Set minute 호출해요 expires에 우리가 get minute 하면 현재 분이 나와요 34분, expires가 현재 34분을 가리키나 봐요. Get minutes하면 34분이 튀어나와요. 그러면 + 5분 뒤를 expire라는 즉 시간을 여기에 5분을 더한 시간을 expire 인스턴스에 셋팅을 하겠다는거죠. expire되는 시간을 여기서 설정하는거에요. Setminute 하고나서 실행하고나서 expires 인스턴스를 보면 시간이 5분 더해져있어요. 이 시간이 되면 쿠키는 유효하지 않다고 판단하는 거에요. 우리 이제 응답을 보내줄 차례에요. 301하고 302 뭐였죠? 한번 다시 보세요. 302는 리다이렉션 시키는거에요. 301은 영구이동. 302 리다이렉션 시켜요. Location / 로 이동시켜요. 저희 요청은 현재 이 주소에서 요청이 왔어요. 그러면 다시 로케이션 슬래쉬로 이동시키고, 자신의 홈 디렉토리로 이동시키겠다는거죠. 자 로케이션 슬래쉬 즉 리다이렉션 할 때는 로케이션 필드를 채워주시면 되요. 그리고 셋 쿠키라는 곳에 내임과 쿠키 네임, 익스파이어 만료되는 날짜, http only 이건 기억 나시죠 http에서만 사용할 수 있다. 쿠키 옵션에 어제 잠깐 우리 http only 이것은 자바스크립트에서 쿠키에 접근할 수 없어요. 그래서 우리가 쿠키를 조작해서 이 기간을 늘려서 예를 들어서 이벤트의 시간을 늘린다던가 진짜 이벤트요 할이하는 이벤트 그런걸 늘릴 수 있어요. 자바스크립트에서 쿠키에 접근할 수 없도록 하는거구요, 저희가 넣어놓은 옵션들 패스에 넣어놨죠. 우리 url 을 이걸 안적어도 기본값이 슬래쉬가 들어가요. 이 경우에는 모든 url에서 슬래시 밑에 있는 모든 url에서 쿠키를 전송할 수 있어요. 그렇게 하고 엔드 보내면 끝나죠. Redirection 슬래쉬로 이동시키고, 저희 페이지 보면 네트워크를 볼게요. 저희가 이거 요청했구요 302로 받았네요 스테이터스 코드 그런데 낫 파운드네요. 아직 겟 요청에 대해서 아직 받아가지를 못 했어요. 그래서 여기 걸어놨거든요, 그리고 여기 한번 더 돌아요. 현재 네임이 jy죠. Jy가 있을 때 라이트헤드에 200, 오케이를 넣어주구요 쿠키점 네임 이 네임이 뭔지 부이세요 jy에요. 우리는 아까 쿠키를 돌려 보낼 때 여기서 네임에 인코드 우ㅠ알아이 네임을 찾았죠. 그게 제이와이네요 그게 리다이렉션을 시켰구요 로케이션 슬래쉬를 리다이렉션 하라고 시켰죠. 이 요청을 다시 보내는거에요. 슬래쉬를 다시 보내는거에요. 여기서 우리가 리다이렉션 했던 여기서 요청했구요 요청을 받아서 슬래쉬를 이동 했구요, 이동을 한 뒤에 이도을 여기로 한 뒤에 다시 슬래시부터 시작하는거에요. 리다이렉션을. 그래서 스타트 윗 로그인 스타트로 시작해서 여기로 들어왔구요. 그런데 쿠키로 네임이 나와있ㅇ요 . 팔스를 하니까 쿠키에 네임이 남아있다. 왜냐 여기서 쿠키의 네임을 결정했기 때문. 여기서 우리가 쿠키를 보내줬어요. 네임 어딨죠 여깄네요, 이 때 쿠키 셋을 해줬어요. 이렇게, 이렇게 쿠키를 세팅해줬죠. 리퀘스팅을 다시 보낼 때 늘 모든 쿠키를 다시 보내죠. 그래서 이 때 네임이 들어가요. 쿠키중에 네임 쿠키 쿠키인데 네임은 jy라는 쿠키가 다시 들어가요 여기서. 그래서 맨 위에서 쿠키 파싱을 하죠. 키 밸류를 형태로 패스에 네임을 찍으면 그 jy라고 찍혀요. 값이 true 니까 이 안으로 들어갔어요. 마지막으로 cookies.nae이라고 찍어서 라임색 1번, 빨간색 둘다 헤더가 동시에 가는거에요. 그리고 그린색 2번 이런 형태로 작동하고나서 레스점 엔드를 할 때 로컬 호스트에서 다 받아갈 것 같아요. 그래서 여기서 한번 더 실행시키고 나면 로컬호스트에서 200 ok를 받았구요, 그리고 요청한 내용을 보면 여기서 리다이렉션을 받았네요. 리다이렉션을 했는데 not found 나왔고, 여기서 우리가 셋 쿠키를 다시 해서 보냈어요. 리스폰스네요. 리다이렉션을 동시에 받았어요 로케이션을. 로케이션 이전했을 때 로케이션 남겨뒀어요. 자 쿠키 우리가 리다이렉션을 하는 경우 직접 한 경우는 처음이죠. 리다이렉션 여기, 로케이션 어디갔죠, 로케이션 리스폰스 로케이션 여기 200 오케이의 예죠. 200 오케이 따로 있나요 없죠. 우리 리스폰스 헤더에 로케이션이 찍혀있어요 이건 리다이렉션 한다는거죠. 셋 쿠키 익스파이어 프라이데이 옥토버 퉤티퉤티 033806 gtm 까지 그리고 httponly 자바스크립 느 패스는 여기서 요청이 가능ㅎ k겠다. 슬래시에서 요청이 가능하겠다 라는 거에요. 패스는 전송된 유알엘을 측정하는거에요. 자 그러면 이렇게 해서 쿠키 정리했구요, 그리고나서 맥스 에이지도 넣을 수 있구요. 이거는 초를 입력하면 되요. 데이터객체 만들어서 시간 설정 하구요 setMinute 혹은 expires.setDate (month, seconds, hours, 등등등) 이렇게 get set 해서 사용하면 되요. 이렇게 사용하시구, 쿠키도 깔끔하게 정리가 됐어요. 그리고 secure 옵션도 https 일 때만 쿠키를 사용하는거에요. 우리 일반적으로 전부 다 패키지를 뜯어볼 수 있어요. https 를 다같이 적는게 좋아요 쿠키 사용할 때. 관련정보가 같이 가기 때문에 그래서 이렇게 쿠키 서버 2 실행하고 우리 확인했쬬 넘어온거 확인합시다. 어플 탭으로 가요. 어플 탭이 여깄네요. 가서 쿠키, 우리가 요청한 주소 밑에 쿠키가 톡 쌓여있어요. 쌓인 쿠키가 네임이 어디갔죠 네임 새로고침 네임이 없네. 자 응답이 한번 전체 끝내볼게요. 전체를 한번 끝내고. 쿠키 자 이거, 나중에 쿠키 받은거 있는지 다시 한번 확인해봅시다 실행해서. Abc 로그인 네임이 들어왔네요. 지금은 네임이 들어왔죠. 네임이 에이비시라고 들어왔어요. 들어왔고, 자 이렇게 이렇게 그러면 네임이 들어왔어요. 여기 있는게 쿠키가 쿠키 값을 읽어서 우리 html에 표시하도록 만들어서 보내줬죠. Html 파일 만들어줬어요. 여기서 쿠키를 읽어서 수정해준거에요. 우리가 새로운 리소스를 찍어준겁니다. 어떻게요? 이렇게요 페이 로드 에 프리뷰 리스폰스 이렇게요. 리다이렉션을 시키구요, 이 페이지는 리다이렉션을 시키고, 이 페이지는 애초에 다르죠. 쿠키2점 html과 달라요 로그인 버ㅡㄴ도 없구요 이건 새로운 페이지에요 그래서 소스 가보면 인덱스라고 전혀 모르는 파일로 있을 거에요. 쿠키 마무리 하구요, 세션 세션도 중요해요. 쿠키는 정보노출이 쉬워요 수정하기도 쉽습니다. 그래서 중요한 정보ㅡㄴ 서버에서 관리하는게 기본이에요. 사요자 정보는 서버에 저장하죠. 클라이언트에는 세션 키만 저장하고 세션 객체 생숭 후 uniquelent 를 만들어 속성 명으로 사용. uniqueInt 를 클라이언트에 보낸다. 같이 사용해볼게요. 쿠키 저장 끝내고, 폴더 하나 더 만들게요. 쿠키 말고, 폴더 세션에 session.js 로

그걸 ㅗ최종적으로 저희 acc에 계속 저장 했었죠. 그리고 최종적으로 리턴이 되구요, 그 다음에 이거는 쉽구요 넘어가서 우리 아까는 쿠키를 만들었어요 세션은 빈 오브젝트로 만들거에요 이렇게 진행하면 저희 세션이 정말 빈 오브젝트로 만들어졌죠. 프로토타입 오브젝트로 이 세션을 찍어 냈어요. 그리고 크리에이트 서버 하고 여기서 멈춰볼게요. 우리 콜백 함수 실행시켜주는 첫 부분에서 멈출거구요, 리퀘스트를 보내야죠. 브라우저에서 8085 8085로 리퀘스트 엔터 하나라도 files면 조건을 확인하지 않아요. 그리고 const data 데이터에 html 파일 셋 쿠키라는 필드를 설정하나요 아니요 설정하지 않아요. Html 파일이 넘어가구요. 쿠키 말고 자 그리고 여기 한번 더 요청이 자동으로 들어오죠. 자 이것을 다시 다시 다시 보내주면 정상적으로 다 받아갔어요. 이 상태에서 우리가 이름을 입력해서 로그인합시다. Abc 로그인하고 로그인 클릭 자 그러면 parsing 하고나면 fix에 자, 기존에 있떤 값이 있어요. 우리 요청 url 슬래쉬 로그인 스타트 하녜요. 슬래시 로그인. 자 로그인으로 스타트 하니까 if 문 내부로 잡혀 들어가요. 이제 들어가서 그리고 url parse는 잉해하시겠죠 우리가 쿼리 스트링 뽑아져 나와요 뽑아져 나왔고 쿼리를 보면 네임 이퀄 에이비시라고 인풋 태그에 있떤 아이디 네임이였죠. 네임은 우리가 입력해준 값. 이걸 parse 하면 키 오브젝트로 나오죠. 그리고 네임이 정상적으로 abc 라는 값이 들어왔네요. 그리고 날짜 객체를 만들어요. 날짜 객체 만드는 것 까지 같아요. Expires 보면 날짜 시간 둘 다 들어와있을거에요. 현재 시간에 5분을 더해서 expires 인스턴스에 시간을 설정하는 셋타임 셋미닛 분만 바꿔졌어요. 현재보다 5분 커졌죠. 그리고, 유니크 int 데이트나우를 넣어줘요. 데이트점나우를 하게되면 현재 시간이 유닉스를 기준으로 초로 들어간다. 유닉스가 만들어진 데가 언제에요, 예전에 에니악이라는 것이 만들어졌죠. 컴퓨터라는 유닉스 운영체제가 나온 시간 (생성된)시간이 있어요. 유닉스 다음에 리눅스가 빠져나왔고, 리눅스를 모티브로 윈도우가 만들어졌어요. 그리고 유닉스 모티브로 hpux, macos 혹은 ios 유닉스는 1970년 1월 1일에 태어났다. 그때를 기준으로 초를 잰다. 그러면 초가 얼마나 지났는지 계산함으로써 사용할 수 있다. 유니크인트에 넣어주구요, 우리 세션을 어떻게 만들었냐 세션에 유니크 아이엔티 이 키값으로 네임과 익스파이어스가 들어갈 것이에요. 만료된 시간에 네임도 들어간다. 헤드를 만들어주는데, 302 뭐에요. 301은 뭐에요 3으로 시작할 때 1번은 가장 타이트 하고 가장 무거운 것이다. 이게 숫자가 커질수록 조금씩 가벼워진다. 301이 경로를 완전 바꾸는 리다이렉트 302는 임시로 바꾸는 리다이렉트 우리가 옮겨가야할 주소를 나타내주네요 슬래시로 나타내줘요. 셋 쿠키 같이 들어갔어요. 셋 쿠키에서 그다음 이전 우리 소스를 가져옵시다. 쿠키스투점 제이에스 여기도 셋 쿠키라는 필드를 세웠어요. 프로토콜 메세지에, 셋 쿠키를 세워서, 뭐가 달라요,익스파이어스 똑가팅 들어가고 httponly 도 있는데, session 이 다르다. 세션이라는 키 값을 뭘 넣어줬냐, uniqueInt 이렇게 해서 우리 여기서 메시지 받을 준비 해야죠. 즉 여기가 뭐가 됐죠, session=${}가 name={} 이제 저희 네트워크에서 받아올 준비를 해야죠. 이거 이제 지우구요, 엔드 찍는 순간 날아가겠죠. 잘 날아갔어요. 밑에 패널을 보면 셋 쿠키, 리퀘스트 리스폰스를 아직 못 받았네요. 리스폰스 조금만 더 진행해볼게요. 여기서 리다이렉트 메시지를 여기서 셋 쿠키를 받았네요. 이 쿠키를 받고나서 한번 더 요청해요. 리다이렉션 하라고 했잖아요 슬래시로, 슬래시로 가던데 여깄어요. 그리고 여기서 다시 요청이 갔기 때문에 … 하나씩 쭉쭉 내려가면 이제 쿠키에 달라요. 쿠키에 세션 아이디가 들어왔어요. 세션 아이디를 갖고 현재 로그인으로 시작하는 건 아니고, 쿠키에 세션이 존재하죠. 그러면 그 다음 조건 확인. 세션에 쿠키점 세션 값. 그 값에 익스파이어스를 찾아본다구요. 값을 찍어 볼게요. 그러면 현재 만료됐네요. 설명 하다보니까 만료가 됐어요. 만료돼지 않았다면 만료가 되지 않았다면 이것을 계싼하고 뉴데이트를 만들게 되겠죠. 들어가봅시다. 현재는 엘스 외로 들어갔어요. 들어가서 쿠키 투점 html 찍어서 전달전달 그리고 끝, 끝나면 최종적으로 localhost 받아오구요, 페이지가 원래대로 받았어요. 현재는 세션에 없죠. 그래서 리퀘스트 url에 슬래시 브라우저에서 자동으로 패비콘.ico 이것은 요청하라는 메시지 안넣었는데 브라우저가 자동으로 요청함. 그래서 한번 더 사이클이 도는 것이다. 이것은 빨리 끝내줘요. 끝내주고, 자 이제 다시 슬래시를 요청했네요 슬래시 요청하고 끝내주고, 그러면 현재 금방에 우리가 만료 시간이 지나서 로그인이 아직 안됬죠. 땡땡땡님 안녕하세요 라고 나온다. 거기까지는 빨리 가보겠다. 에이비시로 다시 로그인할게요. abc클릭. 그리고 여기 첫번째는 빨리 건너뛰고, 두번째는 세션 id가 정상적으로 넘어왔네요. 그래서 요 내부로 들어온거에요. 여기 있고 마지막으로 이 값이 뉴데이트 현재 데이트 값보다 커야해요. 그렇죠, 우리의 값이 요 값보다 커요. 그러니까 트루가 되서 들어온 것이다. 이렇게 들어오고, 들어와서 이제 뭘 해줘야 하냐면 헤드를 만들어서 정상적으로 들어왔다는걸 넘겨주면 된다. 헤드를 만들어주고 이것을 찍어보면 abvc님 안녕하세요 라고 뜬다. 세션이라는 객체는 오브젝트 인스턴스는 첫번째로 키가 쿠키스 점 세션, 즉 이 값으로 키가 들어가구요, 자, 그러면 세션에 형태는 이러하다. 세션이 몇 개나 있나요. 두번 접속해서 이전에 요청했던 세션과 expire에서 만료가 된 28분거, 두번째 세션이 남아있네요. 두번째 남아있는 세션은 expire 느 이 세션을 이용해서 들어온 것이다. 이렇게 해서 정상적으로 abc님 안녕하세요 라는 페이지로 연결이 됐따. 한번 더 온 것은 favicon parsing 하면 패비콘 요청이 있네요 .이거 끝내고 정상적으로 로그인이 된 거에요. 유니크한 키가 된거에요. 속성 값으로 정보를 저장한ㄷ. 첫번째 익스파이어, 두번째 네임. 이 세션이 성공적으로 가지도 있다면 요 저ㅗ건에서 기존에 사용하던 기존에 진행하고 있단 아이템 사용 항목이나, 읽고있던 데이텅나 바로 연결할 수 있따. 세션 밑에 하나 더 저장하면 된다. 당신이 고른 장바구니 항목등. 이렇게 해서 세션 확인했고, 어플리케이션 들어와서 저거 확ㅇ니해야죠. Application 클릭해서 들어가봅시다. Session 이라는 쿠키가 쿠키 이름이 세션이고 밸류가 나와있네요. 세션이랑 쿠키 않에 우리 값이 들어가있어요. 이 값은 우리 유니크한 세션값. 그래서 계속해서 이름을 session 이라고 하기 때문에 새 쿠키 만들면 그 이전 쿠키가 어떻게 되냐 덮어씌워지게 된다. 그리고 session 들어갔어요. 이제 https 로 만들어주는게 기본이죠. https 로 ㅁ나들어 줄 것이다. 이제는 https 서버를 만들어갑시다. Rsa ca 2018 발급일과 만료일이 있따 보통 1년 단위고, 지문은 해쉬 함수다. 그냥 지문 이 자체로서 이 지문이다. 그렇게 생각해주시면 되구요, 디지설트 그룹 글로벌 루트 ca에서 rsa ca 래요. 이게 root ca 에요. Root ca는 최상위 인증기간이에요. Ca는 certificate authority이거 맞나요. Root ca 가 하나 있구요, 일반적인 ca가 하나 존재합니다. 이랍넞ㄱ인 ca는 여러 개 있어요. 계층으로 보면 rootca 하나, digicert global, ca1에 해당하는게 geoTrust https 통신을 하려면 이런 인증서를 갖고 있어야 한다 이말이죠. 자, 사실 우리 일반 사용자는 여깄어요. 일반 사용자, 즉 네이버나 등등 ca의 인증을 받고 이 ca는 루트 ca에 인증 받아요 그래서 이렇게 계층 구조를 띄고 있는거에요. 제가 ca1 geotrust geo trust에 여기서 인능서를 발급 받았다면 geotrust내에 제 인증서가 같이 드렁가 있어요. 자 그러면 정부기관 한번 가볼게요. 정부24 여기 콕 하고 들어가면 cert에 인증서 경로, key에 비밀키 경로, ca에 상위 인증서 경로 하나씩 넣어주면 된다. 그러면 https 프로토콜로 통신을 하고 listen 443을 해주는게 기본적인 의례다. 그러면 직접 인증서를 발급 받아야 하는데, 인증서 let’s encrypt 인증서 발급

http 1.1과 2의 차이” 하나는 베이스라인 하나는 멀티플렉싱멀티플렉싱이니까 최소한 여러 개 연. 처음에 요청할 때 이렇게 진행이 된다. 첫번째로 서로 연결을 수립하구요, 여기까지는tcp가 해주구요, 일단 수립이 됐어요. 그러면 왼쪽이 클라이언트에요. getd로 뭘 요청해요 index.html 을 요청한다. 서버는 응답하겠죠. 그 다음에indes.html 을 실행시켜주다보니까 hrf나 src나 다른 소스가 필요해요. 그래서 요청을 했다. Get style.css가 필요하다. 다시 응답해줘요. 그리고 get/scripts.js 필요하다 응답해주면 이제 다시 전체가모였죠 페이지 렌더링이 보여지게 된다. 그리고 이걸 받아다 쓰면 연결종료. 연결 종료도 최종적으로 tcp가 알아서 해준다. 이런 단계를 처음에 get 요청하는것 까지는 같아요. Index.html 필요한 것들 xic css랑 js 한번에 요청했어요. 두 단계가 하나로 뭉치면서 두단계가 줄어들었죠. 여기서부터 페이지 렌더링이 시작되요. 그리고나서 연결이 유지시켜놔요. 연결 종료하구요, tcp ㅈㅁㄴㅇ라ㅓos가 일을 해야하고 ㅇtcp 연결 수립작업 ㄷ사ㅣ 해야하는데 tcp 연결 몇번 주고 받는다. 연결 수립할 때. 그것조차 타임로스라는거죠. 그래ㅓㅅ http2 버전에서는 속도가 개선이 됐어요. 그러면 브라우저도 http2 프로토콜을 지원해야하구요, 지금은 지원하고 있습니ㅏㄷ. http2 서벌 만드셔도 괜찮아요. http2 서버를 보면, 소스는 그렇게 다르지 않아요 여기랑. httpsServer구요, 그리고 클러스터가 중요하다. 웹서버 속도. 이런건 영어로 검색하면 더 잘나와요. Web server rankfor speed framework 순위네요 framework 순위가 자 여긴 진짜 서버 순위네요. Node server rank for speed. Web server language performance

스레드는 프로세스 안에 있는 거에요. 원래는 스레드라는 개념이 없다가 생겨나서 그래요. 프로세스 라는 개념이 os마다 프로세스 정보가 있고, Port 를 배치한 테이블이 또 있어요. 운영체제 안에 포트를 배치한 테이블. 이 테이블을 갖고 pid가 8000번인 애는 port 8080을 가져간다고 요청했었어 라고 pid가 작은 것은 5000번 큰 것은 8080번 이렇게 가져간다고 등록을 해놓는다. 우리가 프로세스를 여러 개 가져서 노드에 서버를 프로세스를 여러 개 띄워야 하고 포트 번호를 각각 띄워야 해요. 네트워크 아키텍쳐에서 직접 밸런싱을 해줘야 합니다. 한번은 8080으로 한번은 8081로 프로세스를 스레드로 하게 되면 같은 포트 하나로 … 스레드를 지금 16개 만들 것이다. 포트 8080, 8081인가 보네요. 8081로 요 전체 스레드를 띄워서 처리하기 때문에 훨씬 더 빨라질 것이다. 현재는 master에요. 클러스터.마스터. 클러스터 라는 모듈은 이벤트 에미터에요. 쭉 낼려서 아 클러스터요. 이벤트 에미터 클래스네요. 그 속성의 값으로 요런거 들고 있네요. 뭐 등등 핵심은 워커 중에 이벤트 에미터라는거죠. 이벤트를 발생시켜 주고 이즈 마스터라는 변수에 이즈 마스터 트루가 들어가 있죠. 왜요 메인 프로세스가 첫번째 실행되는 메인 프로세스가 진행되어 있다. 콘솔 로그. 프로세스 점 파이아디 찍으면 프로세스 아이디, 28816이에요. 작업관리자 찍으면 세부 정보, PID 찍으면 프로세스 id 28816 여기네요. 여기 켜져있네 28816으로, 우리가 디버그를 함으로 켰기 때문에 28816으로 잘 실행되고 있네요. 그리고, 이렇게 로그를 딱 찍어보면 28816 나오겠죠. 그러고나서 몇번부터 반복해요, 16번 반복할 거에요. Let I 는 0부터, +1 꺼내서, i가 16보다 작을 때 까지 0부터 하면 총 16번이죠. 축약형 이제 잘 하시죠, I + 가 이게, 이렇게 옆으로 넘어가는 것과 같아요. 이거랑 같아요. 클러스터 포크. 클러스터 포커 한번 시작해볼게요. 우리가 반복되는 개수갯 알고 싶어서 i라는 변수를 찍어놓을게요. I는 0이에요. 클러스터.fork 이거를 실행하면 새로 스레드가 만들어지기 때문에 실행하진 않을게요. 클러스터 안에 새로운 스레드가 있다. 파라미터한테 on으로 메시지를 보내고, 자식한테 보내고, 스레드에서 했던 내용과 비슷해요. 자 그러면 클러스터 내부에 포크 한번 더 했구요, 워커는, 워커스에 첫번째 워커, 두번째 워커. 포트 하면 세번째 워커 생겨날거에요. 포크 실행하고 클러스터 보면 세번째꺼 들어갔죠. 이 워커가 인스턴스 안에 쭉쭉 넣어줍니다. 그래서 바로 다음단계 안에 건너뛸게요. 워커를 보면 클러스터에 클러스터 점 워커스네요. 워커스 보게 되면 16번 반복하면서, 워커를 총 16개 만들어냈어요. 16개 만들어냈고 하나하나 실행될 때 마다