

Getting Started & CLI (Ace)

AdonisJS is a TypeScript-first, full-stack MVC and API framework for Node.js.

- **Create Project:** `npm create adonisjs@latest my-app`
- **Start Dev Server:** `node ace serve --hmr`
- **Production Build:** `node ace build`
- **Make Controller:** `node ace make:controller User`
- **Make Model & Migration:** `node ace make:model User -m`
- **Run / Rollback Migrations:** `node ace migration:run / node ace migration:rollback`
- **Make Middleware:** `node ace make:middleware Auth`
- **Start Interactive REPL:** `node ace repl`

Routing

Define application endpoints in `start/routes.ts`.

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

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

```
// Route Groups & Prefixes
router.group(() => {
  router.get('/profile', [UsersController, 'profile'])
}).prefix('/api/v1').use(middleware.auth())
```

```
// Resource Routing
router.resource('posts', PostsController).apiOnly()
```

Controllers & Request Handling

```
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) {
    // Validate with VineJS
    const payload = await request.validateUsing(createUserValidator)
    const user = await User.create(payload)
    return response.created(user)
  }
```

Lucid ORM & Database

Powerful Active Record SQL ORM with relational mapping.

Model Definition & Relations

```
import { BaseModel, column, hasMany } from '@adonisjs/lucid/orm'
import type { HasMany } from '@adonisjs/lucid/types/rerelations'
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>
}
```

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' })
```

Validation (VineJS)

```
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()
  })
)
```

Auth & Middleware

- **Middleware Definition** (`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()
  }
}
```

- **Guards:** Built-in Session, API Tokens, Access Tokens

Template Engine (Edge)

- **Variables:** `{{ user.username }}`
- **Conditionals:** `@if(user.isAdmin) ... @else ... @end`
- **Loops:** `@each(post in posts) ... @end`
- **Components:** `@component('components/button', { text: 'Submit' })`