

Laboratório 8 – Angular, HttpClient e Forms

1 Exercícios

1. Acesse o serviço REST <http://jsonplaceholder.typicode.com/> e escreva uma aplicação Angular (via Angular CLI já com um módulo de roteamento adicionado) que consuma o serviço via GET, PUT, POST, PATCH e DELETE para a coleção de “todos” (coleção de tarefas) e “users” (coleção de usuários). Esta aplicação Angular deve possuir as seguintes características:

- Uma classe *service* que consome o serviço web REST através de *HttpClient* e expõe objetos *Observable* para componentes.
- *View* para consultar todas as tarefas, mostrando título e status da tarefa bem como o nome do usuário responsável por ela. Utilize uma tabela para mostrar as informações.
- *View* para consultar todas as tarefas de um determinado usuário, utilizando um *drop-down list* para a seleção do usuário desejado. Utilize uma tabela para mostrar as informações de forma semelhante à *view* de listagem completa.
- *View* para cadastrar uma nova tarefa. O formulário deve conter um *drop-down list* para a seleção do usuário desejado, caixa de texto para entrada do título da tarefa (campo obrigatório), botão para limpar o formulário e botão para submissão do formulário. A validação de dados do formulário deve estar ativa.
- *View* para editar os dados de uma tarefa. É permitido alterar o usuário responsável pela tarefa, seu título e seu status. Utilize um *check-box* para permitir a alteração do status da tarefa como completado ou não.

2. Retome a aplicação construída no exercício 1. Acrescente configurações adequadas de roteamento para cada *view* desenvolvida anteriormente. Faça com que a tela que lista todas as tarefas seja a tela principal da aplicação. Configure nesta página botões para navegar para as demais *views* (por exemplo, botões “filtrar” e “adicionar”). Altere a tabela para incluir a navegação para a *view* de edição de uma tarefa específica ao se clicar em um botão de edição disponível em cada linha da tabela.

3. Acesse o serviço REST <https://swapi.co/documentation> e consulte o repositório NPM <https://www.npmjs.com/> por implementações de bibliotecas para consumir o serviço via Angular (procure por “swapi”). Configure um novo projeto usando Angular CLI, instale a biblioteca e crie pelo menos dois componentes diferentes que consomem o serviço. Procure separar corretamente as responsabilidades entre as classes envolvidas.