

Frontend

- Based on React and use TypeScript.
- Can add and mark todo items as completed;
- Created two components for better maintainability; One for adding todo item, another for displaying todo items.
- Added a `` and a state to the `Add Item` button for debouncing.
- All the error message will be displayed in the alert.

- Created a generic `apiClient.ts` and `HttpService.ts` to make calling APIs easier and reusable.
- The base API URL defined in `.env`
- Created two end to end tests using Playwright; Use command `npm run test:e2e` to run the tests. More related commands defined in `package.json`.

```
PS C:\Code\developer-assessment\Frontend-React> npm run test:e2e

> todolist.app@0.1.0 test:e2e
> playwright test

Running 2 tests using 1 worker

✓ 1 todoitem.test.ts:3:5 > Add a todo item and verify it appears in the list (989ms)
✓ 2 todoitem.test.ts:24:5 > Add todo item, mark it as completed, and verify the status (840ms)

2 passed (2.7s)
```

Backend

- Applied the so-called “Clean Architecture”.
- Used `Mapster` for object mapping.
- Used `FluentValidation` for business logic validation.
- Created `AuditLoggingMiddleware.cs` for logging API Request and Response.
- Created unit tests for the controller only, as do not want to go too crazy on unit tests.

Test	Duration	Items	Err
▲ ✓ TodoList.Api.UnitTests (8)	150 ms		
▲ ✓ TodoList.Api.Tests.Controllers (8)	150 ms		
▲ ✓ TodoItemsControllerTests (8)	150 ms		
✓ GetTodoItem_ReturnsNotFo...	76 ms		
✓ GetTodoItem_ReturnsOkRes...	52 ms		
✓ GetTodoItems_ReturnsOkRes...	12 ms		
✓ PostTodoItem_ReturnsBadRe...	1 ms		
✓ PostTodoItem_ReturnsCreate...	< 1 ms		
✓ PutTodoItem_ReturnsBadRe...	< 1 ms		
✓ PutTodoItem_ReturnsBadRe...	9 ms		
✓ PutTodoItem_ReturnsNoCon...	< 1 ms		