

Sem validação ❌

```
app.post("/user", async (req, res) => {  
  const response = await createUser({  
    email: req.body.email,  
    password: req.body.password,  
  })  
  
  return res.status(201).json(response)  
})
```

**está criando usuário
com informações
diretas do client**

Dessa forma, ele permite todo tipo de input:

email: vitor@mail.com
password: vitor#123

email: ###@99.com
password: 2368482

email: " ".
password: x

email: markis@balloon.net
password: markis#123

email: vitor@mail
password: vitor#123

email: 
password: 

email: vitor@mail.com
password: ""

email:
password:

Com validação



```
const createUserSchema = z.object({  
  email: z.string().email().max(48),  
  password: z.string().min(6).max(21)  
})
```



Zod

```
app.post("/user", async (req, res) => {  
  const payload = createUserSchema.parse(req.body)  
  const response = await createUser(payload)  
  
  return res.status(201).json(response)  
})
```

valida o req.body primeiro

Agora sim permite apenas **inputs** válidos!

email: vitor@mail.com
password: vitor#123





email: ###@99.com
password: 2368482

email: “ .”
password: x

email: markis@balloon.net
password: markis#123



email: vitor@mail
password: vitor#123

email: 
password: 

email: vitor@mail.com
password: “”

email:
password:

Gostou?

Deixe uma reação!

