**Chapter 11. JSON-RPC 서버 만들기 / 11-1 JSON-RPC 웹 서버 적용하기**

* RPC(Remote Procedure Call) : 서버에 데이터 요청, 응답 받는 과정을 라이브러리로 처리해 로컬에서 함수 호출하는 것과 같은 방식으로 처리한다.
* JSON-RPC는 json 포맷으로 데이터를 주고받는 모듈로, 자바스크립트에 적합
* 즉, 서버에 함수를 하나 만들고 클라에서 일반 함수처럼 서버에 데이터를 요청하기만 하면 req, res 과정은 라이브러리가 담당한다.
* E.g., 서버에 add 함수를 선언하고 클라(웹 문서)에서 이 함수를 호출한다. 🡪 라이브러리는 stub(스텁)을 통해 서버로 요청을 보냄 🡪 서버는 add함수를 실행하 결과를 res로 보내준다. 이 과정은 라이브러리로 이루어짐.
* 라우터 미들웨어 다음에 jayson 모듈을 사용 🡪 특정 패스 요청 시에만 JSON-RPC로 처리되도록 설정 가능.
* JSON-RPC사용을 위한 등록 함수는 주로 Handler(핸들러)로 칭한다.

[app.js]

* 핸들러 로드용 파일(handler\_loader), jayson 모듈 require를 진행한다.
* 그리고 핸들러의 경로를 설정한다. 여기서 기본적으로 ‘/api’이므로 localhost:3000/api로 요청할 수 있다.
* handler\_loader.js는 기존 샘플파일을 사용하고, handler\_info.js에 echo 파일과 메소드를 입력한다.
* 결국 웹 문서가 서버에 req를 보내면, 일단 handler\_info에 정의된 파일명과 메소드를 handler\_loader를 통해 서버에 전달한다. 그럼 서버는 /api 패스로 라우팅을 하게 되고, handlers 폴더에 정의된 다양한 함수 클래스 중 하나의 이름으로(echo 등 미리 설정한 method명) 호출해 처리 결과를 응답해 서버에 전달하고, 이를 클라에 서버가 전달하게 된다.

[echo 함수 만들어 실행하기]

handler\_info.js 내 handler\_info 객체에 echo 파일과 메소드 echo를 설정해 놓는다. 그리고 handlers 폴더에 echo.js를 작성한다.

echo 함수는 params와 callback을 파라미터로 받는데, params는 클라로부터 전달받은 배열 객체이며 callback은 클라이언트로 응답을 보내는 용도이다. callback은 다시 null과 params를 파라미터로 받게 되는데 null은 오류 처리용, params는 정상 데이터를 전달하는 용도이다.

* echo에서는 클라의 데이터를 다시 그대로 보내기 위해서 callback( ) 호출 시 클라로부터 전달받는 params를 그대로 사용한다.
* 핸들러 함수는 결과 값을 callback( ) 함수를 통해 전달하므로 return 키워드, 즉 반환 값이 없다. 주의!