nodeJS+Express 웹서버 설정

1장 npm 시작

* Npm init: 노드설정을 하기 위한 기본 파일인 package.json이 생성됨
* Key&Name Value로 되어 있다.
* https://www.inflearn.com/course/node-js-%EC%9B%B9%EA%B0%9C%EB%B0%9C/lecture/6123?tab=curriculumNpm install express –save
* 패키지에 설치된 정보를 넣어놓고 자동으로 설치될 수 있게 해주자는 것이다.
* Node.js가 의존하고 있는 패키지 정보를 다 저장한다.
* Node\_mondules : 설치 이후 필요로 하는 모듈들을 저장함.
* 모든 모듈에 접근할 이유는 없다.

2장 Express 기반 웹서버 구동

* Var express=require(‘express’)🡪express관련된 모듈들을 불러오는 것이다.
* 노드의 콜백함수들은 전부 비동기로 시행된다.
* 동기 함수들이 시행되고 나중에 비동기 함수들이 시행된다.
* Npm install nodemon –save 내 파일만, npm install nodemon -g 모든 PC에 어떤 디렉토리 시행.
* [https://www.youtube.com/watch?v=DvlyzDZDEq4&t=495s](https://www.youtube.com/watch?v=DvlyzDZDEq4&t=495s%20(nodemon)
* Nodemon 권한문제 해결 유튜브🡪nodemon은 server가 가동중에 지속적으로 코드상의 변경내용이 반영될 수 있도록 해준다.

3장 URL Routing

* http메서드 중에 모든 요청에 대해 get으로 대처함.
* 모든 요청에 대해 처리를 일일히 해주지 않으면, 처리해서 응답값을 주지 않는다.
* 모든 문서의 절대 경로 값을 지정해줘야 한다.
* \_\_dirname을 쓰면 최상위 디렉토리의 절대경로를 표현해준다.
* Nodemon은 npm I nodemon -g로 글로벌 전역변수로 지정하는 것이 좋다.

4장 static 디렉토리 설정

* 모든 url 경로 처리를 다 해줘야 한다.
* 자바스크립트나 이미지, CSS는 정적인 파일로 본다.
* Static을 express에 지정하면 따로 경로 지정해주지 않아도 요청 응답을 한다.

Request, Response 처리 기본

1. POST 요청처리

* POST 방식으로 Form 데이터를 써보라고 했을 때, 서버에서 어떻게 처리하라고 하는지
* Public에 Static이기 때문에 HTML로 만들면 바로 접근 가능하다.
* Form은 method가 있고, action은 폼의 어디로 갈 것이냐를 다루는 것이다.
* Post는 http의 메서드 중 하나이다. 보통은 GET으로 처리하는 데, GET은 URL에 데이터를 붙여서 작동하게 만드는 것이다. post방식은 URL에 담겨져 있는 것이 아니다.GET방식은 길이 제한도 있고 중요한 정보가 노출이 될 수 있다. 그래서 POST방식을 이용하는 게 정상적인 방식이다.
* 경로를 지정해주는 것을 라우팅 처리라고 한다.
* Post인 경우에는 이렇게 해서 따로 받을 수 없다. 그래서 body파워라는 것을 설치해야 한다.
* Npm install body-parser –save(Package.json에 등록을 해두는 것이다.)\
* 경로를 정확하게 해주지 않으면 404 오류가 생긴다.
* Express 서버에 bodypser를 쓴다고 알려줘야 한다.
* JSON의 형태대로 오는 데이터도 있다. 클라이언트도 브레이스해서 키 밸류로 오는 것이 있다.
* 클라이언트에서 전송된 폼이 서버에서 오게 만들 수 있다. 매우 고급 스킬이다.
* 클라이언트에서 보낸 정보를 가지고 HTML형태로 클라에서 응답을 준다.

1. View engine을 활용한 응답처리

* Html의 데이터들이 바뀌어서 적절한 데이터를 섞어서 보내줄 수 있다.
* ejs라는 것이 있는 데, express와 연동해서 쓰는 여러 템플릿 중 하나이다.
* 템플릿마다 치환하기 위한 데이터를 <%=%>를 사용한다.
* 브라우저의 개발자 도구와 비주얼 스튜디오 코드를 활용해서 단위테스트를 할 필요 없이 각각 중요하거나 의심가는 지점에서 확인할 수 있다.
* 정말로 어려운 기술이기 때문에 현직자도 동영상을 다시 찍을 만큼 신중을 기한다. 내가 못하는 건 당연한거다.
* <https://expressjs.com/en/resources/template-engines.html>
* Express view engine의 템플릿을 사용하면 좋은 것들을 많이 만들 수 있다.
* Res.render를 써서 데이터랑 값이 합쳐진 상태로 클라이언트에게 내려주겠다. 이렇게 생각해야 한다.
* 나중에 다른 곳에서 작업하더라도 Package.json의 정보를 읽어서 다 시행시켜 줄 수 있는 것이다.
* Node\_modules는 굉장히 무거운 파일이라 깃허브에 올리지 않는다. 그래서 패키지.json을 올린다.
* Npm i를 치면 종속파일을 다 봐서 깔아준다. 그래서 –save를 해줘야 한다.

1. JSON을 활용한 Ajax처리(비동기 자바 처리)

* Ajax처리를 Node WebServer와 연동시킬 수 있다.
* Asynchronous JavaScript and XML(비동기 적이고 확장적인 풀스택 일처리)
* 새로고침 없이 리퀘스트를 서버에서 유효한 지 검사하고 응답값을 클라이언트에게 준다.
* url에 엠퍼센드 쓰고 뭐 이런 식으로 해도 괜찮다.
* 크롬 개발자도구에서 Console창으로 서버에서 데이터를 보낸 것을 확인할 수 있다.
* Form을 직렬화(b=&이런 형식으로 받을 수 도 있다.)