01

Requisição http

Transcrição

[00:00] A próxima funcionalidade que nós vamos implementar agora no nosso aplicativo é o cálculo da média dos alunos. Quando nós cadastramos os alunos, tem um campo que nós utilizamos para preencher a nota, então vou vir, por exemplo, no João, ele tem a nota 8. Agora nós vamos utilizar todas essas notas que nós cadastramos dos alunos para calcular a média.

[00:20] Mas vamos pensar como nós podemos calcular a média desses alunos? Como nós vimos, o Core Data nos retorna uma lista que tem todos os alunos. Então nós poderíamos, por exemplo, fazer um for, percorrer toda essa lista, pegar as notas dos alunos e depois somar essas notas e dividir pelo número de elementos que tinha dentro do array, ou seja, pelo número de alunos, nós tiramos uma média simples.

[00:45] O problema dessa abordagem é que, por exemplo, se o aplicativo tiver 100 mil alunos, um milhão de alunos, não é indicado deixar esses processamentos matemáticos ou processamentos muito pesados para parte do app, para parte mobile fazer. O mais comum é o quê? Enviarmos essas informações, um servidor processar, calcular a média e devolver. E o iOS, o Android ou qualquer outra parte mobile, apenas exibir essa informação, então nós evitamos deixar essa parte de processamento para o iOS ou para o Android fazer.

[01:21] Então é o que nós vamos fazer agora, vamos aprender como utilizamos requisições http, ou seja, mandar essas informações pro servidor para mostrar depois as informações da média dos alunos. Então vamos fazer isso.

[01:35] Para implementarmos essa funcionalidade, nós vamos utilizar esse botão aqui, esse livro que tem aqui, que nós ainda não usamos. Nós precisamos criar uma ação para esse botão. Vamos abrir o storyboard e nós podemos criar a ação. Selecionei, agora vou dividir a tela e nós podemos criar uma ação para esse botão.

[02:02] Aqui embaixo eu vou dar um espaço, vou segurar o “Ctrl” e puxar para cá. Em “Connection” eu vou mudar para “Action” e em “Type” eu vou mudar para “UIBarButtonItem”, e agora nós podemos colocar o nome. Eu vou colocar, por exemplo, “button”, que é o nome do componente visual, seguido do que ele vai fazer, então “CalculaMedia” e vou dar um connect. Agora nós já temos então um botão onde nós vamos chamar essa funcionalidade.

[02:33] Para implementar toda essa requisição que nós vamos criar, eu vou criar uma nova classe aqui em “Home” mesmo, eu vou criar um novo grupo que eu vou chamar de “Model” e aqui dentro eu vou criar uma nova classe, que vai se chamar “CalculaMediaAPI”. Vou dar um next e um create.

[02:55] Aqui dentro eu vou criar um método então, “func calculaMediaGeralDosAlunos”. Como nós vamos fazer uma requisição para um servidor, a primeira coisa que nós temos que fazer é criar uma variável para fazer essa requisição, então “var requisicao” é do tipo “URLRequest”, e eu vou inicializá-la passando uma URL, mas nós não temos ainda nenhuma URL, não é? Então vamos precisar criar. Mas o que é essa URL?

[03:28] Para acessarmos o servidor, nós precisamos saber qual é o endereço em que nós vamos mandar essas informações dos alunos. Normalmente é o pessoal de back-end que faz nas empresas, e eles nos disponibilizam a documentação da API ou do webservice para verificarmos quais os parâmetros que temos que mandar, qual é o método da requisição e tudo mais.

[03:51] No nosso caso eu tenho uma imagem do postman, que é a que vamos seguir para implementar essa requisição. O primeiro ponto importante que nós temos que ver é de onde a vamos acessar esse servidor da Caelum. Então eu tenho aqui uma URL em que eu vou acessar esse serviço e é ela que nós vamos utilizar lá no swift. Ela é o seguinte,

"https://www.caelum.com.br/mobile" (https://www.caelum.com.br/mobile%E2%80%9D). É essa URL que nós vamos utilizar. Então vamos passar para lá.

[04:22] Vou criar a URL aqui, "let url", como ele espera uma URL do tipo URL mesmo, nós vamos criar aqui uma variável do tipo "URL". E eu vou inicializá-la passando uma string, então em "string" eu vou passar a mesma coisa que está aqui, "https://www.caelum.com.br/mobile" (https://www.caelum.com.br/mobile%E2%80%9D).

[04:59] Agora que nós já temos a URL, podemos utilizá-la aqui dentro, então vou passar a URL. Só que repara que ela está como optional, porque não é certeza que a classe URL vai conseguir converter essa string que nós passamos em URL, ele não sabe. Então nós vamos tratar isso, vamos extrair esse valor. Como vamos fazer isso? Através de um "guard let".

[05:20] Vou pôr aqui "guard let url", e ele vai tentar converter. Se não der certo eu dou um "return", porque não faz sentido nós prosseguirmos com as nossas implementações se ele não conseguir criar a URL. Agora sim eu posso utilizar a URL que nós extraímos com o "guard let".

[05:44] Agora nós já utilizamos esse endereço, que é o que nós vamos enviar os nossos alunos. Só que vamos pensar, como nós fazemos para enviar esses alunos? Tem um padrão que nós temos que seguir. E está aqui na documentação. Se nós repararmos, nós temos um JSON, em que logo de cara já dá para percebermos que é um dicionário, porque ele tem uma key e um valor, em que key é uma string que se chama "list" e um array de alunos. Então nós vamos ter que criar essa estrutura e enviar pro servidor.

[06:16] O primeiro passo nós vamos fazer essa parte de dentro, onde temos as informações dos alunos que nós vamos mandar para lá. Então vamos fazer isso. Eu vou criar aqui um dicionário, "var dicionarioDeAlunos" e eu vou inicializá-la já aqui.

[06:34] Nós temos que copiar exatamente igual está na documentação, senão na hora em que formos fazer a requisição, ele vai dar algum erro indicando que nós passamos os parâmetros errados. Eu tenho que passar aqui o id, o nome, o endereço, vamos fazer isso?

[06:50] Eu tenho aqui o "id", que vai ser, no caso, "1". Temos lá o "nome", aqui eu vou passar o meu nome. Temos o "endereco", vou passar o nome de alguma rua, "Rua blablabla". Temos o próximo, que é o telefone, o site e a nota. Então temos o "telefone", o "site", então vírgula. Sempre depois de cada linha nós temos que pôr vírgula. O "site", vou pôr aqui "www.alura.com.br" (http://www.alura.com.br%E2%80%9D). E temos a "nota", eu vou pôr um valor para testarmos, por exemplo, "8". Então criamos um dicionário igual, está aqui com essas informações.

[07:46] Nós já terminamos de configurar os parâmetros? Ainda não. Repara que nós temos que criar essa estrutura de cima, é uma hierarquia que temos que seguir. Os alunos, essa estrutura que nós criamos, está dentro de uma lista, então precisamos colocar essa estrutura que nós criamos dentro de uma lista. Vamos criar aqui um array para isso. Vou criar "var listaDeAlunos" é do tipo "Array".

[08:14] Só que dentro desse array nós vamos ter o dicionário, que é essa estrutura que nós criamos aqui, isso é um dicionário. Então ele é um dicionário, "Dictionary" de "String", e aqui podemos passar um "Any", porque nós não sabemos se toda vez vai ser "String" aqui.

[08:34] Agora que nós já temos o array, vamos utilizá-lo. Então "listaDeAlunos.append" e eu vou passar o quê aqui? Eu vou passar o dicionário que nós criamos, então "dicionario", agora eu tenho que convertê-lo para o tipo do dicionário que ele está esperando, que é um dicionário de "String:Any", não é isso que nós criamos aqui em cima? "String:Any". Então eu adicionei esse dicionário na nossa lista.

[09:06] Feito isso, nós precisamos dessa próxima estrutura, que vai ser realmente o JSON com todas as informações que nós fizemos até agora. Vou criá-lo aqui embaixo, "var json" é do tipo "Dictionary" em que vai ter uma "String" e aqui vamos colocar "AnyObject". E eu vou inicializá-lo vazio.

[09:30] Agora precisamos criar essa estrutura que nós ainda não fizemos, então vou chamá-lo aqui, "json =". O primeiro passo é colocarmos o "list", não é isso que ele tem aqui? "list:", e ele tem um array aqui dentro, então vou pôr aqui um array. Qual vai ser aqui? Vai ser o "aluno", não é isso que está aqui? Então eu vou passar "aluno" e o valor dela vai ser a lista que nós criamos, então a "listaDeAlunos", tudo isso dentro de um array.

[10:13] Agora nós terminamos de configurar os parâmetros para utilizarmos na nossa requisição. Qual é o próximo passo? O próximo passo é terminarmos de configurar a requisição, nós temos aqui a requisição e ainda não terminamos de configurá-la, inclusive eu vou passá-la para baixo, para mexermos nela agora. Embaixo do "json" eu vou passar a requisição.

[10:37] Aqui ele está reclamando porque nós utilizamos o tipo "AnyObject", então podemos utilizar o "Any", que nós criamos o nosso JSON. Agora temos a requisição, precisamos falar para requisição que nós vamos enviar esses parâmetros. Como nós fazemos isso? Nós chamamos a "requisicao.httpBody", é onde vamos passar corpo da requisição, que nada mais é do que esses parâmetros.

[11:05] Só que repara que ele está esperando uma variável do tipo "Data" e o que nós temos não é data, nós temos um dictionary, então precisamos converter o nosso dicionário no tipo data. Como nós fazemos isso? Da seguinte maneira, eu vou chamar aqui "let dicionario = JSONSerialization.data", exatamente isso que nós precisamos. Nós vamos passar o quê? O "json" que nós temos. E aqui eu não vou precisar de nenhuma opção.

[11:39] Feito isso, ele transformou para nós o dicionário em data. Só que ele é uma throw function, então nós precisamos tratar isso, porque pode ser que dê algum problema na hora que ele for converter, e nós tratamos isso dentro de um "do", então vou pôr aqui "do", ele vai tentar converter tudo isso daqui. Se ele conseguir, nós seguimos com a implementação de converter o dicionário. Vou passá-lo para dentro e se não der certo, ele vai cair no "catch", então nós vamos saber qual é o erro, vou dar um "print(error.description".

[12:15] Agora sim, eu vou chamar a "requisicao" que nós temos ".httpBody" e vou pôr o "dicionario", que na verdade não é mais dicionário, ele é o "data" da nossa requisição.

[12:32] Além disso nós precisamos indicar para nossa requisição qual é o verbo que ela vai utilizar, ou seja, nós vamos mandar os alunos para lá, quando nós recebemos a informação, utilizamos o "get", quando nós mandamos, utilizamos o "post", quando nós atualizamos algum dado, utilizamos o put e quando vamos deletar, nós temos o "delete", que nada mais é do que o verbo da requisição. Então vou chamar aqui "requisição.httpMethod" e ele vai ser um "POST", porque nós precisamos enviar o aluno pro servidor da Caelum.

[13:07] E para finalizar a nossa requisição, nós precisamos explicar pro nosso servidor qual o tipo de dado que nós estamos enviando, então eu vou adicionar um valor em que eu vou falar para ele qual é o tipo de dado, que nada mais é do que um JSON, esse "application/json", esse tipo de dado que nós vamos enviar. Eu vou pôr aqui "application/json" e aqui nós explicamos que é no header, "Content-Type".

[13:39] Agora sim nós podemos testar a nossa requisição. Como vamos fazer isso? Criando uma tarefa, então "let task" é igual a requisição que nós vamos utilizar, que é "URLSession" e vamos inicializá-la passando a nossa requisição, então precisamos achar o método construtor, em que nós passamos a nossa requisição e ele nos devolve qual foi a resposta, se foi erro, se foi sucesso e tudo mais. Então vamos utilizar esse método, onde passamos esse "URLRequest".

[14:13] Qual é o URLRequest que nós temos? A requisição, então vamos passá-lo aqui, "requisicao" e ele vai nos devolver três informações, o "data", que é a resposta realmente do servidor e o "error" também.

[14:32] Primeira coisa que temos que verificar é se ocorreu algum erro na nossa requisição, então vamos verificar, “if error” for igual a “nil”, ou seja, será que existe erro na nossa requisição? Se ele entrar no “if” significa que deu tudo certo, só que ele retorna um “data”, e como nós vamos mostrar esse data na tela? Esse tipo data? Nós precisamos de um dicionário, que é o que estamos enviando, e o servidor vai nos devolver a média dos alunos.

[15:02] Então precisamos converter novamente esse data para dictionary, e como nós fazemos isso? Vamos criar um “dicionario” e vamos chamar o “JSONSerialization”, e vamos inicializá-lo chamando essa opção “.jsonObject”. Repara que ele espera um data, então nós temos um data aqui. Então vou passar essa variável do tipo “data” e em options eu não vou passar nada.

[15:31] Como ela é uma throw function, nós vamos ter que tratar isso novamente, então o que eu vou fazer aqui? Eu vou colocar essa tentativa que nós estamos fazendo, de transformar o data em um dictionary, e se der algum erro, nós vamos descobrir através desse “catch”, então vou dar um “print” e eu vou passar o “error.description”.

[15:54] Agora que já configuramos toda a requisição, precisamos chamar a nossa tarefa e dar esse “.resume”. Só que como nós vamos testar isso? Porque até então nós não estamos chamando essa opção, aqui eu só vou colocar um “print” para nós identificarmos se ele está realmente trazendo a média, então vou dar um “print” desse dicionário aqui embaixo. E aqui eu vou passar para “let”, como não estamos mudando o valor.

[16:23] Na ação do botão no nosso View Controller, nós não estamos chamando essa classe, então vamos chamá-la agora. Vou chamar aqui “CalculaMediaAPI” e vou chamar a função, que é “calculaMediaGeralDosAlunos”. Vamos rodar o app e testar para ver se está funcionando? Vamos lá.

[16:41] Com o simulador aberto, eu vou clicar no botão calcular média. Cliquei e olha só, que legal, ele nos retornou um dicionário, que é igual, nós convertermos a variável.data para dicionário, e ele printou para nós que a média é 8, porque eu deixei fixo aqui, por enquanto, que a nota do Ândriu é 8.

[17:02] É assim que nós fazemos a requisição programaticamente, sem usar nenhum framework como o Alamofire e tudo mais, para API da Caelum, para descobrirmos a média. O próximo passo é exibirmos a média em um UIAlertController, que nós já vamos ver.