Choose any kind of item you want and like to manage, and name the types according to.

URL	Method	Authorize	Description	Request body	Response body
/api/item	GET	User/Admin	Get all user's items		list of items
/api/item/{id}	GET	User/Admin	Get a user's item by ID		item
/api/item	POST	User/Admin	Add a new item to user	item	item location
/api/item/{id}	PUT	User/Admin	Update user's item	item	
/api/item/{id}	DELETE	User/Admin	Delete user's item		
/api/user	GET	User/Admin	Get my user		user
/api/user	GET	Admin	Get all users		list of users
/api/user	POST	Admin	Add a new user	user	user location
/api/user/{id}	DELETE	Admin	Delete user + his item's		
/api/login	POST		Login	user	JWT

Server-side notes:

- Only administrators can add/delete users.
- Administrator can edit all users, including all properties. Regular user can only edit his own user details, and cannot make himself administrator.
- Users can handle only their own items, and cannot see items of other users.
- Both items and users, should be saved in a JSON file.
- Both items and users, should be accessed via an injected service through an interface, so it would be easy to change it to a DB-based app.
  Add an extension method of IServiceCollection for an easy registration.
- Each request should be written to a log file, including start date & time, name of controller & action, name of the logged-in user if there is one, and the duration of operation in milliseconds.

Client-side notes:

- The default page should show the user's items and let add / update / delete items.
- In case there is no logged in user (no token saved in local storage, or the token expired), instead of default page, user should see a login page.
- If the user has administrative privileges, there should be a link from the items list page to the users list page, and vice versa.
- Add a postman button in you login page, with a single request to login including body

Challenge:

- Allow users log-in using their own google account.