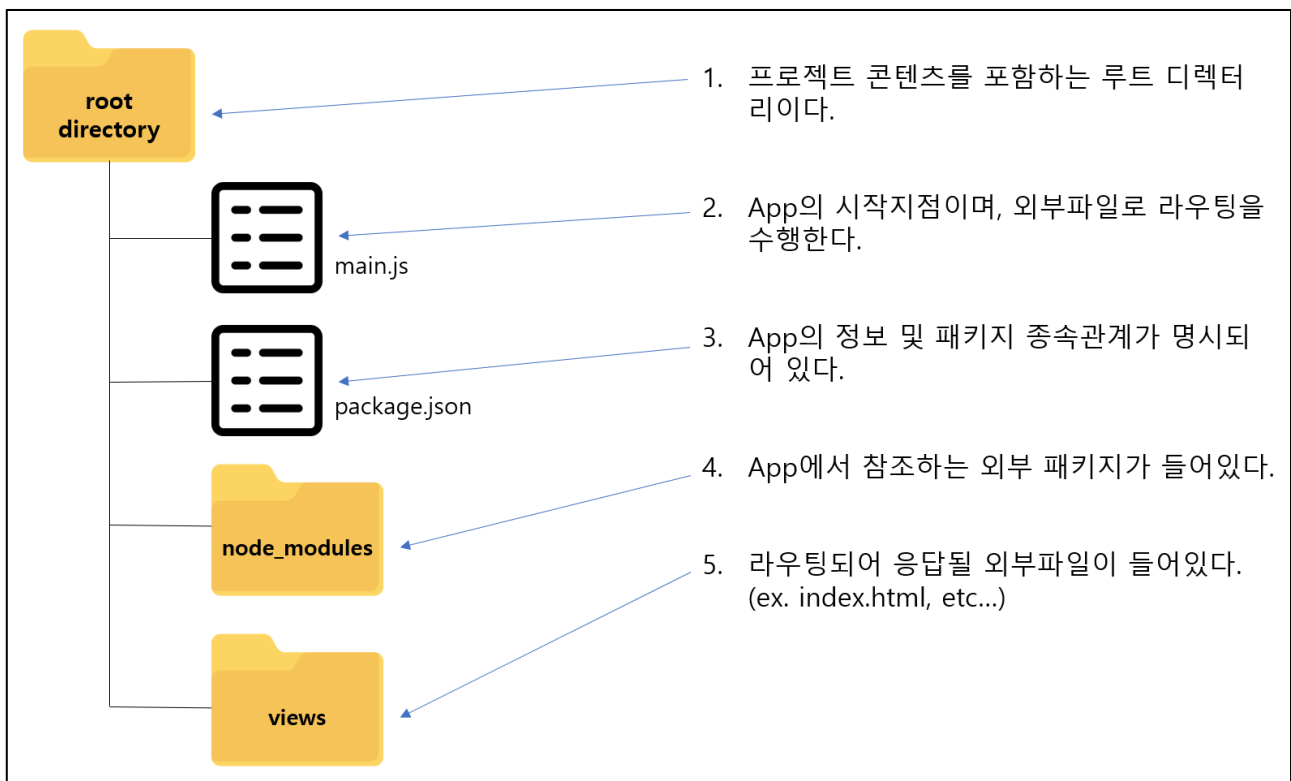


라우트와 외부파일

라우트와 외부파일

- 앞서 수행하였던 라우팅(5절) 예제는 직접 HTML태그를 JavaScript파일에 기입하여 페이지를 응답하였다. 하지만, 이러한 방법은 귀찮으며 JavaScript파일의 구조가 복잡해지는 단점이 존재한다.
 - 앞서 단점을 보완하고 구조적으로 이해하기 쉽게 파일로 응답하는 것을 해당 6절에서 수행한다.
 - JavaScript파일만이 아닌 다른 파일에 접근하기 위해 사용하는 모듈은 fs이다.
fs모듈은 Node.js를 설치하는 과정에 자동으로 설치됨으로 require("fs")를 통해 모듈을 불러올 수 있다.
- ❖ 예제를 진행하기 전 4챕터(웹 서버 만들기)에서 생성한 루트 디렉터리를 복사해 새 디렉터를 생성한다. 그 후 아래와 같은 구조로 **views**를 생성한다.



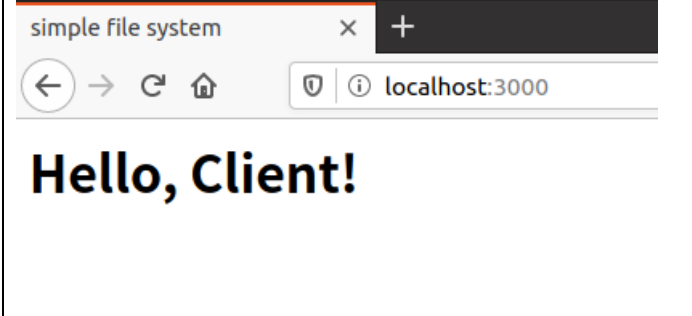
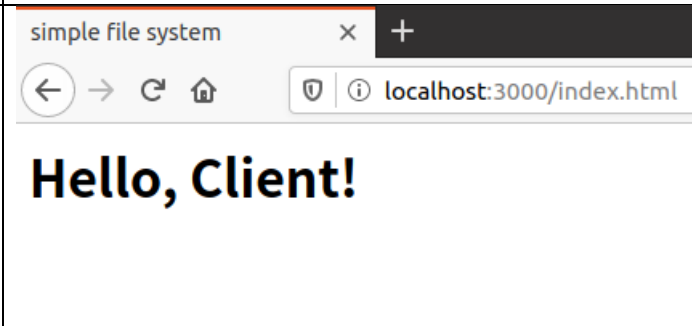
예제1) 외부파일로 응답하기

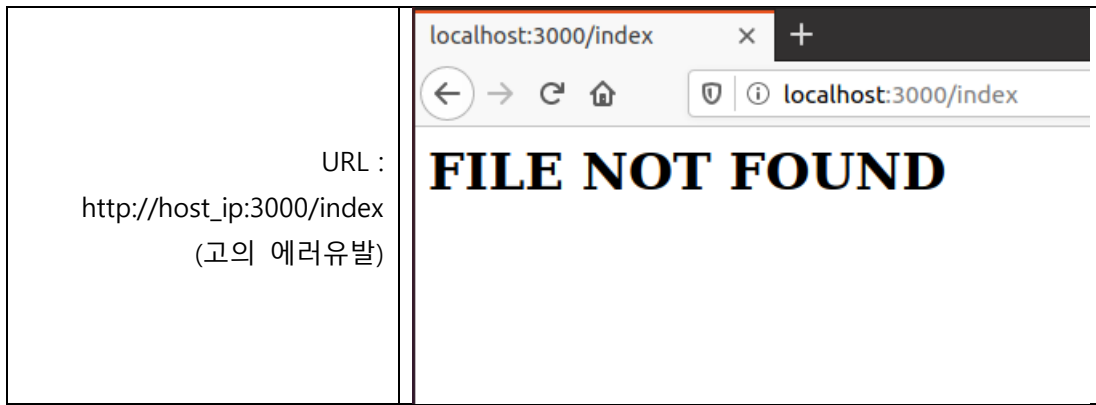
1. 전 챕터(4)에서 만들었던 HTTP 웹 서버의 루트 디렉터를 복사하여 새로운 디렉터리 생성
2. 복사된 디렉터리내 main.js의 이름을 main_fs.js로 변경
: mv main.js main_fs.js
3. 외부파일이 저장될 views디렉터리 생성
: mkdir views

4. main_fs.js에 아래의 내용과 같이 입력

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> "use strict"; const port = 3000; const http = require("http"); const httpStatus = require("http-status-codes"); const fs = require("fs"); // 외부파일 접근 모듈 const app = http.createServer(); const getJSONString = function(obj){ return JSON.stringify(obj, null, 2); }; const getViewURL = function(url){ // url -> views/url var path = "views" + url; return path; }; app.on("request", function(req, res){ console.log("Method : ", req.method); console.log("URL : ", req.url); response.writeHead(httpStatus.OK, { "Content-Type" : "text/html" }); </pre> | <pre> var url = req.url; if(url == "/") url = "/index.html"; // "/" -> "/index.html" url = getViewURL(url); fs.readFile(url, function(err, data){ // 외부파일 접근 if(err){ // url에 해당하는 외부파일이 없을 시 404 res.writeHead(httpStatus.NOT_FOUND); res.write("<h1>FILE NOT FOUND</h1>"); } else{ // url에 해당하는 외부파일이 있을 시 200 res.writeHead(httpStatus.OK, { "Content-Type" : "text/html" }); res.write(data); // 외부파일로 응답작성 } res.end(); // 응답 및 연결종료 }); app.listen(port); console.log("The Server has started and is listening on port number : %d", port); </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

✓ 클라이언트 웹 브라우저 실행화면

| | |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URL : http://host_ip:3000/ |  <p>The screenshot shows a web browser window titled 'simple file system'. The address bar displays 'localhost:3000'. The main content area shows 'Hello, Client!' in a large, bold, black font.</p> |
| URL : http://host_ip:3000/index.html |  <p>The screenshot shows a web browser window titled 'simple file system'. The address bar displays 'localhost:3000/index.html'. The main content area shows 'Hello, Client!' in a large, bold, black font.</p> |



✓ 서버 콘솔 실행화면

```
wlgns12www@nodejs-test:~/nodejs/6_fs/simple_fs$ ls
main_fs.js  node_modules  package-lock.json  package.json  views
wlgns12www@nodejs-test:~/nodejs/6_fs/simple_fs$ node main_fs.js
The Server has started and is listening on port number : 3000
Method : GET
URL : / —————> 응답 : ./views/index.html
Method : GET
URL : /index.html —————> 응답 : ./views/index.html
Method : GET
URL : /index —————> HTTP 404 Error 응답
```

참고문헌

- 조나단 웨슬러, 김성준. (2020.01.31). NODEJS로 프로그래밍 시작하기. 100-105.