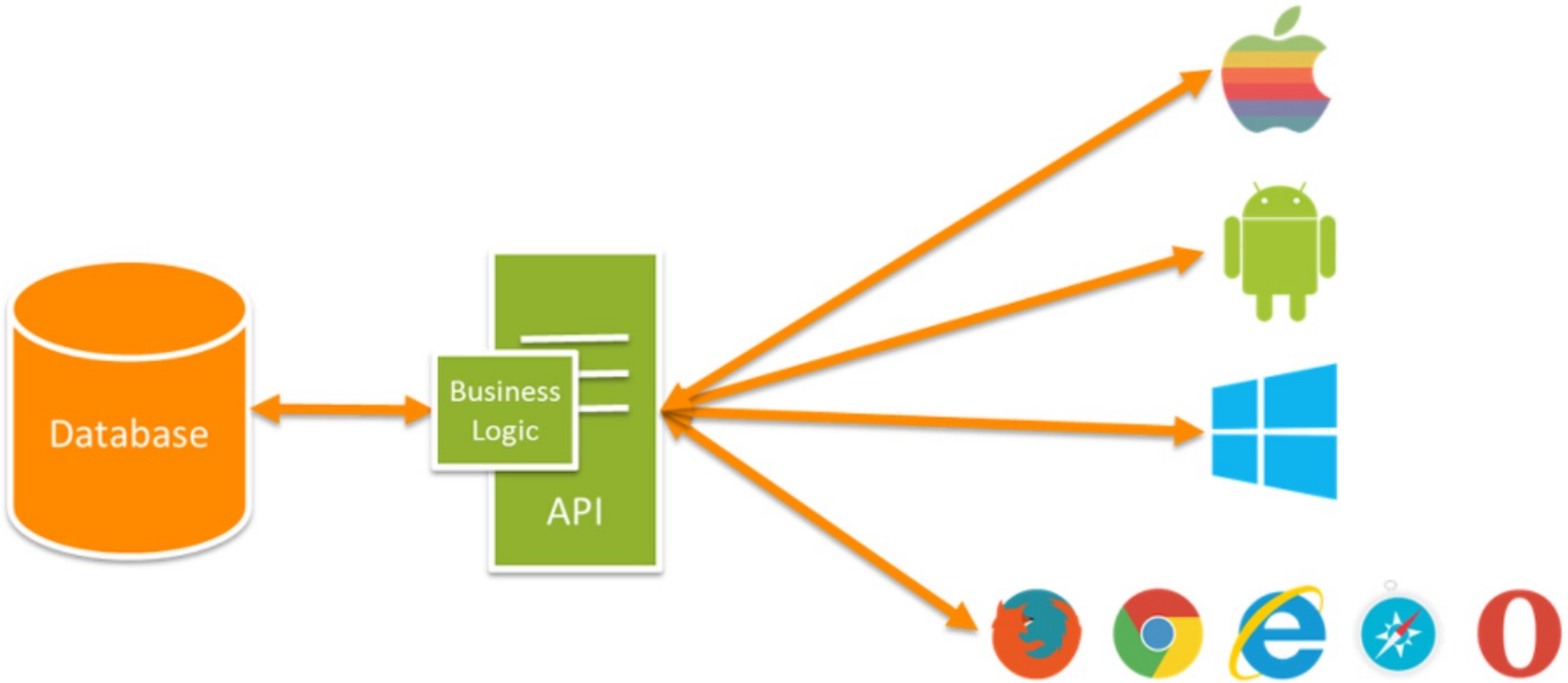
(ex.)

User + POST + “Austin”

REST API 장점


REST API Model
GET
POST
PUT
DELETE
HTTP REQUEST
API
JSON | XML | HTML
HTTP RESPONSE
CLIENT
REST API
SERVER

REST API 장점



2. node.js 개발환경 설정

Node.js 개발환경 설정

1. Node.js 설치 (`node -v`으로 확인)
2. NPM 설치 (Node Package Manager, `npm -v`으로 확인)

3. node.js와 express의 관계 이해

Node.js와 express

Node.js와 express

Node.js와 express

Framework – 일정한 형태와 필요한 기능을 갖추고 있는 뼈대

Library – 특정 기능을 모아둔 코드 및 함수의 집합

4. node.js express 프로젝트

1. 전체 디렉토리 만들기

Mkdir node-practice

Cd node-practice

Code .

1. 전체 디렉토리 만들기

Mkdir node-practice

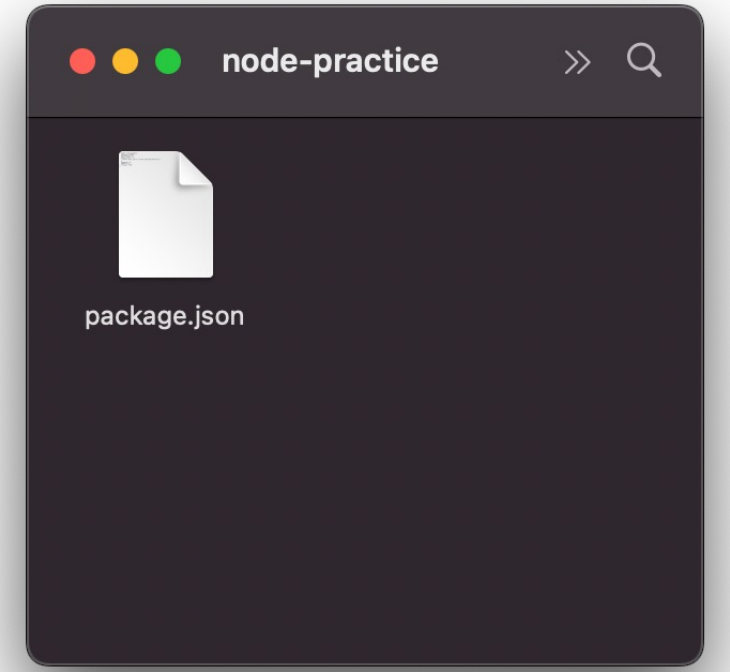
Cd node-practice

Code .

2. NPM 기본 설정하기

Npm init

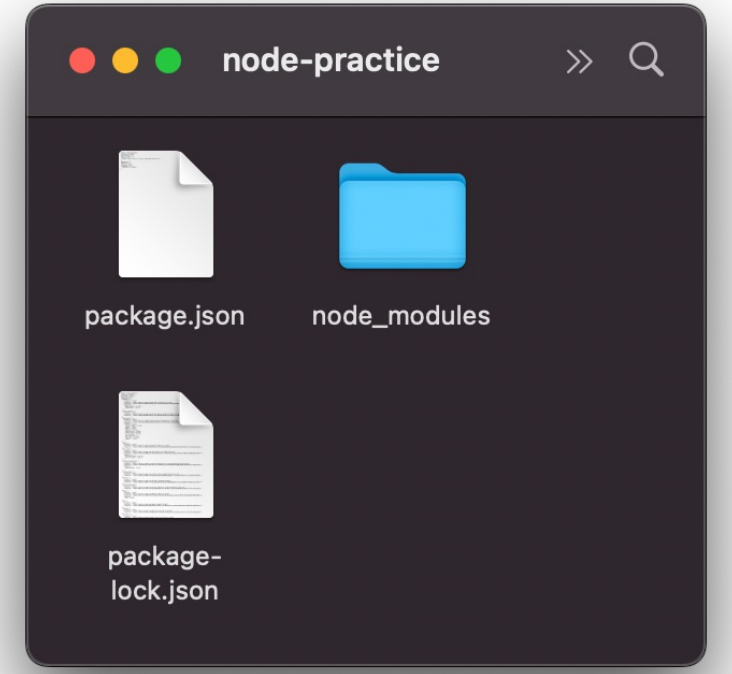
or npm init –y



2. NPM 기본 설정하기

Npm install express --save

or npm i express -s



5. 기본 예제 따라하기

2. NPM 기본예제 설정하기

```
const express = require("express");
const app = express();
```

1. 프레임워크를 불러오는 단계

```
app.get("/", (req, res) => {
    res.send("Hello, world")
});
```

2. 메소드를 설정하는 단계

```
const PORT = 3000
app.listen(PORT, () => {
    console.log("서버 가동")
})
```

3. 서버를 실행하는 단계

2. NPM 기본예제 설정하기

```
const express = require("express");
const app = express();
```

```
app.get("/", (req, res) => {
    res.send("Hello, world")
});
```

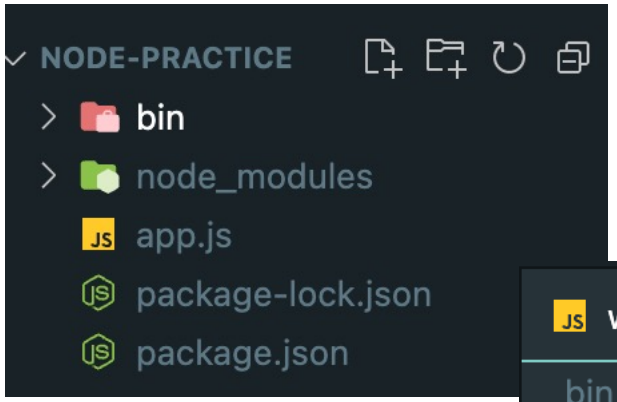
MVC

```
const PORT = 3000
app.listen(PORT, () => {
    console.log("서버 가동")
})
```

bin

5. MVC 패턴 최적화하기

1. NPM 기본예제 설정하기



www.js

bin > www.js

```
1  "use strict";
2
3  const app = require("../app")
4
5
6  const PORT = 3000
7  app.listen(PORT, () => {
8      console.log("서버 가동")
9  })
```

app.js

app.js

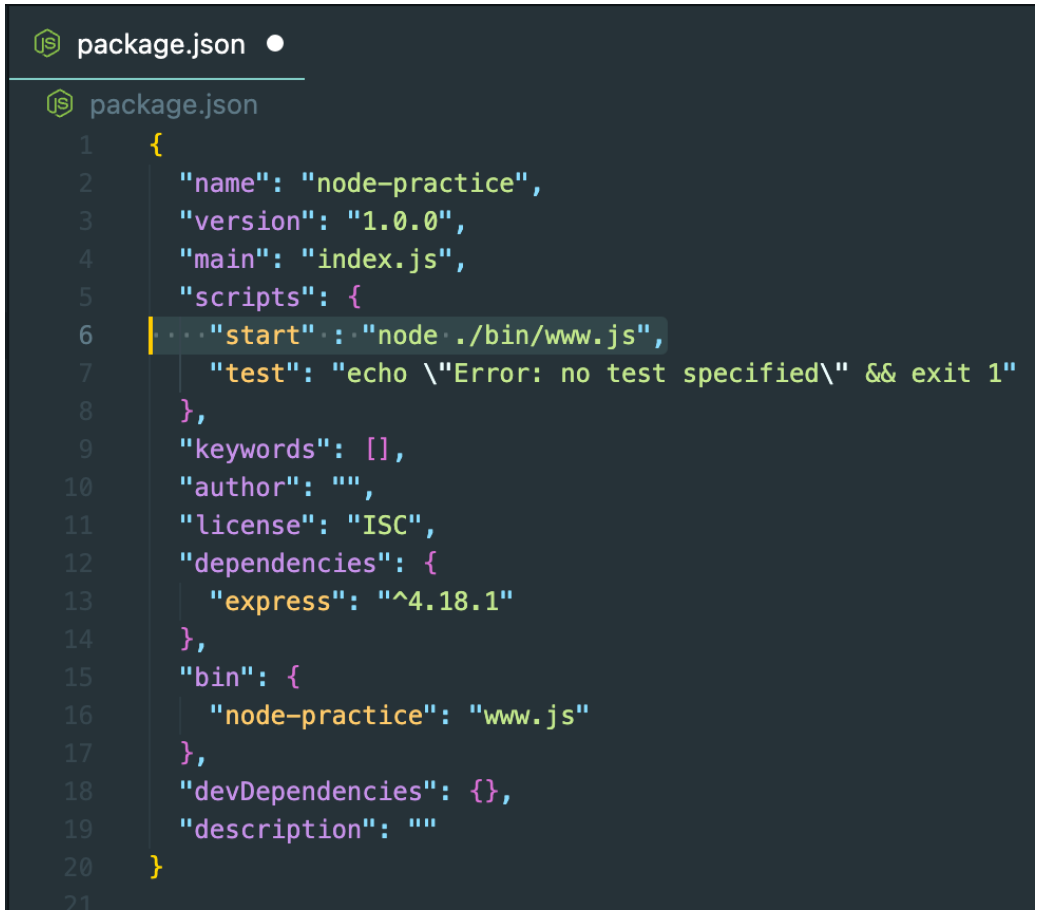
```
1   const express = require("express");
2   const app = express();
3
4
5   app.get("/", (req, res) => {
6       res.send("Hello, world")
7   });
8
9   // const PORT = 3000
10  // app.listen(PORT, () => {
11  //     console.log("서버 가동")
12  // })
13
14  module.exports = app;
```

2. 서버 실행 부분 분리하기

Npm init

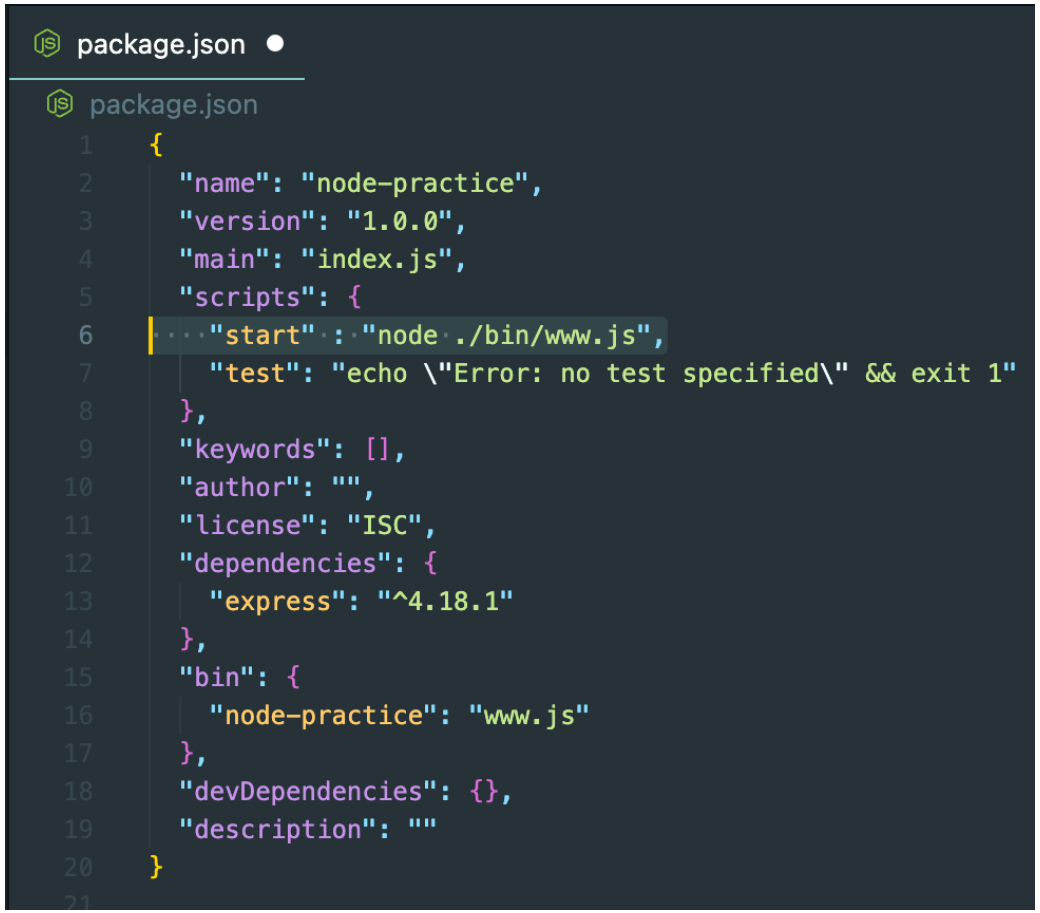
4. Package 수정하기

Npm start

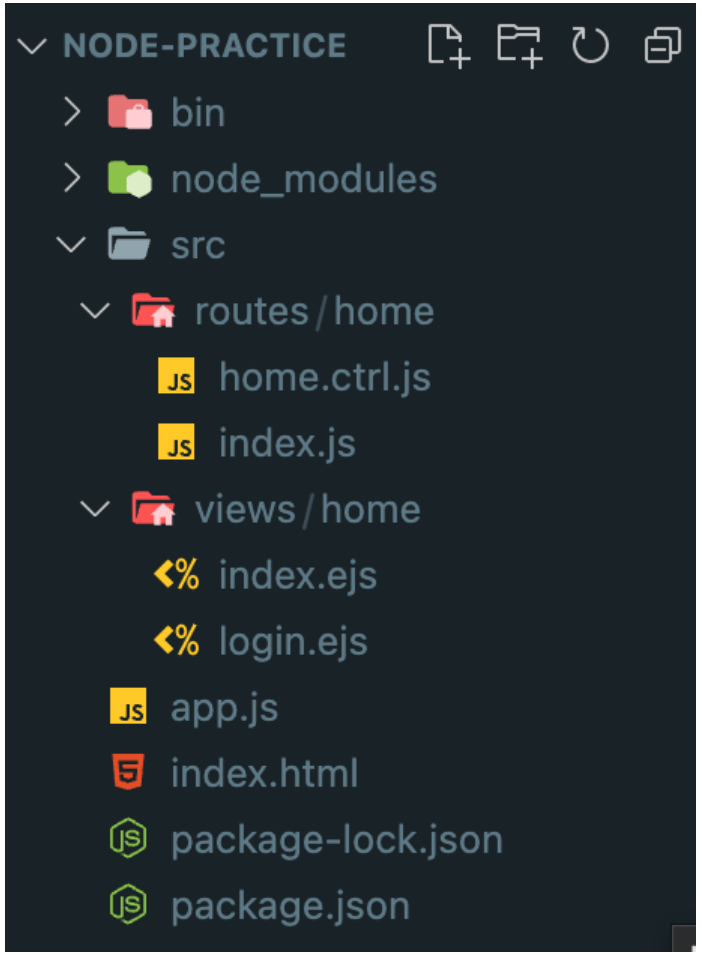


4. Package 수정하기

Npm start



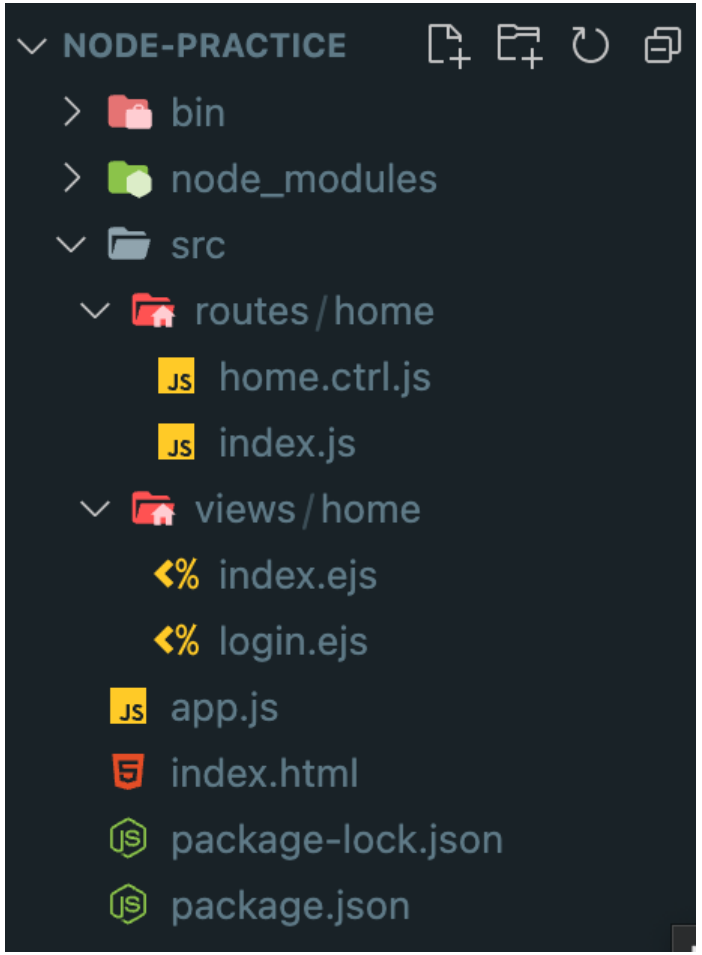
5. MVC 분리하기



app.js

```
const express = require("express");
const app = express();

//·MVC·중·V
app.set("views",·"./src/views");
app.set("view·engine",·"ejs");


//·MVC·중·C
const·home·=·require("./src/routes/home");
app.use("/",·home);

// app.get("/", (req, res) => {
//     res.send("Hello, world")
// });

// const PORT = 3000
// app.listen(PORT, () => {
//     console.log("서버 가동")
// })

module.exports = app;
```

5. MVC 분리하기



index.ejs

src > views > home > index.ejs

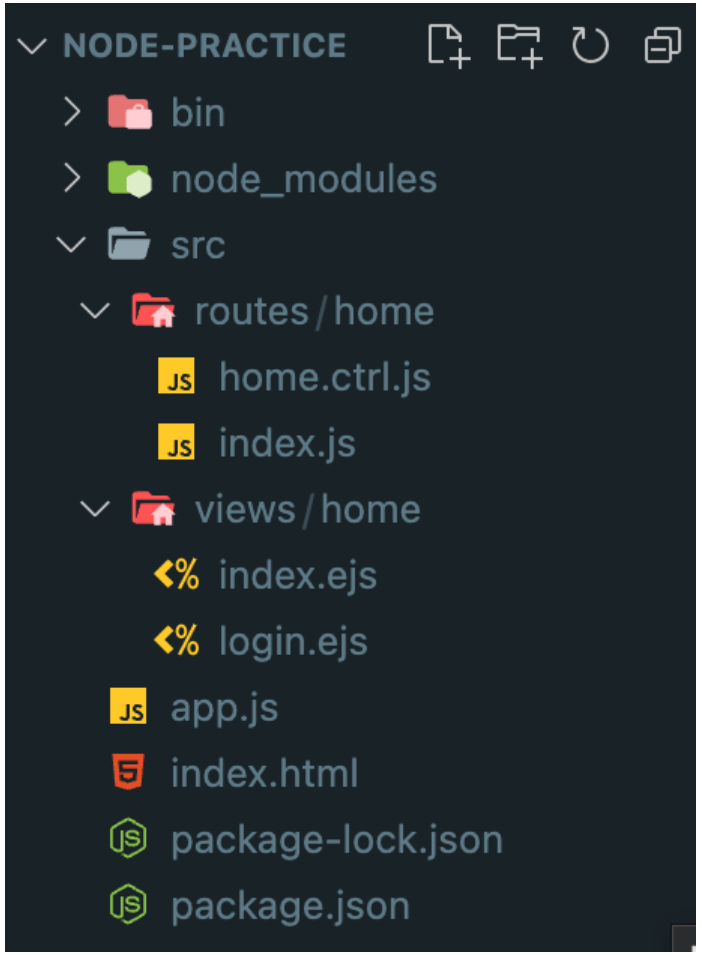
```ejs
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home</title>
  </head>
  <body>
    여기는 루트입니다
  </body>
</html>
```

login.ejs

src > views > home > login.ejs

```ejs
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <script src="js/home/login.js" defer></script>
    <title>Document</title>
  </head>
  <body>
    <input id="id" type="text" placeholder="아이디" /><br />
    <input id="psword" type="text" placeholder="비밀번호" /><br />
    <button>로그인</button>
  </body>
</html>
```

5. MVC 분리하기



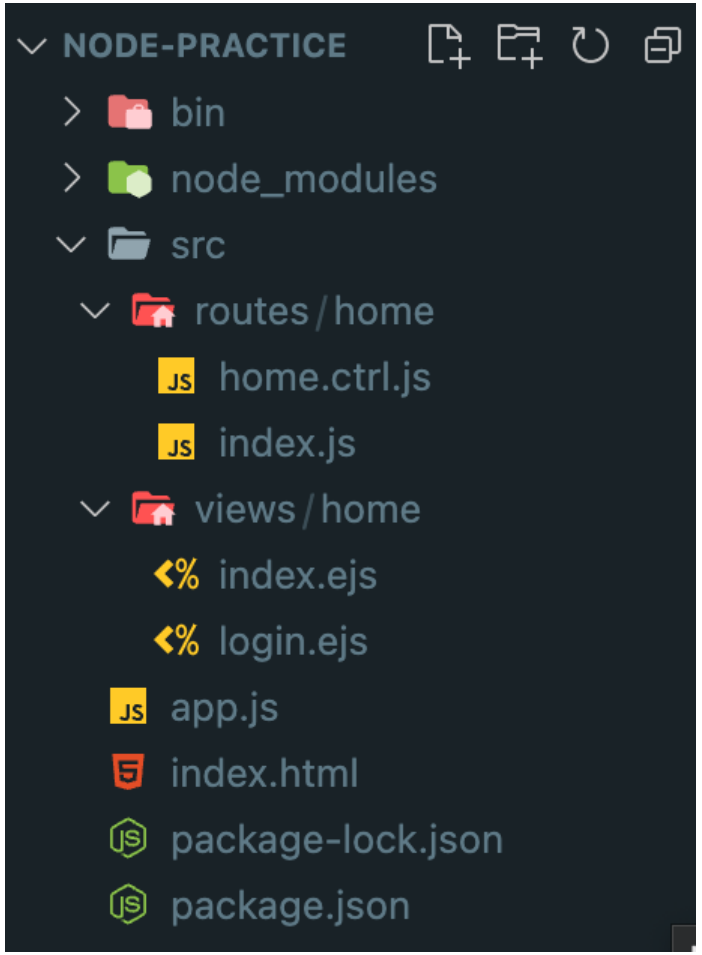
src > routes > home > index.js

```
"use strict";

const express = require("express");
const router = express.Router();

const ctrl = require("./home.ctrl");

// 메소드 정의
router.get("/", ctrl.output.home);


module.exports = router;
```

src > routes > home > home.ctrl.js

```
"use strict";

const output = {
    home: (req, res) => {
        res.render("home/index");
    },
}

const input ={}

module.exports = {
    output,
    input
}
```

5. MVC 분리하기



src > routes > home > index.js

```
"use strict";

const express = require("express");
const router = express.Router();

const ctrl = require("./home.ctrl");

// 메소드 정의
router.get("/", ctrl.output.home);


module.exports = router;
```

src > routes > home > home.ctrl.js

```
"use strict";

const output = {
    home: (req, res) => {
        res.render("home/index");
    },
}

const input ={}

module.exports = {
    output,
    input
}
```