**POSTMAN API TRELLO**

Links para instalações e execuções de comandos:

Postman:

<https://www.postman.com/downloads/>

Trello Dev Portal:

<https://trello.com/>

<https://developer.atlassian.com/cloud/trello/guides/rest-api/api-introduction/>

<https://developer.atlassian.com/cloud/trello/rest/api-group-actions/>

NodeJS:

<https://nodejs.org/en/>

Newman:

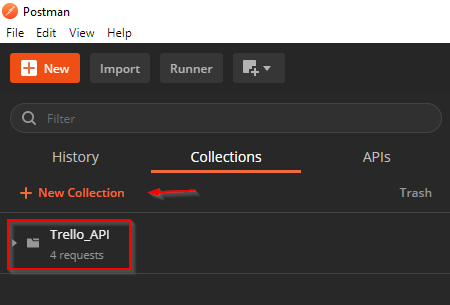
<https://www.npmjs.com/package/newman>

<https://learning.postman.com/docs/running-collections/using-newman-cli/command-line-integration-with-newman/>

<https://www.npmjs.com/package/newman-reporter-htmlextra>

**NOVO PROJETO**

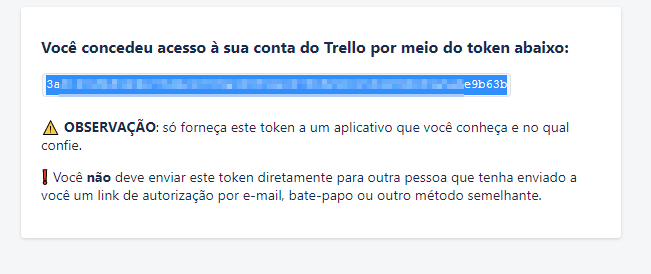
Criando um novo projeto + New Collection >> Adicionar o nome do projeto.



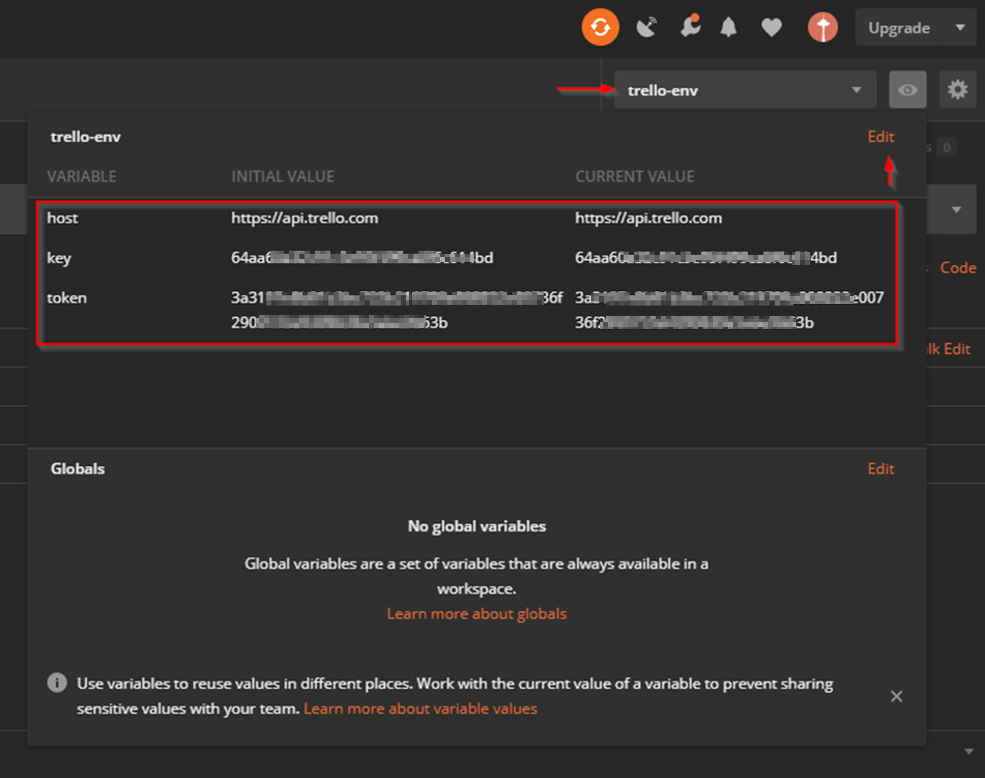
Para gerar o key acessar o o link <https://trello.com/app-key/>



Clicar em Token e permitir o aplicativo.



Adicionar as variaveis auxiliares como endereço da API, key e token

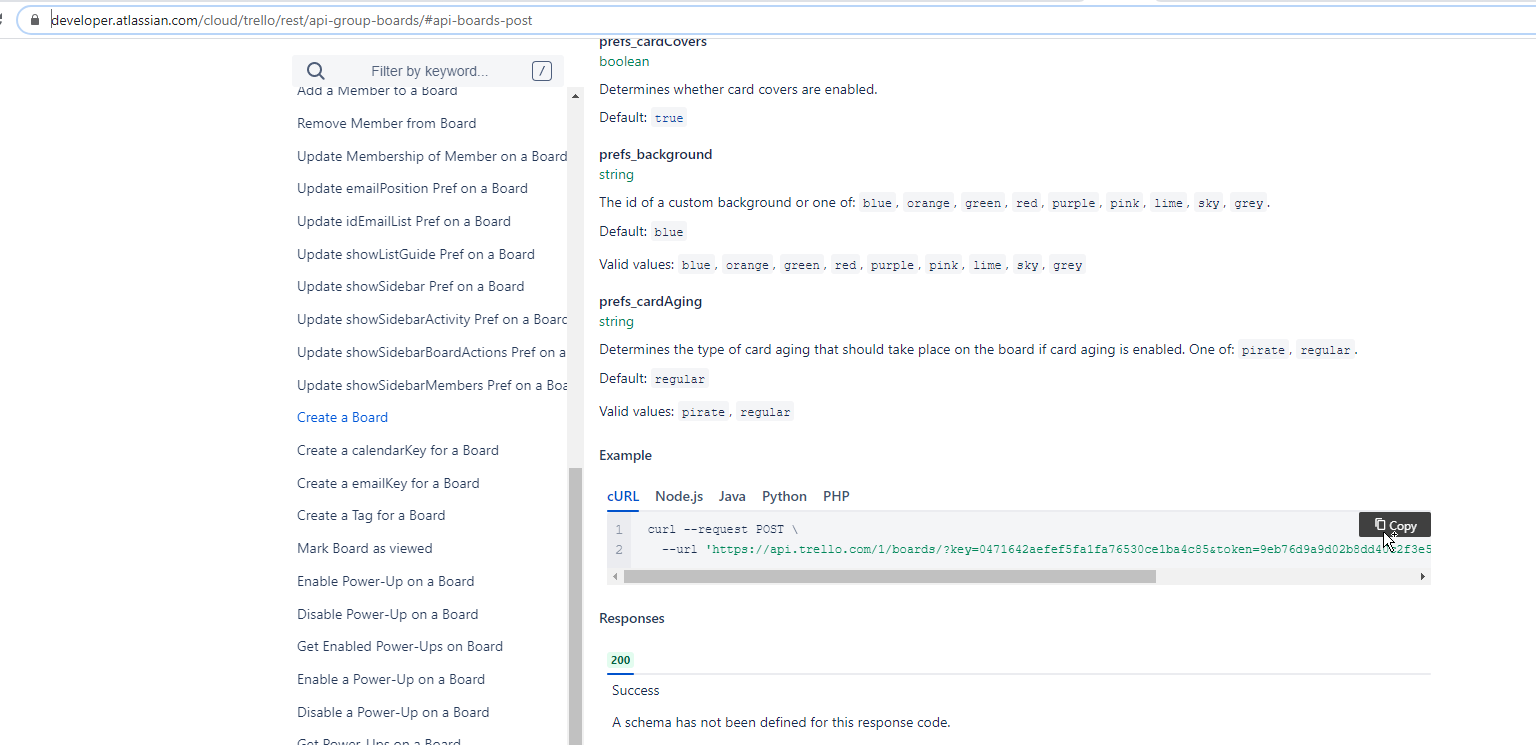


**CRIANDO UM BOARD NO TRELLO**

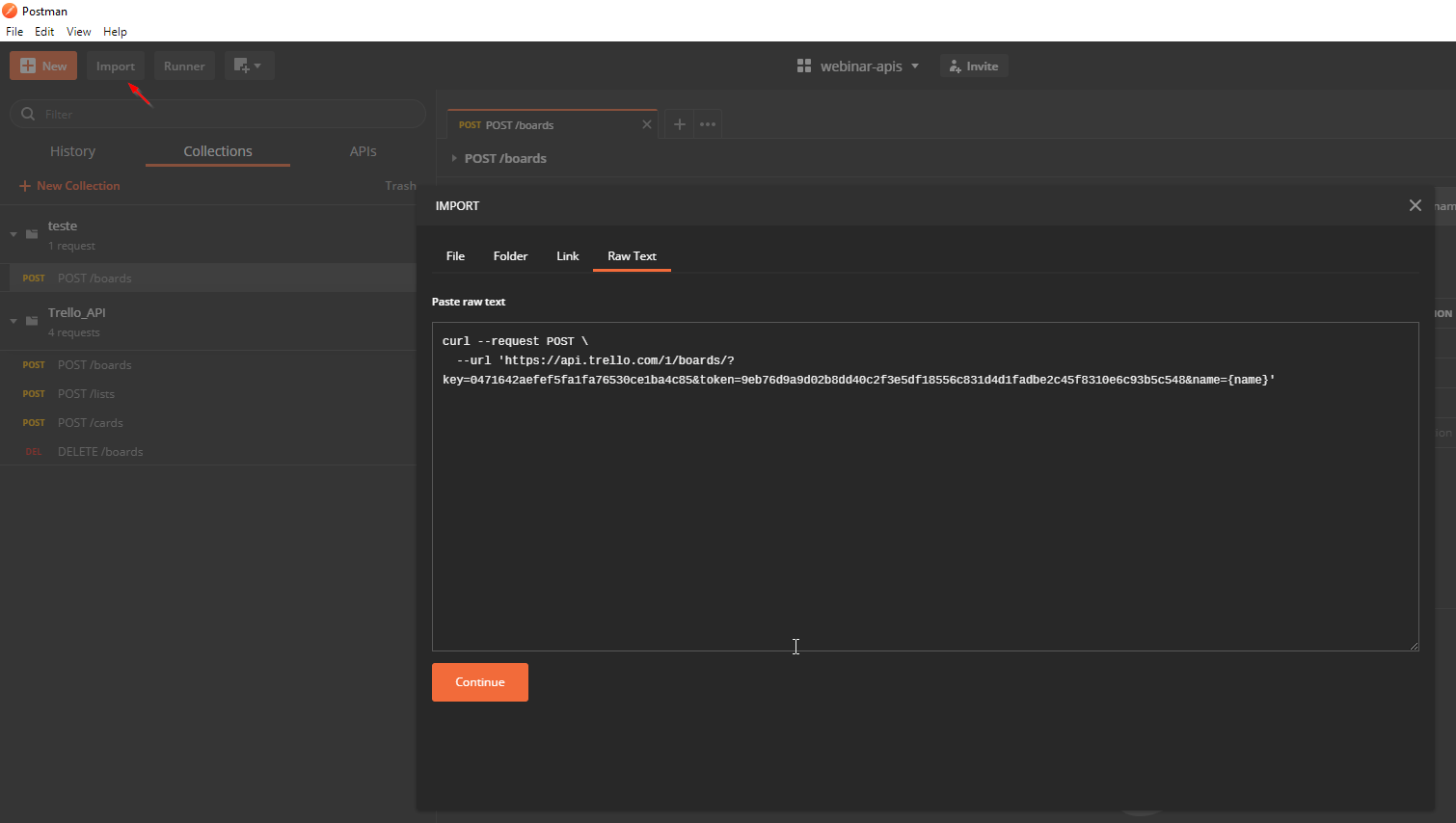
Acessar a documentação da API

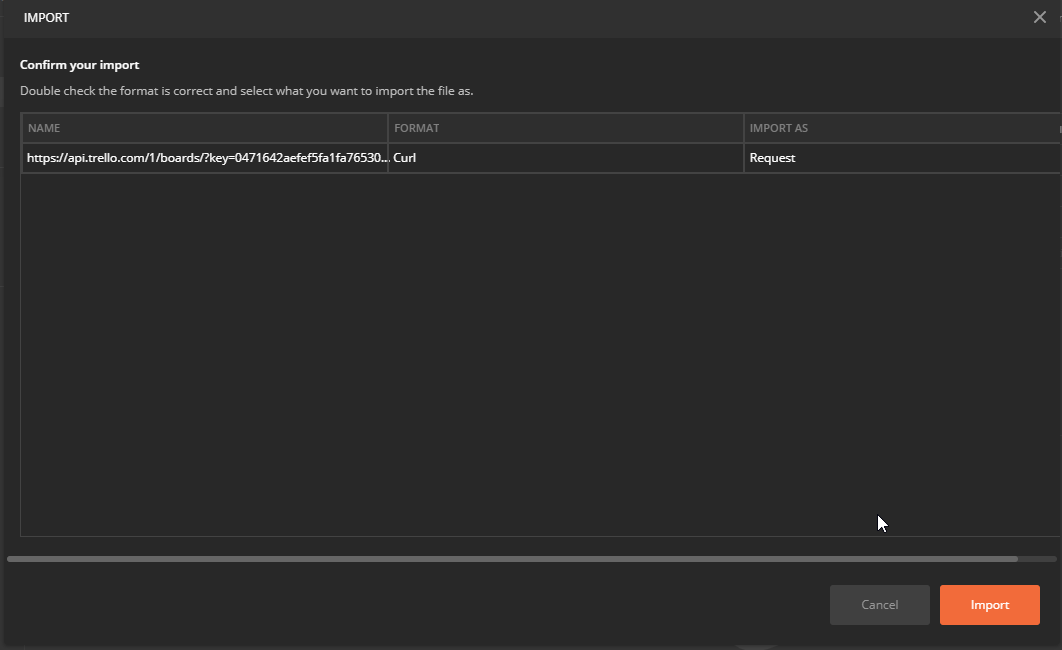
<https://developer.atlassian.com/cloud/trello/rest/api-group-boards/#api-boards-post>

Encontrar a opção de criar um Board e copiar o CURL



No POSTMAN clicar no "import" selecionar "Raw Text" colar o comando cURL e "Continue".





Hit import and you will have the command in your Postman builder!

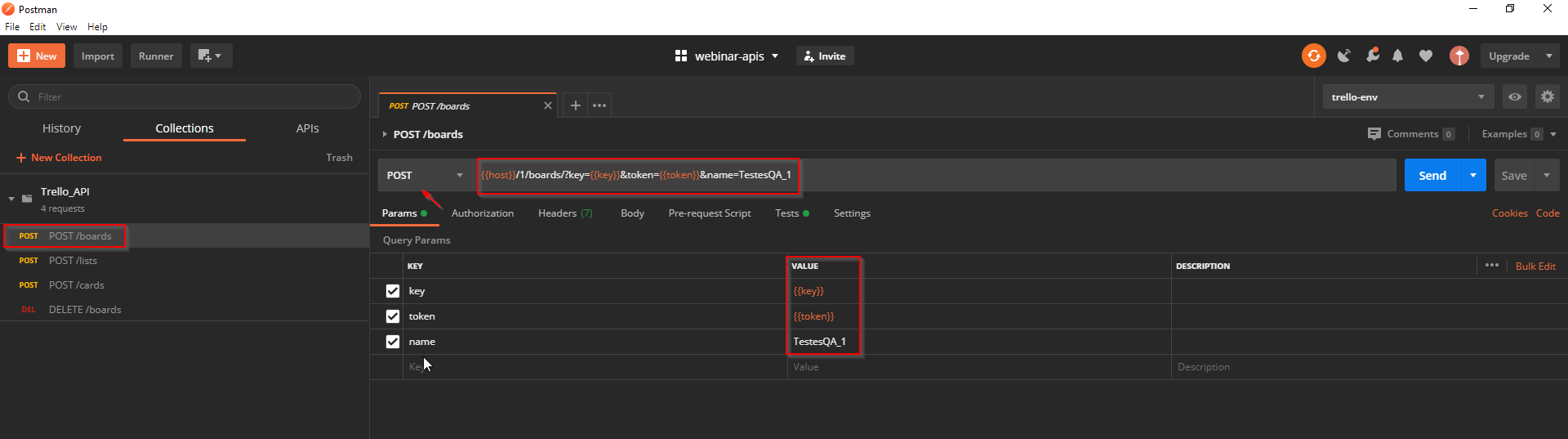
Click Send to post the command

curl --request POST \

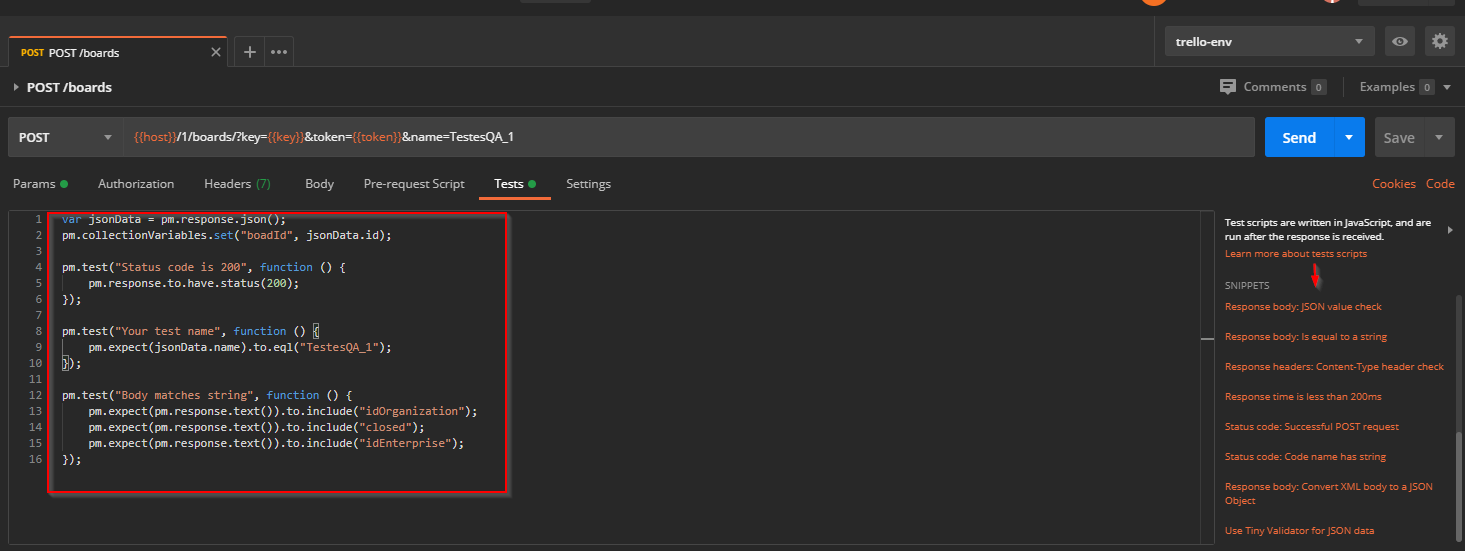
--url 'https://api.trello.com/1/boards/?key=0471642aefef5fa1fa76530ce1ba4c85&token=9eb76d9a9d02b8dd40c2f3e5df18556c831d4d1fadbe2c45f8310e6c93b5c548&name={name}'

Utilziar as variaveis de apoio no endereço da API {{host}} {key}} {{token}} e o parametro name com o nome de BOARD.

{{host}}/1/boards/?key={{key}}&token={{token}}&name=TestesQA\_1



Para validar o retorno da execução poderá ser adicioando Snnipets



Exemplo do código utilizado para a validação do POST de /boards:

var jsonData = pm.response.json(); // salvando o retorno em uma variavel

pm.collectionVariables.set("boadId", jsonData.id); // salvando a tag de retorno boadId em uma variavel

pm.test("Status code is 200", function () {

    pm.response.to.have.status(200); //validando o retorno do codigo 200

});

pm.test("Your test name", function () {

    pm.expect(jsonData.name).to.eql("TestesQA\_1"); //validando o nome do board que foi criado

});

pm.test("Body matches string", function () {

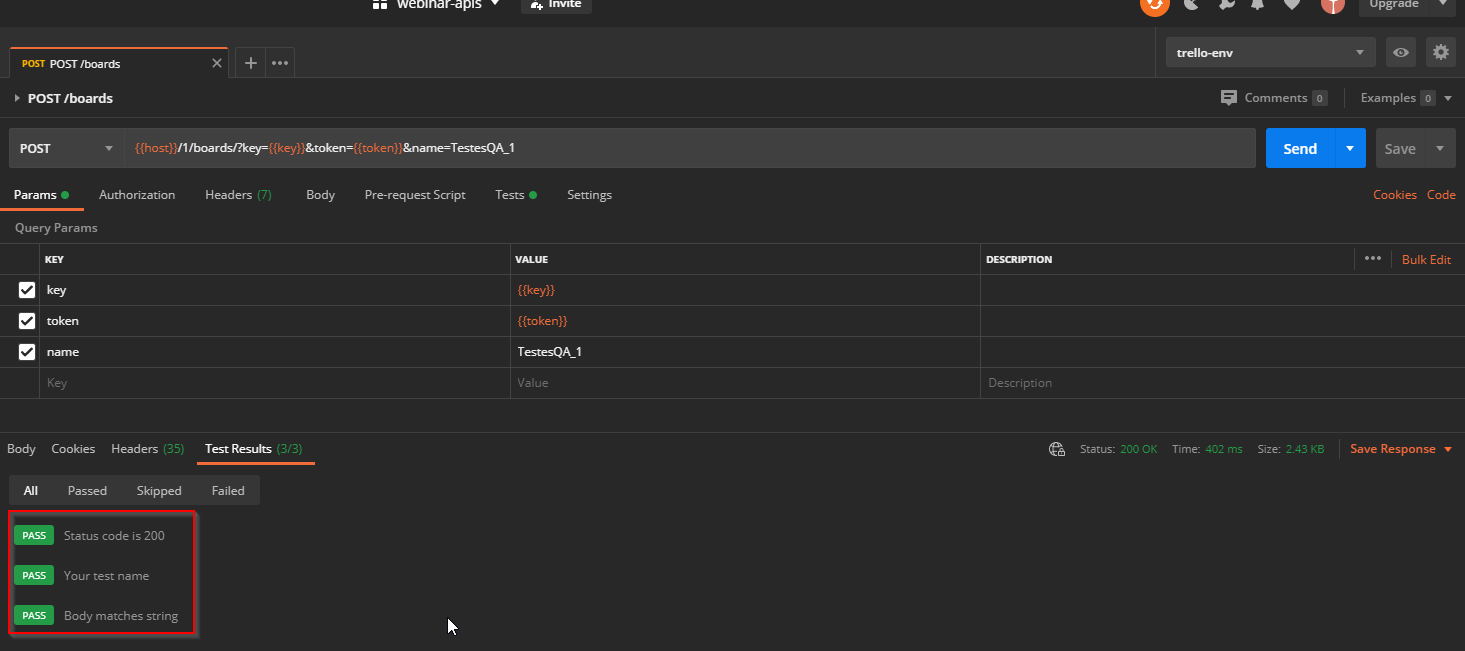
    pm.expect(pm.response.text()).to.include("idOrganization"); // validando se no arquivo Body retorna idOrganization

    pm.expect(pm.response.text()).to.include("closed"); // validando se no arquivo Body retorna closed

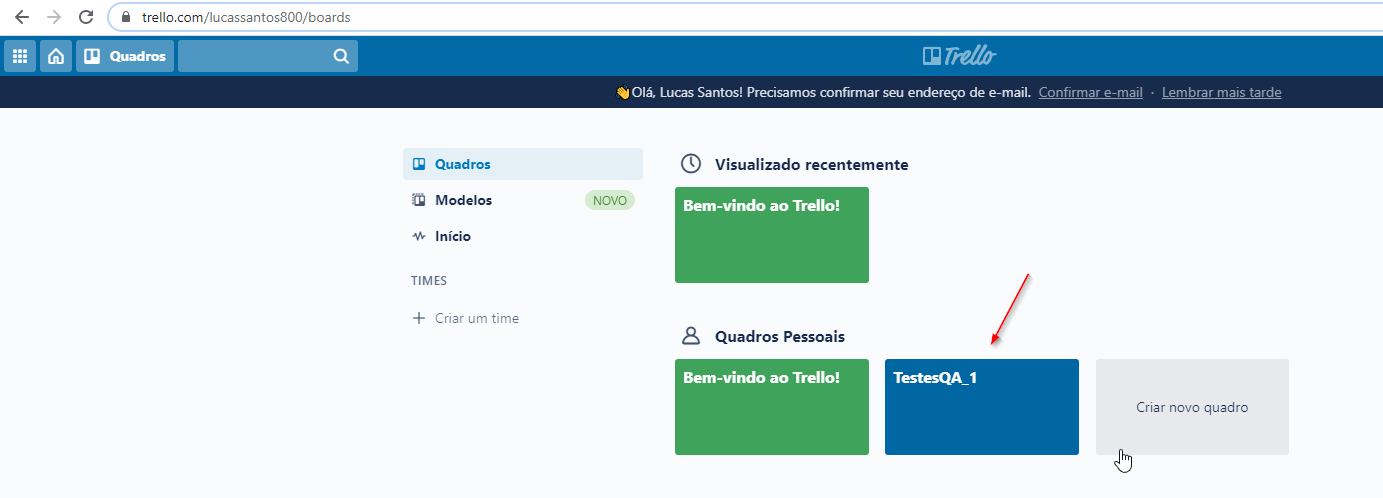
    pm.expect(pm.response.text()).to.include("idEnterprise");  // validando se no arquivo Body retorna idEnterprise

});

Executando o teste clicando em SEND.



Verificando no site.

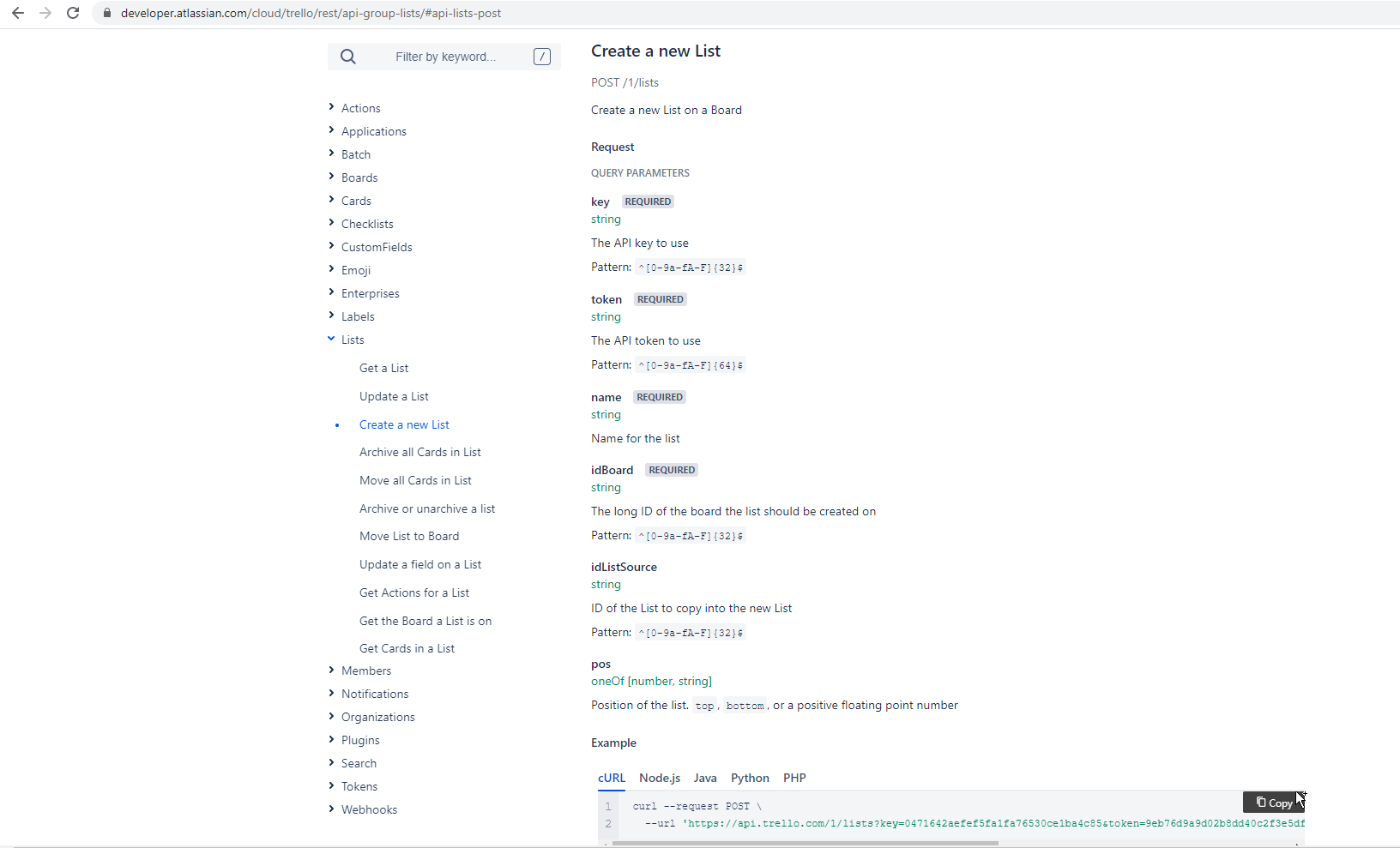


**CRIANDO UMA LISTA NO BOARD**

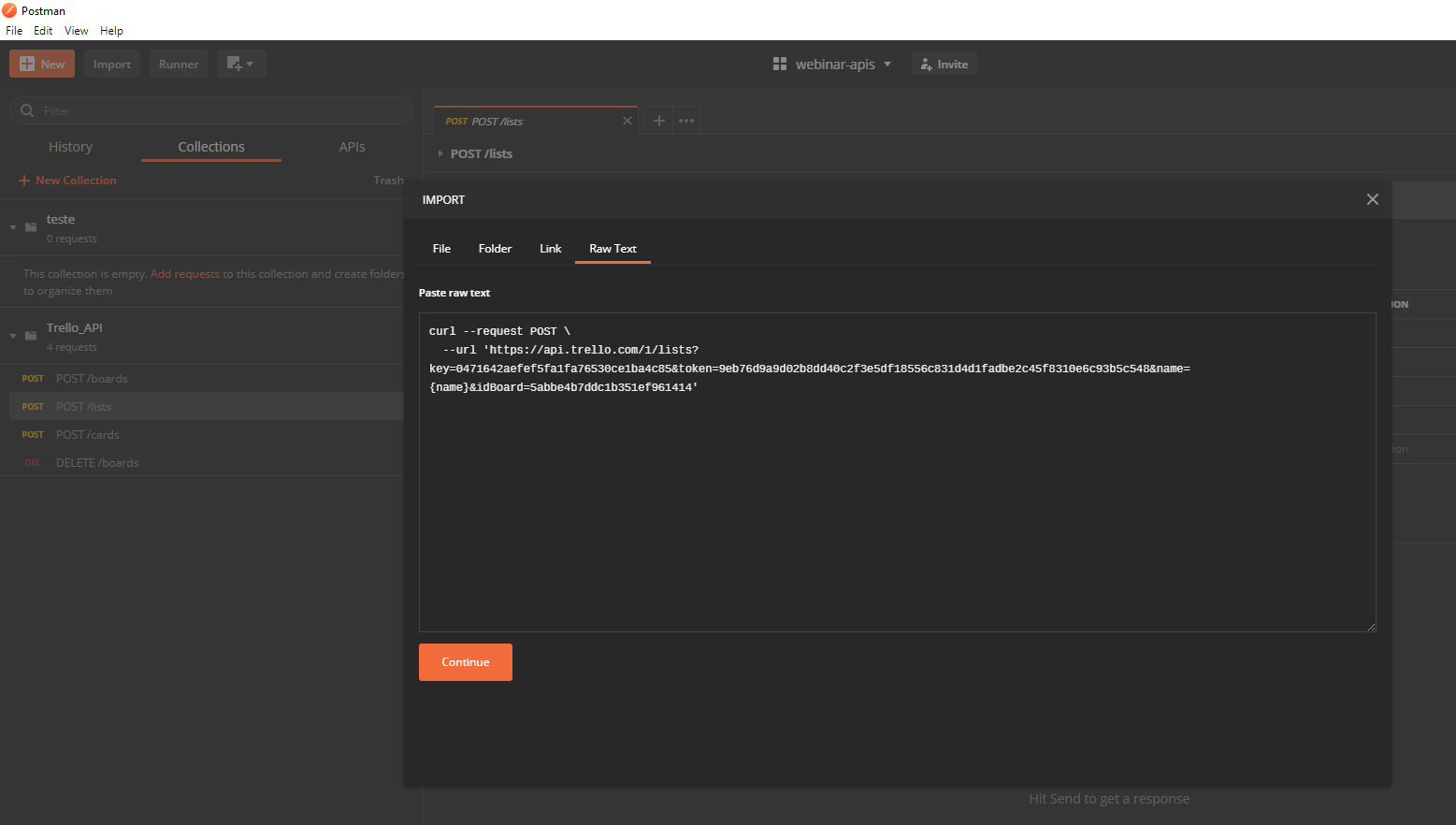
Acessar a documentação da API

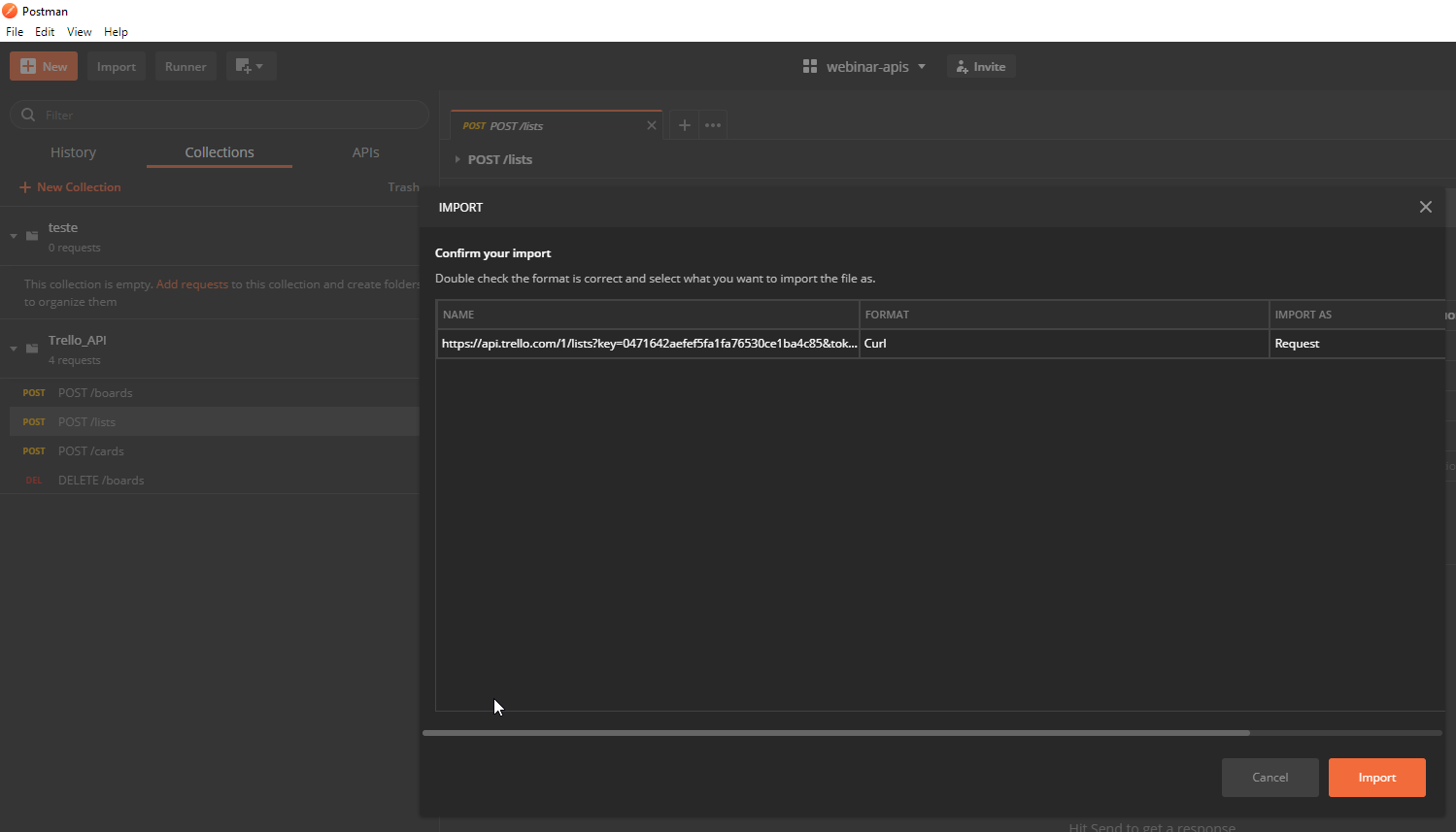
<https://developer.atlassian.com/cloud/trello/rest/api-group-lists/#api-lists-post>

Encontrar a opção de criar uma Lista ao board e copiar o CURL



No POSTMAN clicar no "import" selecionar "Raw Text" colar o comando cURL e "Continue".



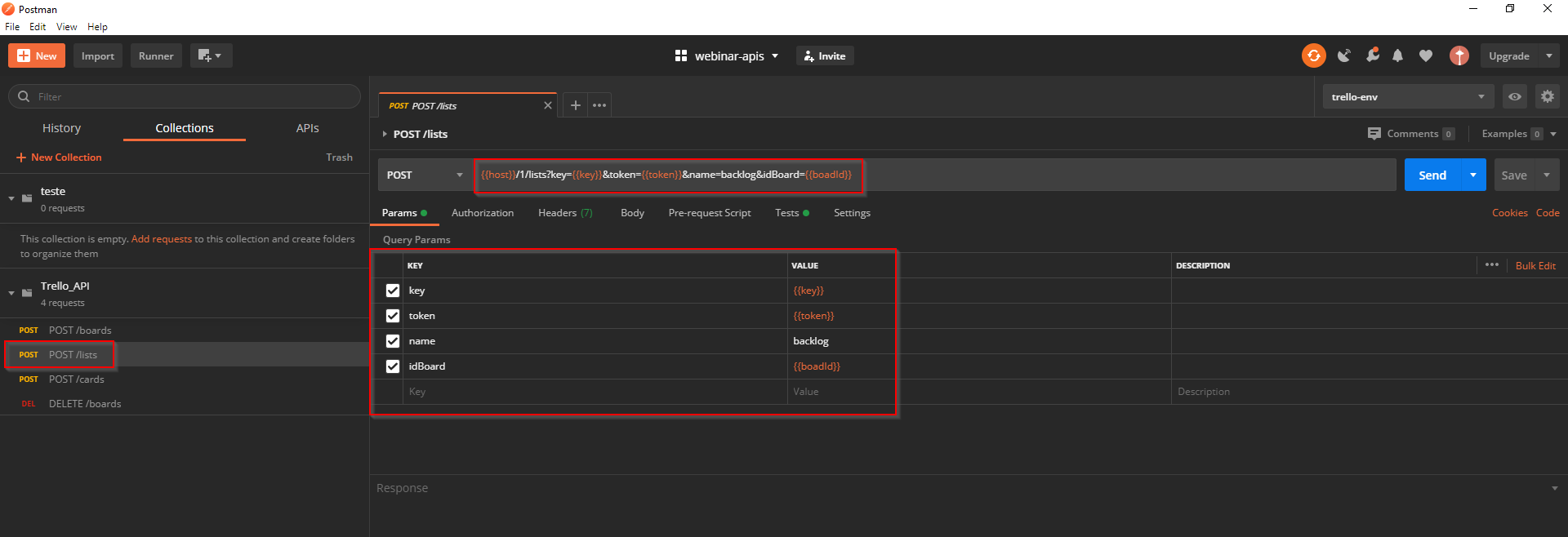


curl --request POST \

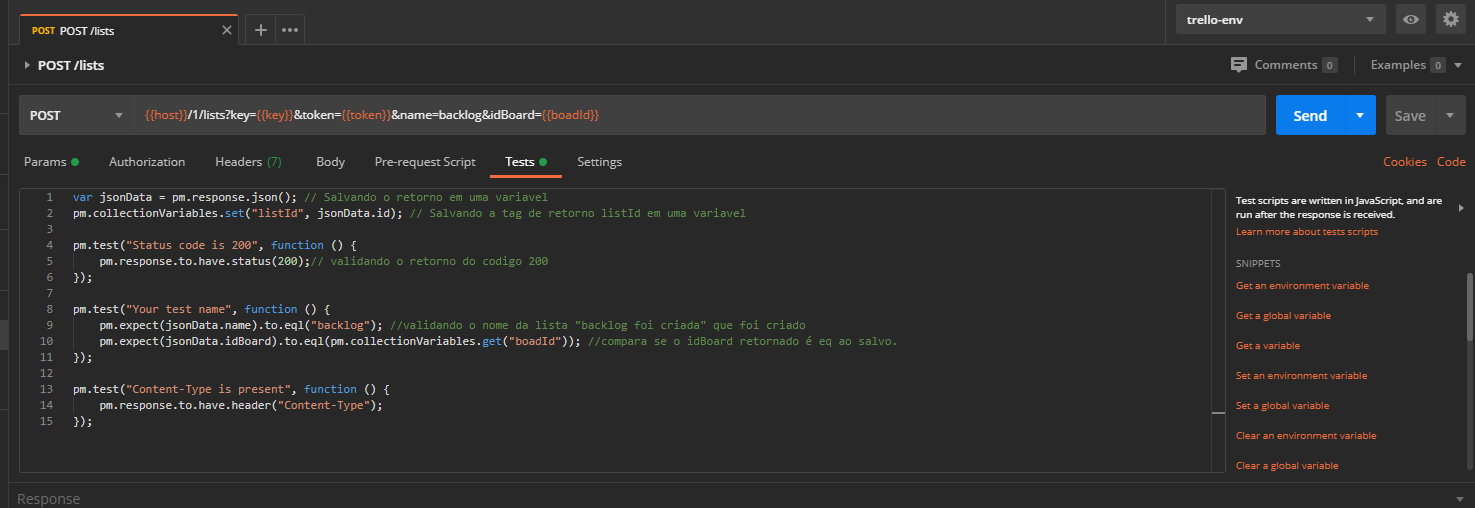
--url 'https://api.trello.com/1/lists?key=0471642aefef5fa1fa76530ce1ba4c85&token=9eb76d9a9d02b8dd40c2f3e5df18556c831d4d1fadbe2c45f8310e6c93b5c548&name={name}&idBoard=5abbe4b7ddc1b351ef961414'

Utilziar as variaveis de apoio no endereço da API {{host}} {key}} {{token}} e a variavel {{boadId}} salva anteriormente na criação do board.

{{host}}/1/lists?key={{key}}&token={{token}}&name=backlog&idBoard={{boadId}}



Para validar o retorno da execução poderá ser adicioando Snnipets



Exemplo do código utilizado para a validação do POST de /boards:

var jsonData = pm.response.json(); // salvando o retorno em uma variavel

pm.collectionVariables.set("listId", jsonData.id); // salvando a tag de retorno listId em uma variavel

pm.test("Status code is 200", function () {

    pm.response.to.have.status(200);// validando o retorno do codigo 200

});

pm.test("Your test name", function () {

    pm.expect(jsonData.name).to.eql("backlog"); //validando o nome da lista "backlog foi criada" que foi criado

    pm.expect(jsonData.idBoard).to.eql(pm.collectionVariables.get("boadId")); //compara se o idBoard retornado é eq ao salvo.

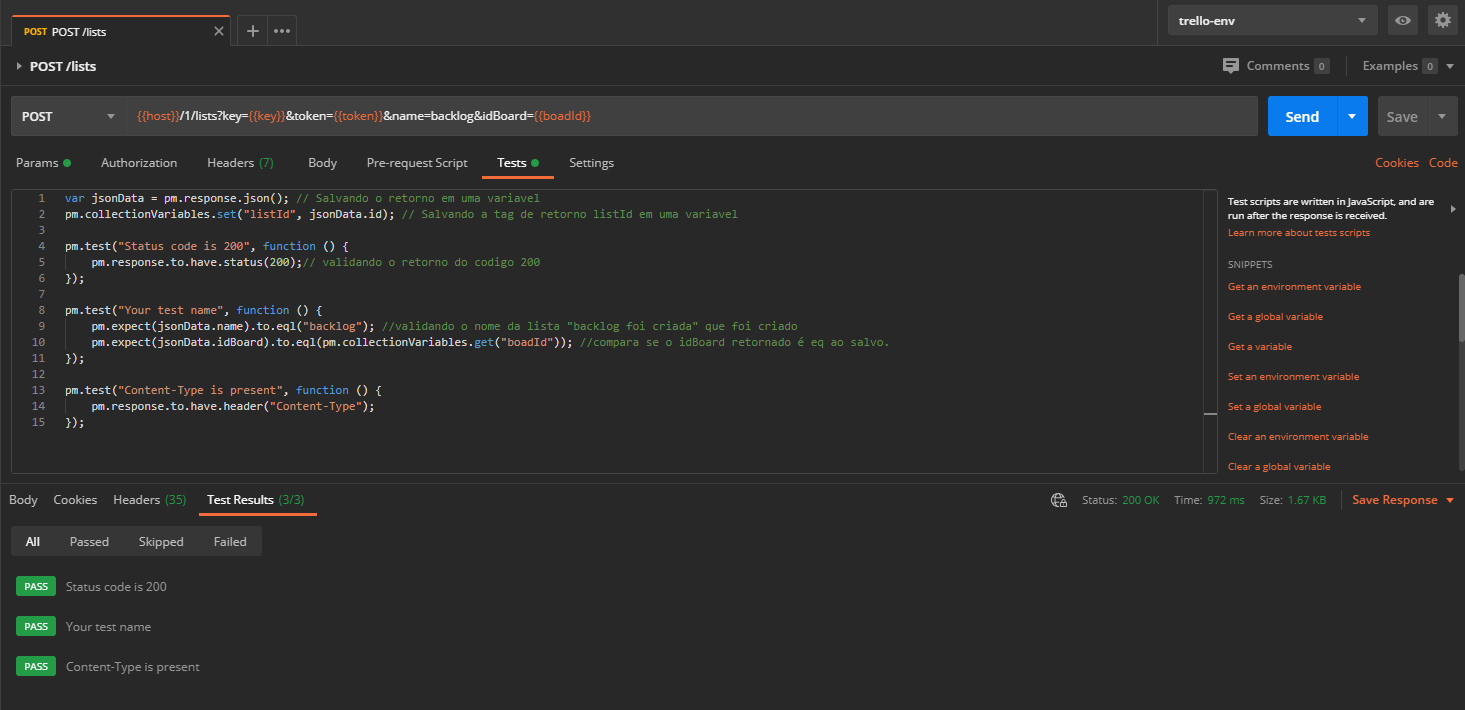
});

pm.test("Content-Type is present", function () {

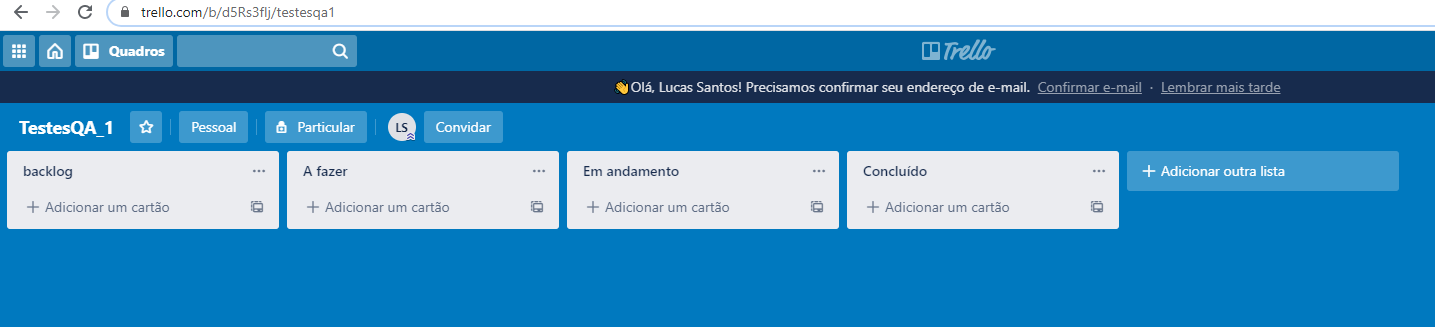
    pm.response.to.have.header("Content-Type");

});

Executando o teste clicando em SEND.



Verificando no site.

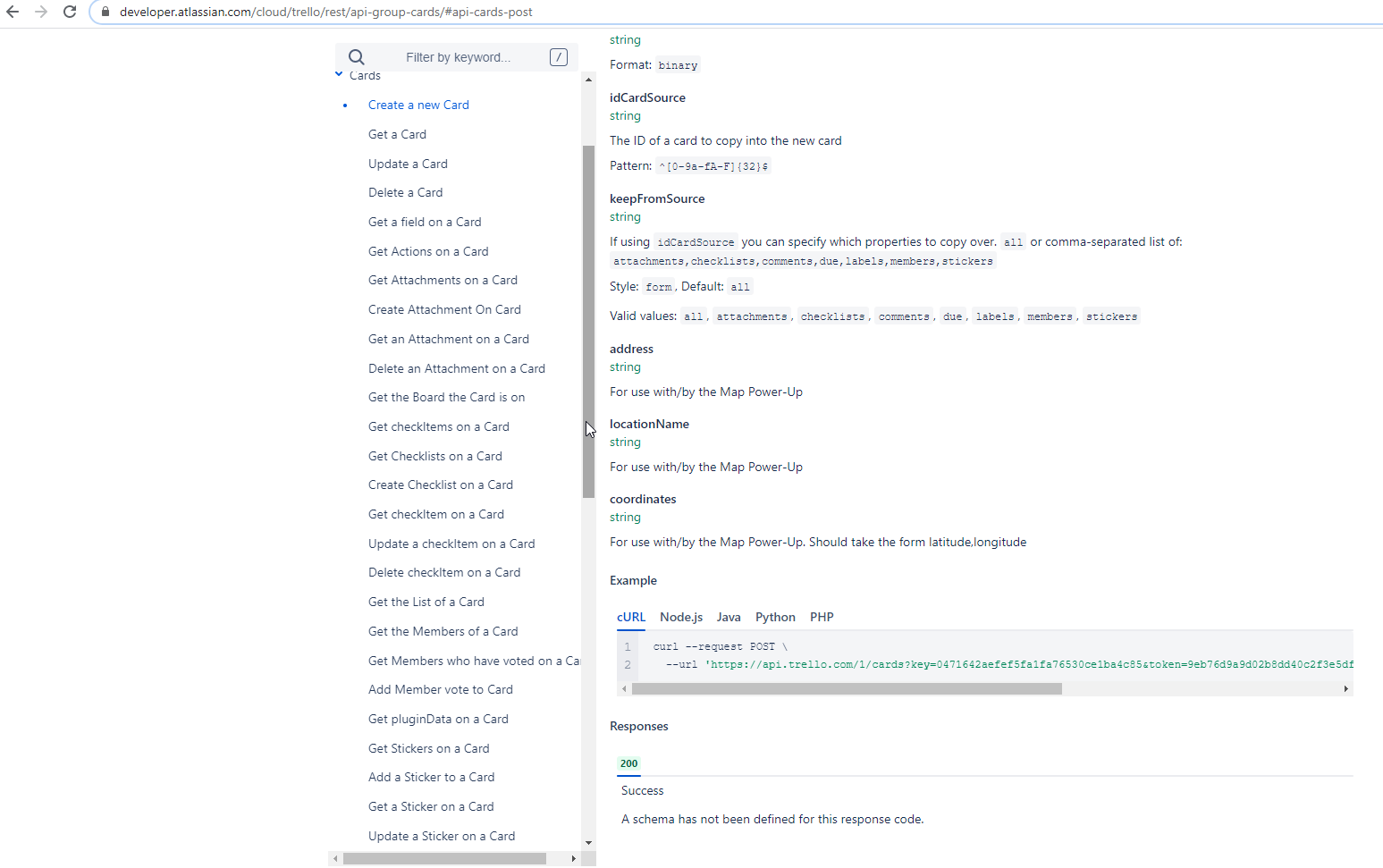


**CRIANDO UM CARD NA LISTA**

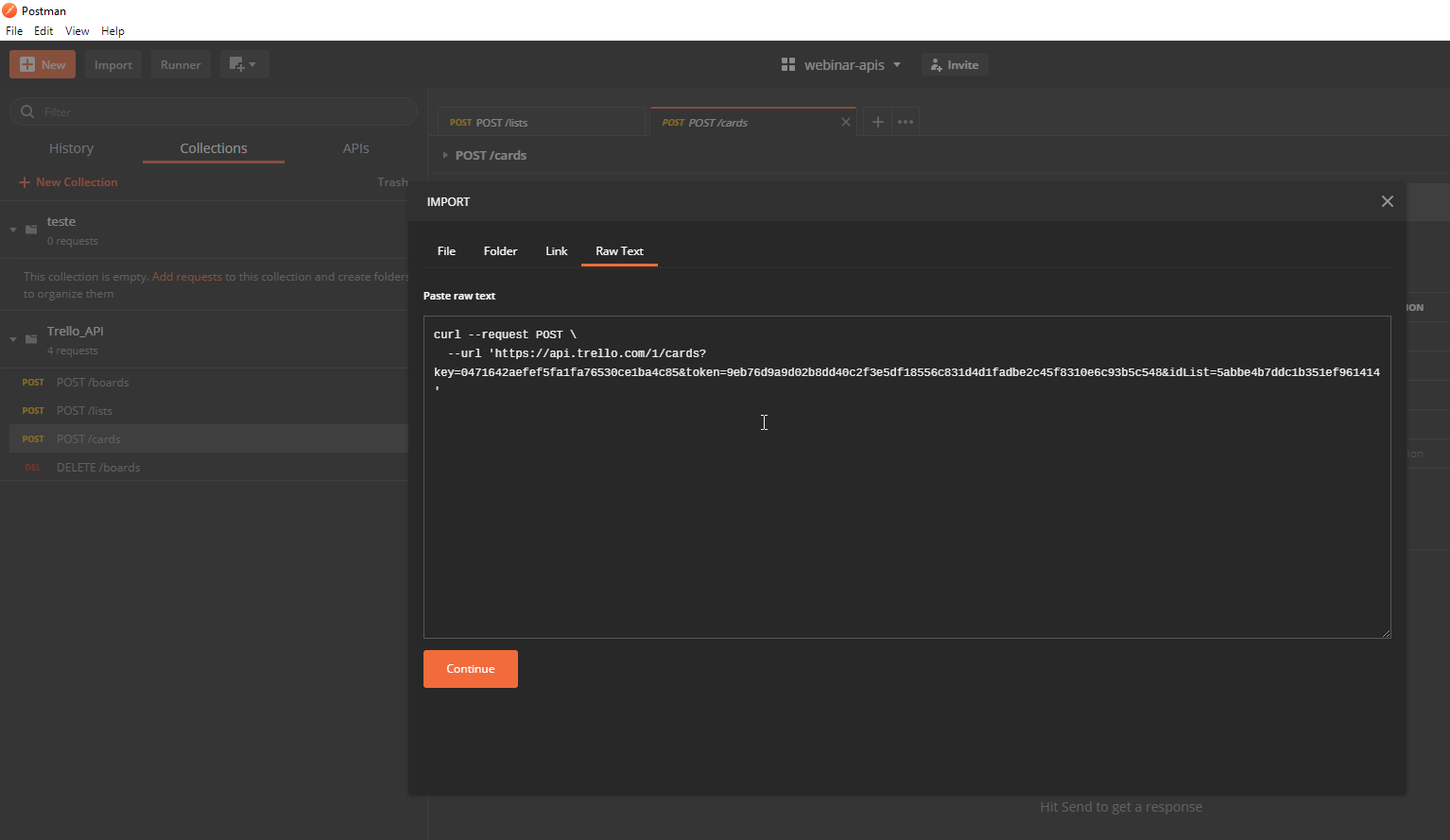
Acessar a documentação da API

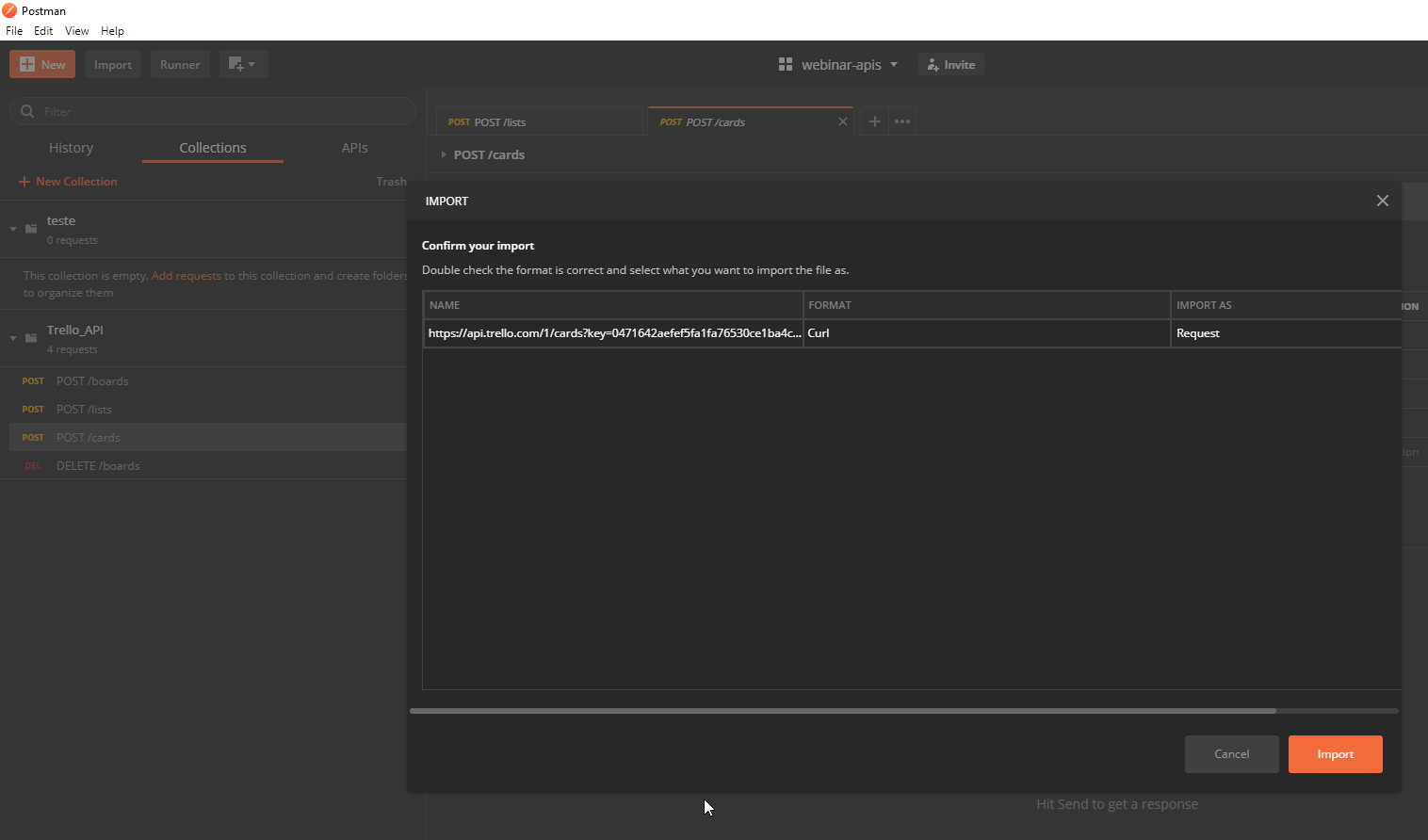
<https://developer.atlassian.com/cloud/trello/rest/api-group-lists/#api-lists-post>

Encontrar a opção de criar um card a uma lista e copiar o CURL



No POSTMAN clicar no "import" selecionar "Raw Text" colar o comando cURL e "Continue".



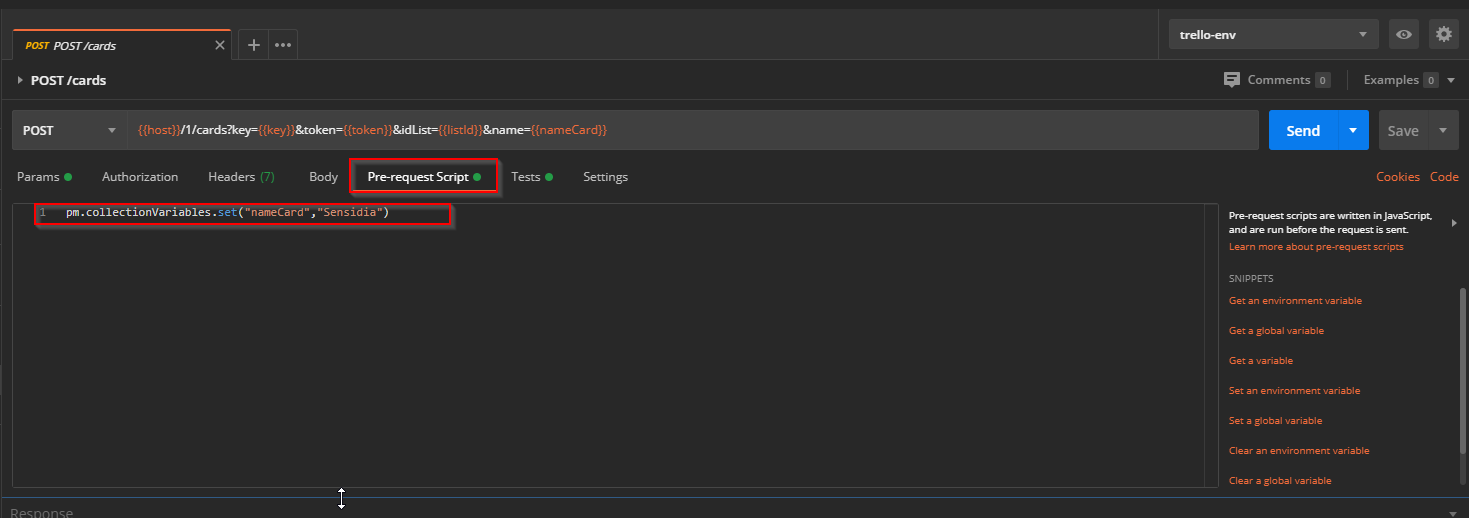


curl --request POST \

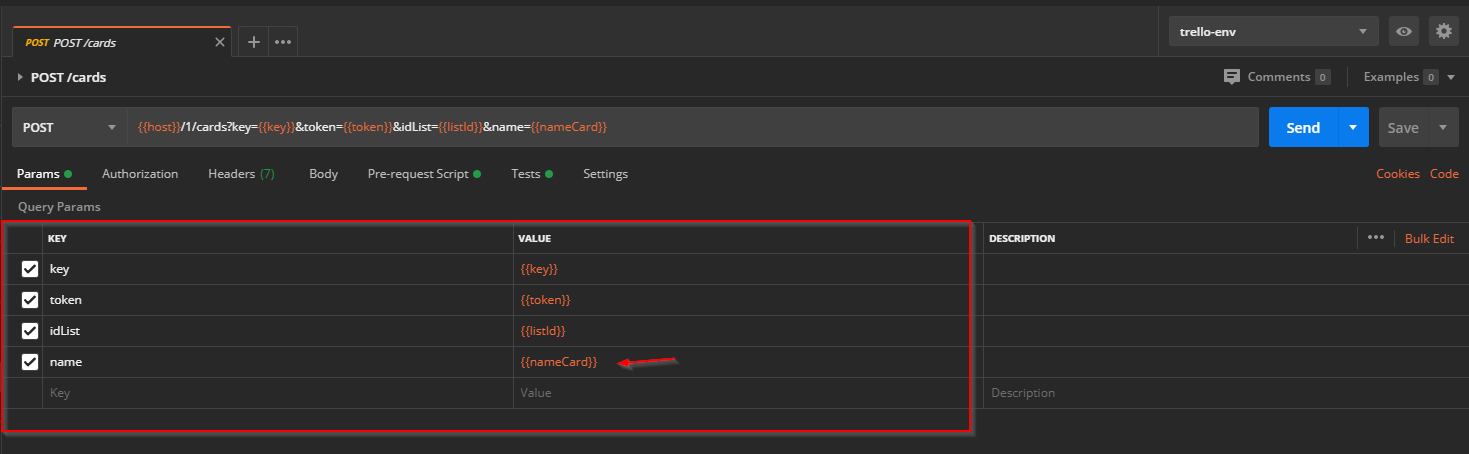
--url 'https://api.trello.com/1/cards?key=0471642aefef5fa1fa76530ce1ba4c85&token=9eb76d9a9d02b8dd40c2f3e5df18556c831d4d1fadbe2c45f8310e6c93b5c548&idList=5abbe4b7ddc1b351ef961414'

Utilziar as variaveis de apoio no endereço da API {key}}, {{token}}, {{listId}} salva anteriormente na criação da lista.

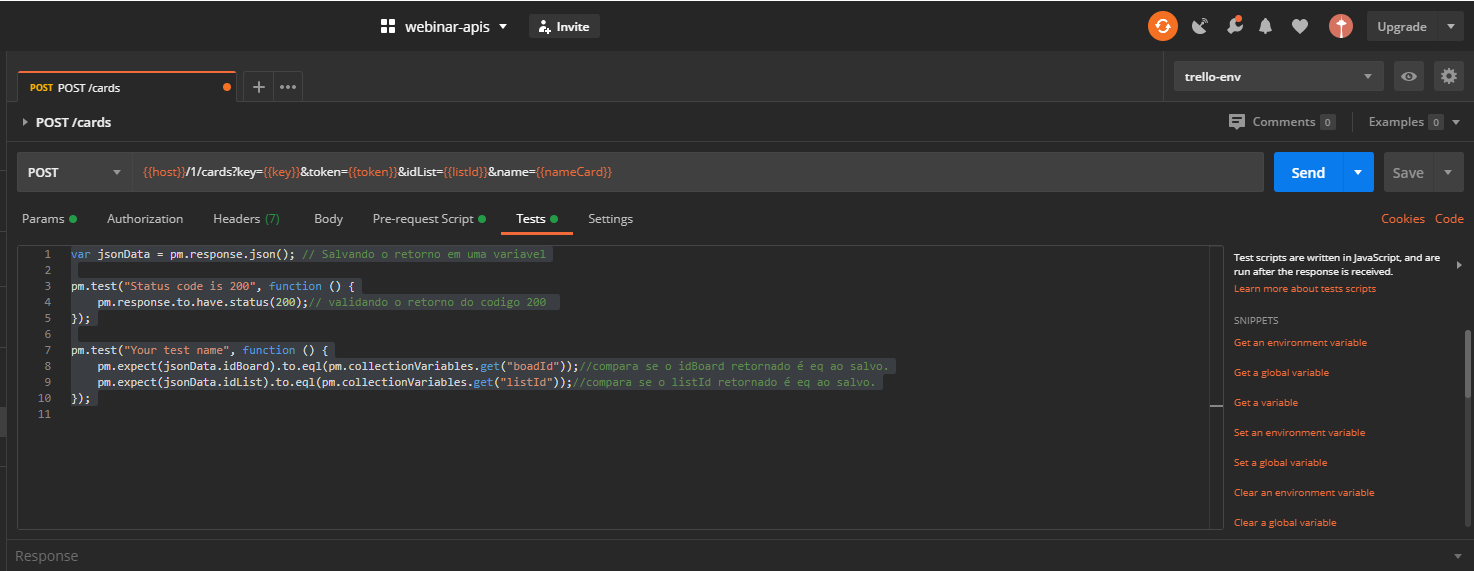
A variavel {{nameCard}} será criada no Pre-requesito Scritp.



Em seguida a variavel poderá ser adicionada como parametro pela API

{{host}}/1/cards?key={{key}}&token={{token}}&idList={{listId}}&name={{nameCard}} 

Para validar o retorno da execução poderá ser adicioando Snnipets



Exemplo do código utilizado para a validação do POST de /boards:

var jsonData = pm.response.json(); // Salvando o retorno em uma variavel

pm.test("Status code is 200", function () {

    pm.response.to.have.status(200);// validando o retorno do codigo 200

});

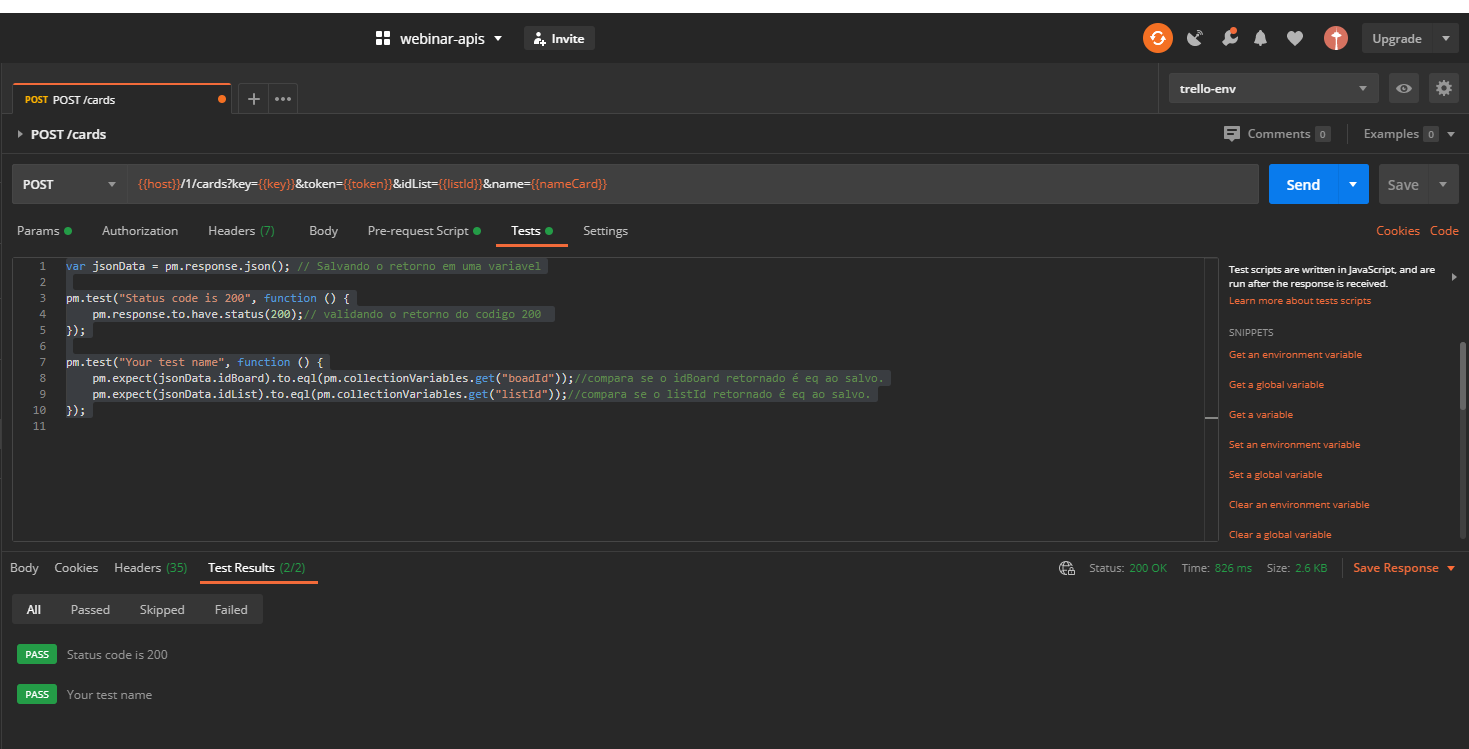
pm.test("Your test name", function () {

    pm.expect(jsonData.idBoard).to.eql(pm.collectionVariables.get("boadId"));//compara se o idBoard retornado é eq ao salvo.

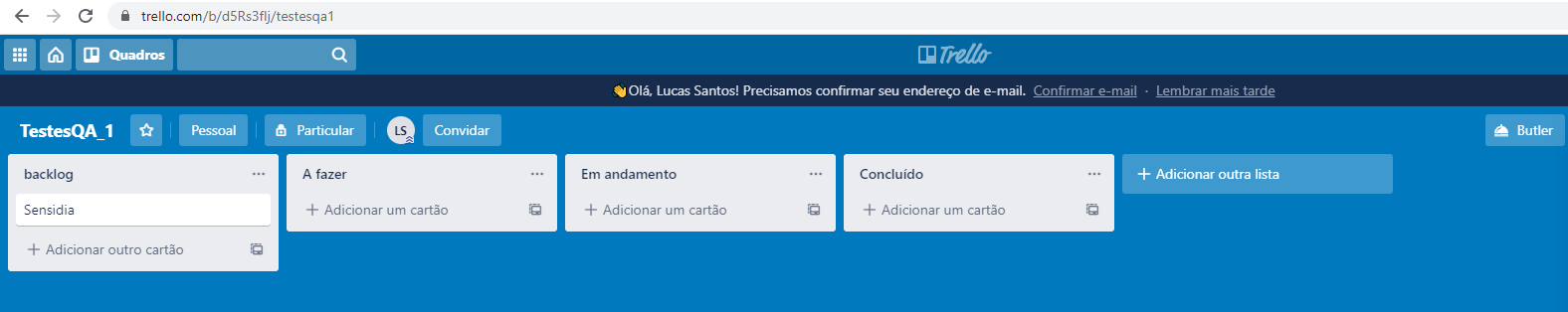
    pm.expect(jsonData.idList).to.eql(pm.collectionVariables.get("listId"));//compara se o listId retornado é eq ao salvo.

});

Executando o teste clicando em SEND.



Verificando no site.

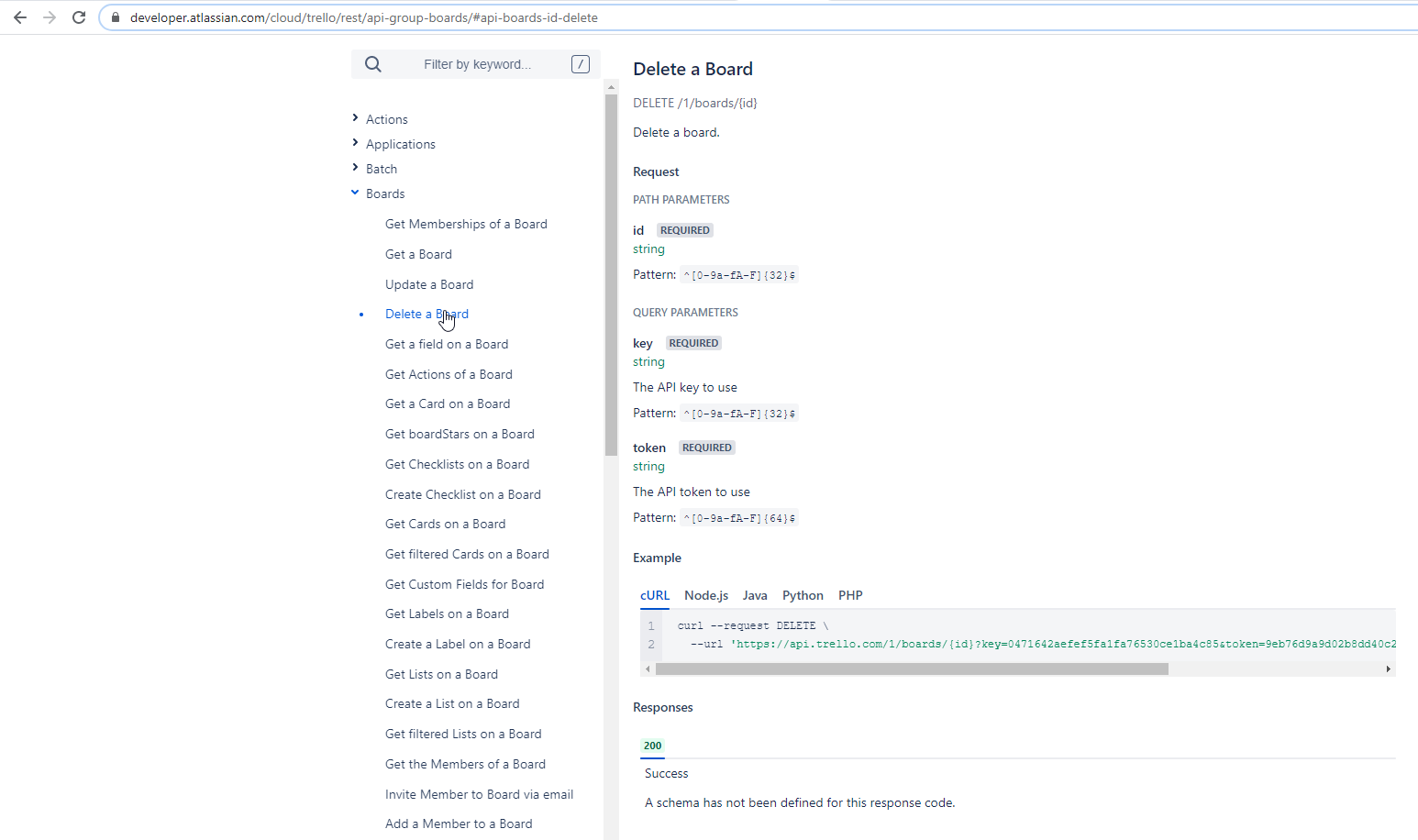


**DELETANDO O BOARD**

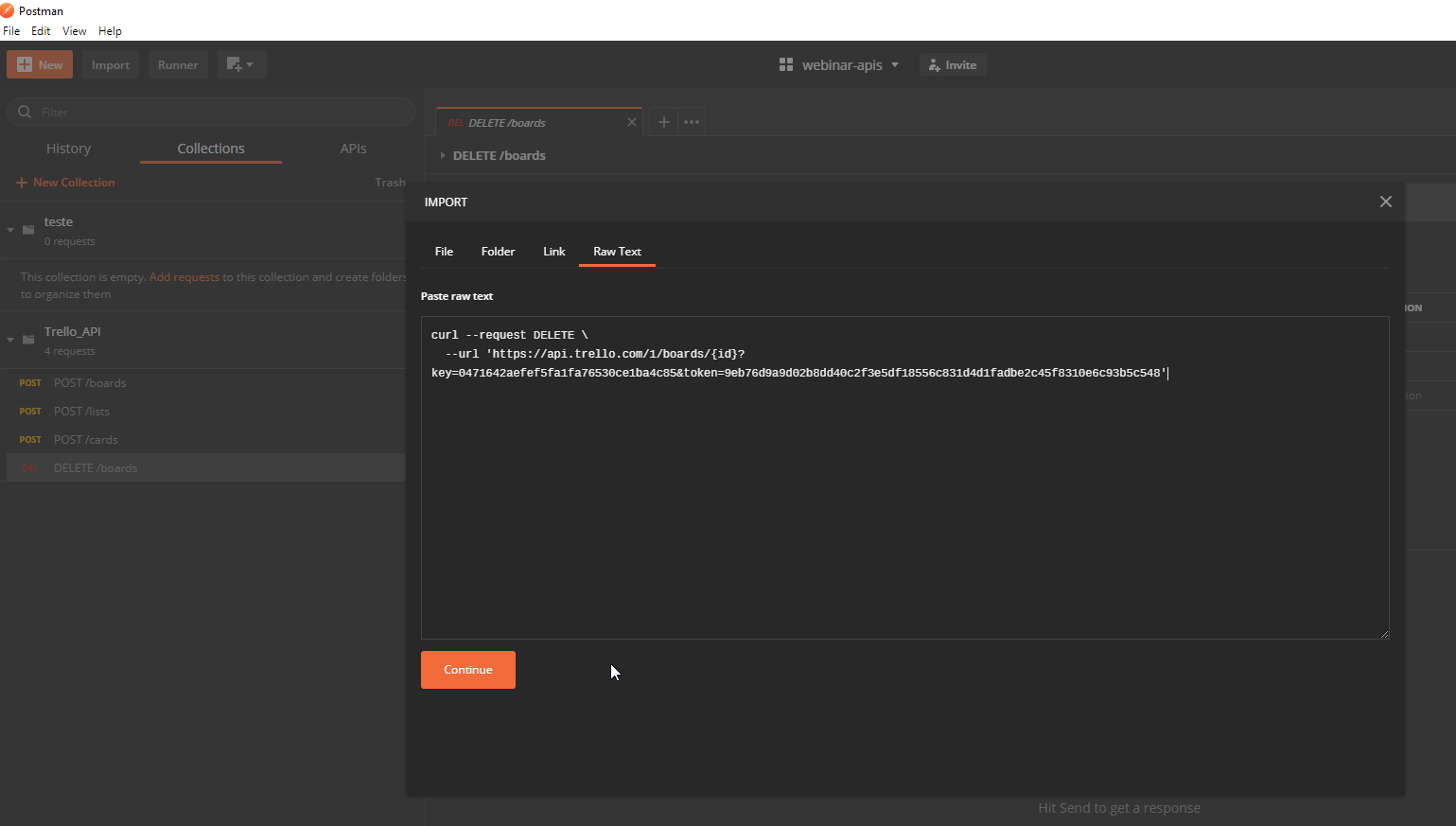
Acessar a documentação da API

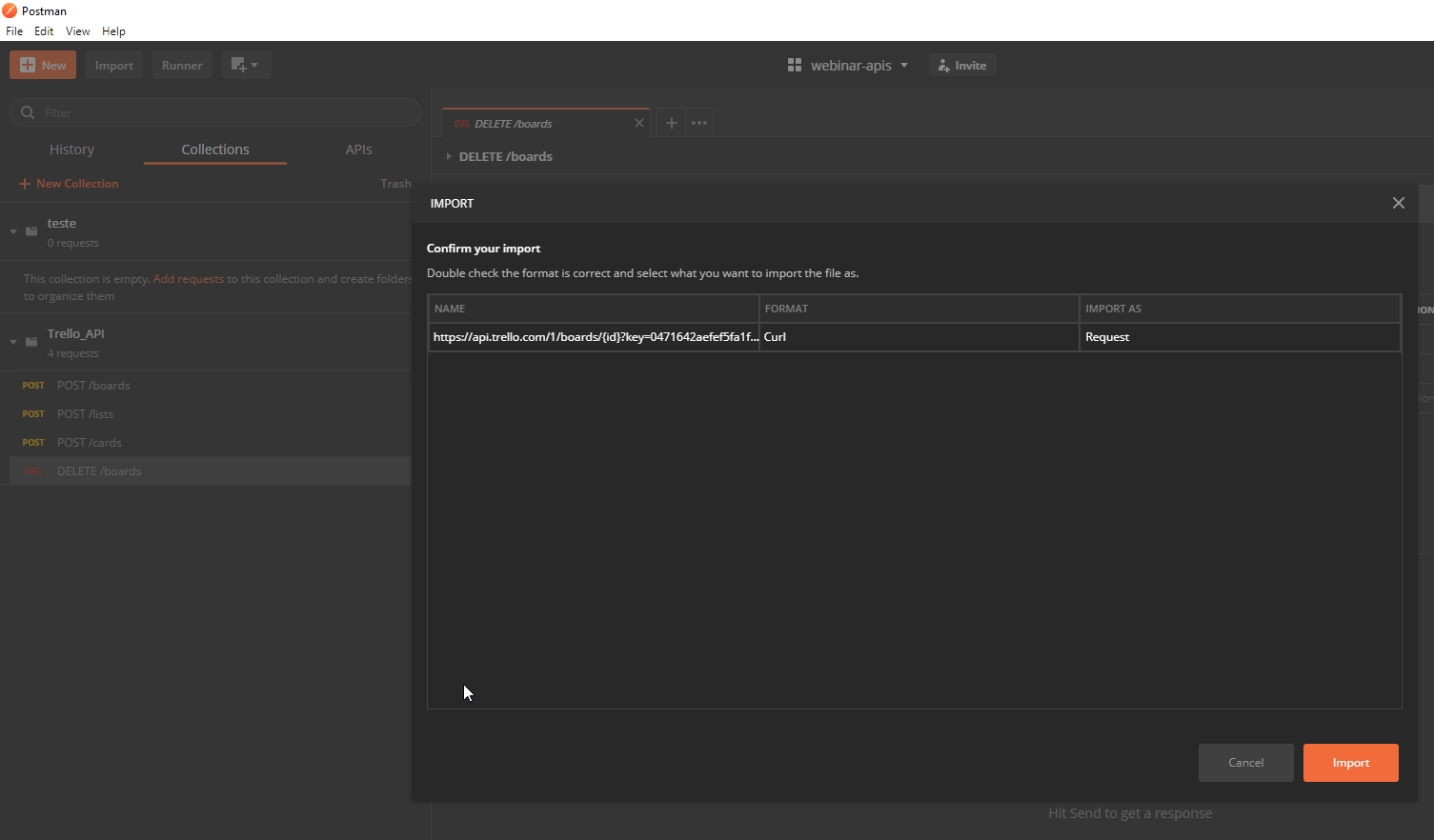
<https://developer.atlassian.com/cloud/trello/rest/api-group-boards/#api-boards-id-delete>

Encontrar a opção de criar um card a uma lista e copiar o CURL



No POSTMAN clicar no "import" selecionar "Raw Text" colar o comando cURL e "Continue".



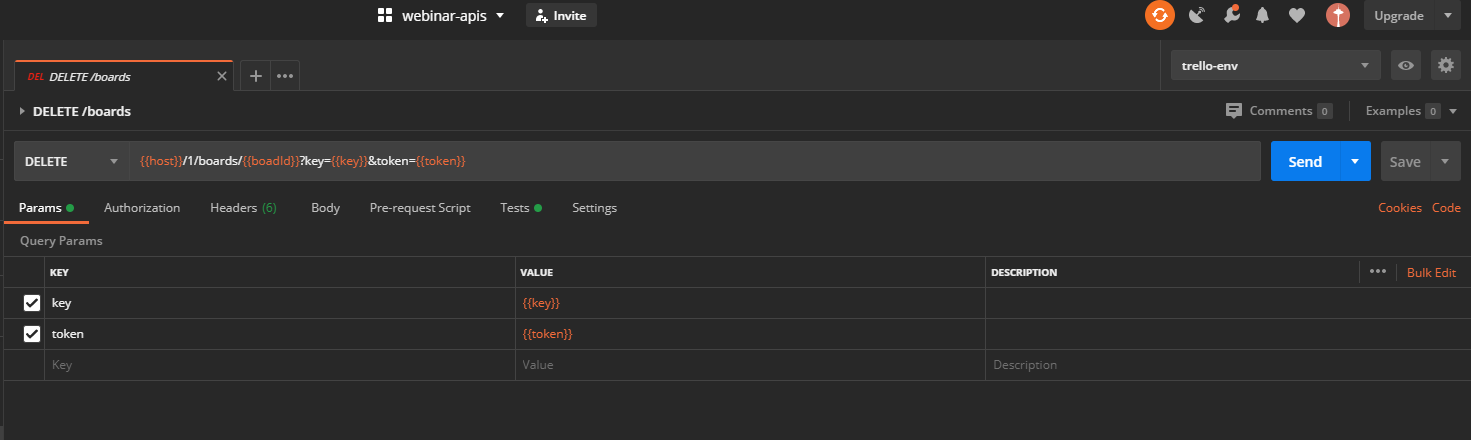


curl --request DELETE \

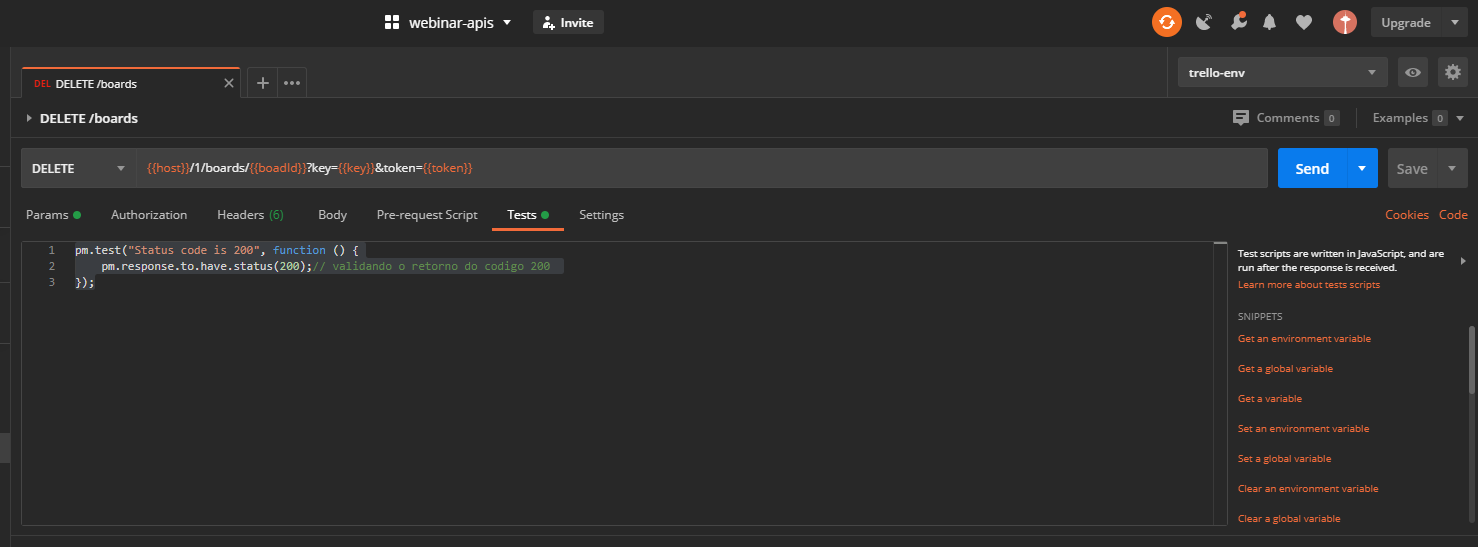
--url 'https://api.trello.com/1/boards/{id}?key=0471642aefef5fa1fa76530ce1ba4c85&token=9eb76d9a9d02b8dd40c2f3e5df18556c831d4d1fadbe2c45f8310e6c93b5c548'

Utilziar as variaveis de apoio no endereço da API {key}} e {{token}}salva anteriormente na criação da lista.

Em seguida a variavel poderá ser adicionada como parametro pela API

{{host}}/1/boards/{{boadId}}?key={{key}}&token={{token}} 

Para validar o retorno da execução poderá ser adicioando Snnipets



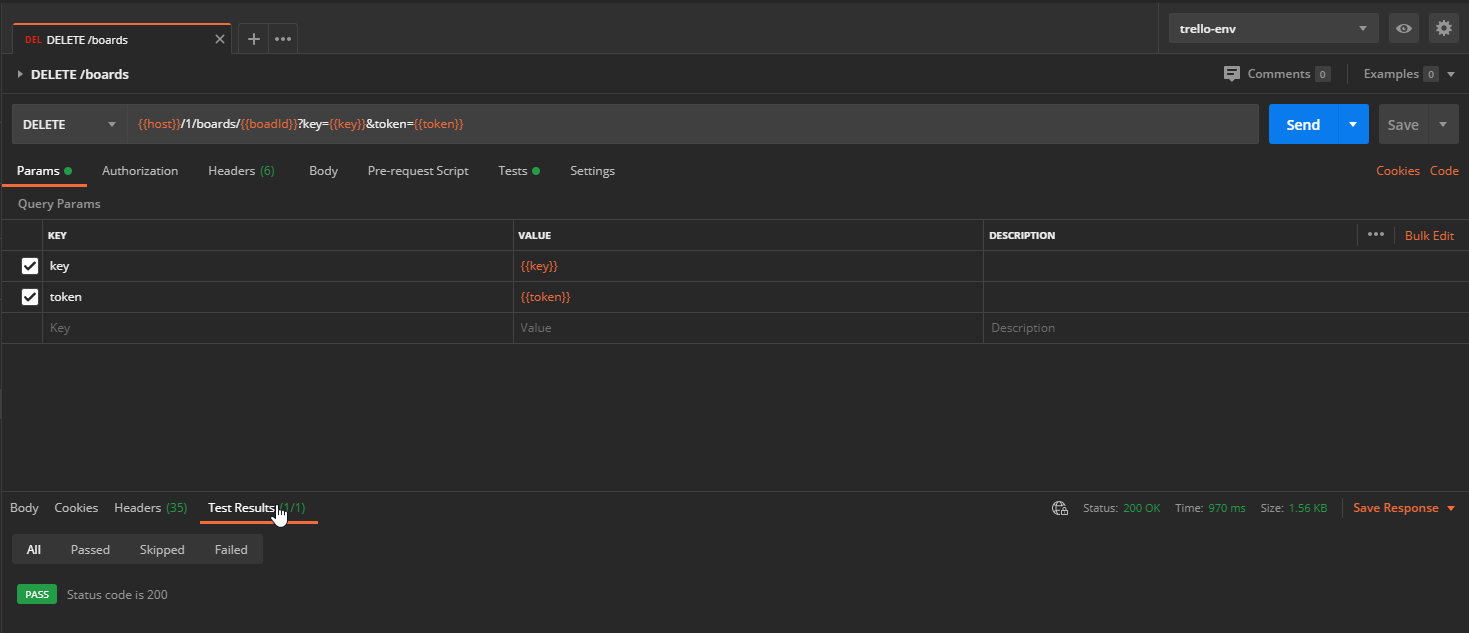
Exemplo do código utilizado para a validação do POST de /boards:

pm.test("Status code is 200", function () {

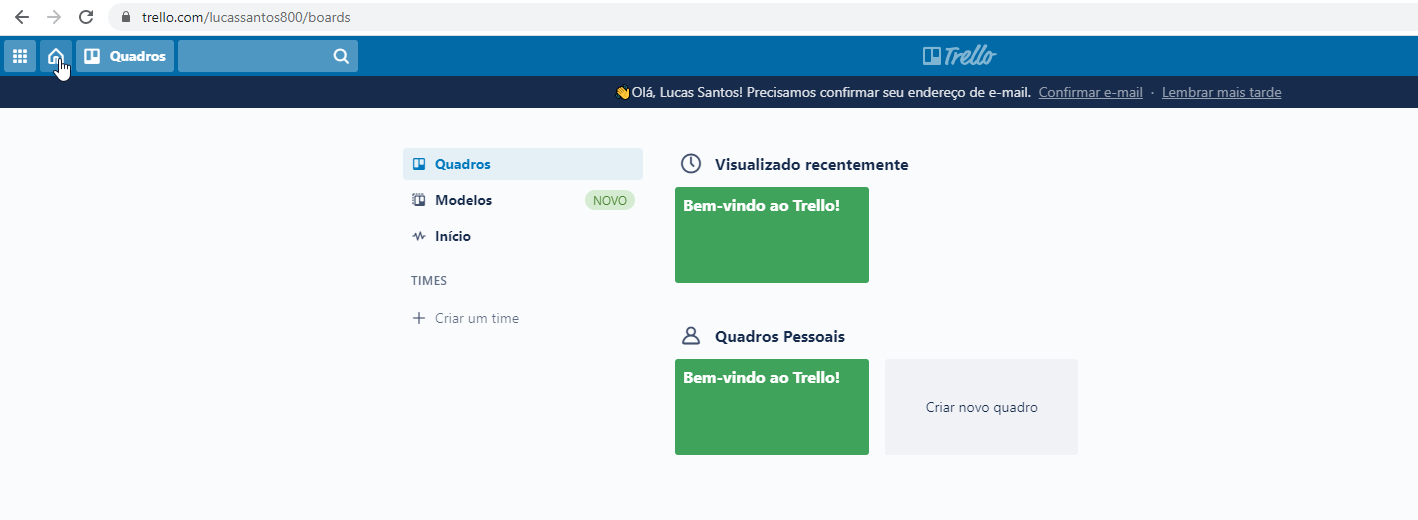
    pm.response.to.have.status(200);// validando o retorno do codigo 200

});

Executando o teste clicando em SEND.

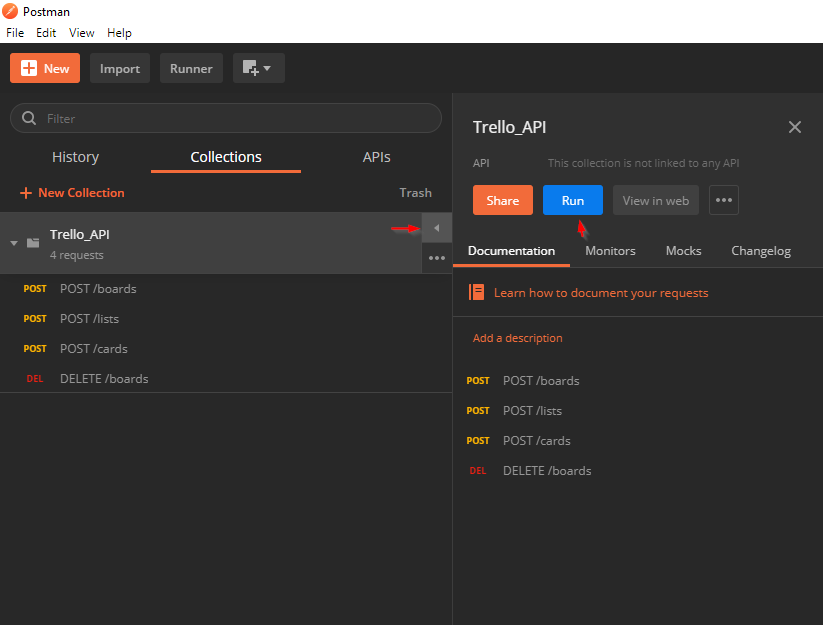


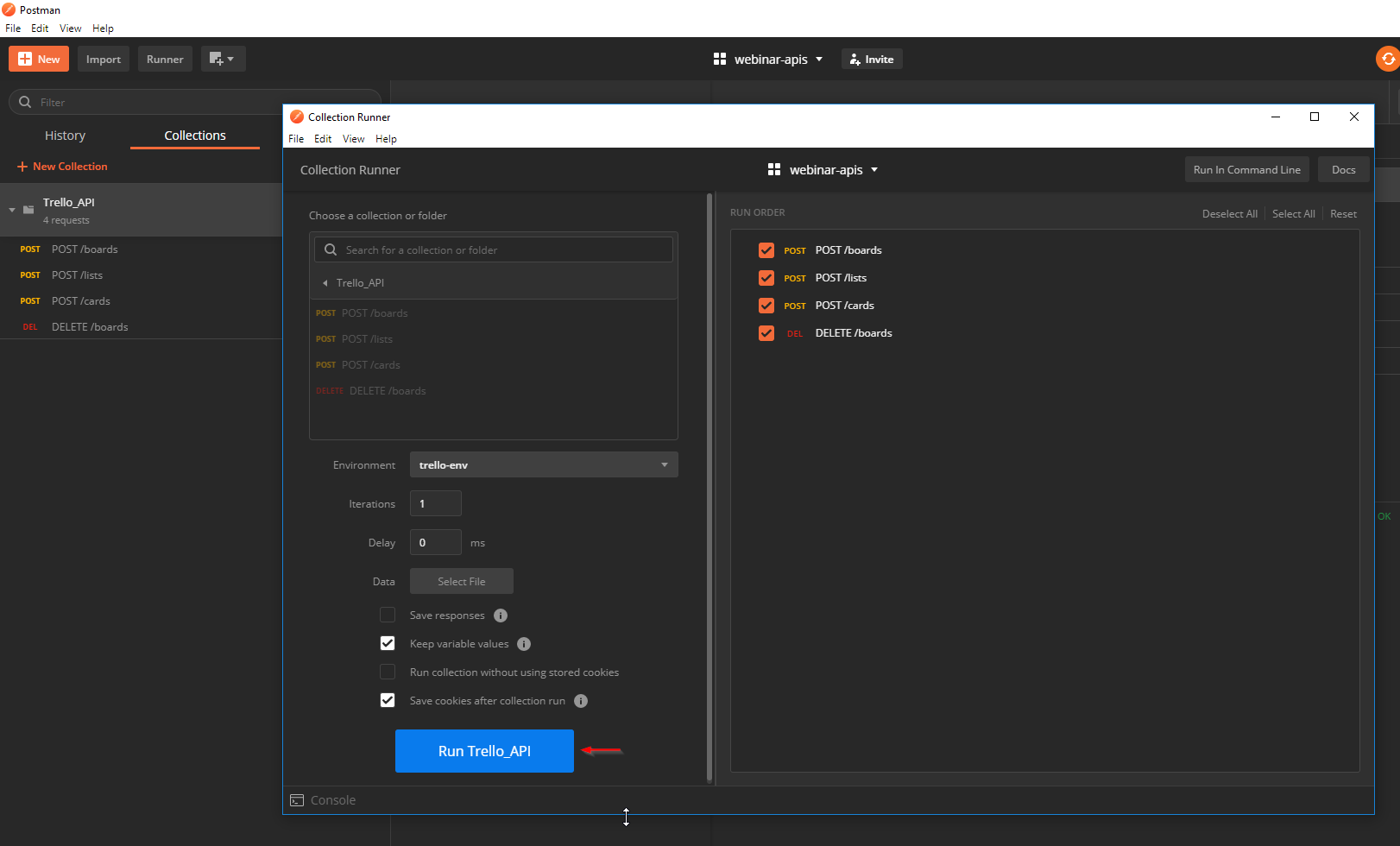
Verificando no site.

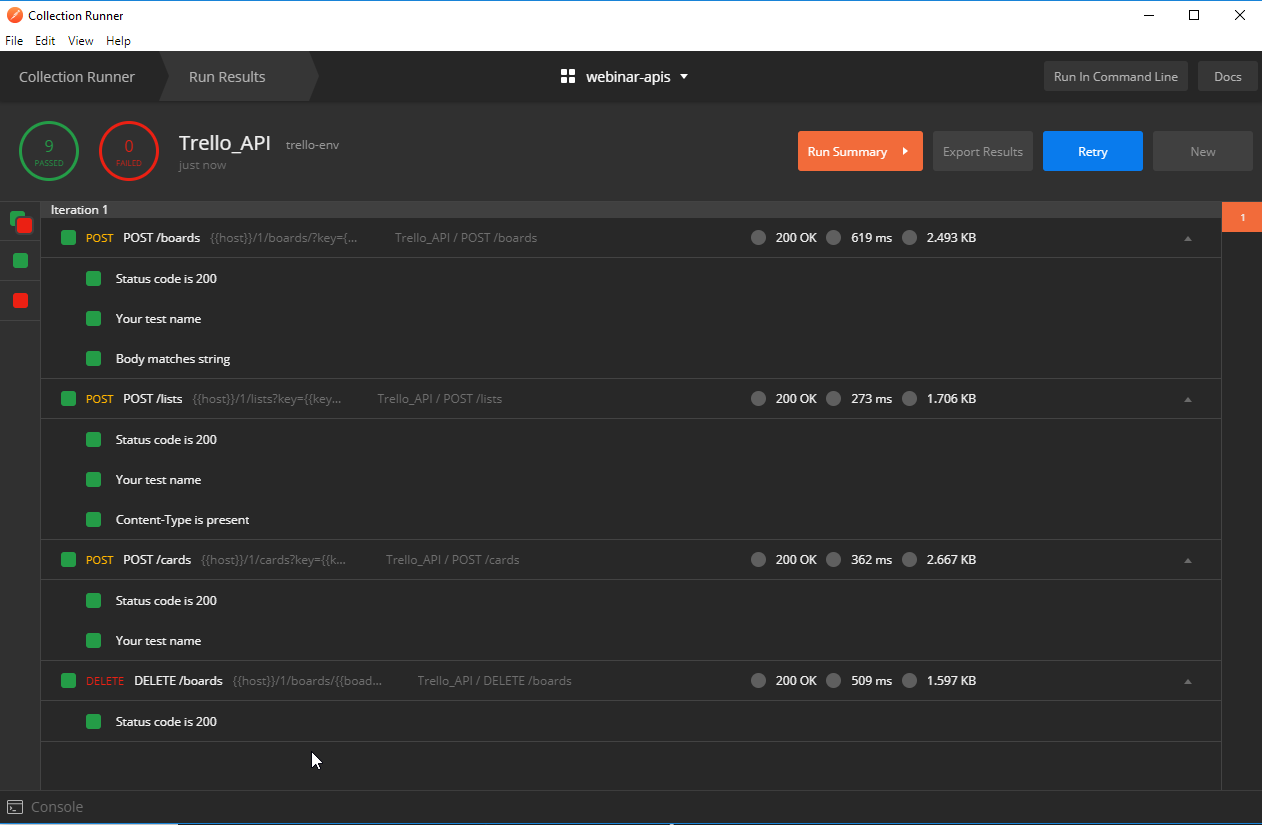


**EXECUTANDO O PROJETO**

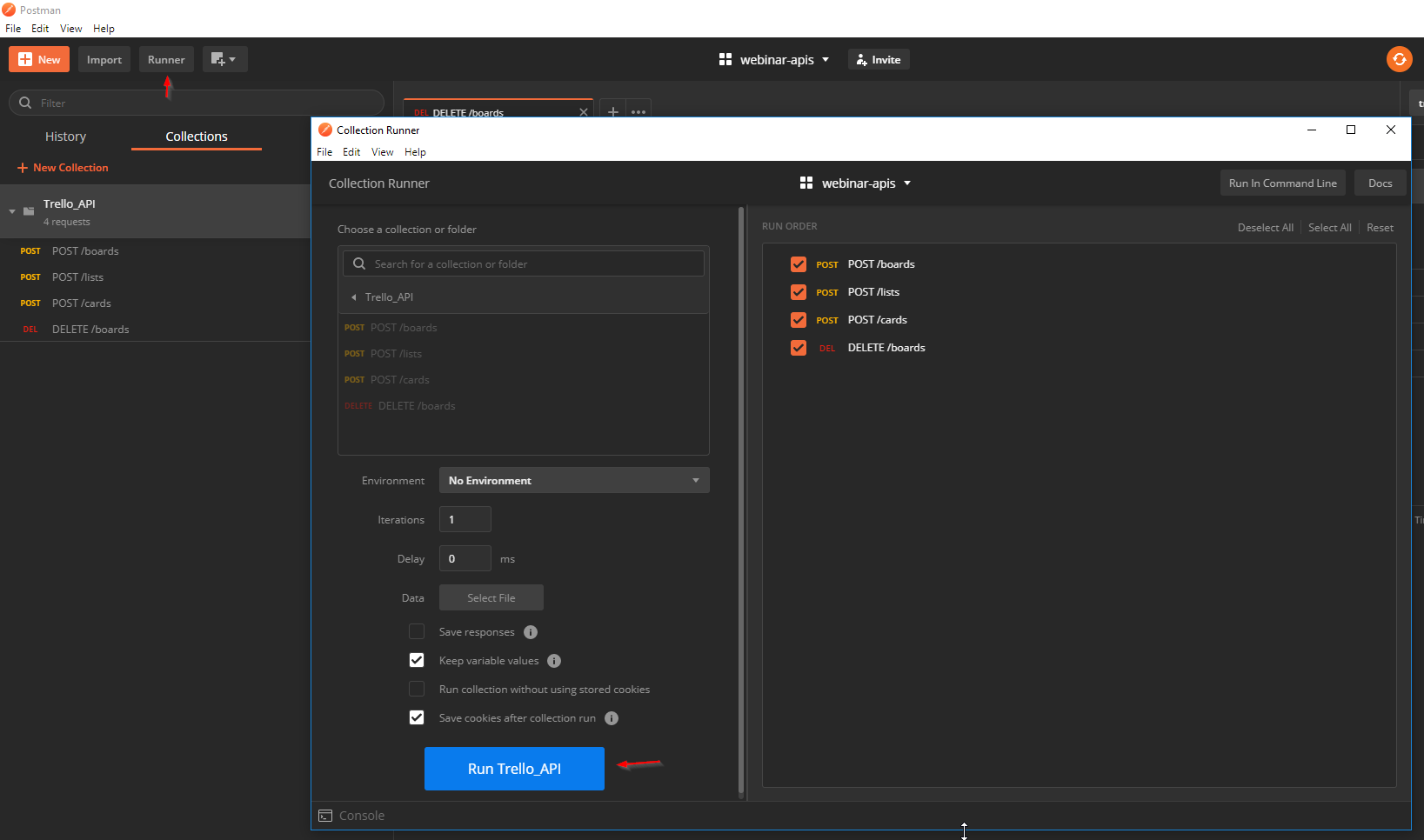
Executando pelo projeto.

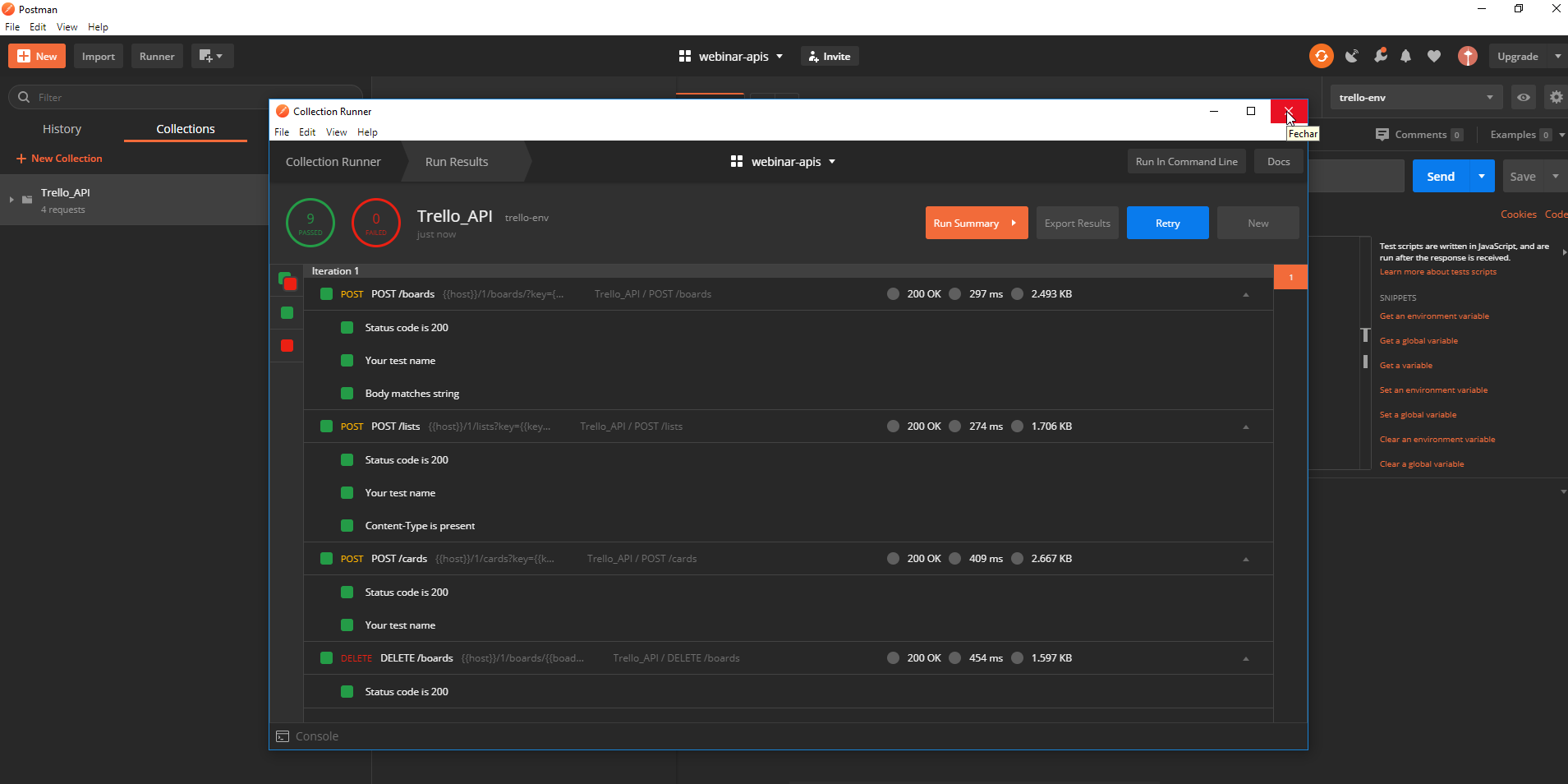






Outra maneira de executar.





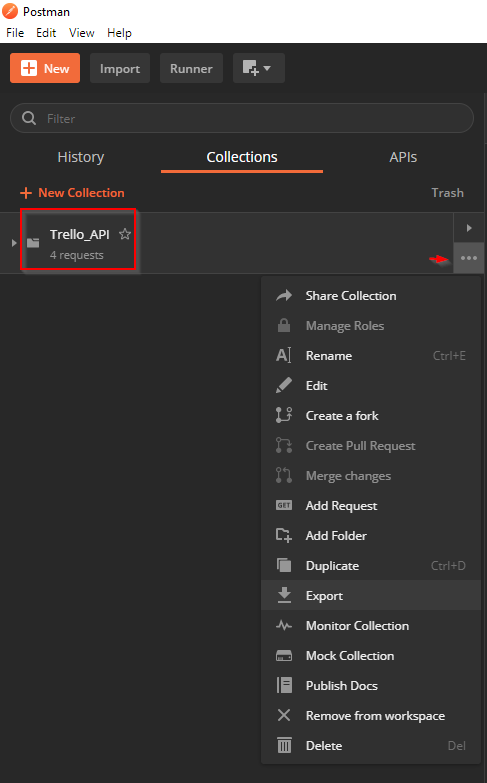
**INSTALAÇÂO NEWMAN**

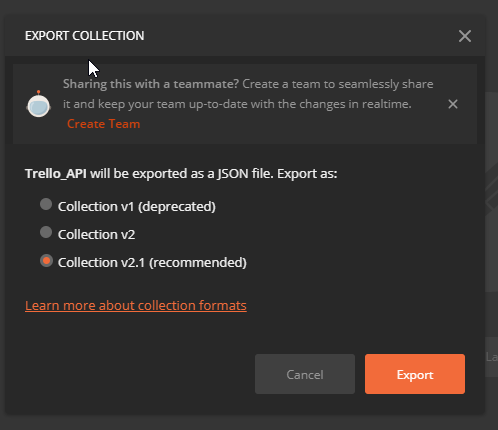
A maneira mais fácil de instalar o Newman é usando o NPM. Se você tiver o Node.js instalado, é mais provável que você também tenha o NPM instalado.

npm install -g newman

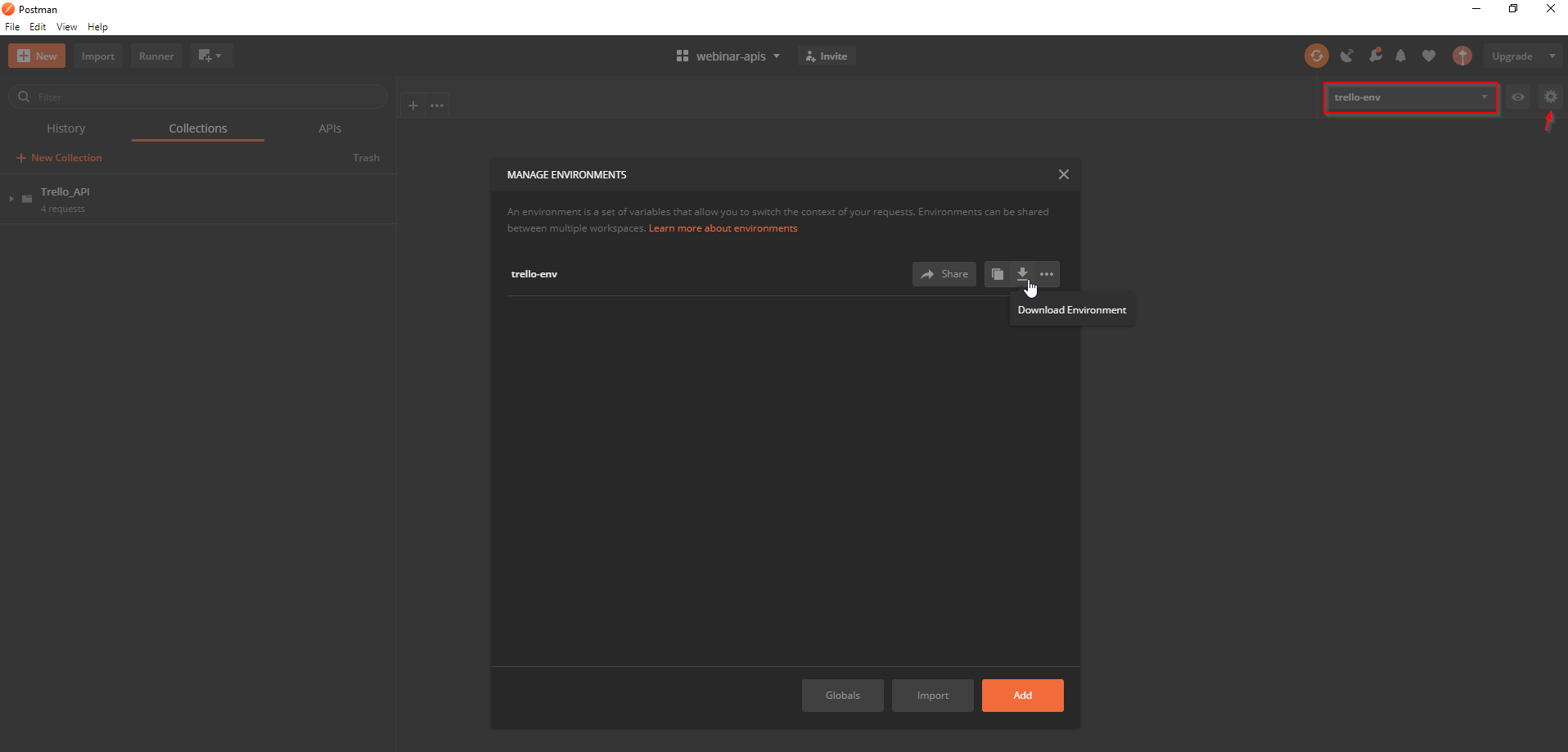
Para executar os testes criado no Postman executando o Newman é necessário exportar a collection e as environment criadas.

Exportando a collection





Exportando as environment



Arquivos exportados na pasta do projeto.



**EXECUTANDO O NEWMAN**

Abrir o cmd no caminho do projeto.

newman run Trello\_API.postman\_collection.json -e trello-env.postman\_environment.json  
