03

Validação do site

Transcrição

[00:00] Até agora, estamos construindo nossa classe sem nos preocuparmos se a url que está sendo enviada para dentro dela é de fato do ByteBank. Mas como definimos se a url vai ser do ByteBank ou não?

[00:15] A minha proposta é: caso a url comece com a string https://bytebank.com/ (https://bytebank.com/), ela é do ByteBank. Mas será que existe alguma outra forma da url conter essa string específica sem pertencer ao ByteBank? Vou mostrar que sim.

[00:33] Vou fazer uma busca rápida no Google por esse argumento. Lembra que sempre que faço uma busca no Google, ele tem algum argumento ali dentro que vai ficar com o nome do que a gente buscou? Pode ser que isso aconteça, e pode ser que por algum motivo alguma url de algum site de buscas ou alguma outra url maliciosa de um site de piratas ou hackers estejam tentando invadir nosso sistema usando esse tipo de artimanha, de colocar nosso endereço dentro do endereço deles achando que nossa classe não estará pronta para isso.

[01:13] Nós vamos agora preparar a classe para esse tipo de problema. Que método conhecemos que identifica se algum texto está dentro de alguma coisa? O find. A primeira coisa que vamos fazer é começar a usar o método find para buscar se alguma url contém a do ByteBank ou não, e em qual posição ela se encontra, porque isso é muito importante. urlByteBank = "https://bytebank.com" (https://bytebank.com%E2%80%9D) url1 = "https://buscasites.com/busca?q= (https://buscasites.com/busca?q=) https://bytebank.com" (https://bytebank.com%E2%80%9D) url2 = "https://bitebank.com.br" (https://bitebank.com.br%E2%80%9D) url3 = "https://bytebank.com/cambio/teste/teste" (https://bytebank.com/cambio/teste/teste%E2%80%9D)

[02:17] Esse é um exemplo parecido com o do Google, mas em escala menor, para deixar a url mais identificável. A url2 tem um nome bem parecido com o do ByteBank, mas não é.

[02:50] Vou começar a utilizar o método find para verificar se nossa url buscada está dentro dessas URLs que acabamos de criar: print (url1.find(urlByteBank)). Ele encontrou na posição 31.

[03:31] Essa url contém a url do ByteBank mesmo sem ser do ByteBank. Nós não queremos que nossa classe aceite esse tipo de coisa. Precisamos que seja do primeiro índice para confirmar que essa url é do ByteBank. Vamos testar de novo com a url2. Ele me retorna -1, o que significa que nem achou. Na url3, ele me dá 0.

[04:08] Se o retorno do método find for 0, posso considerar que essa url de fato é confiável, é do nosso sistema. Mas existe outra forma de fazer esse tipo de busca. Existe um método específico de string que me ajuda bastante com esse tipo de coisa. É o método starts with. Eu verifico se a minha string começa com um caractere específico. Exemplo: print (url3.startswith (urlByteBank)).

[04:52] Ele acusa como verdadeiro. Perfeito. Posso usar ou o find retornando 0 ou posso usar direto esse método starts with, que já me retorna true ou false de cara.

[05:11] Vamos testar com a url1. Lembrando que ela contém de fato a url do ByteBank, mas me retornou false, porque não começa com a url. Novamente, nosso trabalho sempre é colocar essa lógica dentro da classe.

[05:29] Esse método de urlEhValida já parece bem justo para isso. Ele verifica se a url existe. Pode verificar também se além de existir é do ByteBank. Eu posso dizer: if url and url.startswith ("https://bytebank.com" (https://bytebank.com%E2%80%9D)): return true

[06:12] Testando isso, ele funcionou normalmente. Mas e se eu colocar ao invés de Byte, Bite? Ele me mostra um erro e me diz que a url é inválida.

[06:35] Agora nossa classe está pronta para lidar com URLs maliciosas, que não sejam da nossa aplicação e estejam tentando utilizar nossa classe para fazer qualquer coisa. Terminamos nossa classe, ela está pronta, funcional.

[06:54] O que vamos ver na próxima aula é algo um pouco diferente. Nós vamos continuar trabalhando com strings, mas de outra forma. Vamos ver expressões regulares, que são formas de encontrar padrões bem definidos dentro de textos mais humanos, dentro de e-mails, posts de Facebook ou de mensagens de celular. Mas sem spoiler. Façam todas as atividades e tirem as dúvidas que tiverem. Obrigado por assistir e até a próxima aula.