
HTML에서 웹앱까지

12주차_02

한 동 대 학 교
김군오 교수

학습 목표: 동시통역 채팅 프로그램 만들기

학습내용:

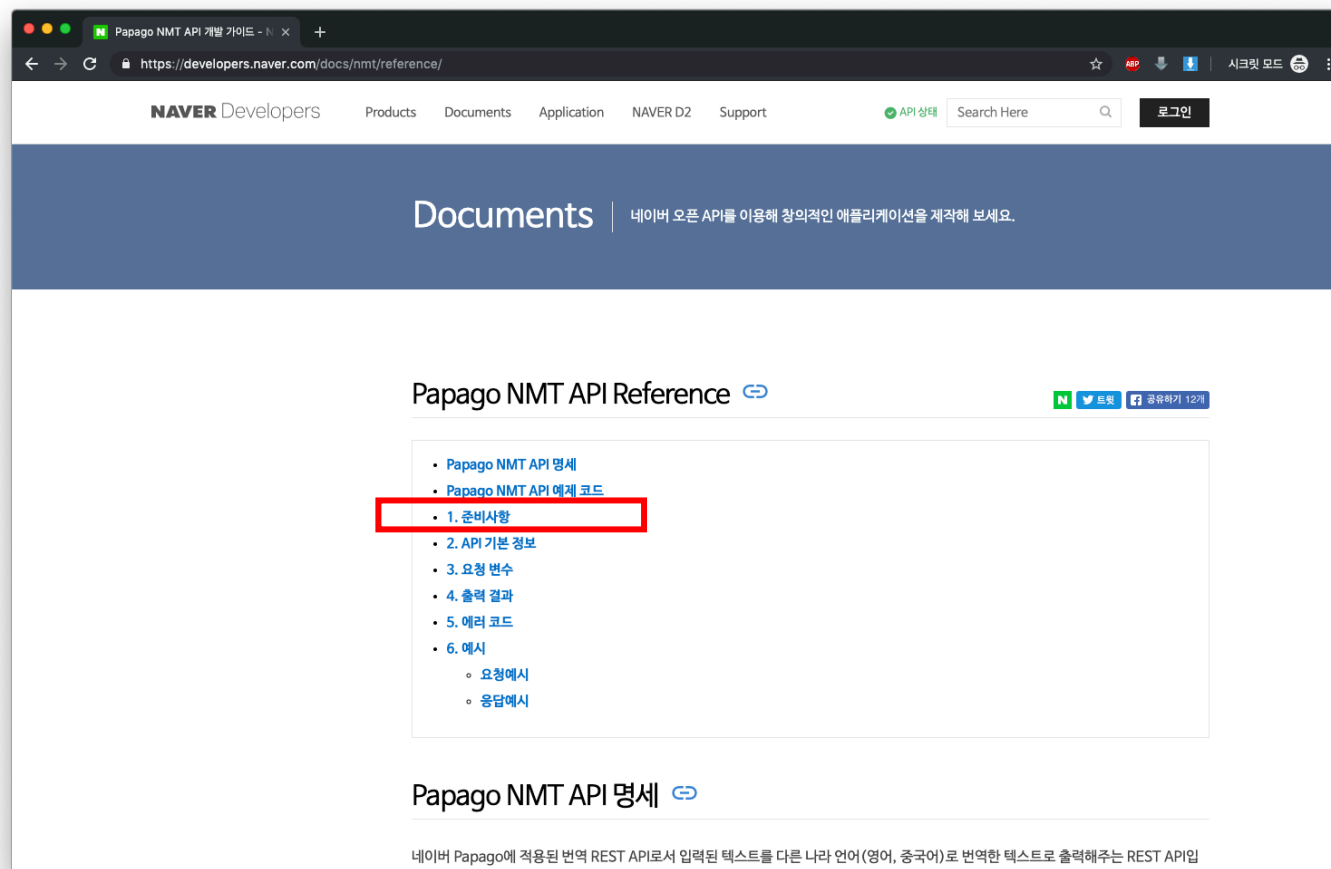
- 네이버 파파고 번역 API
- 클라이언트 구현
- 서버 구현

NodeJS 동시통역 채팅 프로그램

- 채팅 웹 페이지를 Node JS 서버를 통해 호스팅
- 파파고 Open API를 이용하여 메시지 번역 기능 구현

NodeJS 동시통역 채팅 프로그램

<https://developers.naver.com/docs/nmt/reference/>



NodeJS 동시통역 채팅 프로그램

https://developers.naver.com/apps/#/register?api=ppg_n2mt

The screenshot shows the 'Application' registration page on the Naver Developers portal. The page title is 'Application' with a subtitle 'API 이용을 위해 애플리케이션을 등록하고 API 설정을 할 수 있습니다.' (You can register an application and set the API for API use). The left sidebar contains links: '내 애플리케이션' (My Application), '애플리케이션 등록' (Application Registration), 'Clova Platform Console β', 'API 제휴 신청' (API Partnership Application), and '계정 설정' (Account Settings). The main content area is titled '애플리케이션 등록 (API 이용신청)' (Application Registration (API Use Application)). It includes a description: '애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.' (If you register the basic information of the application, a sub-menu will be created in the sub-menu of the My Application menu on the left, with the name of the application you registered). Below this is a form with three rows: 1. '애플리케이션 이름' (Application Name) with a text input field and a red border, followed by a red circle with an exclamation mark. The input field has a placeholder '애플리케이션 이름' and a list of rules: '네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요.' (Since this is the name displayed to users when logging in with a Naver ID, please use a simple name within 10 characters whenever possible.) and '40자 이내의 영문, 한글, 숫자, 공백문자, "-", "." 만 입력 가능합니다.' (Only English, Korean, numbers, spaces, hyphens, and periods within 40 characters are allowed for input). 2. '사용 API' (API to be used) with a dropdown menu showing '선택하세요.' (Select) and a button 'Papago NMT 번역' (Papago NMT Translation) with a close icon. 3. '비로그인 오픈 API 서비스 환경' (Environment for non-logged-in open API services) with a dropdown menu showing '환경 추가' (Add Environment) and a red circle with an exclamation mark.

애플리케이션 이름	<input type="text" value="애플리케이션 이름"/> ⓘ • 네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요. • 40자 이내의 영문, 한글, 숫자, 공백문자, "-", "." 만 입력 가능합니다.
사용 API	<div>선택하세요. ▼</div> <div>Papago NMT 번역 ✕</div>
비로그인 오픈 API 서비스 환경	<div>환경 추가 ▼ ⓘ</div>

NodeJS 동시통역 채팅 프로그램

https://developers.naver.com/apps/#/register?api=ppg_n2mt

The screenshot shows the Naver Developers application registration page. The browser address bar displays the URL: `https://developers.naver.com/apps/#/register?api=ppg_n2mt`. The page header includes the Naver Developers logo and navigation links: Products, Documents, Application, NAVER D2, and Support. A search bar with the placeholder text "Search Here" is also present.

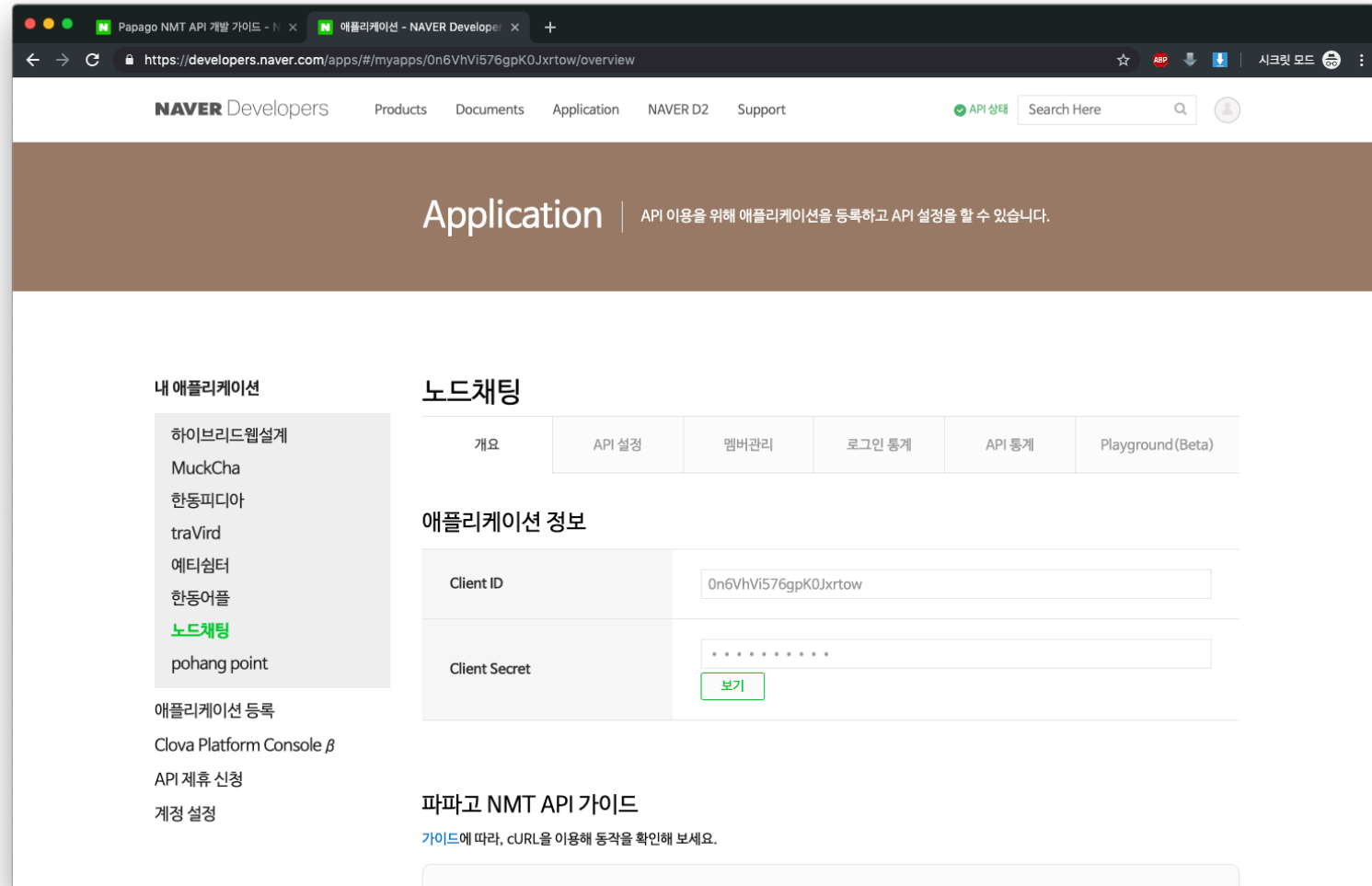
The main content area is divided into three sections:

- 내 애플리케이션 (My Application):** Contains links for "애플리케이션 등록 (Application Registration)", "Clova Platform Console β", "API 제휴 신청 (API Partnership Application)", and "계정 설정 (Account Settings)".
- 애플리케이션 이름 (Application Name):** A text input field containing "노드채팅" (Node Chat) with a green checkmark icon. Below the input field, there are two bullet points: "네이버 아이디로 로그인할 때 사용자에게 표시되는 이름이므로 가급적 10자 이내의 간결한 이름을 사용해주세요." and "40자 이내의 영문, 한글, 숫자, 공백문자, '-', '_' 만 입력 가능합니다."
- 사용 API (Used API):** A dropdown menu showing "선택하세요." (Select) and a text input field containing "Papago NMT 번역" (Papago NMT Translation) with a red 'X' icon.
- 비로그인 오픈 API 서비스 환경 (Non-Login Open API Service Environment):** A dropdown menu showing "환경 추가" (Add Environment) and a "WEB 설정" (WEB Settings) section. The "WEB 설정" section has a title bar with "X" and "↑" icons. It contains a table with the following data:

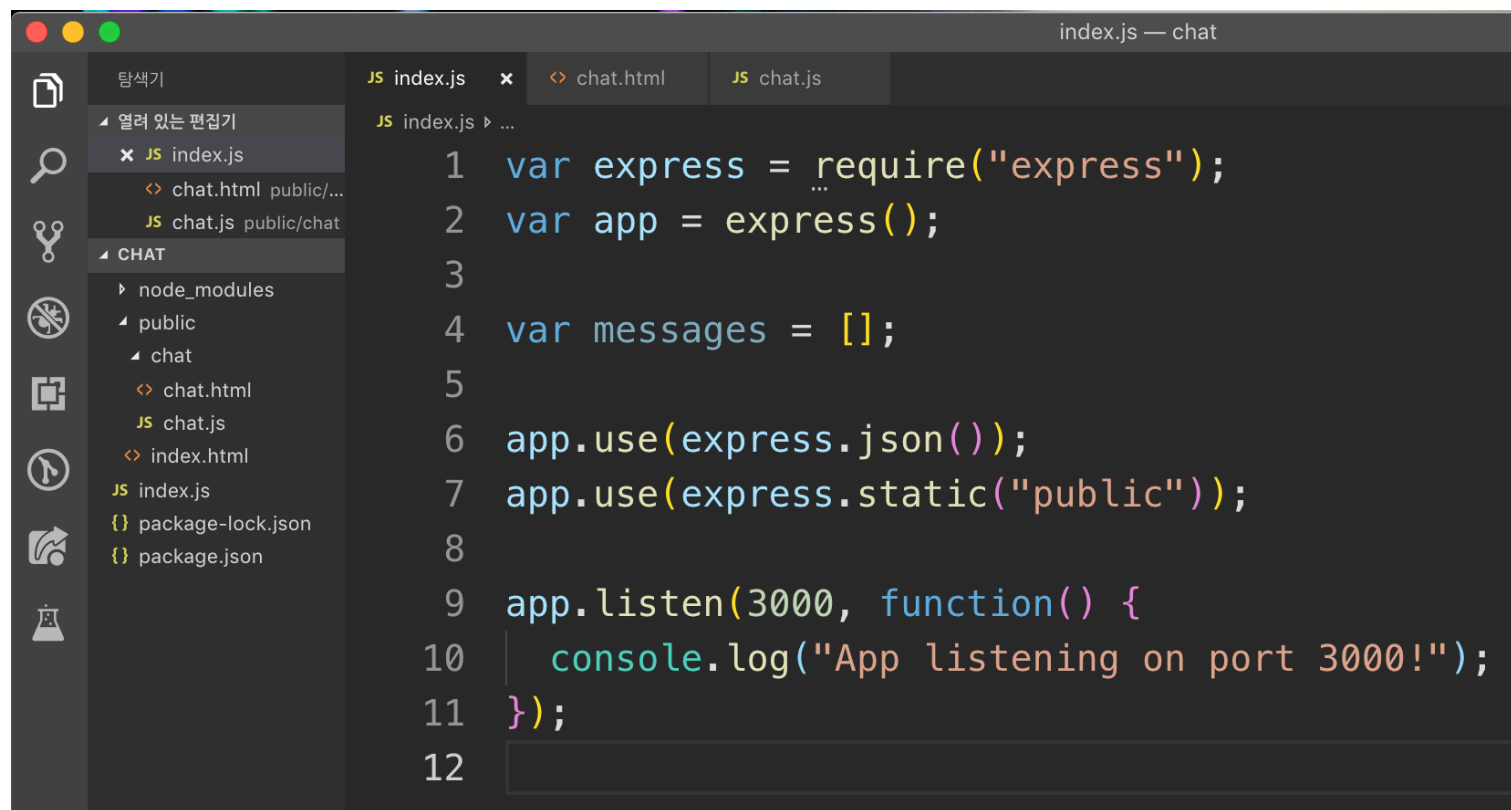
웹 서비스 URL (최대 10개) (Web Service URL (Maximum 10))		
<input type="text" value="http://localhost"/>	-	✓
<input type="text" value="http://127.0.0.1"/>	-	✓
<input type="text" value="http://자신의서버주소"/>	+	✓

Below the table, there are four bullet points: "텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.", "http와 https는 구분하지 않습니다.", "www는 빼고 입력해 주세요. 예) http://naver.com", and "서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)".

NodeJS 동시통역 채팅 프로그램



NodeJS 동시통역 채팅 프로그램

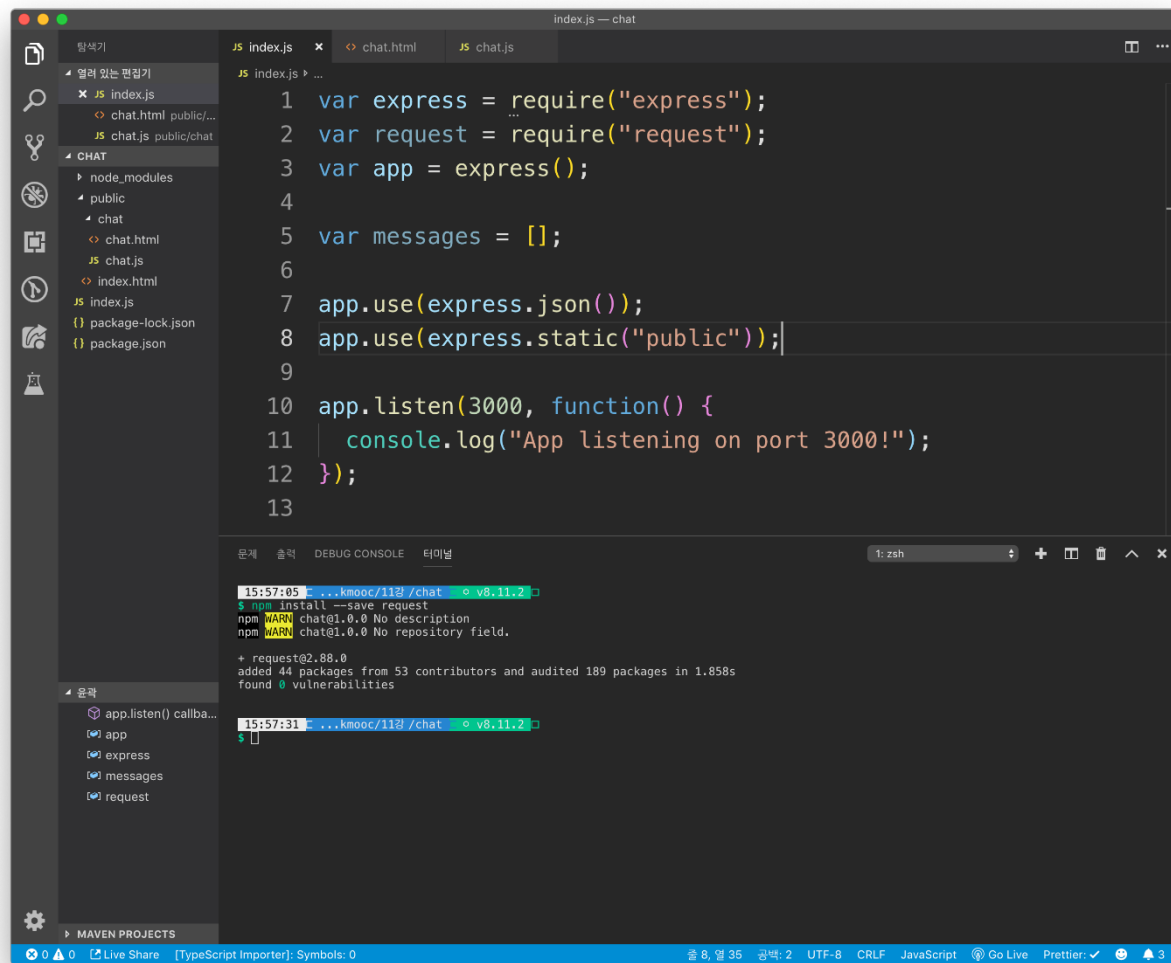


The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer with a tree view containing folders like 'node_modules', 'public', and 'chat', and files like 'index.js', 'package-lock.json', and 'package.json'. The main editor area shows the content of 'index.js' with the following code:

```
1 var express = require("express");
2 var app = express();
3
4 var messages = [];
5
6 app.use(express.json());
7 app.use(express.static("public"));
8
9 app.listen(3000, function() {
10   console.log("App listening on port 3000!");
11 });
12
```


NodeJS 동시통역 채팅 프로그램

npm install --save request



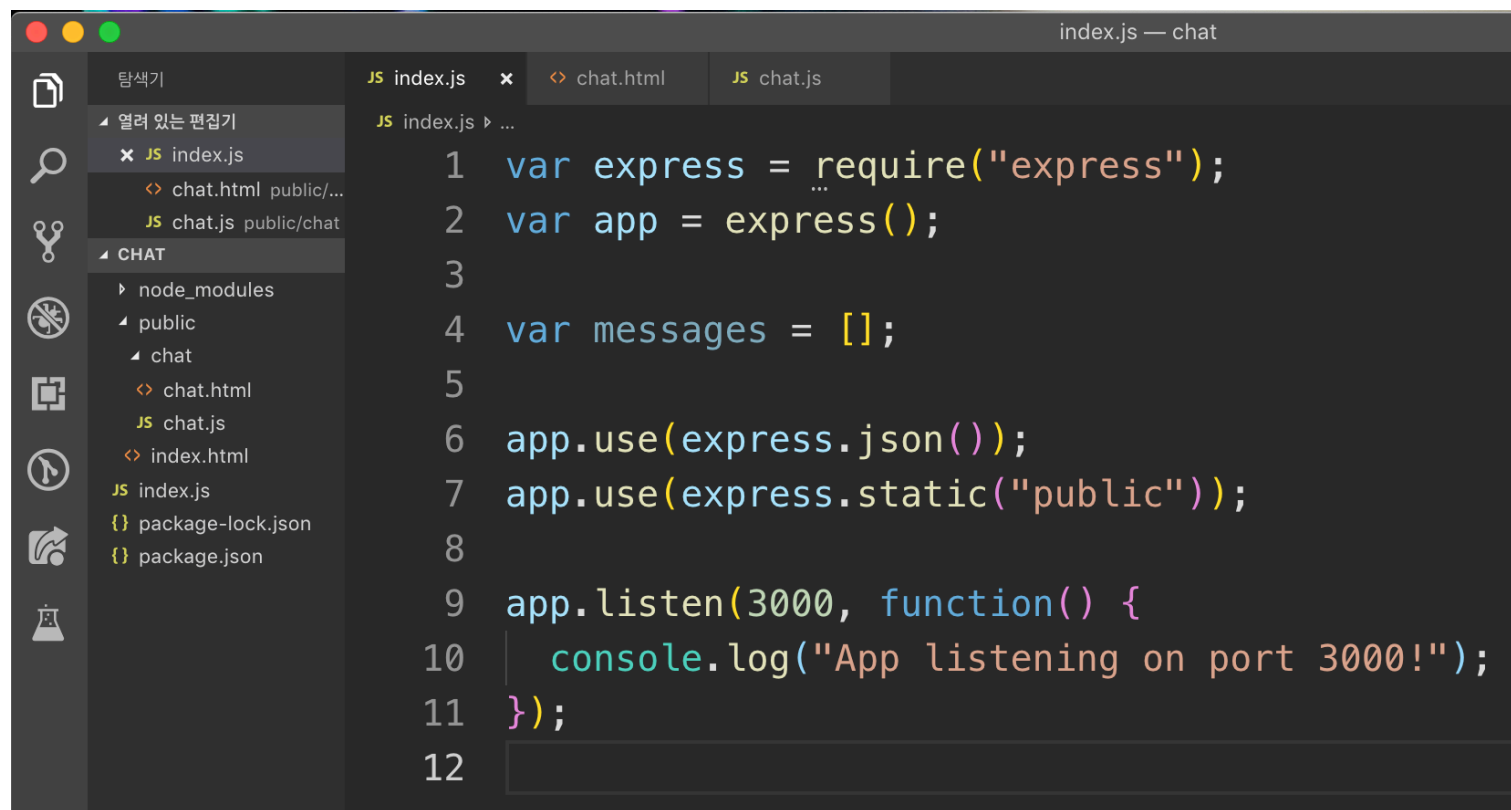
Request 모듈

- NodeJS 앱에서 HTTP 요청하기
- JQuery 의 .ajax 함수와 비슷
- `npm install --save request` 로 설치

Request 모듈 사용하기

```
1 var request = require('request');
2
3 request.get('/books', function(error, response, body) {
4     |
5 });
6
7 request.post('/books', { body: { key: value } }, function(error, response, body) {
8
9 });
10
11 request.put('/books', { body: { key: value } }, function(error, response, body) {
12
13 });
14
15 request.delete('/books', function(error, response, body) {
16
17 });
```

NodeJS 동시통역 채팅 프로그램



The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer with a tree view containing folders like 'node_modules', 'public', and 'chat', and files like 'index.js', 'package-lock.json', and 'package.json'. The main editor area shows the content of 'index.js' with the following code:

```
1 var express = require("express");
2 var app = express();
3
4 var messages = [];
5
6 app.use(express.json());
7 app.use(express.static("public"));
8
9 app.listen(3000, function() {
10   console.log("App listening on port 3000!");
11 });
12
```

NodeJS 동시통역 채팅 프로그램

```
13
14 app.post("/send", function(req, res) {
15     var message = {
16         sender: req.body.sender,
17         ko: req.body.ko,
18         en: req.body.en
19     };
20
21     console.log(message);
22     messages.push(message);
23     res.status(200).send({ message: "Success" });
24 });
25
```

NodeJS 동시통역 채팅 프로그램

<https://developers.naver.com/docs/nmt/reference/>

2. API 기본 정보

API명	메서드	요청 URL	출력 포맷
N2MT	POST	https://openapi.naver.com/v1/papago/n2mt	JSON

6. 예시

요청예시

```
POST /v1/papago/n2mt HTTP/1.1
HOST: openapi.naver.com
User-Agent: curl/7.49.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 client id 값}
X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 client secret 값}
Content-Length: 51
```

3. 요청 변수

요청 변수명	타입	필수 여부	설명
source	string	Y	번역 대상 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
target	string	Y	번역 결과 언어 코드. 1.ko : 한국어 2.en : 영어 3.zh-CN : 중국어 간체 4.zh-TW : 중국어 번체 5.es : 스페인어 6.fr : 프랑스어 7.vi : 베트남어 8.th : 태국어 9.id : 인도네시아어 ko(-)en, ko(-)zh-CN, ko(-)zh-TW, ko(-)es, ko(-)fr, ko(-)vi, ko(-)th, ko(-)id, en(-)ja, en(-)fr 조합만 가능
text	string	Y	번역할 text (1회 호출시 최대 5,000자까지 가능)

NodeJS 동시통역 채팅 프로그램

```
14 app.post("/send", function(req, res) {
15     var message = {
16         sender: req.body.sender,
17         ko: req.body.ko,
18         en: req.body.en
19     };
20     console.log(message);
21
22     var options = {
23         url: "https://openapi.naver.com/v1/papago/n2mt",
24         form: {
25             source: message.ko.length == 0 ? "en" : "ko",
26             target: message.ko.length == 0 ? "ko" : "en",
27             text: message.ko.length == 0 ? message.en : message.ko
28         },
29         headers: {
30             "X-Naver-Client-Id": "Client Id",
31             "X-Naver-Client-Secret": "Client Secret",
32             "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"
33         }
34     };
```

NodeJS 동시통역 채팅 프로그램

```
29     headers: {
30         "X-Naver-Client-Id": "Client Id",
31         "X-Naver-Client-Secret": "Client Secret",
32         "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"
33     }
34 };
35
36 request.post(options, function(error, response) {
37     var result = JSON.parse(response.body).message.result;
38     message.ko = message.ko.length == 0 ? result.translatedText : message.ko;
39     message.en = message.en.length == 0 ? result.translatedText : message.en;
40     console.log(message);
41     messages.push(message);
42     res.status(200).send({ message: "Success" });
43 });
44 });
```


NodeJS 동시통역 채팅 프로그램

응답예시 [↗](#)

```
{
  "message": {
    "@type": "response",
    "@service": "naverservice.labs.api",
    "@version": "1.0.0",
    "result": {
      "translatedText": "tea"
    }
  }
}
```

NodeJS 동시통역 채팅 프로그램

```
29     headers: {
30         "X-Naver-Client-Id": "Client Id",
31         "X-Naver-Client-Secret": "Client Secret",
32         "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8"
33     }
34 };
35
36 request.post(options, function(error, response) {
37     var result = JSON.parse(response.body).message.result;
38     message.ko = message.ko.length == 0 ? result.translatedText : message.ko;
39     message.en = message.en.length == 0 ? result.translatedText : message.en;
40     console.log(message);
41     messages.push(message);
42     res.status(200).send({ message: "Success" });
43 });
44 });
```

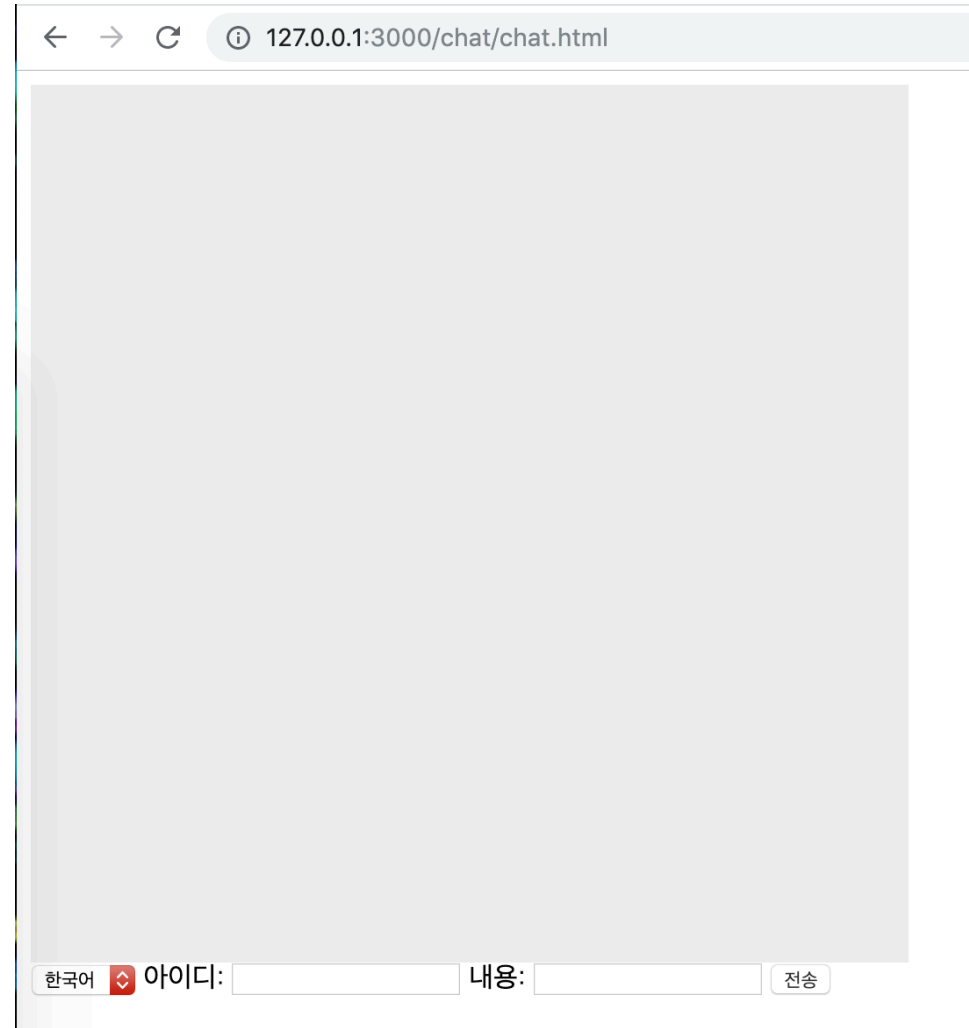
NodeJS 동시통역 채팅 프로그램

```
45
46 app.get("/receive", function(req, res) {
47     var result = { total: messages.length, messages: [] };
48
49     if (messages.length > req.query.from) {
50         result.messages = messages.slice(req.query.from);
51     }
52
53     res.status(200).send(result);
54 });
55
```

NodeJS 동시통역 채팅 프로그램

```
1 <html>
2   <head>
3     <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
4     <style>
5       #messages {
6         width: 500px;
7         height: 500px;
8         overflow: scroll;
9         background: #eeeeee;
10      }
11    </style>
12  </head>
13  <body>
14    <div id="messages"></div>
15    <div>
16      <select id="selectBox">
17        <option value="ko">한국어</option>
18        <option value="en">영어</option>
19      </select>
20      아이디: <input type="text" id="senderId" /> 내용:
21      <input type="text" id="chatInput" />
22      <button id="sendButton">전송</button>
23    </div>
24    <script src="./chat.js"></script>
25  </body>
26 </html>
```

NodeJS 동시통역 채팅 프로그램



NodeJS 동시통역 채팅 프로그램

```
1 var curIndex = 0;
2 var language = "ko";
3
4 setInterval(function() {
5     $.ajax(`http://localhost:3000/receive?from=${curIndex}`).done(function(data) {
6         for (const message of data.messages) {
7             console.log(message);
8
9             $("#messages").append(
10                message.sender + ": " + (language == "ko" ? message.ko : message.en)
11            );
12            $("#messages").append("<br />");
13        }
14        curIndex = data.total;
15    });
16 }, 1000);
17
```

NodeJS 동시통역 채팅 프로그램

```
18 $("#sendButton").click(function() {
19     var message = {
20         ko: "",
21         en: "",
22         sender: $("#senderId").val()
23     };
24
25     if (language == "ko") message.ko = $("#chatInput").val();
26     else message.en = $("#chatInput").val();
27
28     console.log(message);
29
30     $.ajax({
31         url: "http://localhost:3000/send",
32         method: "POST",
33         data: JSON.stringify(message),
34         dataType: "json",
35         contentType: "application/json"
36     }).done(function() {
37         console.log("POST done");
38         $("#chatInput").val("");
39     });
40 });
```

NodeJS 동시통역 채팅 프로그램

```
41
42  $("#selectBox").change(function() {
43      language = $("#selectBox").val();
44      console.log(language);
45      $("#messages").html("");
46      curIndex = 0;
47  });
```


NodeJS 동시통역 채팅 프로그램

← → ↻ 127.0.0.1:3000/chat/chat.html

blue: 안녕하세요
blue: 노드 서버 테스트입니다
홍길동: 오늘 날씨가 좋네요

한국어  아이디:

내용:

전송

← → ↻ 127.0.0.1:3000/chat/chat.html

blue: Hello.
blue: Node server test.
홍길동: The weather is nice today.

영어  아이디:

내용:

전송

요약

- 네이버 파파고 번역 API
- 클라이언트 구현
- 서버 구현

차시 예고

- 12-3 : Cordova 소개 및 설치하기
 - Cordova 소개
 - JDK 설치하기
 - 안드로이드 스튜디오 설치하기
 - 안드로이드 SDK 설치하기
 - Cordova 설치하기

강의를 마치겠습니다
수고하셨습니다

12주차_02 동시통역 채팅 프로그램 만들기