

## HTTP Get Aluracarheroku

### Transcrição

[00:00] Agora que a gente já viu um pouco sobre a nova arquitetura que a gente quer utilizar na nossa solução, a gente tem aqui a nossa lista fixa de veículos. Olha só, a gente tem uma listagem com quatro veículos. Esses quatro veículos estão sendo utilizados na nossa listagem, na nossa view de listagem.

[00:23] Então, a gente vai substituir essa lista, por uma lista vazia, porque inicialmente ela vai ser vazia, mas num outro momento, ela vai ser usada para carregar dados que vão estar num outro servidor. A gente vai carregar isso via web. Então, o que a gente vai fazer é ir lá na nossa listagem viewmodel...

[00:45] Que está consumindo essa lista fixa de veículos, a gente vai trocar essa referência aqui, para essa lista fixa e a gente vai simplesmente instanciar uma lista vazia de veículos. Então, eu vou colocar `new List();`. E a gente vai ter aqui uma listagem vazia.

[01:06] Agora a gente vai rodar a aplicação e ver o que acontece. Provavelmente vai vir uma lista vazia, se vier alguma coisa diferente, é porque realmente tem alguma coisa muito estranha. Então, aqui carregando a aplicação. Legal, agora carregou a primeira página, que é a página de listagem...

[01:24] E realmente não trouxe nenhum veículo. Agora, o segundo passo é acessar algum serviço que vai estar disponível para a nossa aplicação, que vai ser utilizada para carregar a listagem. Como que a gente faz isso? Bom, felizmente, alguém já programou para a gente um web service REST.

[01:49] Que vai fornecer para a gente essa lista de veículos. Então, a gente vai acessar aqui o endereço, aqui no browser. Eu vou colocar `aluracar.herokuapp.com` e nesse endereço a gente vai obter a listagem atualizada de veículos.

[02:06] Bom, então essa listagem vai vir para a nossa aplicação e como a gente vai fazer isso? Bom, primeiro a gente tem aqui no browser algumas informações, primeiro tem o endereço que a gente requisitou, aqui tem o endereço na barra de endereços, `aluracar.herokuapp.com`.

[02:31] E o browser respondeu essa chamada Http no protocolo Http da internet, respondeu com esse resultado. Esse resultado significa uma lista de veículos que está sendo retornada por esse endereço no servidor. Então, o servidor recebeu essa requisição e respondeu com essa listagem.

[02:59] Agora, o que é essa listagem exatamente? Essa listagem está num formato aqui chamado Json, que significa, Javascript Simple Object Notation ou Notação de Objetos Simplificada em Javascript. Então, nesse formato Json, para quem não conhece, ele está começando aqui com um colchete.

[03:33] Ele começa com colchetes aqui e termina com colchetes aqui. Então, o colchete na anotação Json, significa que eu estou recebendo uma lista e essas chaves aqui, que estão aqui no meio, começa aqui, termina aqui, começa aqui de novo, termina aqui, começa aqui, termina aqui. A gente tem várias dessas.

[03:54] Cada uma dessas chaves, significa um objeto. Então, dentro desse objeto a gente tem propriedades. Então, eu tenho nome, preço, depois no outro objeto, eu tenho nome, preço, nome, preço, nome, preço e ele faz isso até o último elemento da lista.

[04:18] Então, cada objeto desse representa um veículo que o servidor está trazendo, que ele está respondendo para a nossa aplicação. E aqui, entre os nomes, nome e o preço, eu tenho os valores. Olha só, eu tenho o valor da propriedade

nome, que é o Azera V6, eu tenho oitenta e cinco mil, que é o preço dele.

[04:39] Depois eu tenho o nome, eu tenho o valor aqui, que é o Onix 1.6, o preço, eu tenho trinta e cinco mil, tá? Então, perceba que as propriedades estão separadas por vírgula. A gente tem as propriedades separadas por vírgula e entre essas vírgulas, a gente tem aqui a propriedade...

[04:59] O nome da propriedade e o valor. Então, o nome da propriedade e o valor são separados por dois pontos, então tudo isso faz parte da sintaxe do Json. Então, agora a gente já sabe como acessar esse serviço, que é um serviço Http rest, por que rest?

[05:23] Bom, se a gente olhar aqui, eu vou entrar aqui na ferramenta do desenvolvedor com o F12 e olha só, eu vou fazer a requisição novamente, só que eu vou selecionar a aba Network aqui do browser. Vou dar F5 para carregar de novo e olha só. Aqui em baixo, ele exibe a minha requisição.

[05:45] E aqui no cabeçalho da requisição, eu tenho aqui o request, o RL, que é o endereço que eu estou chamando para obter o serviço e aqui eu tenho o método de requisição, que é o método get. E aqui em baixo o status code, que é o código do status, retornando ok, com o código 200.

[06:14] Ou seja, ok, significa que ele retornou com sucesso, mas o que eu quero mostrar aqui, na verdade, é que eu estou acessando o serviço através de uma URL e recebendo o resultado e essa requisição foi feita utilizando o get. Então, a gente tem o get para fazer essa requisição.

[06:37] Então, um serviço rest recebe alguns verbos, então a gente tem o get... tem o get que é para a gente fazer geralmente uma busca de dados, uma consulta de dados e a gente tem, por exemplo, o post, que é um outro tipo de verbo do http, que é utilizado para salvar dados.

[07:01] E aqui, na outra aba do response a gente tem os dados aqui. Então, a gente tem aqui exatamente o resultado que está aparecendo aqui no corpo da página. A gente tem esse resultado, que foi obtido através da nossa requisição. Agora, você deve estar se perguntando.

[07:23] Como que a nossa aplicação vai acessar esse endereço, para obter esse resultado, pegar essa lista que está nesse formato Json e consumir isso e transformar isso numa lista. Será que a nossa aplicação vai ter que acessar o browser?

[07:40] É claro que não, a nossa aplicação vai ter que utilizar algum outro mecanismo que vai consumir, vai fazer a requisição, através do protocolo Http. E é isso que a gente vai ver agora.