

Angular



Érico Rosiski Weber - 26/07/2024

Aula 3 - Conceitos básicos

- Angular Router
- API

Angular Router

Navegação entre páginas

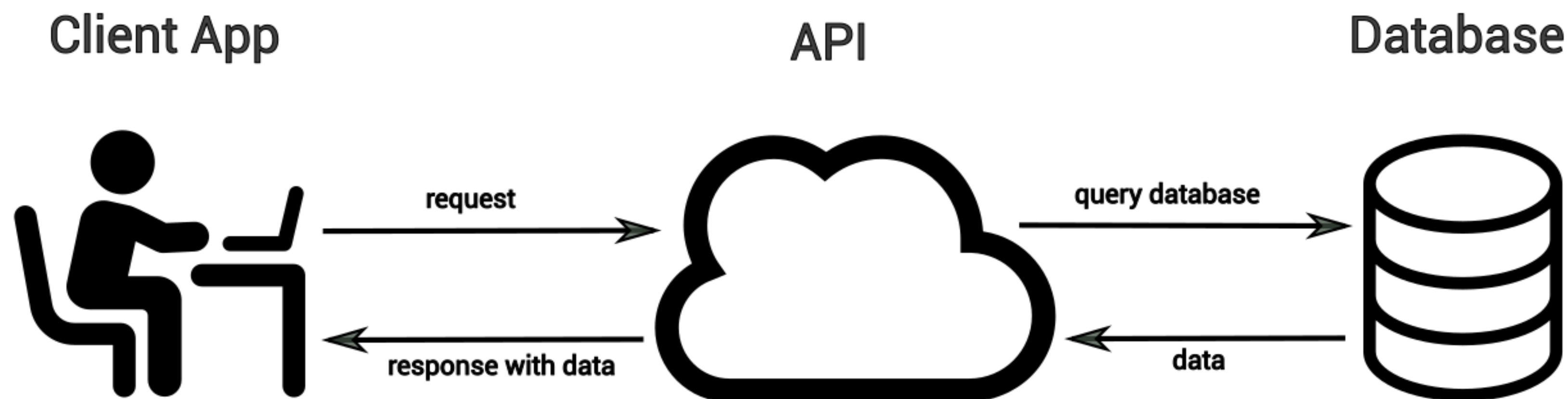
- Novo arquivo para declarar as rotas
- As rotas precisam ser criadas no template principal
- Substituímos os componentes por “<router-outlet>”

API

Requisições HTTP

Comunicação com APIs

Mensagens enviadas pelo cliente para iniciar uma ação no servidor



Exemplo de API (Application Programming Interface)

APIs são mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos.

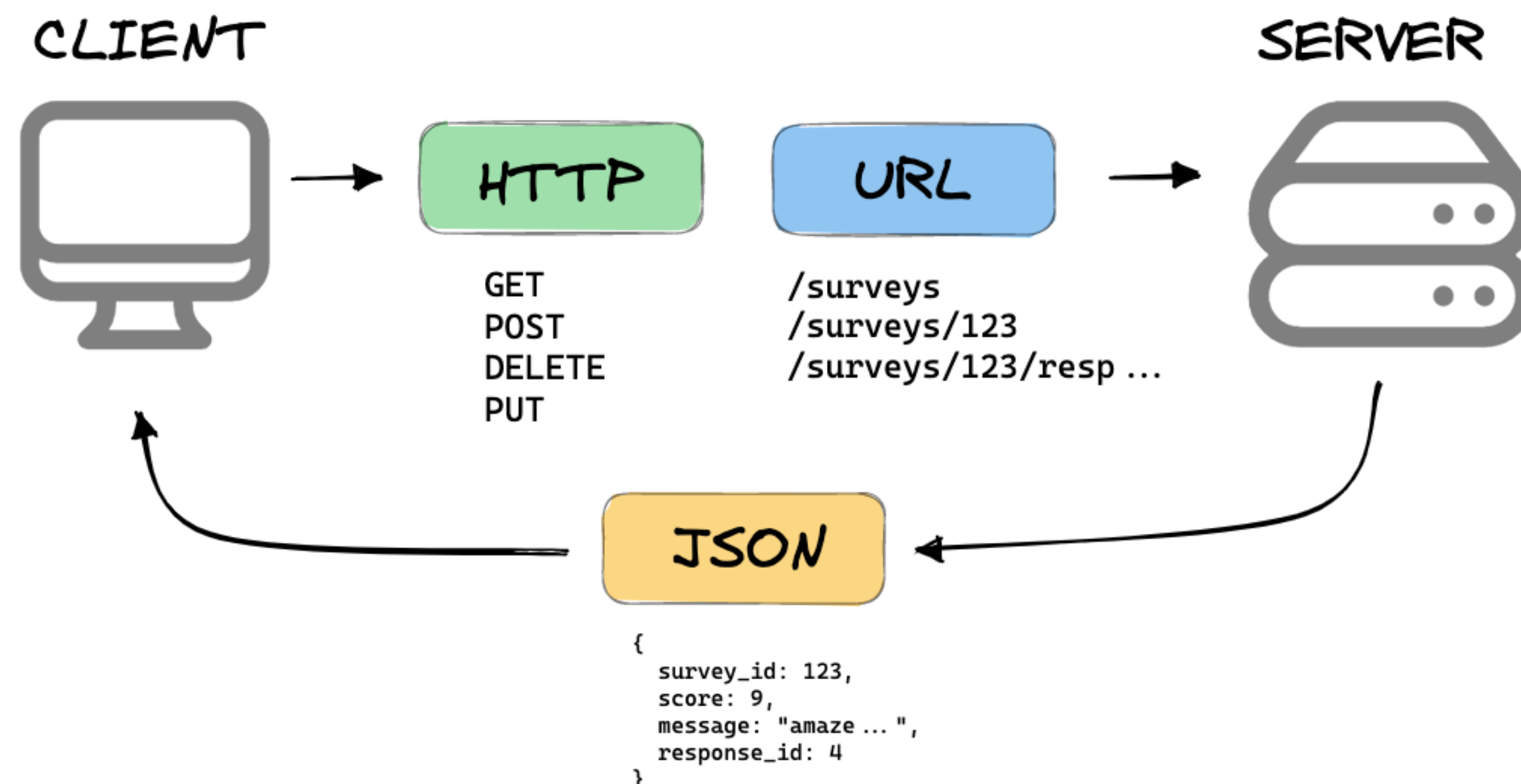
Exemplo:

O sistema de software do instituto meteorológico contém dados meteorológicos diários. A aplicação para a previsão do tempo em seu telefone “fala” com esse sistema por meio de APIs e mostra atualizações meteorológicas diárias no telefone.

APIs REST

Essas são as APIs mais populares e flexíveis encontradas na Web atualmente. O cliente envia solicitações ao servidor como dados. O servidor usa essa entrada do cliente para iniciar funções internas e retorna os dados de saída ao cliente.

WHAT IS A REST API?



Como funcionam as APIs RESTful?

1. O cliente envia uma solicitação ao servidor. O cliente segue a documentação da API para formatar a solicitação de modo que o servidor entenda.
2. O servidor autentica o cliente e confirma que o cliente tem o direito de fazer essa solicitação.
3. O servidor recebe a solicitação e a processa internamente.
4. O servidor retorna uma resposta ao cliente. A resposta contém informações que indicam ao cliente se a solicitação foi bem-sucedida. A resposta também inclui informações solicitadas pelo cliente.

Estrutura API Rest

Solicitação/Requisição:

Método + Cabeçalho HTTP + dados + parâmetros

Resposta

HTTP Status Code + Cabeçalhos + Corpo da mensagem (body)

Estrutura API Rest

Solicitação/Requisição:

Método + Cabeçalho HTTP + dados + parâmetros

Resposta

HTTP Status Code + Cabeçalhos + Corpo da mensagem (body)

Estrutura API Rest

Solicitação/Requisição

Método: Um método HTTP informa ao servidor o que ele precisa fazer com o recurso. A seguir, estão quatro métodos HTTP comuns:

GET:

Os clientes usam GET para acessar recursos localizados no URL especificado no servidor. Eles podem armazenar em cache solicitações GET e enviar parâmetros na solicitação da API RESTful para instruir o servidor a filtrar dados antes de enviar.

POST

Os clientes usam POST para enviar dados ao servidor. Eles incluem a representação de dados com a solicitação. Se enviarem a mesma solicitação POST várias vezes, criarão o mesmo recurso várias vezes.

Estrutura API Rest

Solicitação/Requisição

PUT/PATCH

Os clientes usam PUT para atualizar recursos existentes no servidor. Ao contrário do POST, o envio da mesma solicitação PUT várias vezes em um serviço da Web RESTful tem o mesmo resultado.

DELETE

Os clientes usam a solicitação DELETE para remover o recurso. Uma solicitação DELETE pode alterar o estado do servidor. No entanto, se o usuário não tiver a autenticação apropriada, a solicitação falhará.

Estrutura API Rest

Solicitação/Requisição

Os verbos/métodos HTTP mais comumente usados POST, GET, PUT, DELETE são semelhantes às CRUD operações (Create, Read, Update e Delete) no banco de dados. Segue uma comparação

1. Criar - POSTAR
2. Ler - OBTER
3. Atualização - PUT
4. Excluir - EXCLUIR

Estrutura API Rest

Solicitação/Requisição

Cabeçalhos HTTP: Os cabeçalhos de solicitação são os metadados trocados entre o cliente e o servidor.

Cabeçalho	Descrição
Tipo de conteúdo	Obrigatórios. Defina como "application/json; charset=utf-8".
Autorização	Obrigatórios. Contém o token de autenticação no formato <code>Bearer {OAuth authorization token}</code> . (Consulte a implementação do Azure Active Directory do OAuth 2.0 sobre como obter o token de autorização).
x-ms-client-request-id	Opcional. Fornece uma identidade gerada pelo cliente da solicitação na forma de um GUID. O uso desse cabeçalho é altamente recomendável para correlacionar as atividades do lado do cliente com as solicitações recebidas pelo servidor.

Estrutura API Rest

Solicitação/Requisição

Dados

As solicitações da API REST podem incluir dados para POST, PUT e outros métodos HTTP para funcionarem com êxito.

```
{
  id: "ttl231",
  title: "Pride and Prejudice"
  year: 2093,
  director: "Michael Bay",
  genres: [
    "Horror",
    "Comedy"
  ],
  stars: [
    {
      name: "Kiera Knightley",
      id: 9863
    },
    {
      name: "Danny DeVito",
      id: 2031
    }
  ]
}
```


Estrutura API Rest

Solicitação/Requisição

Parâmetros

As solicitações da API RESTful podem incluir parâmetros que fornecem ao servidor mais detalhes sobre o que precisa ser feito. A seguir, estão alguns tipos diferentes de parâmetro.

- Parâmetros de caminho que especificam detalhes do URL.
- Parâmetros de consulta que solicitam mais informações sobre o recurso.
- Parâmetros de cookies que autenticam clientes rapidamente.

Estrutura API Rest

Solicitação/Requisição

Parâmetros

As solicitações da API RESTful podem incluir parâmetros que fornecem ao servidor mais detalhes sobre o que precisa ser feito. A seguir, estão alguns tipos diferentes de parâmetro.

- Parâmetros de caminho que especificam detalhes do URL.
- Parâmetros de consulta que solicitam mais informações sobre o recurso.
- Parâmetros de cookies que autenticam clientes rapidamente.

Exemplo Básico

GET `/articles?include=author` HTTP/1.1

HTTP/1.1 200 OK

Content-Type: `application/vnd.api+json`

```
{
  "data": [{
    "type": "articles",
    "id": "1",
    "attributes": {
      "title": "JSON:API paints my bikeshed!",
      "body": "The shortest article. Ever.",
      "created": "2015-05-22T14:56:29.000Z",
      "updated": "2015-05-22T14:56:28.000Z"
    },
    "relationships": {
      "author": {
        "data": {"id": "42", "type": "people"}
      }
    }
  }],
  "included": [
    {
      "type": "people",
      "id": "42",
      "attributes": {
        "name": "John",
        "age": 80,
        "gender": "male"
      }
    }
  ]
}
```

Estrutura API Rest

Solicitação/Requisição:

Método + Cabeçalho HTTP + dados + parâmetros

Resposta

HTTP Status Code + Cabeçalhos + Corpo da mensagem (body)

Estrutura API Rest

Resposta

Cabeçalho = Igual Requisição

Body

Resposta Json

```
{
  "metainfo": {
    "status": "Finished",
    "timestamp": "2016-02-22 09:27:08",
    "service_url": "http://qed.epa.gov/rest/hwbi/locations/GA/Clarke"
  },
  "inputs": [
    {
      "state": "georgia",
      "county": "clarke",
    }
  ],
  "services": [
    {
      "ServiceID": "S01",
      "ServiceName": "Capital Investment",
      "Description": "Capital Investment",
      "ServiceType": "Economic",
      "Score": 0.5858767309
    },
    {
      "ServiceID": "S02",
      "ServiceName": "Consumption",
      "Description": "Consumption",
      "ServiceType": "Economic",
      "Score": 0.5130107144
    }
  ],
}
```

Estrutura API Rest

Resposta

HTTP Status Code

Os códigos de status de resposta HTTP indicam se uma solicitação HTTP específica foi concluída com sucesso. As respostas são agrupadas em cinco classes:

1. Respostas informativas (100– 199)
2. Respostas bem-sucedidas (200– 299)
3. Mensagens de redirecionamento (300– 399)
4. Respostas de erro do cliente (400– 499)
5. Respostas de erro do servidor (500– 599)

HTTP STATUS CODES

2xx Success

200**Success / OK**

3xx Redirection

301**Permanent Redirect****302****Temporary Redirect****304****Not Modified**

4xx Client Error

401**Unauthorized Error****403****Forbidden****404****Not Found****405****Method Not Allowed**

5xx Server Error

501**Not Implemented****502****Bad Gateway****503****Service Unavailable****504****Gateway Timeout**

Material Extra

[Vídeo: O que é uma API](#)

Exercícios

CRIAR UM PROJETO ANGULAR NOVO

Exercício 1: Vinculação de dados

Crie um componente AngularJS que vincule dados de um controlador ao front end. Exibe uma lista de itens com a diretiva ng-repeat.

Exercício 2: Tratamento de Eventos

Implemente um componente AngularJS que lide com as interações do usuário, como clique em botão ou envio de formulário. Use as diretivas ng-click e ng-submit para manipulação de eventos.

Exercício 3: vinculação de dados bidirecional

Crie um formulário com campos de entrada que demonstrem a ligação bidirecional de dados entre a visualização e o modelo. Use a diretiva ng-model para ligação bidirecional.

Exercício 4: Diretivas Condicionais

Desenvolva uma diretiva AngularJS personalizada que encapsule uma parte reutilizável da lógica da UI. Implemente a diretiva em um componente front-end.

Exercícios

Instalar Postman ou Insomnia

1. Escolher uma das APIs do artigo:

18 Fun APIs For Your Next Project

2. Utilizar o programa postman ou insomnia para realizar um request de get e receber as informações em JSON

Próxima Aula

- Correção dos exercícios
- Chamar uma API através de um service no Angular :D

Avalie essa aula:
Formulário avaliação da Aula