

## TSConfig



Configuring a TypeScript  
project

# donation-hapi-03-ts

- TypeScript version of all js modules



```

  public
  homer-1.png
  homer-2.jpeg
  # main.css
  views
    partials
      coordinates.hbs
      donation-form.hbs
      donation-list.hbs
      heading.hbs
      login-form.hbs
      menu.hbs
      signup-form.hbs
      splashscreen.hbs
      user-credentials.hbs
      user-details.hbs
      donate.hbs
      layout.hbs
      login.hbs
      main.hbs
      report.hbs
      signup.hbs
  .env
  .env_example
  .eslintrc.json
  .gitignore
  .prettierrc.json
```

- Unchanged

```

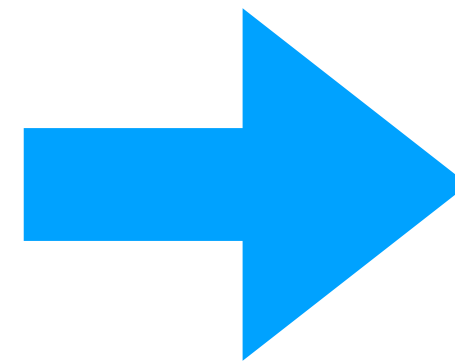
  src
    controllers
      accounts-controller.ts
      donations-controller.ts
    models
      mongo
        candidate-store.ts
        candidate.ts
        connect.ts
        donation-store.ts
        donation.ts
        seed-data.ts
        user-store.ts
        user.ts
        db.ts
    types
      donation-types.ts
      server.ts
      web-routes.ts
  package.json
  tsconfig.json
```

- Modified/new

# JavaScript -> TypeScript

```
✓ controllers
  JS accounts-controller.js
  JS donations-controller.js
✓ models
  ✓ mongo
    JS candidate-store.js
    JS candidate.js
    JS connect.js
    JS donation-store.js
    JS donation.js
    JS seed-data.js
    JS user-store.js
    JS user.js
  JS db.js
  > views
  JS server.js
  JS web-routes.js
  {} package.json
```

- Each module renamed from .js to .ts



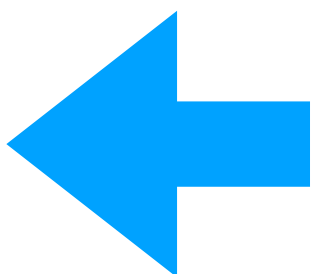
- Refactor to introduce typescript specific syntax
- donation-types introduced
- package.json modified
- tsconfig.json introduced

```
✓ src
  ✓ controllers
    TS accounts-controller.ts
    TS donations-controller.ts
  ✓ models
    ✓ mongo
      TS candidate-store.ts
      TS candidate.ts
      TS connect.ts
      TS donation-store.ts
      TS donation.ts
      TS seed-data.ts
      TS user-store.ts
      TS user.ts
    TS db.ts
  ✓ types
    TS donation-types.ts
  > views
  TS server.ts
  TS web-routes.ts
  {} package.json
  TS tsconfig.json
```

# JavaScript -> TypeScript

- package.json modified
- tsconfig.json introduced

```
✓ src
  ✓ controllers
    TS accounts-controller.ts
    TS donations-controller.ts
  ✓ models
    ✓ mongo
      TS candidate-store.ts
      TS candidate.ts
      TS connect.ts
      TS donation-store.ts
      TS donation.ts
      TS seed-data.ts
      TS user-store.ts
      TS user.ts
    TS db.ts
  ✓ types
    TS donation-types.ts
  > views
    TS server.ts
    TS web-routes.ts
  {} package.json
  TS tsconfig.json
```



# tsconfig.json

```
{
  "$schema": "https://json.schemastore.org/tsconfig",
  "display": "Node 20",

  "compilerOptions": {
    "lib": ["es2022"],
    "module": "es2022",
    "target": "es2022",

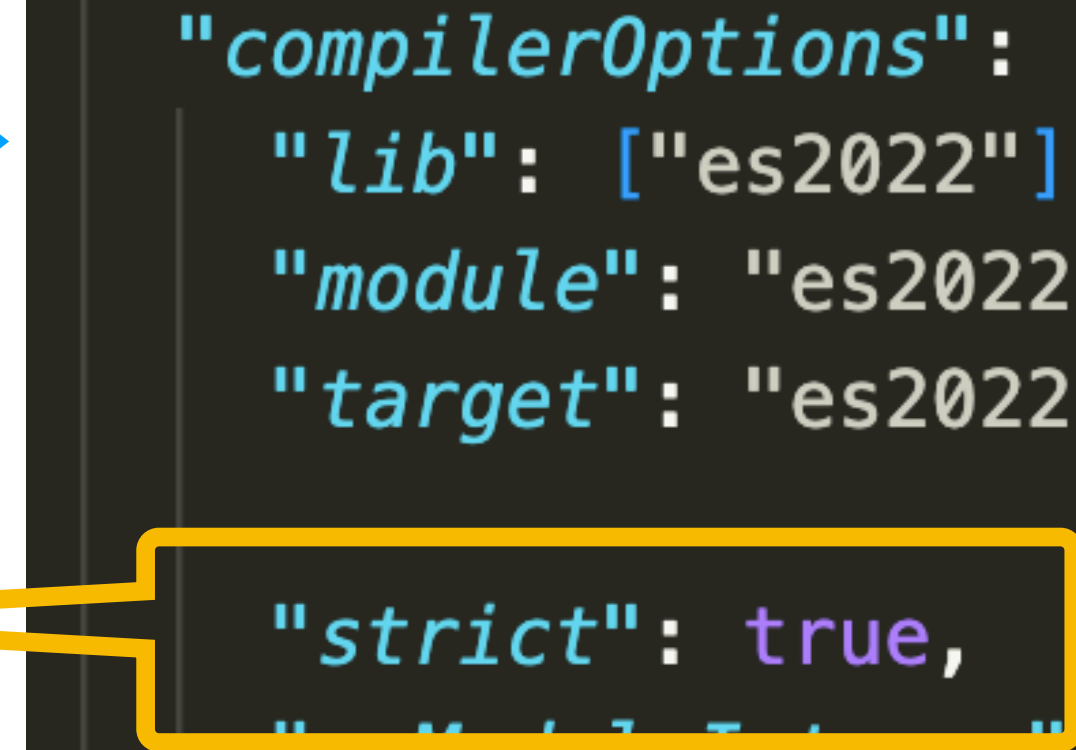
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "moduleResolution": "node",
    "rootDir": "./src"
  }
}
```



# tsconfig.json

- TypeScript usage requires a compiler - tsc
- The settings for tsc is maintained for the project tsconfig.json
- Strict “true” implies stronger static (i.e. compile time) checks.
- A shortcut to enable more fine grained options:

- noImplicitAny
- noImplicitThis
- alwaysStrict
- strictBindCallApply
- strictNullChecks
- strictFunctionTypes
- strictPropertyInitialization



```
{
  "$schema": "https://json.schemastore.org/tsconfig",
  "display": "Node 20",

  "compilerOptions": {
    "lib": ["es2022"],
    "module": "es2022",
    "target": "es2022",

    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "moduleResolution": "node",
    "rootDir": "./src"
  }
}
```

- Keep blanket “true” for code safety

# package.json: scripts

```
{
  "name": "donation-hapi-02-ts",
  "version": "0.3.0",
  "description": "",
  "main": "src/server.js",
  "type": "module",
  ▶ Debug
  "scripts": {
    "build": "tsc",
    "dev:tsc": "tsc --watch",
    "start": "npm run build && node src/server.js",
    "dev": "./node_modules/.bin/nodemon src/server.js --ignore 'src/models/json/*.json'",
    "lint": "./node_modules/.bin/eslint . --ext .js"
  },
}
```

# package.json: scripts

- Node executes .js file, **not** .ts
- **tsc**, the typescript compiler, translates .js to .ts
- TypeScript commands:

```
{
  "name": "donation-hapi-02-ts",
  "version": "0.3.0",
  "description": "",
  "main": "src/server.js",
  "type": "module",
  
  "scripts": {
    "build": "tsc",
    "dev:tsc": "tsc --watch",
    "start": "npm run build && node src/server.js",
    "dev": "npx nodemon src/server.js --ignore 'src/models/json/*.json'",
    "lint": "npx eslint . --ext .js"
  },
}
```

- **build**: run the tsc compiler on the full project

- **dev:tsc**: start the tsc compiler, running in the background

- **start**: run the build command + start the application



# package.json: @types

```
"devDependencies": {  
  "ts-node": "^10.9.2",  
  "@types/hapi__cookie": "^12.0.5",  
  "@types/hapi__inert": "^5.2.10",  
  "@types/hapi__vision": "^5.5.7",  
  "@typescript-eslint/eslint-plugin": "^6.20.0",  
  "@typescript-eslint/parser": "^6.20.0",  
  "eslint": "^8.56.0",  
  "eslint-config-airbnb-base": "^15.0.0",  
  "eslint-config-prettier": "^9.1.0",  
  "eslint-plugin-import": "^2.29.1",  
  "nodemon": "^3.0.3",  
  "prettier": "^3.2.4",  
  "typescript": "^4.9.5"  
},
```

# package.json: @types

- Many JavaScript libraries have accompanying typescript 'types'
- These provide TypeScript type information for functions and objects exported by the library
- Enables full type checking for JavaScript authored libraries

```
"devDependencies": {  
  "ts-node": "^10.9.2",  
  "@types/hapi__cookie": "^12.0.5",  
  "@types/hapi__inert": "^5.2.10",  
  "@types/hapi__vision": "^5.5.7",  
  "@typescript-eslint/eslint-plugin": "^6.20.0",  
  "@typescript-eslint/parser": "^6.20.0",  
  "eslint": "^8.56.0",  
  "eslint-config-airbnb-base": "^15.0.0",  
  "eslint-config-prettier": "^9.1.0",  
  "eslint-plugin-import": "^2.29.1",  
  "nodemon": "^3.0.3",  
  "prettier": "^3.2.4",  
  "typescript": "^4.9.5"  
},
```

- Types for node APIs + the Hapi framework

# package.json: typescript + eslint

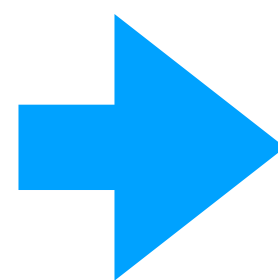
- TypeScript extension for eslint

- TypeScript compiler

```
"devDependencies": {  
  "ts-node": "^10.9.2",  
  "@types/hapi__cookie": "^12.0.5",  
  "@types/hapi__inert": "^5.2.10",  
  "@types/hapi__vision": "^5.5.7",  
  "@typescript-eslint/eslint-plugin": "^6.20.0",  
  "@typescript-eslint/parser": "^6.20.0",  
  "eslint": "^8.56.0",  
  "eslint-config-airbnb-base": "^15.0.0",  
  "eslint-config-prettier": "^9.1.0",  
  "eslint-plugin-import": "^2.29.1",  
  "nodemon": "^3.0.3",  
  "prettier": "^3.2.4",  
  "typescript": "^4.9.5"  
},
```



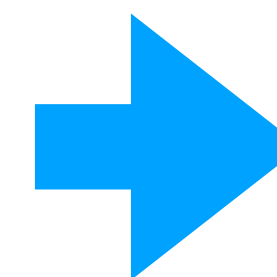
```
✓ controllers
  JS accounts-controller.js
  JS donations-controller.js
✓ models
  ✓ mongo
    JS candidate-store.js
    JS candidate.js
    JS connect.js
    JS donation-store.js
    JS donation.js
    JS seed-data.js
    JS user-store.js
    JS user.js
  JS db.js
  > views
  JS server.js
  JS web-routes.js
```



- Rename all .js to .ts

```
✓ src
  ✓ controllers
    TS accounts-controller.ts
    TS donations-controller.ts
  ✓ models
    ✓ mongo
      TS candidate-store.ts
      TS candidate.ts
      TS connect.ts
      TS donation-store.ts
      TS donation.ts
      TS seed-data.ts
      TS user-store.ts
      TS user.ts
    TS db.ts
  ✓ types
    TS donation-types.ts
  > views
  TS server.ts
  TS web-routes.ts
```

- **dev:tsc**: start the tsc compiler, running in the background
- Each .ts file will be accompanied by the .js file generated by tsc



```
✓ src
  ✓ controllers
    JS accounts-controller.js
    TS accounts-controller.ts
    JS donations-controller.js
    TS donations-controller.ts
  ✓ models
    ✓ mongo
      JS candidate-store.js
      TS candidate-store.ts
      JS candidate.js
      TS candidate.ts
      JS connect.js
      TS connect.ts
      JS donation-store.js
      TS donation-store.ts
      JS donation.js
      TS donation.ts
      JS seed-data.js
      TS seed-data.ts
      JS user-store.js
      TS user-store.ts
      JS user.js
      TS user.ts
    JS db.js
    TS db.ts
  > types
  > views
  JS server.js
  TS server.ts
  JS web-routes.js
  TS web-routes.ts
```

- src
  - controllers
    - JS accounts-controller.js
    - TS accounts-controller.ts
    - JS donations-controller.js
    - TS donations-controller.ts
  - models
    - mongo
      - JS candidate-store.js
      - TS candidate-store.ts
      - JS candidate.js
      - TS candidate.ts
      - JS connect.js
      - TS connect.ts
      - JS donation-store.js
      - TS donation-store.ts
      - JS donation.js
      - TS donation.ts
      - JS seed-data.js
      - TS seed-data.ts
      - JS user-store.js
      - TS user-store.ts
      - JS user.js
      - TS user.ts
    - JS db.js
    - TS db.ts
  - types
  - views
  - JS server.js
  - TS server.ts
  - JS web-routes.js
  - TS web-routes.ts

## VSCode settings.json

```
"files.exclude": {
  "**/.git": true,
  "**/*.js.map": true,
  "**/*.js": { "when": "$(basename).ts" }
},
```

- Hide from VSCode folders:
  - .git folder
  - .js files where a corresponding .ts file exists
  - .js.map files generated for debug purposes by tsc

- src
  - controllers
    - TS accounts-controller.ts
    - TS donations-controller.ts
  - models
    - mongo
      - TS candidate-store.ts
      - TS candidate.ts
      - TS connect.ts
      - TS donation-store.ts
      - TS donation.ts
      - TS seed-data.ts
      - TS user-store.ts
      - TS user.ts
    - TS db.ts
  - types
    - TS donation-types.ts
  - views
    - TS server.ts
    - TS web-routes.ts



## TSConfig



Configuring a TypeScript  
project