Api - parte 2

Observação:

Caso não saiba o que é uma api, leia o seguinte sobre aqui.

Rest e RestFull

Rest é um modelo usado para criar api's, de maneira adequada para a web.

Recursos

Explicando por um exemplo: Imagine que tenhamos a url de um site de utilitários:

`https://utils.com/`

Quando abrimos o site no browser, fazendo uma requisição, é feita uma pesquisa no serviço chamado **DNS** (Domain Name Server), que diz qual servidor está localizada essa aplicação, assim, redirecionando para o site que nos retorna a pagina html (que o navegador traduz e nos exibe).

Se por exemplo acessarmos:

`https://utils.com/tarefas/`

Estaremos requisitando um recurso, que neste caso é o `tarefas/` que nos retorna as tarefas salvas no sistema. Logo podemos compreender que recursos são dados/informações, que por meio de requisições, podem ser cadastradas, alteradas, apagadas, e removidas.

URI's URI's, ou Uniform Resource Identifier, nada mais são do que os identificadores dos recursos, como por exemplo:

`tarefas/`

Essas URI's seguem regras, tais como:

- Cada recurso tenha sua **única** uri relacionada ao mesmo, por exemplo `tarefas/`.
- Caso os recursos sejam criados no plural (Ex.: `tarefas/`), manter **todos** no plural, ou caso seja, escolhido no singular, manter todos em singular (Ex.: `tarefa/`).
- Também **não** manter uma uri singular para cada recurso, pois, para acessar um determinado recurso, será utilizado um id.

  Exemplo do que não fazer: `tarefa/` e `tarefas/`.
- **Não** utilizar as **ações** nos nomes nas uri's, pois assim, não é possível reutilizar as mesmas.

Exemplo do que não fazer: `getTarefas/`, `alterTarefas/`, `deleteTarefas/`, `searchTarefas/`.

Para realizar requisições das ações anteriormente referidas, será utilizado os métodos HTTP, como os listados abaixo:

- **GET:** usado para acessar os dados do recurso.
- **POST:** usado para cadastrar dados no recurso.
- **PUT:** usado para alterar os dados no recurso.
- **DELETE:** usado para apagar os dados no recurso.

Assim será utilizada a mesma uri para todas as ações. Abaixo, são mostrados exemplos, no formato [**ação**] **recurso/** :

- [**GET**] **tarefas/** retorna todas as tarefas
- [**GET**] **tarefas/10** retorna a tarefa de id 10
- [**DELETE**] **tarefas/5** apaga a tarefa de id 5
- [**POST**] **tarefas/** insere dados envidados na tarefa
- [**PUT**] **tarefas/3** altera os dados contidos na tarefa

Obs.: Existem outros métodos, e uma lista deles está em: Lista de Métodos.

Respostas para as requisições Para cada requisição feita, teremos uma determinada resposta:

- **1xx:** Informações gerais
- **2xx:** Sucesso
- **3xx:** Redirecionamento
- **4xx:** Erro no cliente (quem fez a requisição)
- **5xx:** Erro no servidor (erro no servidor que manipula a requisição).

Para mais informações descrevendo a lista das resposta das requisições aqui.

Representações As requisições podem ser retornadas em alguns tipos de representações:

- html
- json
- xml

Criador

Olá me chamo Gustavo, e criei este material, para mais informações, clique no link abaixo:

- LinkTree