

Só quero uma fatia do bolo, não o bolo inteiro!

Alfredo hoje consegue obter uma lista de frases através de `$.get` como no exemplo abaixo:

```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log(retorno); // imprime a lista de frases.  
  
});
```

Contudo, ele quer obter a segunda frase. Para isso ele precisou alterar seu código para:

```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log(retorno[1]); // imprime a segunda frase  
  
});
```

Porém, e se tivéssemos 1.000.000 de frases? Teríamos que trazer uma quantidade de dados gigante para então acessar um dado específico.

Aprendemos que um servidor por estar configurado para aceitar um parâmetro na requisição `get` que indica para ele qual dado retornar ao invés de retornar uma lista inteira.

Qual das opções abaixo altera o código de Alfredo para que retorna a frase com id igual a 5?

A

```
var parametros = {id: 5}; // usamos id porque o servidor espera receber id. Quem  
  
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log(retorno); // imprime a frase!  
  
}, dados);
```

B

```
$.get('http://localhost:3000/frases', 5, function(retorno) {  
  
    console.log(retorno); // imprime a frase!  
  
});
```



```
var parametros = {id: 5}; // usamos id porque o servidor espera receber id. Quem  
  
$.get('http://localhost:3000/frases', parametros, function(retorno) {  
  
    console.log(retorno); // imprime a frase!
```

```
});
```

Está correto! Passamos a variável `parametros` e depois executamos a function com o dado recebido!

Nesta altura do campeonato você já deve ter compreendido que para consumirmos dados de um servidor precisamos conhecer esse servidor. Geralmente quem construiu o servidor gera uma documentação que lista todos os endereços possíveis, quais verbos do HTTP usar e seus parâmetros. Se você é a pessoal que criou o servidor, fica mais fácil ainda saber os endereços e parâmetros.

PRÓXIMA ATIVIDADE