10

PostAsync, Json, Sucesso, Command e Validation

Transcrição

[00:00] Agora que a gente fez o post e conseguiu tratar a falha do agendamento e mostrar para o usuário que teve uma falha, agora a gente vai fazer a parte em que o usuário consegue postar os dados no serviço com sucesso. Então, para isso a gente vai ter que pegar os dados do usuário que ele preencheu na tela...

[00:22] E jogar isso lá para o serviço. Então, como que a gente faz? A gente vai ter que pegar e passar aqui no string content, que é o corpo, que vai ser o corpo da mensagem, o corpo da requisição post e vai ter que jogar nesse corpo um Json, um formato Json correto, como esperado pela aplicação.

[00:46] Então a gente olha aqui nessa imagem, a gente está vendo quais são os dados que a aplicação aceita no corpo do Json, no corpo da requisição. Então, eu tenho: nome, fone, email, carro, preço e agendamento. Então, a gente vai colocar isso no código, então como a gente faz?

[01:05] A gente vai primeiro ter que criar uma string Json a partir de um objeto. Então, para isso, a gente vai utilizar aqui um objeto do tipo anônimo e a gente vai utilizar a biblioteca Newtonsoft, que é ótima para fazer essas conversões entre Json e objetos.

[01:27] Então vamos declarar aqui uma variável chamada Json e aí, a gente utiliza o JsonConvert, que é a classe de conversão e a gente vai serializar um objeto, um objeto do tipo anônimo. Então lá dentro a gente vai colocar o quê? O nome, a gente vai colocar o fone, vai colocar o Email e vai colocar também o carro e o preço desse veículo.

[01:57] Então, a gente vai colocar aqui o carro e o preço do veículo. Então, aí no final, a gente vai passar também a data do agendamento, não é isso? Então, DataAgendamento e lá a gente passa a data do agendamento que está aqui no viewmodel. Só que olha só, a data do agendamento que a gente tem aqui...

[02:22] A gente não tem só data, a gente também tem a hora, então para isso, a gente vai ter que fazer uma coisa diferente, tá? A gente vai ter que pegar a data e agora e transformar numa coisa única, numa variável única. Então, eu vou declarar aqui em cima uma dataHora de agendamento, que vai conter os dados de ambas as variáveis.

[02:44] Então, dataHoraAgendamento vai ser o novo system date time. Então, é o novo DateTime e lá dentro a gente vai passar o quê? O ano, o mês, o dia, hora, minutos e segundos. Então, a gente pega primeiro da DataAgendamento.Year, que é o ano, a gente passa também o mês, que é month...

[03:13] E a gente passa também o dia. E agora, da hora do agendamento, a gente vai pegar horas, minutos e segundos. Então: HoraAgentamento.Hours e aí, também vai pegar os minutos e vai pegar também os segundos. Então, como isso, a gente tem aqui um tipo que combina as duas variáveis, combina as duas propriedades.

[03:47] A gente vai passar isso aqui na data agendamento e tem agora um Json a partir desse objeto. Esse Json, a gente vai passar aqui no string content, que vai vir aqui no corpo da requisição. Então, com isso, agora a gente vai poder fazer uma chamada com os dados completos.

[04:07] Então vamos lá, rodando a aplicação, agora está abrindo a aplicação. Vou escolher aqui um veículo, vou clicar no próximo e aqui vou colocar fulano de tal, vou colocar aqui um telefone, vou colocar o Email e aí, vou selecionar um agendamento, uma data, dia 14 e aqui um horário qualquer.

[04:42] E agora vou clicar em agendar. Aí, eu confirmo o agendamento, “sim” e olha só, “Agendamento salvo com sucesso!”, então a gente conseguiu fazer o ciclo completo, desde a confirmação do usuário, até o envio dos dados que foram preenchidos pelo formulário, a gente enviou os dados lá para a nossa requisição do serviço...

[05:10] Ele respondeu com sucesso e a gente exibiu isso para o usuário. Bom, então o que a gente viu aqui de requisição post para o servidor, para poder gravar o agendamento? A gente viu isso aqui. A gente fez a requisição, fez a requisição post passando aqui a URL do nosso serviço que vai aceitar os dados do agendamento.

[05:39] E aqui no corpo da requisição post, a gente passou o Json com os dados que o usuário preencheu. Então, ele preencheu os dados aqui, preencheu e aí, ele mandou para o servidor e o servidor respondeu. E aí, no caso do sucesso, a gente tem a resposta que é um http 200.

[06:06] Por exemplo, o http dois alguma coisa, no caso, a gente está recebendo o http 200. Então, com esse sucesso, a gente exibe aqui a mensagem: “Ok, você conseguiu agendar o seu test drive com sucesso”. Agora, o que acontece, por exemplo, quando o nosso usuário não preenche os dados?

[06:26] Quando ele não preencheu o nome, nem o fone, nem o Email, ele preencheu só a data e a hora, ou seja, quando ele não preencher algum campo obrigatório. O que acontece? Ele vai clicar no botão, ele consegue clicar no botão e aí é feita a requisição. E aí, vai lá para o servidor, o servidor vai falar:

[06:48] “Opa, tem alguma coisa errada, tem algum dado faltando aí”, então, ele responde com erro. Então, a nossa aplicação vai pegar o erro depois que o servidor processou. Agora, não faz sentido a gente mandar para o servidor uma informação inconsistente, que a gente poderia já tratar no client.

[07:09] A gente já poderia tratar dentro do formulário essa validação de campos obrigatórios que não foram preenchidos. Então, a gente tem que mudar essa estratégia, para a gente poder fazer a validação do lado do cliente. Agora, como a gente vai fazer essa validação no nosso aplicativo?

[07:29] Bom, como a gente viu antes, a gente utiliza para o botão, para a gente executar uma ação através do botão, a gente utiliza um comando, um command. Então, a gente pode utilizar o command dentro do Xamarin Forms para fazer essa validação. E aí, o que acontece?

[07:49] Se o formulário estiver preenchido corretamente, o botão vai ficar habilitado, se tiver algum campo obrigatório faltando a gente pode desabilitar esse botão e aí o usuário não consegue enviar os dados que estão inconsistentes. Então, a gente vai lá na declaração do command.

[08:06] Então a gente vai procurar aqui command, onde a gente declarou o agendar command, a gente vai ter uma opção aqui, eu coloco uma vírgula, eu vou ter uma opção chamada canExecute. Então, o primeiro parâmetro aqui do comando é o execute, que é uma ação.

[08:34] E o segundo é uma função anônima que é do tipo boolean, ou seja, ele vai retornar um valor indicando se pode executar ou não. Então, aqui é o lugar ideal para a gente colocar a nossa validação. Então, eu coloco aqui uma função anônima, vou colocar uma expressão lambda aqui.

[08:57] E aí, eu vou retornar um valor que indica se está válido ou não. Então, por padrão, por (de fo), esse retorno é sempre verdadeiro. Então, por isso, que o botão está sempre habilitado. Agora, a gente vai trocar esse verdadeiro por uma condição e nessa condição o nome do cliente, o nome do usuário não pode estar vazio.

[09:22] Então this.Nome não pode ser nulo, nem vazio. Então, negativo. Então, não pode ser uma string nula ou vazia, o nome do cliente, o nome do usuário. Além disso, também não pode ser uma string nula, vazia, o quê? O telefone, porque eu preciso do telefone para poder fazer o agendamento.

[09:49] E também não pode ser uma string nula, vazia o quê? O Email dele, porque eu preciso comunicar o usuário. Então, aqui faltou o & comercial. Então, agora a gente tem uma condição aqui. Então, esses três campos, eles têm que estar preenchidos, senão o nosso botão não vai ficar habilitado.

[10:15] Agora, só isso não é suficiente para o botão responder essa condição, isso é uma característica do Xamarin Forms. Eu tenho também que colocar em cada uma das propriedades que estão entrando nessa condição, que é o nome, fone e Email, eu tenho que entrar nessas propriedades...

[10:36] E colocar uma notificação, quando essas propriedades foram alteradas, eu preciso notificar o Xamarin Forms que eu tenho que mudar, que eu tenho que atualizar a aparência do botão, para ver se ele vai ficar habilitado ou não. Então, sem isso, a nossa estratégia de habilitar, de validar o formulário através do comando não vai funcionar.

[11:03] Então, para isso, eu vou lá no set do nome, então, entro aqui e coloco o primeiro OnPropertyChanged. Então, pelo visto, OnPropertyChanged não existe aqui no nosso viewmodel do agendamento, por quê? Porque a gente não implementou ainda. Então, olha só, viewmodel não tem o método OnPropertyChanged.

[11:30] E a gente consegue resolver isso como? Simplesmente herdando do base viewmodel. Então o base viewmodel, eu vou colocar aqui, é uma classe abstrata que a gente já viu antes, que ela contém OnPropertyChanged. Então, agora sim, eu posso utilizar OnPropertyChanged aqui na validação do nome.

[11:51] Então, eu passo aqui... não passo nenhum parâmetro, abro e fecho parênteses, agora tem que pegar o comando, o agendamento command, agendar command, aliás. E tenho que fazer o quê? Eu tenho que chamar o método ChangeCanExecute. Só que, lógico, o agendar command...

[12:16] Ele é um ICommand, ele é uma interface, ele não foi declarado como um command, então eu preciso fazer o quê? Eu preciso converter, aqui eu vou fazer uma conversão para command. E aí, sim, eu consigo chamar o ChangeCanExecute. Agora, o que eu vou fazer?

[12:38] Eu vou copiar essas duas linhas, para as outras propriedades que também são obrigatória, que são o fone, certo? E o Email também. Agora a gente vai rodar a aplicação e ver como que é o comportamento dessa estratégia de validação.

[13:01] Agora está rodando o aplicativo, vai carregar a lista, vou selecionar aqui um veículo. Clico no próximo e vou sem preencher nada, vou tentar clicar no agendar. Então, opa, olha só, o agendar está desabilitado, por quê? Porque aqueles campos que são obrigatórios, eles não foram preenchidos ainda.

[13:24] Então, eu vou preencher agora o nome. Então vamos lá: “fulano de tal”, ok, vou clicar... Opa, ainda não está habilidade, por quê? Porque não preenchi nome, nem Email. Então, eu vou colocar aqui... perdão, não preenchi o fone, nem o Email, então vou preencher aqui o fone.

[13:47] Também não está habilitado, agora vou preencher o Email: “fulano@gmail.com”. Dou “ok”. Agora sim, então agora apareceu o agendar para a gente. O agendar está habilitado. Então, vou clicar nele, agora consigo salvar o agendamento. Clico aqui. Muito bem.

[14:07] “Agendamento. Falha ao agendar o test drive! Verifique os dados e tente novamente mais tarde!”. Vou tentar de novo, “Agendamento salvo com sucesso”. Então, agora a gente conseguiu fazer todo esse ciclo do preenchimento do formulário. A gente já tem a validação...

[14:31] Uma validação muito simples, que simplesmente desabilita o botão, se os campos obrigatórios não estiverem preenchidos. A gente consegue fazer a requisição utilizando http post, a gente pega o resultado da requisição e verifica se a resposta foi bem-sucedida ou não...

[14:50] E a gente exibe a resposta de acordo com o resultado, a gente exibe para o usuário, indicando se teve sucesso ou não. Bom, então é isso aí gente, a gente conseguiu nessa aula, fazer com que a nossa aplicação deixasse de ficar isolada ali no cantinho dela.

[15:10] A gente conseguiu conectar a nossa aplicação com o mundo, acessando serviço para buscar lista de veículos, salvar agendamento. Então a aula foi muito intensa, foi muito produtiva. Espero que vocês tenham gostado bastante. Muito obrigado e até a próxima.