

시작하기 및 CLI (Ace)

AdonisJS는 견고한 아키텍처와 타입 안정성을 제공하는 Node.js 기반 풀스택/API TypeScript 프레임워크입니다.

- 프로젝트 생성: `npm create adonisjs@latest my-app`
- 개발 서버 실행: `node ace serve --hmr`
- 프로덕션 빌드: `node ace build`
- 컨트롤러 생성: `node ace make:controller User`
- 모델 및 마이그레이션 생성: `node ace make:model User -m`
- 마이그레이션 실행/롤백: `node ace migration:run / node ace migration:rollback`
- 미들웨어 생성: `node ace make:middleware Auth`
- 대화형 REPL 실행: `node ace repl`

라우팅 (Routing)

`start/routes.ts` 파일에서 엔드포인트를 정의합니다.

```
import router from '@adonisjs/core/services/router'
import UsersController from '#controllers/users_controller'
import { middleware } from '#start/kernel'
```

```
// 기본 HTTP 메서드
router.get('/users', [UsersController, 'index'])
router.post('/users', [UsersController, 'store'])
router.get('/users/:id', [UsersController, 'show'])
```

```
// 라우트 그룹 및 프리픽스
router.group(() => {
  router.get('/profile', [UsersController, 'profile'])
}).prefix('/api/v1').use(middleware.auth())
```

```
// RESTful 리소스 라우트
router.resource('posts', PostsController).apiOnly()
```

컨트롤러 및 요청 처리 (Controllers)

```
import type { HttpContext } from '@adonisjs/core/http'
import User from '#models/user'
import { createUserValidator } from '#validators/user'
```

```
export default class UsersController {
  async index({ response }: HttpContext) {
    const users = await User.all()
    return response.ok(users)
  }
```

```
  async store({ request, response }: HttpContext) {
    // VineJS 유효성 검사 수행
    const payload = await request.validateUsing(createUserValidator)
    const user = await User.create(payload)
    return response.created(user)
  }
}
```

Lucid ORM & 데이터베이스

Active Record 패턴 기반의 강력한 SQL ORM.

모델 정의 및 관계

```
import { BaseModel, column, hasMany } from '@adonisjs/lucid/orm'
import type { HasMany } from '@adonisjs/lucid/types/relations'
import Post from '#models/post'
```

```
export default class User extends BaseModel {
  @column({ isPrimary: true })
  declare id: number

  @column()
  declare email: string

  @hasMany(() => Post)
  declare posts: HasMany<typeof Post>
}
```

쿼리 빌더 (Queries)

- `const user = await User.findBy('email', email)`
- `const user = await User.findOrFail(id)`
- `const posts = await Post.query().where('published', true).preload('user')`
- `await user.related('posts').create({ title: 'Hello' })`

유효성 검사 (VineJS Validation)

```
import vine from '@vinejs/vine'
```

```
export const createUserValidator = vine.compile(
  vine.object({
    username: vine.string().trim().minLength(3).maxLength(30),
    email: vine.string().email(),
    password: vine.string().minLength(8),
    age: vine.number().min(18).optional()
  })
)
```

미들웨어 및 인증 (Auth & Middleware)

- 미들웨어 정의 (`app/middleware/auth_middleware.ts`):

```
import type { HttpContext } from '@adonisjs/core/http'
import type { NextFn } from '@adonisjs/core/types/http'
```

```
export default class AuthMiddleware {
  async handle(ctx: HttpContext, next: NextFn) {
    await ctx.auth.authenticateUsing(['api'])
    return next()
  }
}
```

- 인증 가드: Session, API Tokens, Access Tokens 지원

템플릿 엔진 (Edge Template)

- 변수 출력: `{{ user.username }}`
- 조건문: `@if(user.isAdmin) ... @else ... @end`
- 반복문: `@each(post in posts) ... @end`
- 컴포넌트: `@component('components/button', { text: '전송' })`