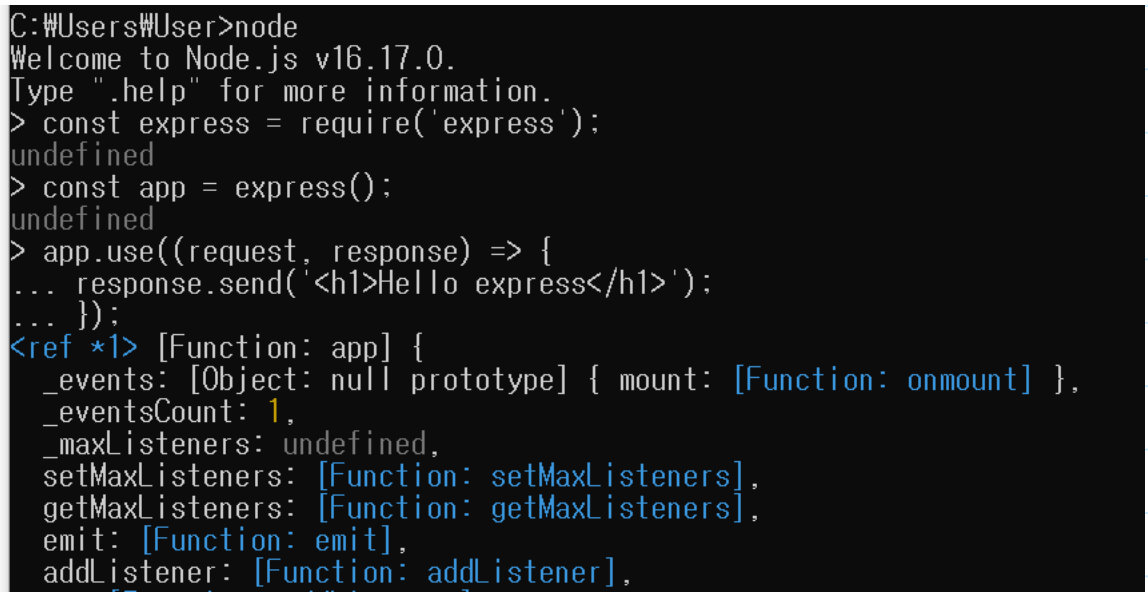
Express 모듈을 사용한 서버 생성과 실행

2022년 10월 8일 토요일 오후 6:08

- Express 모듈은 외부 모듈
- 명령어 : npm install express@4
- 기본 메소드

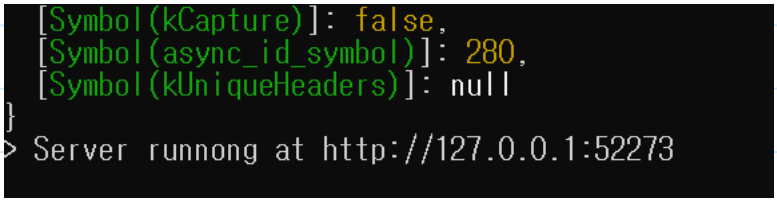
메소드	설명
Express()	서버 애플리케이션 객체를 생성
App.use()	요청이 왔을 때 실행할 함수를 지정
App.listen()	서버를 실행

** 고정적으로 사용하는 코드로 외워야함

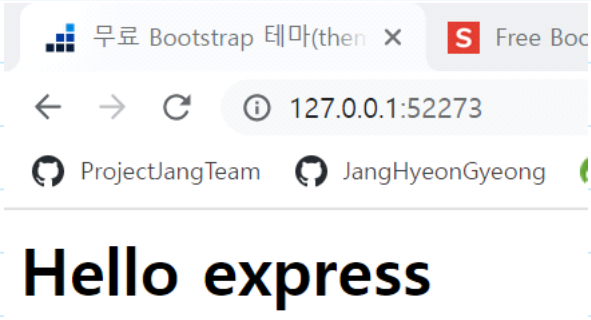


화면 캡처: 2022-10-11 오후 7:06

화면 캡처: 2022-10-11 오후 7:06



화면 캡처: 2022-10-11 오후 7:06



화면 캡처: 2022-10-11 오후 7:06

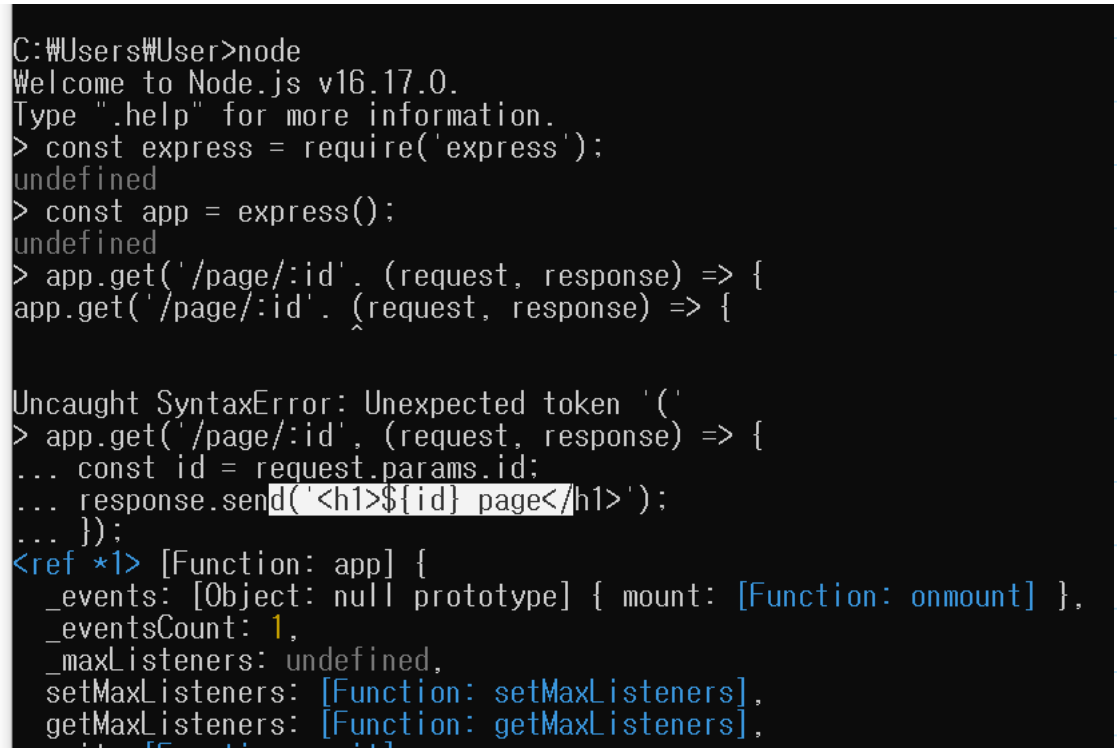
페이지 라우팅

2022년 10월 11일 화요일 오후 7:06

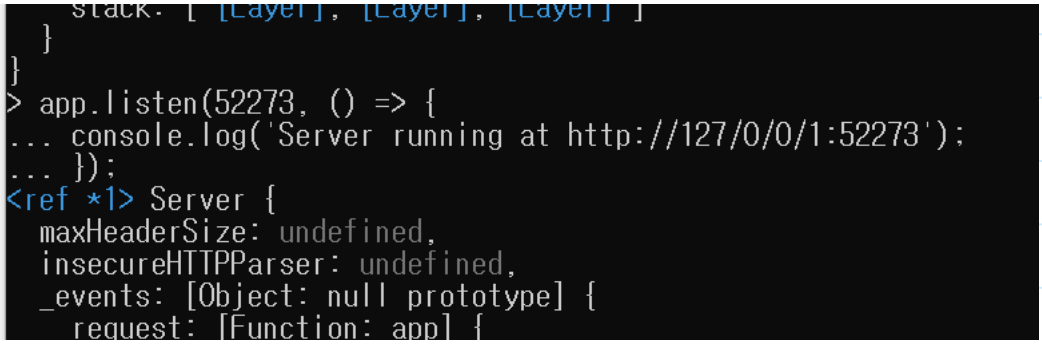
- 클라이언트 요청에 적절한 페이지를 제공하는 기술
- Express 모듈은 app객체의 다음 메소드를 활용해서 페이지 라우팅 한다.
- 페이지 라우팅 메소드

메소드	설명
Get(path, callback)	GET 요청이 발생 했을 때 이벤트 리스너 지정
Post (path, callback)	Post 요청이 발생했을 때 이벤트 리스너 지정
put(path, callback)	Put 요청이 발생 했을 때 이벤트 리스너 지정
delete(path, callback)	Delete 요청이 발생 했을 때 이벤트 리스너 지정
all(path, callback)	모든 요청이 발생했을 때 이벤트 리스너 지정

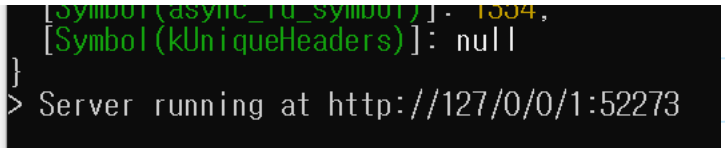
- 페이지 라우팅 할 때 토큰(token)을 활용 할 수 있다.
- 토큰 은 ':토큰_이름' 형태로 설정한다
- 토큰은 다른 문자열로 변환해서 입력할 수 있다.
- Request 객체의 params 속성으로 전달 된다.



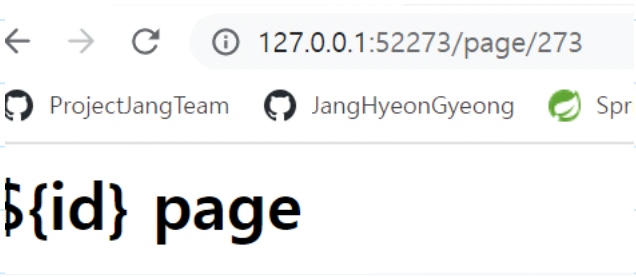
화면 캡처: 2022-10-11 오후 7:16



화면 캡처: 2022-10-11 오후 7:16



화면 캡처: 2022-10-11 오후 7:16



화면 캡처: 2022-10-11 오후 7:16

Response 객체

2022년 10월 11일 화요일 오후 7:17

- Response.send() 메소드를 사용하면 문자열을 클라이언트에게 제공 할 수 있다.

1) Response 객체

- 객체의 기본 메소드를 활용해 상태 코드, 헤더, 본문을 설정
- 아스키문자만 활용 가능
- 메소드

메소드	설명
Send()	데이터 본문을 제공
Status()	상태 코드를 제공
Set()	헤더를 설정

  - send() 메소드 사용해 사용자에게 데이터 본문(body)를 제공
  - Send()메소드를

- Send() 메소드의 매개변수

자료형	설명
문자열	Html을 제공
객체, 배열	Json을 제공

2) Content - type

- 어떤 타입의 파일인지 구분하지 못하기때문에 속성을 헤더에 적어 제공해야 한다
- 헤더에서 확인하고 제공된 데이터의 형태를 확인한다
- 이 때 MIME는 문자열 제공
- MIME형식

MIME 형식	설명
Text/plain	기본적인 텍스트 의미
Text/html	Html데이터 의미
Image/png	Png데이터 의미
Audio/mpe	Mp3음악 파일
Video.mpeg	Mpeg 비디오 파일
Application/json	Json데이터 의미
Multipart/form-data	입력 양식 데이터 의미

- MIME 형식 지정 메소드

메소드	설명
Type()	Content-type을 MIME

**사진, 음악, js파일 넣을 때

- Set() 메소드 구현

```
Response.set('Content-Type', 'image/png');
Response.send(data);
```

3) HTTP 상태 코드

- 인터넷의 '404 Not Found' 문자열은 http상태 코드라고 한다
- 상태코드는 서버가 클라이언트에게 해당경로는 이런 상태라고 알림
- HTTP 상태코드

HTTP 상태코드	설명	예
1XX	처리 중	100 Continue
2XX	성공	200 ok
3XX	리다이렉트	300 Multiple Choices
4XX	클라이언트 오류	400 Bad Request
5XX	서버 오류	500 internal Server Error

- 상태코드 지정할 때 status()메서드 사용

메소드	설명
Status()	상태 코드를 지정

```
App.get('*', (request, response) => {
Response.status(404);
Response.send('해당 경로에는 아무것도 없음');
});

App.listen(52273, () => {
Console.log('Server running at _____');
});
```

4) 리다이렉트

- 웹 브라우저가 리다이렉트를 확인하려면 화면을 출력하지 않고 응답 헤더의 Location 속성을 확인해서 해당위치로 이동
- 특정 경로로 웹 브라우저를 인도 할 때 사용하는 것이 리다이렉트
- Redirect() 메소드

메소드	설명
Redirect()	리다이렉트 한다

```
App.get('*', (request, response) => {
Response.redirect('_____');
});

App.listen(52273, () => {
Console.log('Server running at _____');
});
```

- Set() 메소드 구현

```
Response.status(302);
Response.set('Location', '주소');
Response.send();
```

Request 객체

2022년 10월 12일 수요일 오후 7:38

1. 요청 매개 변수

- 주소분석
  :: https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EB%8B%A4%EC%9D%B4%EC%8A%A8&oquery=%EB%8B%A4%EC%9D%B4%EC%8A%A8+%EB%9D%BC%EC%9D%B4%EB%B8%8C%EC%BB%A4%EB%A8%B8%EC%8A%A4&tqi=h0jnmwp0Jy0sstYMwrosssssXC-337301

분류	값	설명
프로토콜	https	통신에 사용되는 규칙
호스트	search.naver.com	애플리케이션 서버(또는 분산 장치 등)의 위치 의미
URL	search.naver	애플리케이션 서버 내부에서 라우트 위치 나타낸다
요청 매개 변수	where=nexearch&query=%EB%8B%A4%EC%9D%B4%EC%8A%A8&oquery=%EB%8B%A4%EC%9D%B4%EC%8A%A8+%EB%9D%BC%EC%9D%B4%EB%B8%8C%EC%BB%A4%EB%A8%B8%EC%8A%A4&tqi=h0jnmwp0Jy0sstYMwrosssssXC-337301	추가적인 정보 의미

- 여기서 요청 매개 변수를 추출할 때는 query 객체 사용
- 요청 매개 변수는 URL 뒤에 ? 기호를 삽입하고 , "키=값" 형태로 &로 구분해 입력한다

미들웨어

2022년 10월 12일 수요일 오후 7:42

- 미들웨어 설정 메소드

메소드	설명
Use()	미들웨어 설정한다

1. 정적 파일 제공

- 웹 페이지를 만들 때 가장 많이 사용하는 기능으로 정적 파일 제공이 있다.
- 웹 페이지에서 변경되지 않은 요소(이미지, 음악, 자바스크립트 파일, 스타일시트)를 쉽게 제공해주는 기능
- 자주 사용하는 기능으로 express 미들웨어가 자체적으로 미들웨어를 제공한다.
  1) 먼저 정적 파일 제공할 폴더를 만든다
  2) 퍼블릭 폴더에 이미지, 음악 등 원하는 파일 넣는다
  3) 이어서 코드를 작성할 때 app.use(express.static('public')) 코드가 static 미들웨어를 사용하는 부분이다.
  4) 매개 변수로 넣은 문자열에 폴더 이름을 입력한다

2. Body-parser 미들웨어

- 요청 본문을 분석해주는 미들웨어
  1) 클라이언트 서버에서 서버로 데이터 전송
     - Url 입력도 일종의 데이터 전송이라 할 수 있지만, 주소에 정보가 남는다는 단점이 있다. 사용자 추적으로 인한 문제 발생
  2) 요청 본문
     - 요청 본문의 종류(Encoding-type)를 함께 전달 해야 한다

MIME 형식	설명
Application/x-www0form0urlencoded	웹 브라우저에서 입력 양식을 post, put, delete 방식 등으로 전달 할 때 사용하는 기본적인 요청 형식
Application/json	Json 데이터로 요청하는 방식
Multipart/form-data	대용량 파일을 전송할 때 사용하는 요청 방식

  3) Body-parser 미들웨어
     - 위의 복잡한 부분을 쉽게 해준다
     - 다만 대용량 파일을 전송하는 multipart 형태는 분석하지 못한다. 별도 모듈 사용
     - 가장 기본적인 입력 양식을 분석한다

3. Morgan 미들웨어

- 설치 : npm install morgan
- 로그는 관련된 정보를 가진 글자 의미하며 로그 출력 미들웨어는 웹 요청과 관련된 내용을 출력하는 미들웨어이다.
- 모든 웹 서버에서 기본적이면서도 중요한 미들웨어
- 웹 브라우저로 한 번만 접속 했는데 요청이 여러 번 전송되는 모습을 볼 수 있다. 이것은 해당 웹 사이트의 추가 정보(파비콘)를 가져오려고 추가적으로 요청하는 것이다.
- 로그는 보안의 위협을 검출할 수 있는 가장 기본적인 방법
- 실제 웹 서버 운용을 할 때 로그 출력을 실시간으로 감시해서 특정한 오류가 여러 번 발생 상황이 생길 때 경보 하는 기능에 활용된다.

4. 속성 정리

- 서버로 데이터를 전송하는 세가지 방법
    - Params 객체 : URL의 토큰을 나타낸다. 보기가 편함
    - Query 객체 : URL의 요청 매개 변수를 나타낸다. 토큰 보다 많은 데이터를 전달할 수 있고, 주소로 어떤 데이터가 오고 가는지 확인 할 수 있다.
    - Body 객체: 대용량 문자열 등을 전송할 때 사용. 하지만 주소에서 데이터를 기록하지 못하므로 새로 고침이나 즐겨찾기 기능 등을 활용 할 수 없다.
- 위 세가지 모두 서버 개발에 중요한 역할 담당 기억해야함!