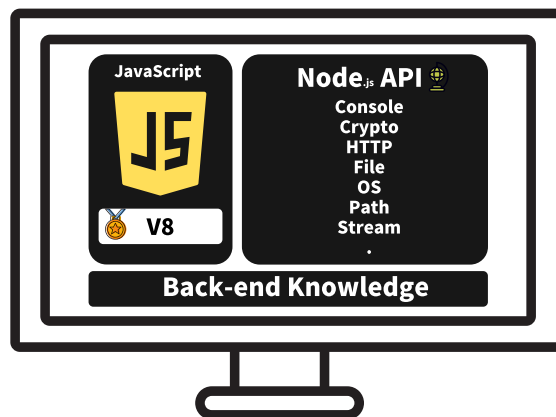


# Backend A-Z

# NODE.JS



**Dream Coding**  
CODE YOUR DREAM

**Backend A-Z**

# COURSE PLANNING

| Chapter                 | Due Date | Done Date |
|-------------------------|----------|-----------|
| 1.Introduction          |          |           |
| 2.What's Node.JS?       |          |           |
| 3.Tools · Preparation   |          |           |
| 4.Node Modules          |          |           |
| 5.NPM 구석 구석             |          |           |
| 6.Debugging 치명적인 매력     |          |           |
| 7.파일정리 스크립트 만들기         |          |           |
| 8.Web 기초 지식 HTTP        |          |           |
| 9.Node로 서버 만들기          |          |           |
| 10. Restful APIs        |          |           |
| 11.Express.js 가좌아       |          |           |
| 12.Dwitter - 프로젝트 시작!   |          |           |
| 13.리팩토링 - MVC           |          |           |
| 14.Validation - 유효성     |          |           |
| 15.Authentication - 로그인 |          |           |
| 16.Dwitter - 로그인 적용     |          |           |
| 17.Configuration        |          |           |
| 18.Socket 실시간 트윗        |          |           |
| 19.Database 정리          |          |           |
| 20.MySQL                |          |           |
| 21.Sequelize            |          |           |
| 22.MongoDB              |          |           |
| 23.Mongoose             |          |           |
| 24.Deployment - 배포      |          |           |
| 25.Goodbye              |          |           |

# COURSE CONTENT

## 1. INTRODUCTION

강의 진행 방식과 어떻게 강의를, 각 챕터를 효율적으로 들을 수 있는지 알아 봅니다.

강의를 통해 궁극적으로 얻고 싶은 목표와 어떻게 강의를 원하는 목표기간안에 200% 내것으로 만들 건지 목표를 설정해 볼거예요.

- ☐ 강의를 통해 성취하고 싶은 궁극적인 목표 설정
- ☐ 강의를 어떻게 들을 건지 목표와 계획을 설정
- ☐ 강의를 듣고 나중에 더 알아 보고 싶은 토픽을 적을 노트 준비 (디지털 노트 또는 메모장)

## 2. WHAT'S NODE.JS ?

개념

Node.js가 탄생한 배경과 어떤 분야에서 활용될 수 있는지, 그리고 배웠을때 나에게 어떤 좋은 역량 무기가 될 수 있는지 알아 봅니다. 그리고 조금더 깊숙하게 들어가서 Node.js 주요 특징 들과 내부 동작 방식에 대해 이해합니다. 그리고 이런 특징들이 서버로 동작했을때 어떤 장/단점이 있는지도 알아봐요.

- ☐ Node.js는 무엇인지, 그리고 무엇을 할 수 있는지 말하기
- ☐ Node.js 특징과 내부 동작 방식에 대해 큰 그림 이해하기
- ☐ Node.js 서버의 특징과 장점과 단점 말하기

내부 동작 방식과 서버의 특징 개념은 다소 어려울 수 있기 때문에 가볍게 들어 보시고, 강의를 다 완강한 후 최종 정리 하는 느낌으로 다시 들어 보세요 :)

## 3. TOOLS · PREPARATION

실습

Node.js 개발을 위한 환경설정과 필요한 툴들에 대해 알아보고 설치합니다.

노드를 REPL로 그리고 파일로도 코딩해서 노드 맛보기를 합니다. Hello World!

본격적으로 강의를 시작하기 전에 필요한 것들을 함께 챙겨볼거예요


- ☐ 기본 필수 툴들을 설치
- ☐ REPL이란 무엇인지 알아보고 직접 실행해 보기
- ☐ 파일로 노드 코드를 동작시켜 보기

# COURSE CONTENT

## 4. NODE MODULES

실습

노드 APIs 에서 자주 쓰이는 유용한 모듈 들에 대해 알아 보고 직접 코딩을 하면서 익숙해질 수 있도록 해볼거예요. 글로벌, 콘솔 로그, 운영 체제, 프로세스, 파일, 버퍼등 다양한 모듈 들을 직접 사용해 봅니다.

- ☐ 중요한 사이트 북마크 해두기 
- ☐ 어떻게 효율적으로 모듈 들을 알아보고 공부해 볼 수 있는지 팁 확인
- ☐ 스스로 모듈 들을 살펴보고 필요한 내용들을 습득 할 수 있기
- ☐ 실습 노트 (프로젝트) 만들어 두고 계속 사용해 보기

## 5. NPM 구석 구석

실습

NPM은 무엇인지, 사용하는 방법에 대해 정확하게 알아 봅니다. 프로젝트 패키지 들을 관리하고, 어떻게 유용한 외부 라이브러리를 찾고 다운로드 받을 수 있는지에 대해 정확하게 정리해 볼거예요. 그리고 라이선스 종류와 버전 관리를 위한 정보들도 다뤄봅니다.

- ☐ NPM은 무엇이고 어떻게 활용되는지 확인해 보기
- ☐ 프로젝트에 필요한 외부 라이브러리를 다운로드 받고 관리 할 수 있기
- ☐ 프로젝트 버전명 관리 하는 방법에 대해 알아 보기

## 6. DEBUGGING 치명적인 매력

실습

개발에 필수적인 디버깅이란 무엇인지 알아 보고, 디버깅을 위한 정말 중요한 툴 디버거 기본 사용법과 활용법에 대해 알아 봅니다. 🍷 꿀팁들 많이 담아 갈 준비를 해주세요 :)

- ☐ 디버깅의 목적이 무엇인지 이해하기
- ☐ 디버거의 사용법을 익히고 직접 실습해 보기
- ☐ 엘리가 알려주는 꿀팁들 사용해 보기
- ☐ 앞으로 개발할때 유용하게 활용해 보기

# COURSE CONTENT

## 7. 파일정리 스크립트 만들기

과제

여태까지 배운 노드 모듈과 개발 툴들을 활용해서 직접 프로젝트를 해봅시다 📌

프로젝트를 위한 모든 지식은 앞에서 배웠기 때문에, 1시간 정도 (초급자는 세시간까지) 시간을 두시고 꼭 스스로 공식 사이트와 검색을 통해서 구현해 봅니다. 잊지 마세요, 우리의 목표는 문제 해결 능력이라는 사실!

- ☐ 첫번째 영상에서 요구사항들을 확인하고 직접 정의해 보기
- ☐ 요구 사항을 토대로 1시간 time blocking 해두고 구현해 보기
- ☐ 콘솔 로그, 디버깅을 적극 활용해서 기능 구현을 해보기
- ☐ 다 완성한 프로젝트 깃허브 링크 또는 프로젝트를 압축해서 토론창에 공유하기
- ☐ 엘리의 솔루션을 보면서 어떻게 개선할 수 있을지 확인해 보기

## 8. WEB 기초 지식 HTTP

개념

HTTP란 무엇인지 그리고 HTTP와 HTTPS의 차이점, HTTP v1, v2, v3의 차이점들에 대해 알아 봅니다.

웹서버에서 정말 중요한 HTTP 요청 함수 (request method), 반응 코드 (status code)를 어떻게 정확하게 활용할 수 있는지 그리고 헤더는 어떻게 사용되는지 기본적인 내용을 정리 하고 추가적으로 면접에서도 자주 등장하는 조금 난이도 있는 부분들도 정리해 봅니다 :)

- ☐ HTTP의 역사와 HTTPS와의 차이점 이해하기
- ☐ HTTP 세가지 다른 버전의 차이점 이해하기
- ☐ Request method, Response Code, 그리고 Headers는 무엇인지 이해하기
- ☐ 필요할때 어떻게 더 공부 할 수 있는지 팁 북마크 📌

## 9. NODE로 서버 만들기

실습

앞에서 배운 노드 모듈과 Web 기초 지식을 활용해서 노드로 첫번째 서버를 만들어 봅니다.

클라이언트에 HTML 보내주기 부터, Templating Engine인 EJS를 활용해보고, Restful APIs를 위한 JSON 데이터도 보내주는 서버를 만들어 보도록 할거예요.

- ☐ HTML 파일 보내보기
- ☐ Templating Engine, EJS 사용해 보기
- ☐ JSON으로 데이터 보내보기
- ☐ Postman 사용해 보기

# COURSE CONTENT

## 10. RESTFUL APIS

개념

진정한 Restful APIs에 대해 알아보고 통상적으로 쓰이는 APIs 구현 방법에 대해서도 정리해 보아요 :) 커피 한잔 마시면서 대략적인 그림을 이해해 보시고 마지막에 보여드리는 예제를 통해서 좋은 영감을 받아 볼거예요.

- ☐ Restful APIs 원칙은 무엇인지 이해하기
- ☐ APIs 디자인 할때 피해야 하는 지켜야 하는 규칙 이해하기
- ☐ 진정한 Restful APIs 예제를 보면서 어떤 부분을 도입할 수 있을지 생각해보기
- ☐ 기본적인(통상적인) APIs 예제를 보면서 기본 정의법 이해하기

## 11. EXPRESS.JS 가짜아

실습

드디어 Express! 앞에서 노드로만 만든 서버와 Express로 만드는 서버는 어떻게 다른지 차이점을 포인트로 두면서 강의를 따라와 보세요 ☺ Express는 무엇인지, 어떻게 사용하면 좋을지 큰그림부터 확인하고 디테일한 내용들을 함께 실습하면서 익숙하게 사용해 볼거예요 :)

- ☐ Express 전반적인 큰 그림 이해하기 (포인트 잡아놓기)
- ☐ 미들웨어는 무엇인지 정확하게 이해하기
- ☐ 기본 사용방법, 에러 처리 방법, 라우트에 대해 실습하기
- ☐ 유용한 내부 그리고 커뮤니티에서 많이 사용되어지는 미들웨어 사용하기

## 12. DWITTER - 프로젝트 시작!

과제

앞에서 배운 모든 지식들을 활용해서 첫번째! Restful APIs를 제공하는 서버를 만들어 봅니다. 기능 사항들을 분석하면서 APIs를 설계하고 Postman 활용 꿀팁을 이용해서 서버를 만들어 볼거예요. 자바스크립트를 이용해서 리액트 기반의 프론트엔드 코드도 업데이트 해서 서버로 부터 트윗 목록을 받아오고, 작성 및 수정하는 기능을 완성해 볼거예요. 우리의 목표는 배운 지식을 활용해서 기능을 구현 할 수 있는 문제 해결 능력 인거 아시죠? 꼭 적절한 시간을 분배해서 스스로 먼저 구현해 보는 시간을 가져 보세요. (적절한 구현 시간: 3~5시간)

- ☐ 기능 요구 사항 분석후 알맞는 APIs설계하기
- ☐ 백엔드 개발을 위한 포스트맨 활용법을 이용해서 서버 구현 하기
- ☐ 자바스크립트를 활용하여 프론트엔드 서비스층 구현하기

# COURSE CONTENT

## 13. 리팩토링 - MVC

과제

앞에서 기능 구현을 목표 로만 작성한 서버와 프론트엔드 서비스층의 코드를 리팩토링을 통해 개선해 봅니다. MVC 패턴과 코드를 재사용 할 수 있는 다양한 기법들을 활용해서 코드를 개선하는 팁을 배우고 또 여러분 스스로 사고하면서 코드를 개선할 수 있도록 챌린지를 진행해 볼거예요 :)

- ☐ 첫번째 영상에서 영감을 받은 힌트를 통해 직접 코드 개선해 보기 (고민해 보기)
- ☐ 솔루션 영상들을 보면서 어떻게 개선 되었는지 확인하기
- ☐ 앞으로 코드를 작성할때 어떻게 적용할 건지 포인트 잡아놓기

## 14. VALIDATION - 유효성

과제

유효성 검사란 무엇인지, 백엔드와 프론트엔드에서 어떻게 처리하는게 좋은지 얘기해 보고, 실습을 통해 기본적인 내용을 함께 정리 한 다음 스스로 챌린지를 해보면서 Dwitter 프로젝트에 적용해 봅니다.

- ☐ Validation, Sanitization 이란 무엇인지, 중요성 확인
- ☐ Validation 실습을 통해 필요한 내용 정리 하기
- ☐ Sanitization 실습을 통해 필요한 내용 정리 하기
- ☐ Dwitter에 어떻게 적용할 수 있을지 고민하고 적용하기

## 15. AUTHENTICATION - 로그인

실습

웹서비스에서 기본적으로 사용되어지는 Authentication (인증, 로그인)에 대해 알아보고 여러가지 테크닉에 대해 알아 봅니다. 기본적인 개념 부터 시작해서 전통적으로 사용되는 세션과 쿠키 그리고 JWT 방식은 무엇인지 그들의 차이점, 장/단점에 대해 정리를 해볼거예요 :) Bcrypt란 무엇인지 정리도 해보고 직접 JWT, Bcrypt 실습을 통해 재밌게 알아볼거예요

- ☐ 세션과 쿠키 vs JWT 방식의 차이점을 이해
- ☐ Bcrypt와 JWT 실습하면서 사용하는 법 익히기
- ☐ 자바스크립트를 활용하여 프론트엔드 서비스층 구현하기

# COURSE CONTENT

## 16. DWITTER - 로그인 적용

과제

앞에서 배운 로그인, 인증과 관련된 기술들을 적용해서 Dwitter에 사용자 가입, 로그인 기능을 완성해 봅시다. 사용자의 패스워드를 어떻게 안전하게 보관할 것인지, 어떻게 사용자를 인증할 것인지 고민해 보면서 프로젝트를 만들고, 사용자 관련 데이터와 트윗의 데이터를 어떻게 설계하는게 좋을지 고민해 볼거예요 :)

- ☐ 로그인 관련 APIs 설계하기
- ☐ 사용자 인증 미들웨어 만들어 보기
- ☐ Postman을 이용해서 간편하게 로그인된 사용자 요청 테스트하기
- ☐ User와 Tweet의 모델을 어떻게 설계하면 좋을지 고민하기
- ☐ 직접 스스로 도전하면서 가입, 로그인 서비스 완성하기 (백엔드와 프론트엔드)

## 17. CONFIGURATION

과제

서버에서 중요한 보안 관련해서 어떻게 중요한 정보들을 숨길 수 있는지, 어떻게 하면 서버의 설정을 유연하게 만들 수 있을지 고민하면서 기존에 작성한 프로젝트를 개선해 볼거예요. 실습을 통해서 필요한 내용들을 확인해 보고 스스로 프로젝트에 적용하면서, 엘리는 어떻게 코드의 활용성을 높였는지 팁을 확인해 보세요 :)

- ☐ config의 중요성 확인하기 (하지 않았을때의 두가지 문제점 정리)
- ☐ 어떻게 config를 처리 할 수 있는지 확인하기
- ☐ 더 나아가 어떻게 코드를 개선할 수 있는지, 활용성을 높일 수 있는지. 🍷팁 확인

## 18. SOCKET 실시간 트윗!

과제

재미로 추가한 이번 챕터는 소켓을 이용해서 실시간으로 업데이트 되는 트윗을 사용자가 받아 볼 수 있도록 구현해 볼거예요. 소켓은 무엇인지, 어떻게 구현 할 수 있는지 기본적인 사용법을 함께 익혀보고 그 뒤에 여러분들 스스로 실시간 트윗 기능을 Dwitter 프로젝트에 적용해 봅시다. 유후!

- ☐ HTTP와 Socket의 차이점 이해하기
- ☐ 소켓으로 백엔드, 프론트엔드 기본 동작 사항 실습 하기
- ☐ 실습한 내용을 토대로 Dwitter에 스스로 적용해 보고 솔루션 듣기 :)



# COURSE CONTENT

## 19. DATABASE 정리

개념

☞와 함께 하는 개념만 정리하는 챕터예요. Database는 무엇인지, 그 종류에는 어떤것들이 있는지 알아 봅니다. SQL과 NoSQL은 무엇인지 차이점에 대해 알아보고 ORM/ODM은 무엇인지도 알아봅니다. 그리고 배운 내용을 토대로 해서 프로젝트별로 어떤 데이터베이스를 선택하면 좋은지 정리해 볼거예요.

- ☐ 데이터베이스란 무엇인지 말해보기
- ☐ SQL 데이터베이스 특징에 대해 이해하기
- ☐ NoSQL 데이터 베이스 특징에 대해 이해하기
- ☐ ORM과 ODM은 무엇인지, 장/단점에 대해 이해하기
- ☐ 프로젝트별로 어떻게 적절한 데이터베이스를 찾으면 좋은지 팁 확인

## 20. MYSQL

실습

MySQL 데이터베이스를 직접 설치해서 사용해 봅니다. 기본적인 관계형 스키마를 정의하고 Join Query를 이용해서 관련 있는 데이터들을 읽어오고, 또 업데이트 해보는 기본적인 CRUD 동작 방법을 익혀보고 Dwitter에 적용해 볼거예요. 함께 실습하면서 전반적인 사용법에 대해 익혀보는 시간을 가져보자구요.

- ☐ MySQL 설치및 테이블 (스키마) 정의하기
- ☐ Dwitter 적용 - 사용자 정보 읽고 추가하기
- ☐ Dwitter 적용 - 사용자와 트윗의 정보를 함께 Join해서 읽어오는 동작 외 CRUD 구현

## 21. SEQUELIZE

과제

MySQL의 대표적 ORM인 Sequelize 사용법을 익혀보고 적용해 봅니다. 어떻게 찾아보고 공부하면 되는지 문서를 읽는 팁을 들어보고 스스로 구현해 볼 수 있도록 해볼거예요. 이 챌린지를 통해서 개발자로서 정말 중요한 필요한 지식과 해결법을 빠르게 찾아서 적용하는 능력을 연습 하실 수 있어요.

- ☐ Sequelize에 대해 알아보기
- ☐ 스스로 Dwitter에 적용해서 Postman으로 기능 테스트 하기
- ☐ 솔루션을 확인하면서 내용 점검

# COURSE CONTENT

## 22. MONGODB

실습

MongoDB를 사용해 봅시다. 데이터베이스 개념 챕터에서 배운 내용들을 직접 실습을 통해 내것으로 소화하는, 직접 체험해 보는 시간을 가져볼 거예요. 앞에서 사용한 MySQL과 어떤 차이점이 있는지 그 부분을 포인트로 잡고 챕터를 진행하시면 재밌을 거예요.

- ☐ MongoDB 설치해서 사용해 보기
- ☐ Dwitter 적용 - 사용자 정보 읽고 추가하기
- ☐ Dwitter 적용 - 사용자와 트윗의 정보 CRUD 구현
- ☐ MYSQL과 어떤점이 다른지, 어떤 것을 유의 해야 하는지 포인트 정리 하기

## 23. MONGOOSE

과제

MongoDB의 대표적 ODM인 Mongoose 사용법을 익혀보고 적용해 봅시다. 어떻게 찾아보고 공부하면 되는지 문서를 읽는 팁을 들어보고 스스로 구현해 볼 수 있도록 해볼거예요. MongoDB만을 이용해서 구현했을때와 Mongoose를 사용했을때 어떤점이 달라지는지 개발자의 생산성의 차이점을 확인해 보세요

- ☐ Mongoose에 대해 알아보기
- ☐ Sqelize편에서 연습했던 문서를 통해 구현하는 능력 연습하기
- ☐ 솔루션을 확인하면서 내용 점검

## 24. DEPOLYMENT - 배포

실습

드디어 우리가 열심히 만든 Dwitter를 배포해 보는 시간입니다. 역시 이번 챕터도 단순히 배포 하는것이 아니라, 배포 시 꼭 확인해야할 체크리스트 📝 정리를 확인하고, 호스팅을 어떻게 고르는지도 정리 해 보고, Dwitter는 어떻게 프론트엔드/백엔드 호스팅을 선택했는지 확인해 볼거예요.

- ☐ 서버 배포전 꼭 체크해야 할 체크리스트 확인 ☑
- ☐ 다양한 호스팅 선택 방법 확인하기
- ☐ 호스팅 전 코드 업데이트 사항 확인
- ☐ 백엔드 배포하기
- ☐ 프론트엔드 배포하기
- ☐ 축하하고 자랑하기 ☑

# COURSE CONTENT

---

## 25. GOODBYE

---

강의전 세운 목표를 되돌아 보고 앞으로 어떻게 더 공부해 나가면 좋은지 얘기해 보아요 :)

- ☐ 앞으로 프로젝트를 어떻게 확장해 나갈 건지 계획
- ☐ 백엔드 로드맵 확인 후 더 공부 하고 싶은 내용 적어 보기
- ☐ 스스로 만들어 보고 싶은 프로젝트 아이디어 정리
- ☐ 멋진 수강평 남겨놓기 😊
- ☐ 보너스 챕터에 유용한 내용들 확인해 보기

