

PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite



Exercícios

- Implemente duas rotas de atualização:
 - Uma rota usando `@app.put` para realizar a atualização total de um contato, substituindo completamente os dados (nome e telefone).
 - Outra rota usando `@app.patch` para realizar a atualização parcial, permitindo alterar apenas o telefone de um contato existente.
 - Utilize `request.get_json()` para acessar os dados enviados no corpo da requisição. Lembre-se de tratar casos de erro, como dados ausentes ou contato inexistente.

Atualizando um contato usando @app.put

```
@app.put('/contato/<nome>')
def atualizar_contato_total(nome):
    dados = request.get_json()

    if not dados or 'nome' not in dados or 'telefone' not in dados:
        return {'erro': "Campos 'nome' e 'telefone' são obrigatórios."}, HTTPStatus.BAD_REQUEST

    nome = nome.lower()
    novo_nome = dados['nome'].lower()
    novo_tel = dados['telefone']

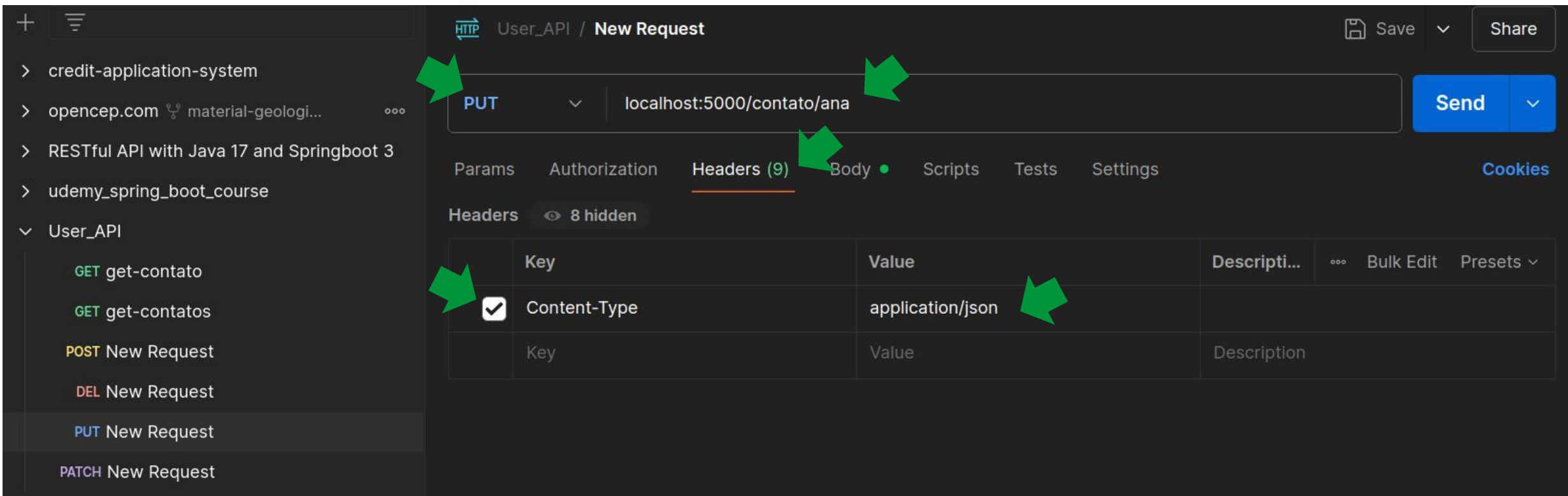
    if nome not in agenda:
        return {'erro': f"Contato '{nome}' não encontrado."}, HTTPStatus.NOT_FOUND

    if novo_nome != nome and novo_nome in agenda:
        return {'erro': f"Contato '{novo_nome}' já existe."}, HTTPStatus.CONFLICT

    del agenda[nome]
    agenda[novo_nome] = novo_tel

    return {
        'mensagem': "Contato atualizado com sucesso.",
        'nome': novo_nome.capitalize(),
        'telefone': novo_tel
    }, HTTPStatus.OK
```

Atualizando um contato usando @app.put



HTTP User_API / New Request

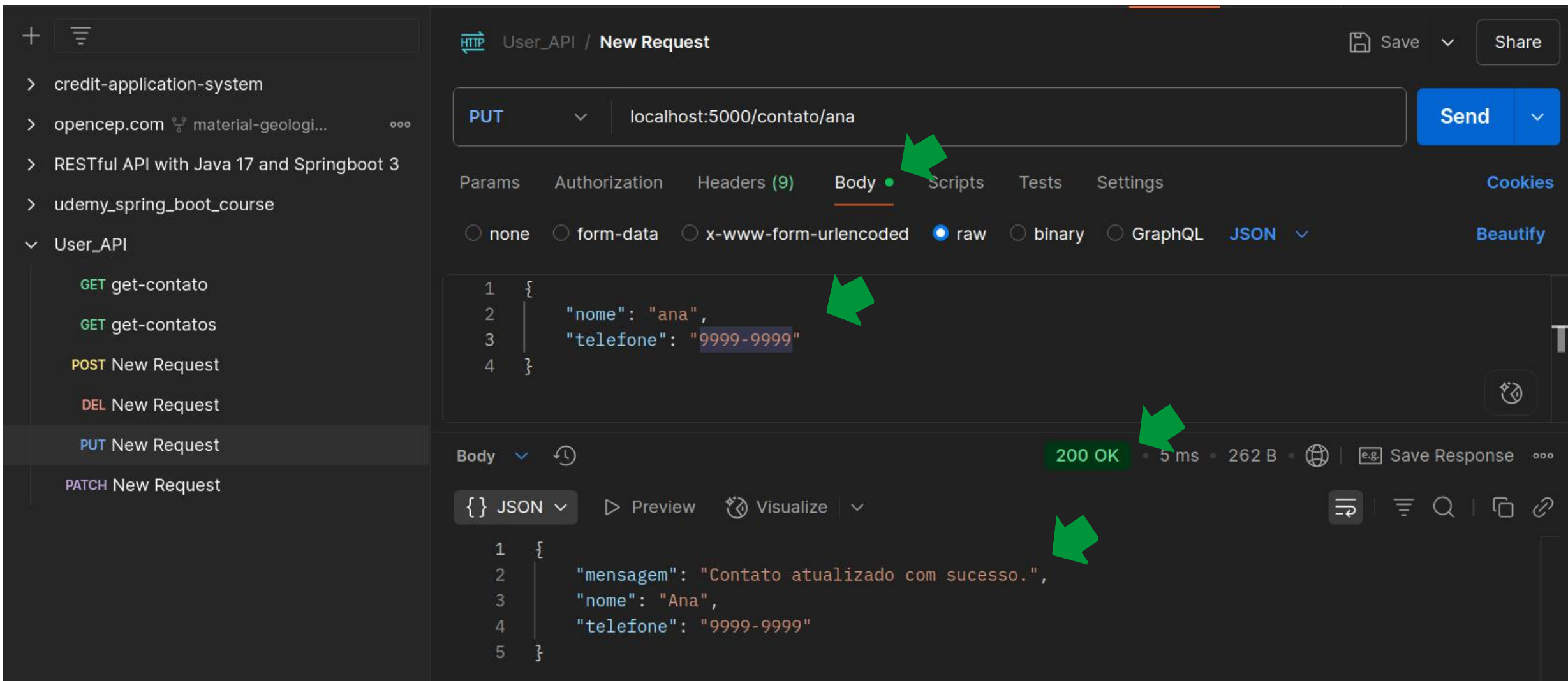
PUT localhost:5000/contato/ana

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Headers 8 hidden

	Key	Value	Descripti...	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Atualizando um contato usando @app.put



The screenshot displays the Postman interface for a PUT request. The left sidebar shows a collection named 'User_API' with several endpoints. The main panel shows the 'New Request' tab for the endpoint 'localhost:5000/contato/ana'. The request method is 'PUT'. The 'Body' tab is selected, showing a JSON payload:

```
{  "nome": "ana",  "telefone": "9999-9999"}
```

. The response section shows a '200 OK' status with a response time of 5 ms and a size of 262 B. The response body is a JSON object:

```
{  "mensagem": "Contato atualizado com sucesso.",  "nome": "Ana",  "telefone": "9999-9999"}
```

. Green arrows highlight the request URL, the JSON body, the '200 OK' status, and the response body.

credit-application-system
opencep.com material-geologi...
RESTful API with Java 17 and Springboot 3
udemy_spring_boot_course
User_API
GET get-contato
GET get-contatos
POST New Request
DEL New Request
PUT New Request
PATCH New Request

User_API / New Request

PUT localhost:5000/contato/ana

Params Authorization Headers (9) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

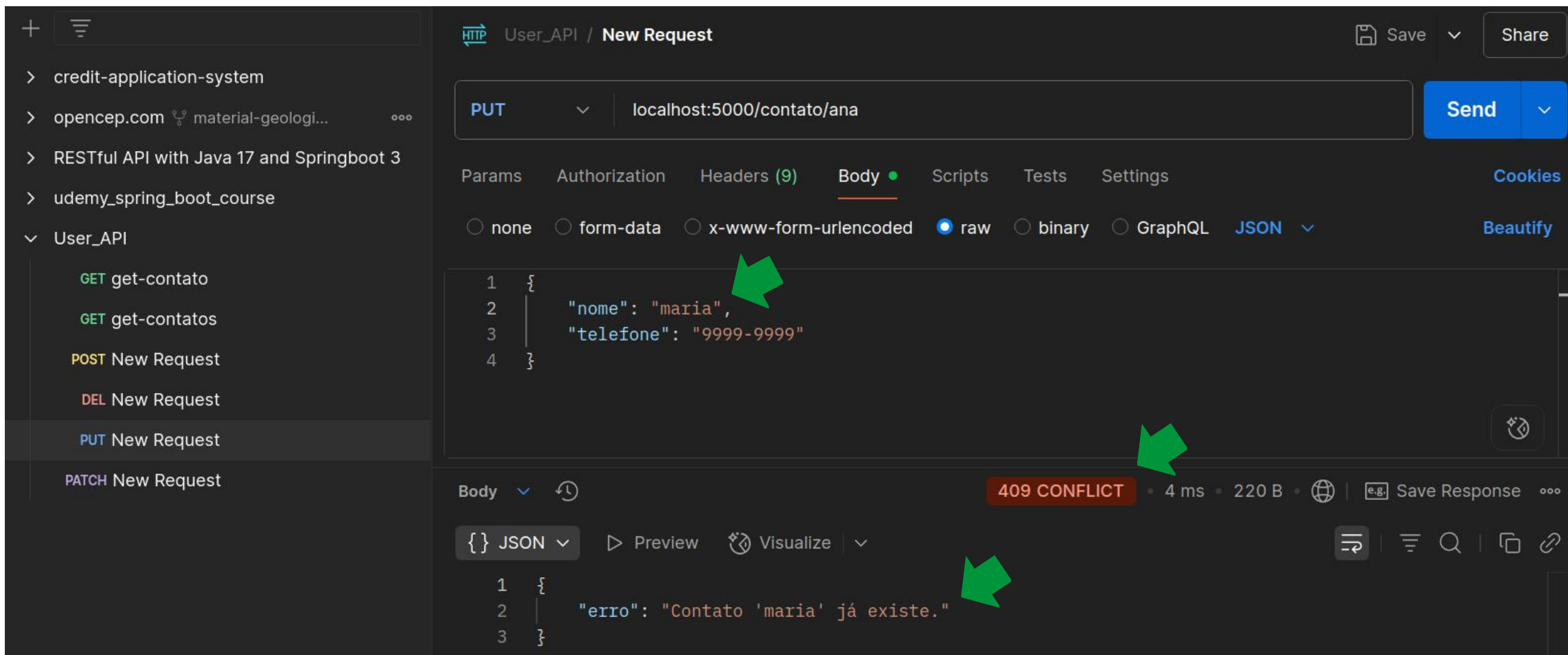
```
1 {
2   "nome": "ana",
3   "telefone": "9999-9999"
4 }
```

Body 200 OK 5 ms 262 B Save Response

{ JSON Preview Visualize

```
1 {
2   "mensagem": "Contato atualizado com sucesso.",
3   "nome": "Ana",
4   "telefone": "9999-9999"
5 }
```

Atualizando um contato usando @app.put



The screenshot displays a REST client interface with a sidebar on the left containing a project tree. The main area is titled "User_API / New Request". The request method is set to "PUT" and the URL is "localhost:5000/contato/ana". The "Body" tab is selected, showing a JSON payload:

```
{  "nome": "maria",  "telefone": "9999-9999"}
```

. The response section shows a "409 CONFLICT" status with a message:

```
{  "erro": "Contato 'maria' já existe."}
```

. Green arrows highlight the JSON body in the request and the error message in the response.

Left sidebar (Project Tree):

- credit-application-system
- opencep.com material-geologi...
- RESTful API with Java 17 and Springboot 3
- udemy_spring_boot_course
- User_API
 - GET get-contato
 - GET get-contatos
 - POST New Request
 - DEL New Request
 - PUT New Request
 - PATCH New Request

Main Area: User_API / New Request

Request Method: PUT

URL: localhost:5000/contato/ana

Body Tab (JSON):

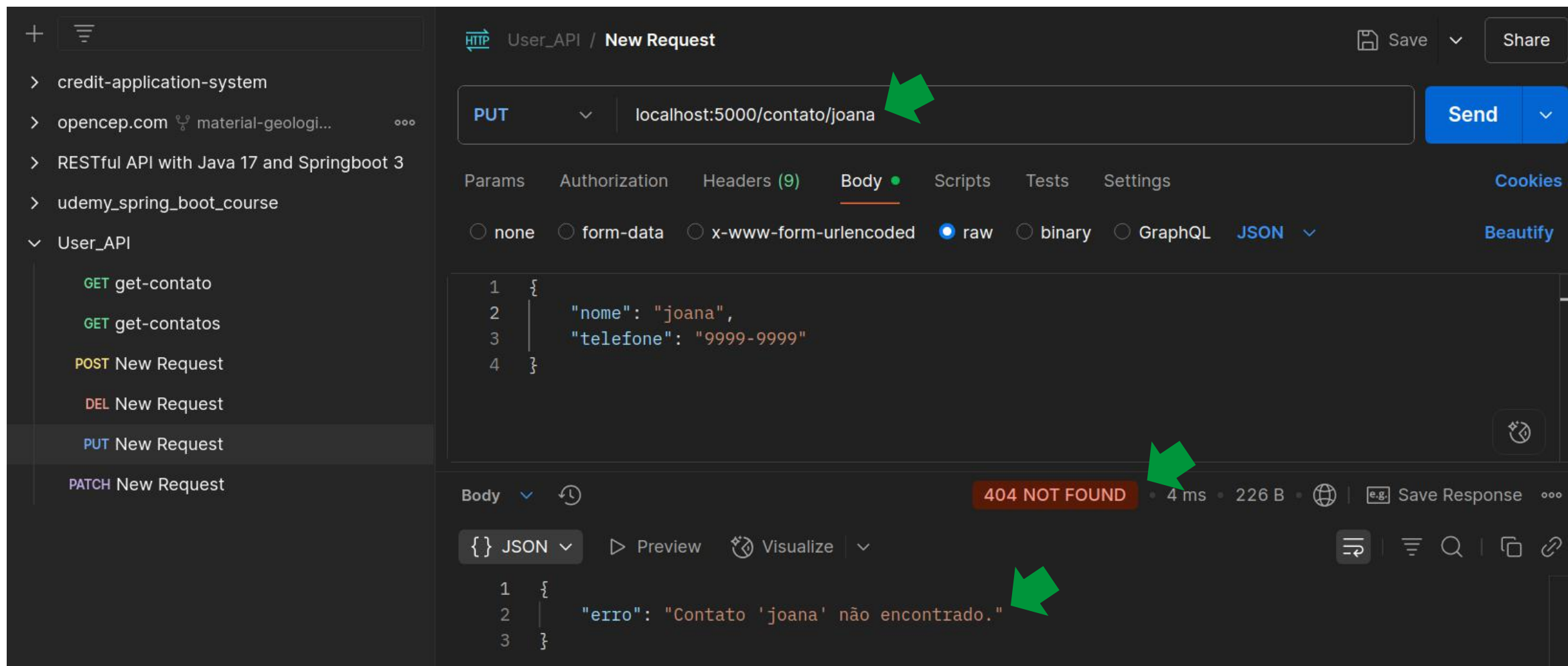
```
1 {
2   "nome": "maria",
3   "telefone": "9999-9999"
4 }
```

Response: 409 CONFLICT (4 ms, 220 B)

Response Body (JSON):

```
1 {
2   "erro": "Contato 'maria' já existe."
3 }
```


Atualizando um contato usando @app.put



The screenshot shows the Postman interface for a new request. The left sidebar lists the project 'User_API' with several endpoints. The main area shows a PUT request to 'localhost:5000/contato/joana' with a JSON body. The response is a 404 NOT FOUND error with a message in Portuguese.

Request Details:

- Method: PUT
- URL: localhost:5000/contato/joana
- Body Type: raw (JSON)
- Body Content:


```
1 {  
2   "nome": "joana",  
3   "telefone": "9999-9999"  
4 }
```

Response Details:

- Status: 404 NOT FOUND
- Time: 4 ms
- Size: 226 B
- Body Type: JSON
- Body Content:

```
1 {  
2   "erro": "Contato 'joana' não encontrado."  
3 }
```

Atualizando um contato usando @app.patch



```
@app.patch('/contato/<nome>')
def atualizar_contato_parcial(nome):
    dados = request.get_json()

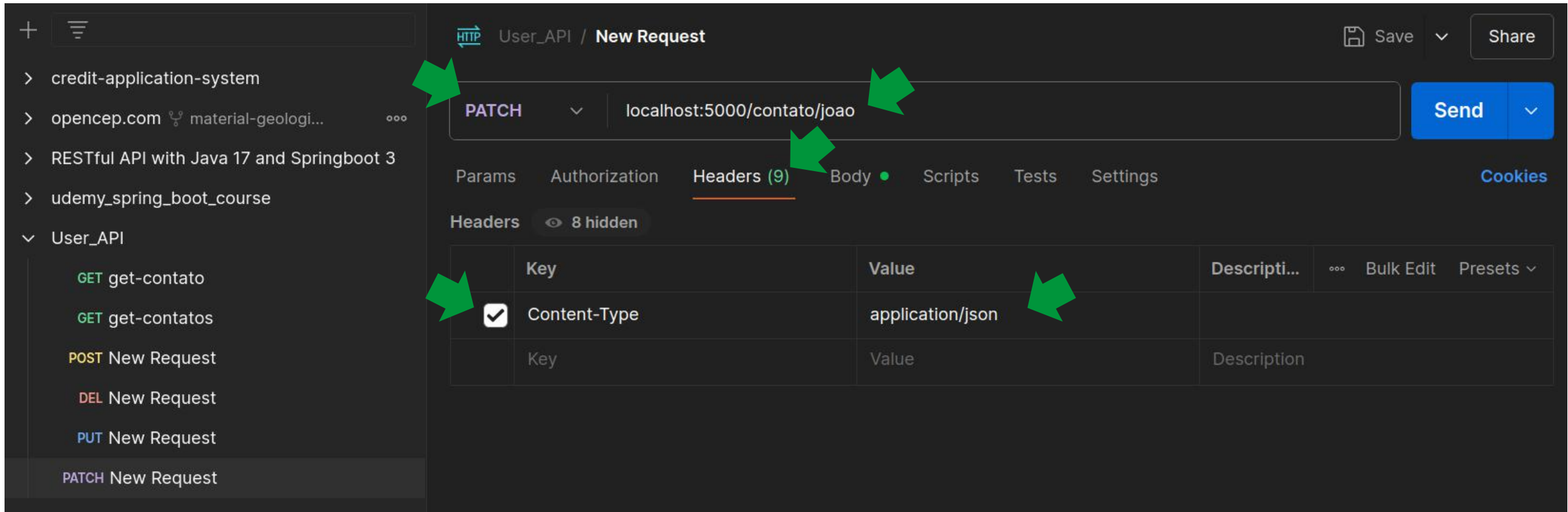
    if not dados or 'telefone' not in dados:
        return {'erro': "Corpo da requisição deve conter 'telefone'."}, HTTPStatus.BAD_REQUEST

    nome = nome.lower()
    novo_tel = dados['telefone']

    if nome not in agenda:
        return {'erro': f"Contato '{nome}' não encontrado."}, HTTPStatus.NOT_FOUND

    agenda[nome] = novo_tel
    return {
        'mensagem': "Telefone atualizado com sucesso.",
        'nome': nome.capitalize(),
        'telefone': novo_tel
    }, HTTPStatus.OK
```


Atualizando um contato usando @app.patch



The screenshot shows the Postman interface for a new request. The left sidebar lists the API collection 'User_API' with several endpoints. The main panel is titled 'User_API / New Request' and shows a PATCH request to 'localhost:5000/contato/joao'. The 'Headers' tab is selected, showing a table with one header: 'Content-Type' with the value 'application/json'. Green arrows highlight the PATCH method, the URL, the Headers tab, the Content-Type header, and the value 'application/json'.

HTTP User_API / New Request

PATCH localhost:5000/contato/joao

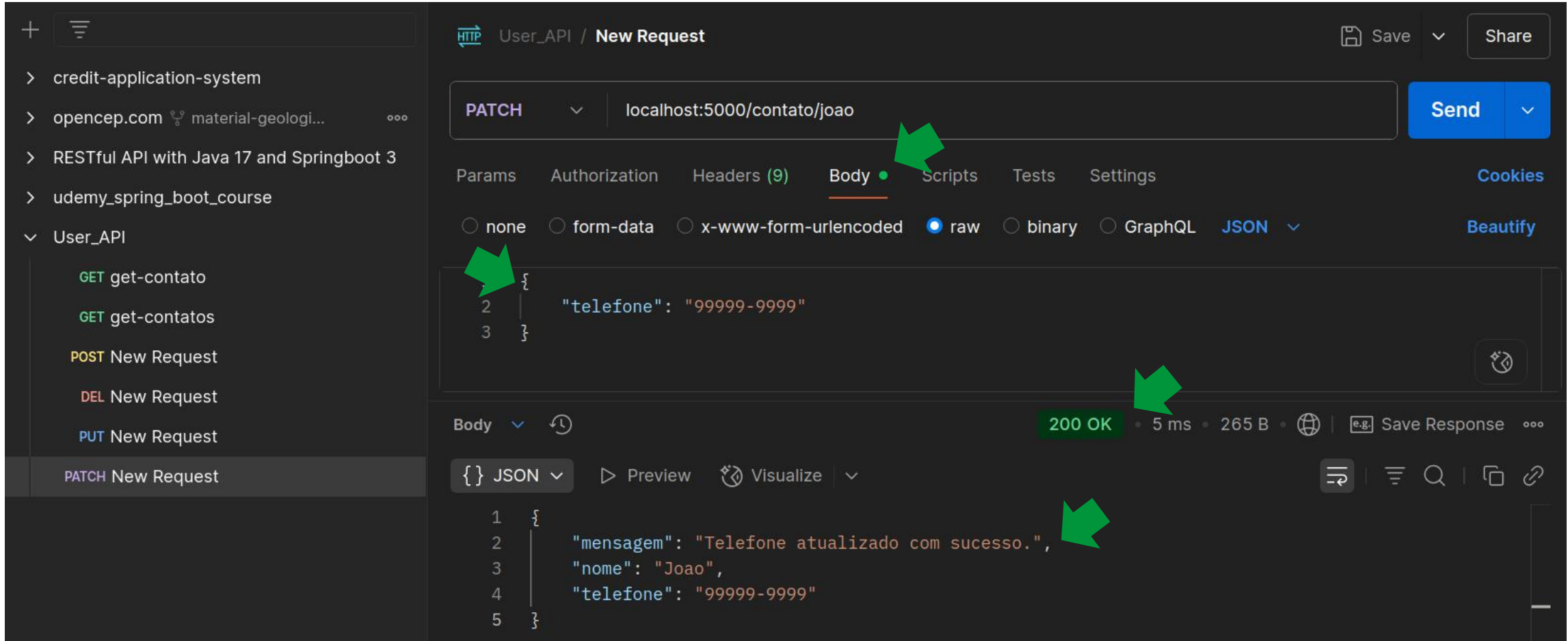
Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Headers 8 hidden

	Key	Value	Descripti...	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Atualizando um contato usando @app.patch



The screenshot shows the Postman interface for a new request. The left sidebar lists the API collections, with 'User_API' expanded. The main area shows a PATCH request to 'localhost:5000/contato/joao'. The 'Body' tab is selected, and the request body is a JSON object: `{ "telefone": "99999-9999" }`. The 'raw' radio button is selected, and the 'JSON' format is chosen. The 'Send' button is visible. Below the request, the response is shown as '200 OK' with a status of '5 ms' and a size of '265 B'. The response body is a JSON object: `{ "mensagem": "Telefone atualizado com sucesso.", "nome": "Joao", "telefone": "99999-9999" }`. Green arrows highlight the PATCH method, the request URL, the request body, the response status, and the response body.

HTTP User_API / New Request

PATCH localhost:5000/contato/joao

Params Authorization Headers (9) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

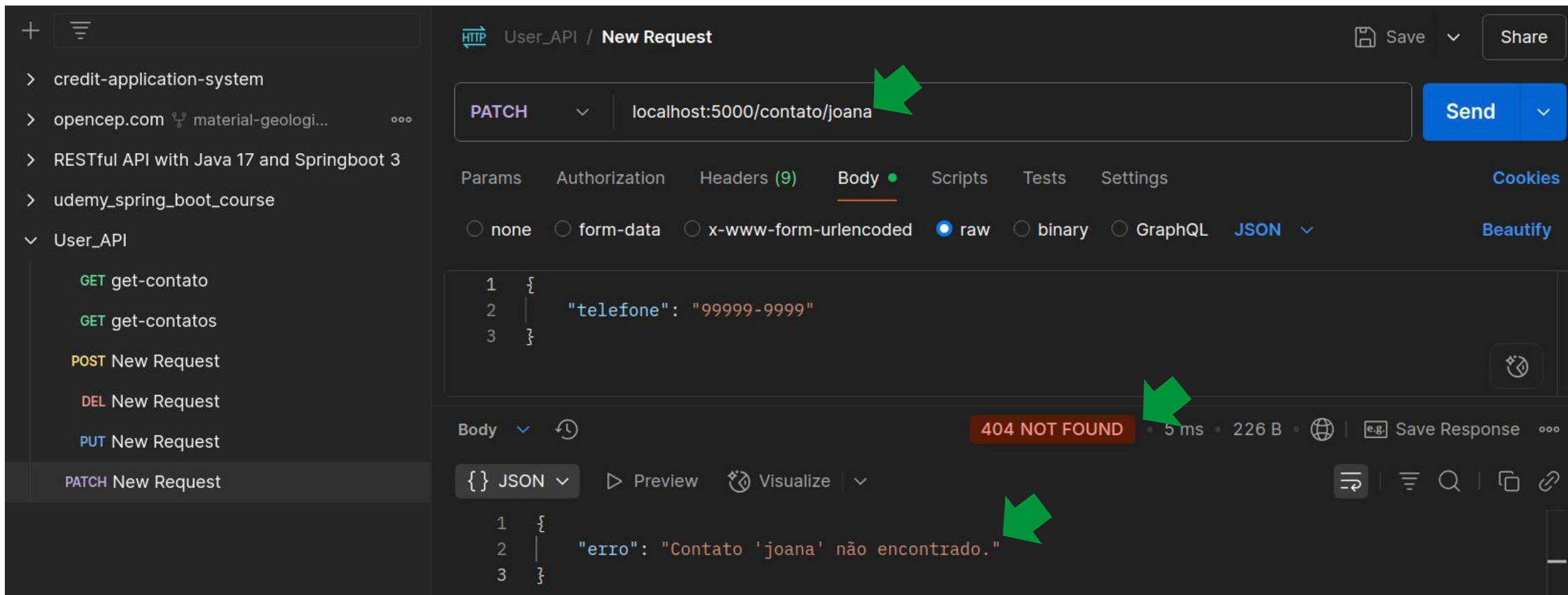
```
{
  "telefone": "99999-9999"
}
```

Body 200 OK • 5 ms • 265 B • Save Response

{ JSON Preview Visualize

```
{
  "mensagem": "Telefone atualizado com sucesso.",
  "nome": "Joao",
  "telefone": "99999-9999"
}
```

Atualizando um contato usando @app.patch



The screenshot shows the Postman interface for a new PATCH request. The URL is `localhost:5000/contato/joana`. The request body is a JSON object: `{ "telefone": "99999-9999" }`. The response is a **404 NOT FOUND** status with a response time of 5 ms and a size of 226 B. The response body is a JSON object: `{ "erro": "Contato 'joana' não encontrado." }`.

Left sidebar (API Explorer):

- credit-application-system
- opencep.com material-geologi...
- RESTful API with Java 17 and Springboot 3
- udemy_spring_boot_course
- User_API
 - GET get-contato
 - GET get-contatos
 - POST New Request
 - DEL New Request
 - PUT New Request
 - PATCH New Request

Right sidebar (Request Editor):

- HTTP User_API / New Request
- Method: PATCH
- URL: localhost:5000/contato/joana
- Body: raw (selected), JSON (selected)
- Body content: `{ "telefone": "99999-9999" }`
- Response: 404 NOT FOUND (5 ms, 226 B)
- Response body: `{ "erro": "Contato 'joana' não encontrado." }`

Testando a API com o Curl

- **GET /contato/<nome>:** `curl -X GET http://localhost:5000/contato/ana`
- **GET /contatos:** `curl -X GET http://localhost:5000/contatos`
- **POST /contato:** `curl -X POST http://localhost:5000/contato \`
 `-H "Content-Type: application/json" \`
 `-d '{"nome": "pedro", "telefone": "95555-1234"}'`
- **DELETE /contato/<nome>:** `curl -X DELETE http://localhost:5000/contato/ana`

Testando a API com o Curl

- **PUT /contato/<nome>:** `curl -X PUT http://localhost:5000/contato/pedro \`
`-H "Content-Type: application/json" \`
`-d '{"nome": "paulo", "telefone": "96666-7890"}'`
- **PATCH /contato/<nome>:** `curl -X PATCH http://localhost:5000/contato/paulo \`
`-H "Content-Type: application/json" \`
`-d '{"telefone": "97777-0000"}'`

Dúvidas



PROGRAMAÇÃO WEB II

Curso Técnico Integrado em Informática
Lucas Sampaio Leite

