

API externa

Clarice precisava buscar os dados de uma API que retorna os dados dos alunos matriculados em sua turma.

Vejamos um exemplo:

```
fetch('http://endereco-da-api.com.br/alunos/1')
  .then(res => {

    if(res.ok) {
      return res;
    } else {
      throw new Error(res.statusText);
    }

  })
  .then(res => res.json())
  .then((aluno: any) => {
    console.log(aluno.nome);
    console.log(aluno.matricula);

    // faz algo com o aluno
  })
  .catch(err => console.log(err.message));
```

5 meses depois, a API que ela utilizava mudou a propriedade `matricula` para `inscricao`. Com certeza a aplicação de Clarice se comportou erradamente. Então, ela foi e alterou seu código para:

```
fetch('http://endereco-da-api.com.br/alunos/1')
  .then(res => {

    if(res.ok) {
      return res;
    } else {
      throw new Error(res.statusText);
    }

  })
  .then(res => res.json())
  .then((aluno: any) => {
    console.log(aluno.nome);

    // modificou aqui! Veja que ela escreveu errado!
    console.log(aluno.inscrica);

    // faz algo com o aluno
  })
  .catch(err => console.log(err.message));
```

No entanto, ao realizar a correção, ela escreveu errado o nome da propriedade `inscricao`. Novamente, ela só descobriu o problema em runtime para então realizar o ajuste final:

```
fetch('http://endereco-da-api.com.br/alunos/1')
.then(res => {

  if(res.ok) {
    return res;
  } else {
    throw new Error(res.statusText);
  }

})

.then(res => res.json())
.then((aluno: any) => {
  console.log(aluno.nome);

  // modificou aqui! Veja que ela escreveu corretamente.
  console.log(aluno.inscricao);

  // faz algo com o aluno
})
.catch(err => console.log(err.message));
```

Se Clarice tivesse definido uma interface (um formato) para os dados recebidos do servidor, além das alterações serem mais fáceis de realizar, ela não teria como cometer o erro que cometeu escrevendo `inscrica` em vez de `inscricao`.

Marque a alternativa que cria a interface `Aluno` que define as propriedades mais recentes da API. Inclusive, aquela que tipa corretamente o retorno.

Seleciona uma alternativa

A

```
export interface Aluno {

  nome: string;
  inscricao: string
}

fetch('http://endereco-da-api.com.br/alunos/1')
// código omitido
.then(res => res.json())
.then(aluno: Aluno => {

  console.log(aluno.nome);
  console.log(aluno.inscricao);

  // faz algo com o aluno
})
.catch(err => console.log(err.message));
```

B

```
export interface Aluno {  
  
  nome: string;  
  inscricao: string  
}  
  
fetch('http://endereco-da-api.com.br/alunos/1')  
// código omitido  
.then(res => res.json())  
.then(aluno: any => {  
  
  console.log(aluno.nome);  
  console.log(aluno.inscricao);  
  
  // faz algo com o aluno  
})  
.catch(err => console.log(err.message));
```

C

```
export interface Aluno {  
  
  public nome: string;  
  public inscricao: string  
}  
  
fetch('http://endereco-da-api.com.br/alunos/1')  
// código omitido  
.then(res => res.json())  
.then(aluno: Aluno => {  
  
  console.log(aluno.nome);  
  console.log(aluno.inscricao);  
  
  // faz algo com o aluno  
})  
.catch(err => console.log(err.message));
```