****

**CASE STUDY FOR** QA Developer

*Neste documento apresenta as evidências de execução dos testes relacionados oas desafios 1, 2 e 3. Para informações de como executar os testes consulte o arquivo README.*

**THE CHALLENGE 1 - TEST MOBILE APPLICATION MANUALLY**

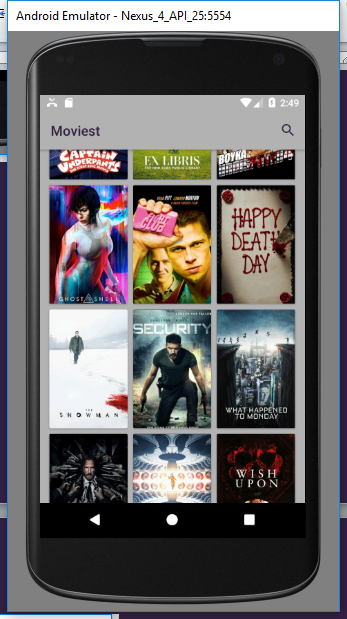
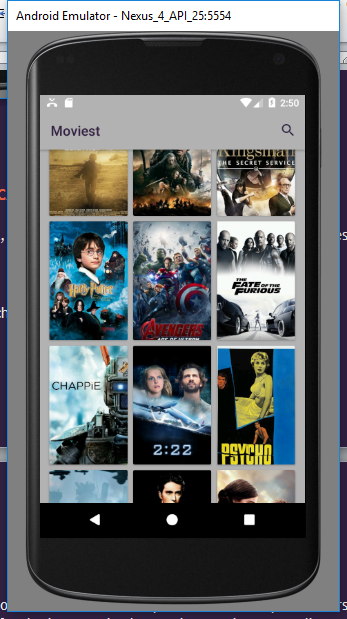
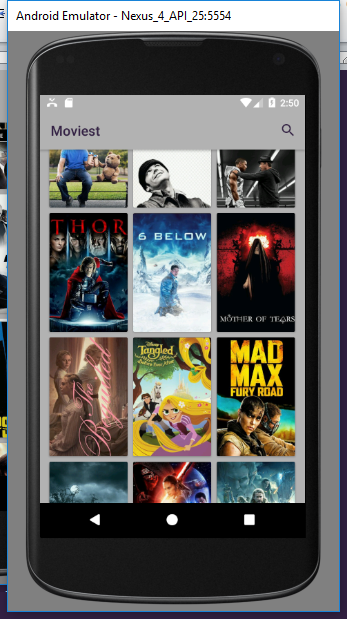
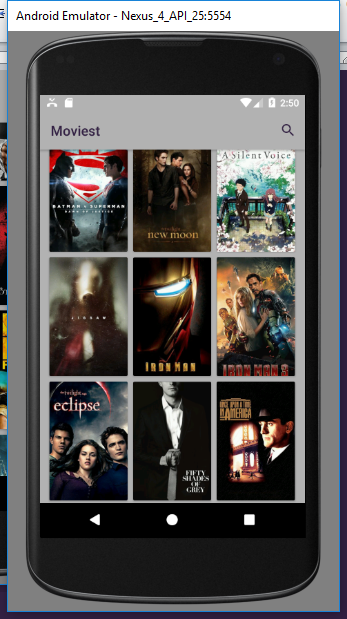
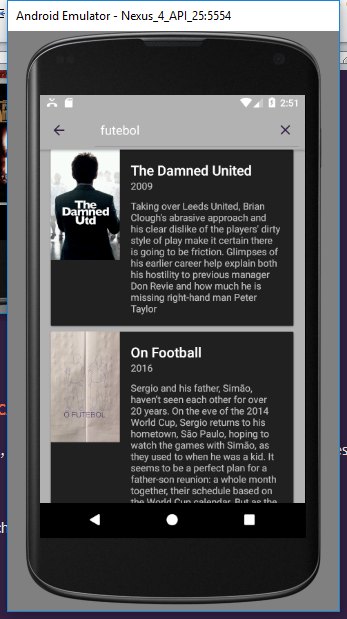
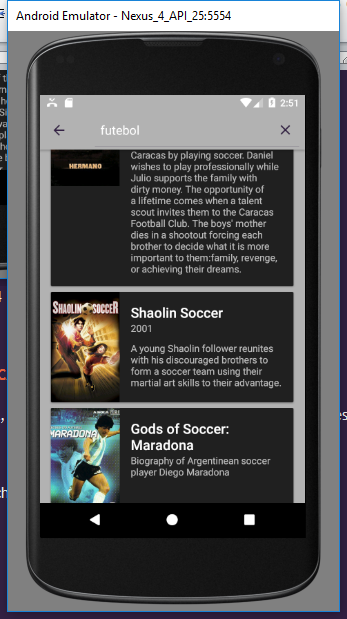
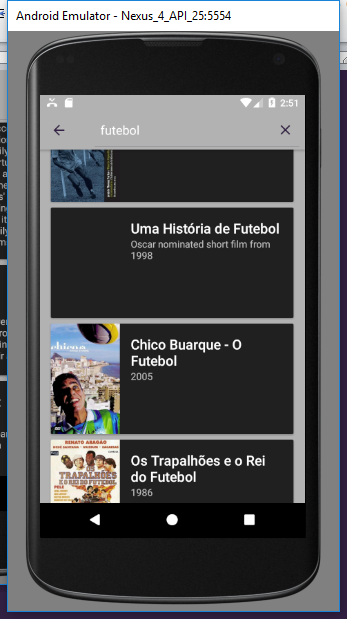
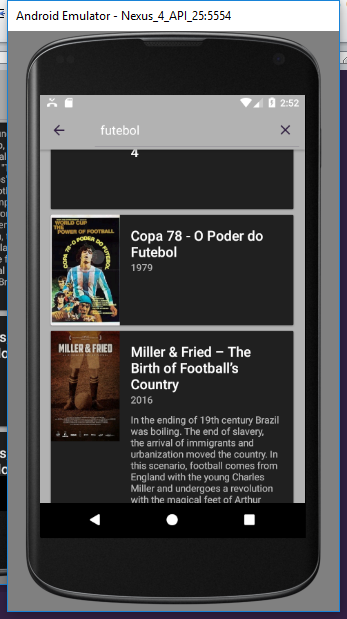
You received 2 projects. Only one of them follow all the requirements below. You should discover which one is the right one and what item is broken. After you find out, use the right project in the CHALLENGE 2.

***USER STORY*:** I as a movies fan want to be able to view a list of current popular movies and read more about some movie.

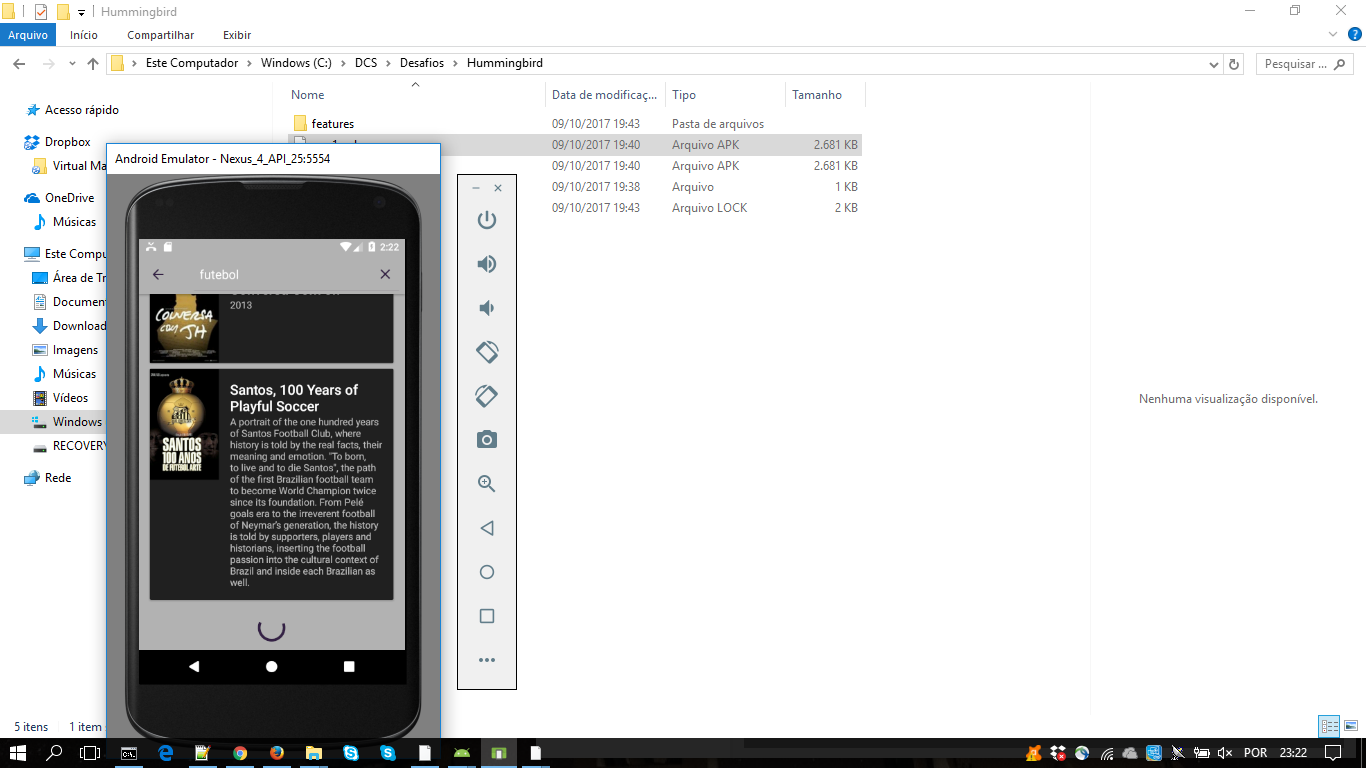
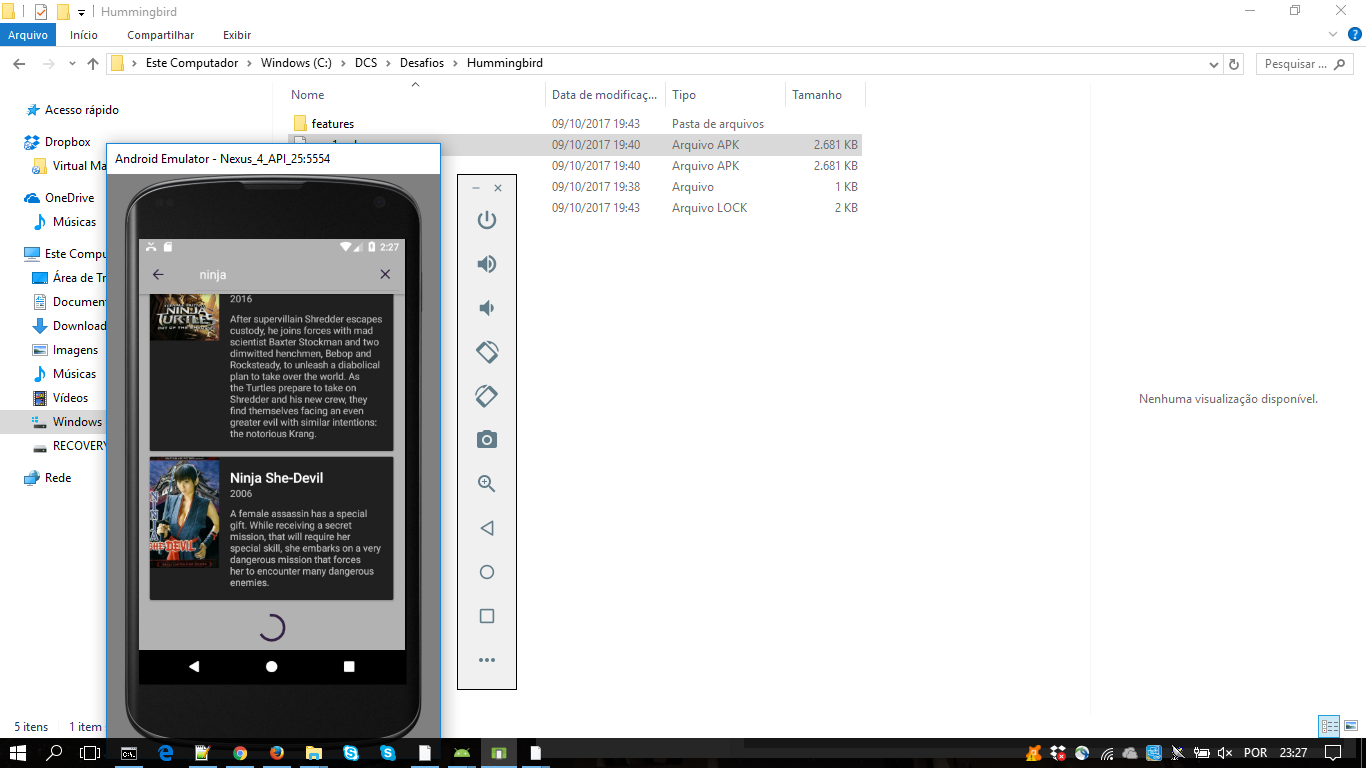
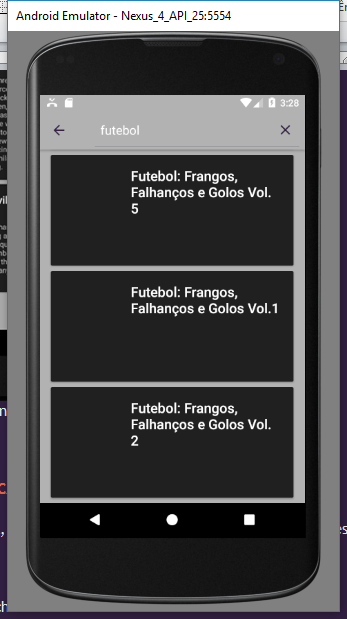
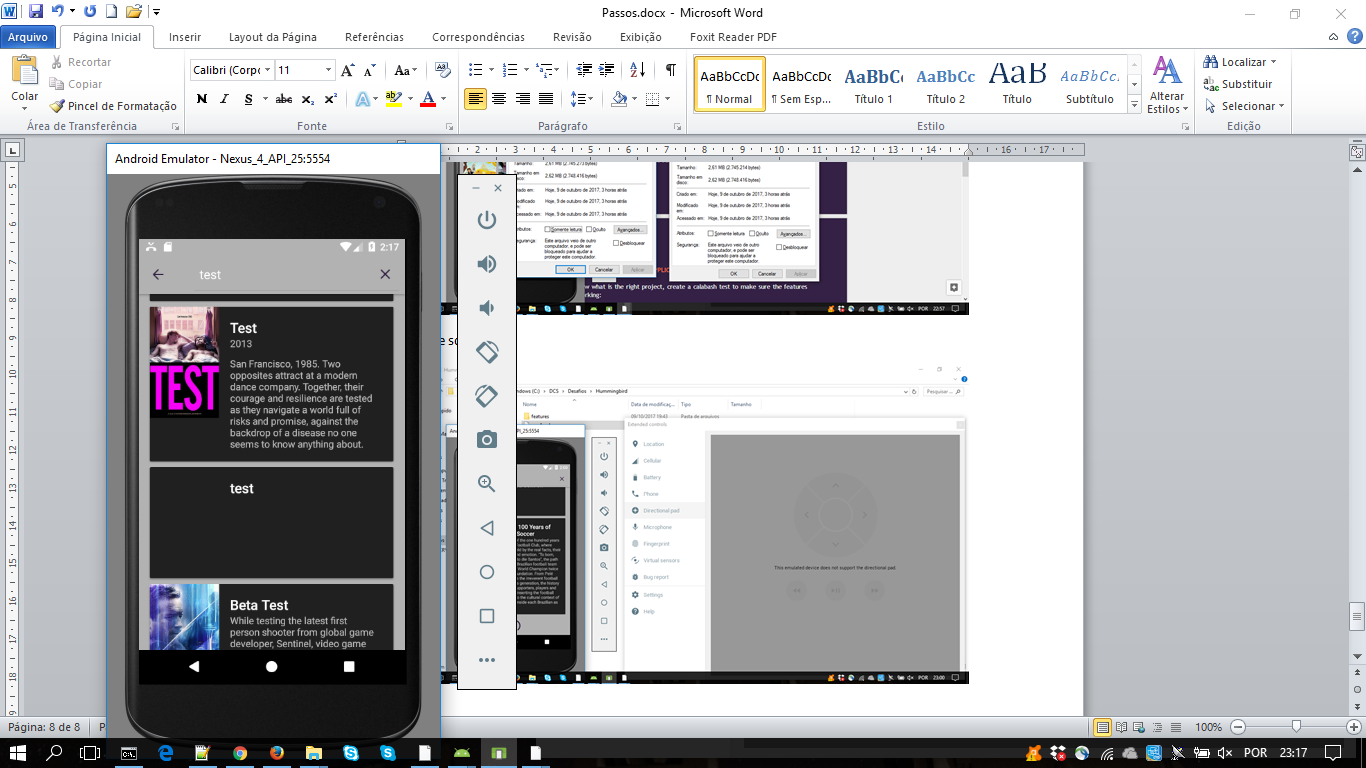
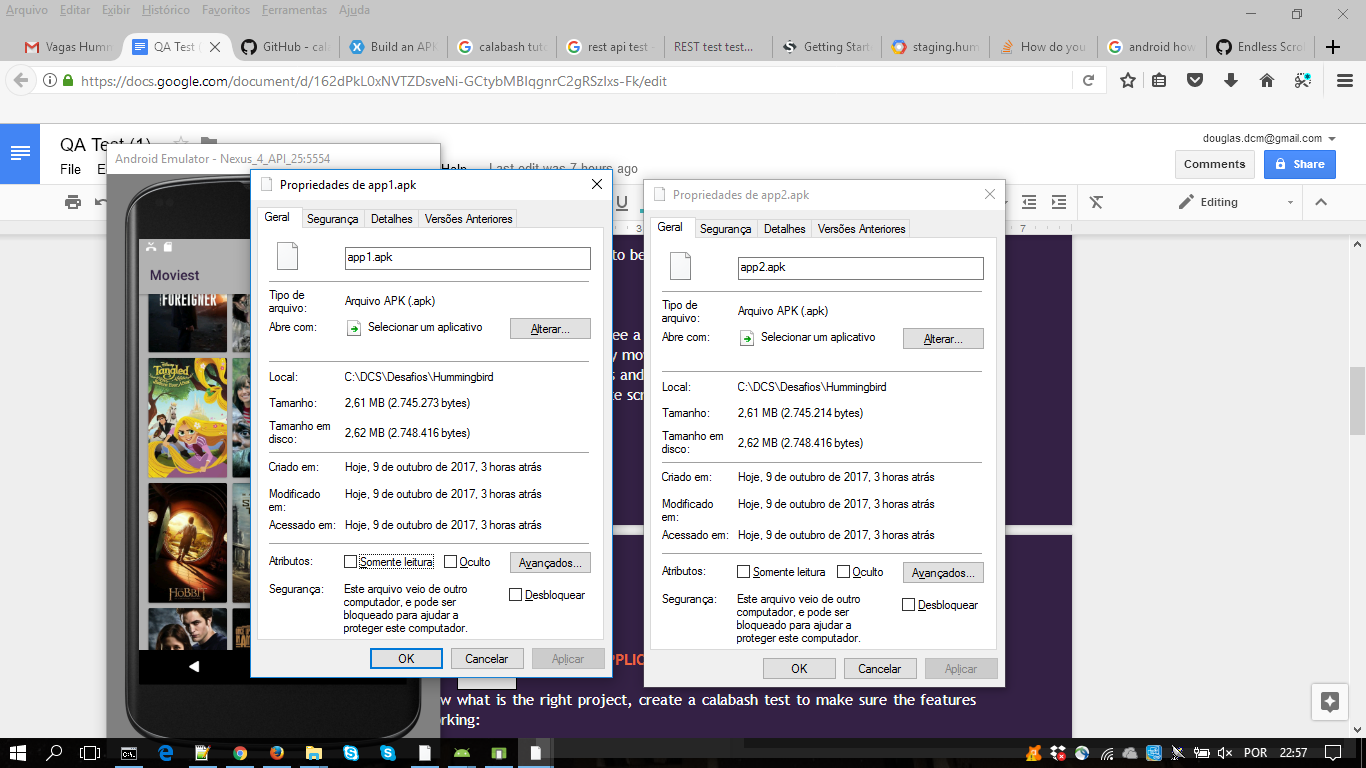
*RULES:*

1. In the home screen, I want to see a list of movies with infinite scrolling
2. You should be able to search any movie over internet
3. You can search different movies and results must be right for each search term
4. Search should alse have a infinite scrolling

***Teste manual***

1. *Navegação pelo aplicativo*
2. *Tela principal*
3. **
4. *Scroll infinito (4 interações)*
5. **
6. **
7. **
8. **
9. *Pesquisar*
10. **
11. *Scroll infinito (tela de pesquisa, 4 interações)*
12. **
13. **
14. **
15. **

***Bugs***

1. *Foi encontrada uma não conformidade com as especificações no app1.apk: a tela de pesquisa não permite scroll infinito limitando o número de filmes exibidos. Para o termo “futebol” foram exibidos 23 itens ao par que no app2.apk foram exibidos mais de 30 itens. Para o termo “ninja” no app1.apk foram exibidos 18 itens enquanto no app2.apk o número foi bem maior (não foi possível definir).*
   1. **
   2. **
2. *Na tela de pesquisa alguns itens não exibiram a imagem do filme*
   1. **
3. *Ao fazer a pesquisa pelo termo “test” foi encontrado um item sem foto e descrição.*
   1. **
4. *O tamanho em bytes dos arquivos app1.apk e app2.apk não são iguais*
   1. **

**THE CHALLENGE 2 - TEST MOBILE APPLICATION WITH CALABASH**

Now you know what is the right project, create a calabash test to make sure the features below are working:

1. Search
2. Infinity Scrolling in Home & Search

*Os arquivos para execução seguem anexos. Neste serão exibidos apenas os arquivos \*.feature e \*\_steps.rb*

***Arquivo search.feature***

*Feature: Search*

*Scenario: I as a movies fan want to be able to view a list of current popular movies and read more about some movie.*

*When I click on magnifying glass button*

*And I fill the search field with "Ninja"*

*Then A list of movies related to "Ninja" is displayed Now you know what is the right project, create a calabash test to make sure the features*

***Arquivo scrolling\_home.feature***

*Feature: Scrolling Home*

*Scenario: I as a movies fan want to be able to view a list of current popular movies and read more about some movie.*

*And I scroll down infinitely*

***Arquivo scrolling\_search.feature***

*Feature: Scrolling Search*

*Scenario: I as a movies fan want to be able to view a list of current popular movies and read more about some movie.*

*When I click on magnifying glass button*

*And I fill the search field with "Blade Runner"*

*Then A list of movies related to "Blade Runner" is displayed*

*And I scroll down infinitely*

***Arquivo calabash\_steps.definition***

*require 'calabash-android/calabash\_steps'*

*#search*

*When("I click on magnifying glass button") do*

*sleep 1*

*touch("\* id:'action\_search'")*

*end*

*When("I fill the search field with {string}") do |string|*

*enter\_text "\* id:'search\_src\_text'" , "#{string}"*

*end*

*Then("A list of movies related to {string} is displayed") do |string|*

*#query("\* text:'#{string}'")*

*element\_exists("\* text:'#{string}'")*

*end*

*#scrolling search and home*

*And("I scroll down infinitely") do*

*q = query("\* id:'progress\_bar'")*

*minutos = 15*

*tempo = minutos \* 60*

*p q*

*segundos = 1*

*hide\_soft\_keyboard*

*p "I will scroll down for #{minutos} minutes or util the end of list..."*

*while q.empty? and segundos < tempo #or end\_of\_line*

*scroll("android.support.v7.widget.RecyclerView", :down)*

*t = 2*

*sleep t*

*q = query("\* id:'progress\_bar'")*

*p "Elapsed #{segundos} seconds from #{tempo}."*

*if not(q.empty?)*

*p "End of list."*

*end*

*segundos = segundos + t*

*end*

*end*

**THE CHALLENGE 3 - TEST API**

We created a simple API for this test. You should create a script to validate all parameters rules and verify api responses. We added few logic errors in the staging version, according to the documentation below, you should find all of them. The production version should not have the same errors.

*USER STORY*: As a user, I would like to create new api feature flags, change existing ones and be able to get all of them according my app version or platform

BASE URL: http://production.hummingbird-api.appspot.com

BASE URL WITH ERRORS: http://staging.hummingbird-api.appspot.com

GET: /featureflag

Return all save feature flags in a json format

*Parameters*:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | Type | Description | Rules |
| appVersion | String | Number of the app version | All app versions should follow this pattern: 12.3.100 |
| platform | String | Application platform | ANDROID or iOS |

*Success Response* (request status code == 200):

{

"example1": false,

"example2": true

}

*Failed Response* (request status code == 400):

{

"errorCode": 1023

"errorMessage": "This is an example of error message"

}

POST: /featureflag

Add a new featureflag to datastorage or override an existent one

*Parameters:*

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | Type | Description | Rules |
| appVersion | String | Number of the app version | All app versions should follow this pattern: 12.3.100 |
| platform | String | Application platform | ANDROID or iOS |
| enabled | String | Set featureflag on or off | TRUE or FALSE |
| name | String | Name of the featureflag key | Any string with 3 <= characters <= 20 |

*Success Response* (request status code == 200):

{

"successMessage": "Your feature flag was created with success"

}

*Failed Response* (status code == 400):

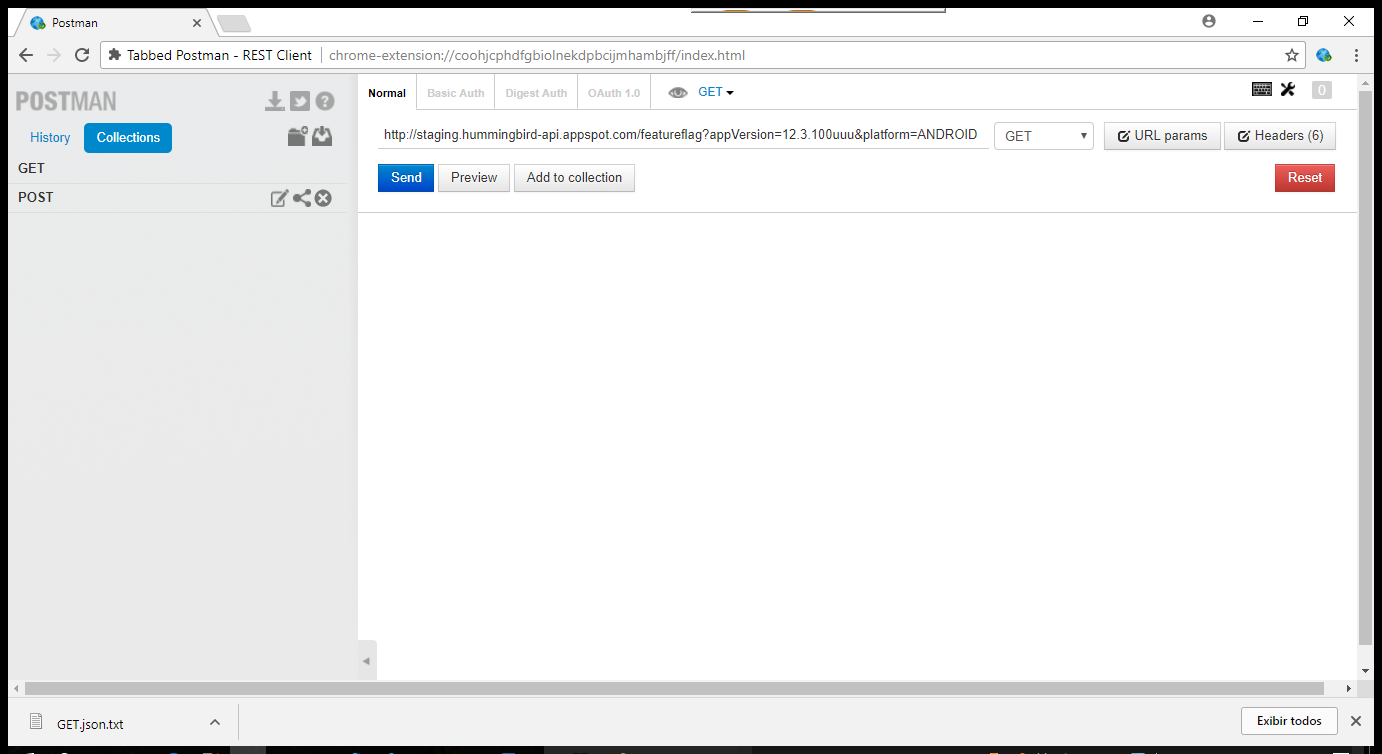
{

"errorCode": 1023

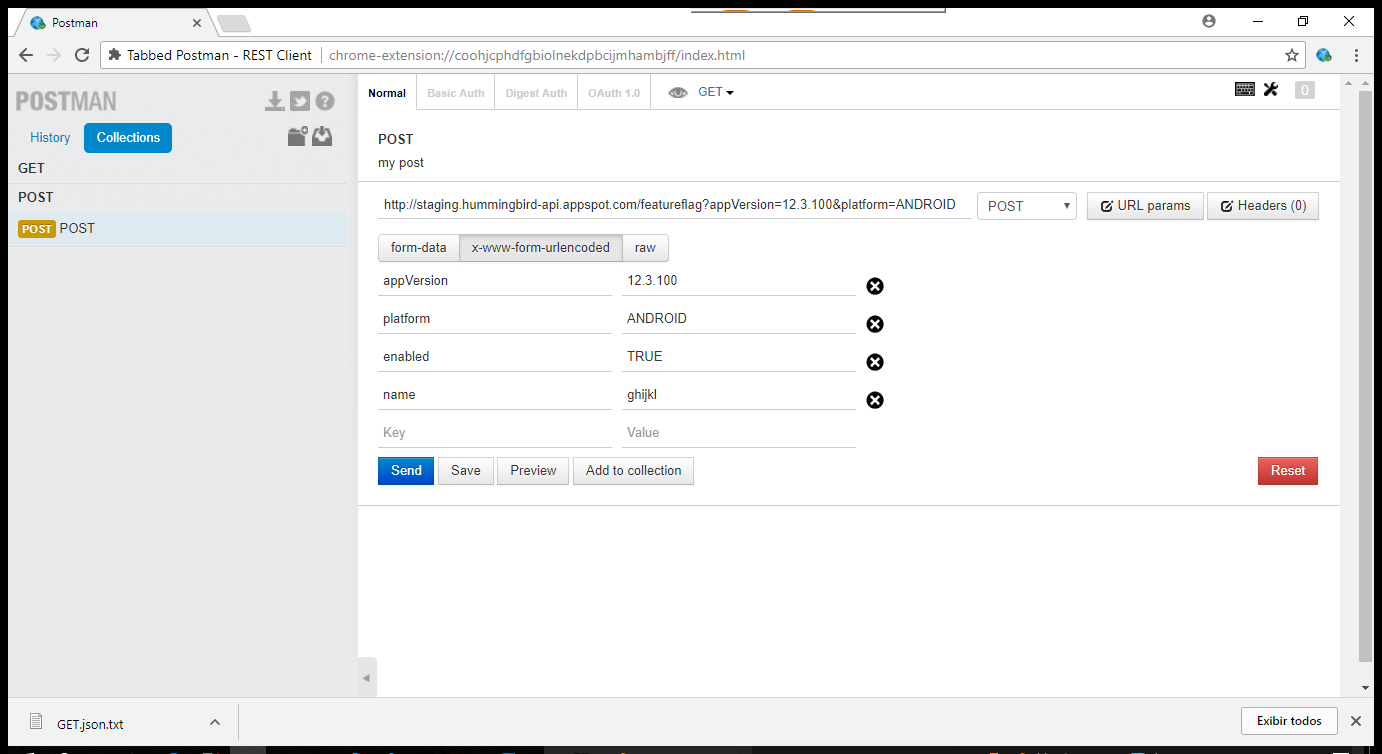
"errorMessage": "This is an example of error message"

}

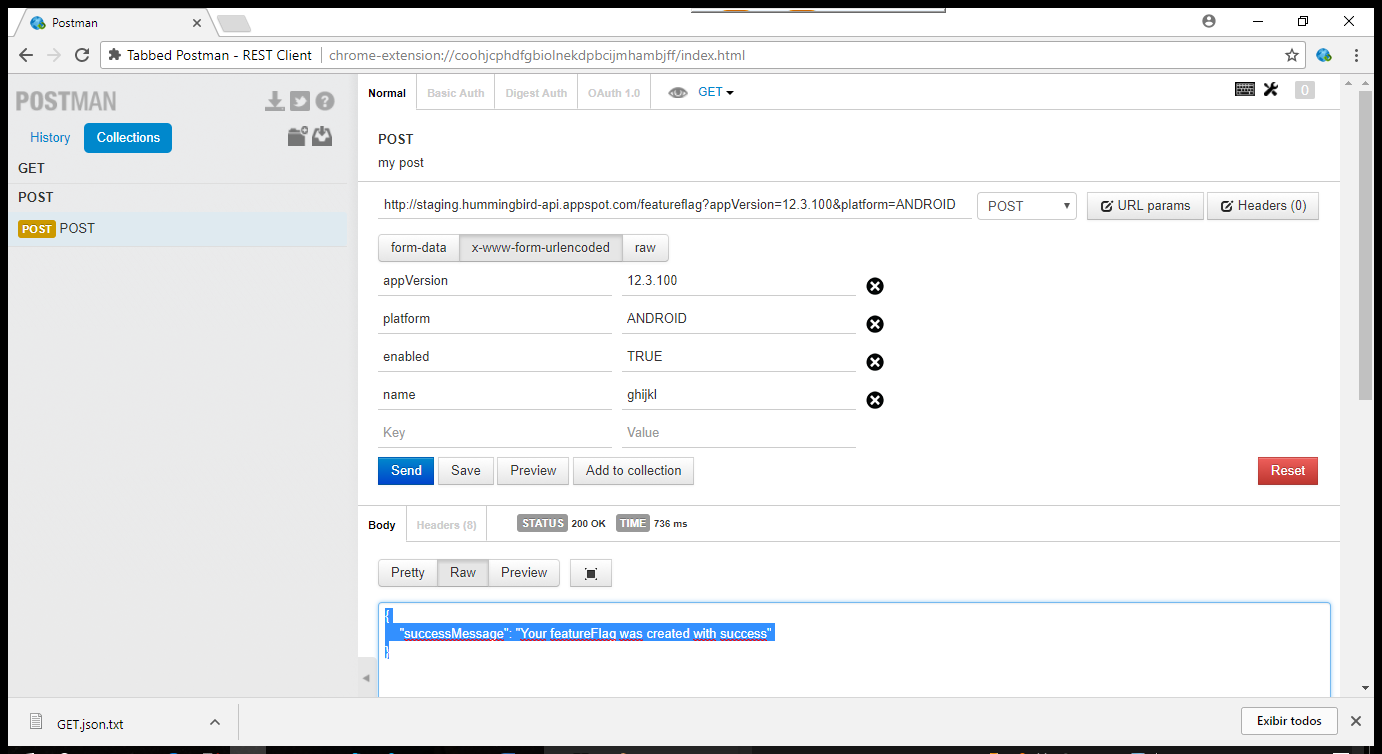
*Execução dos testes segue abaixo.*



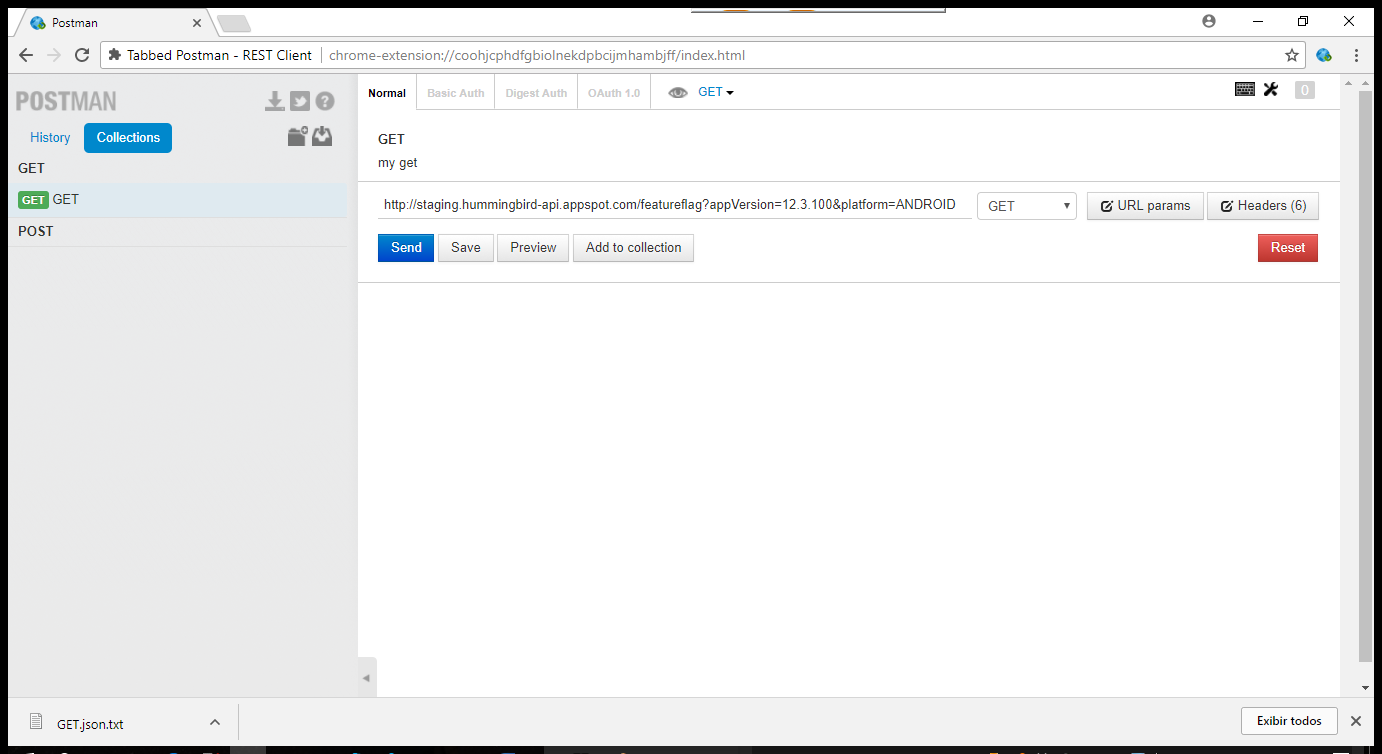
*Executar primeiro script POST clicando em SEND*

**

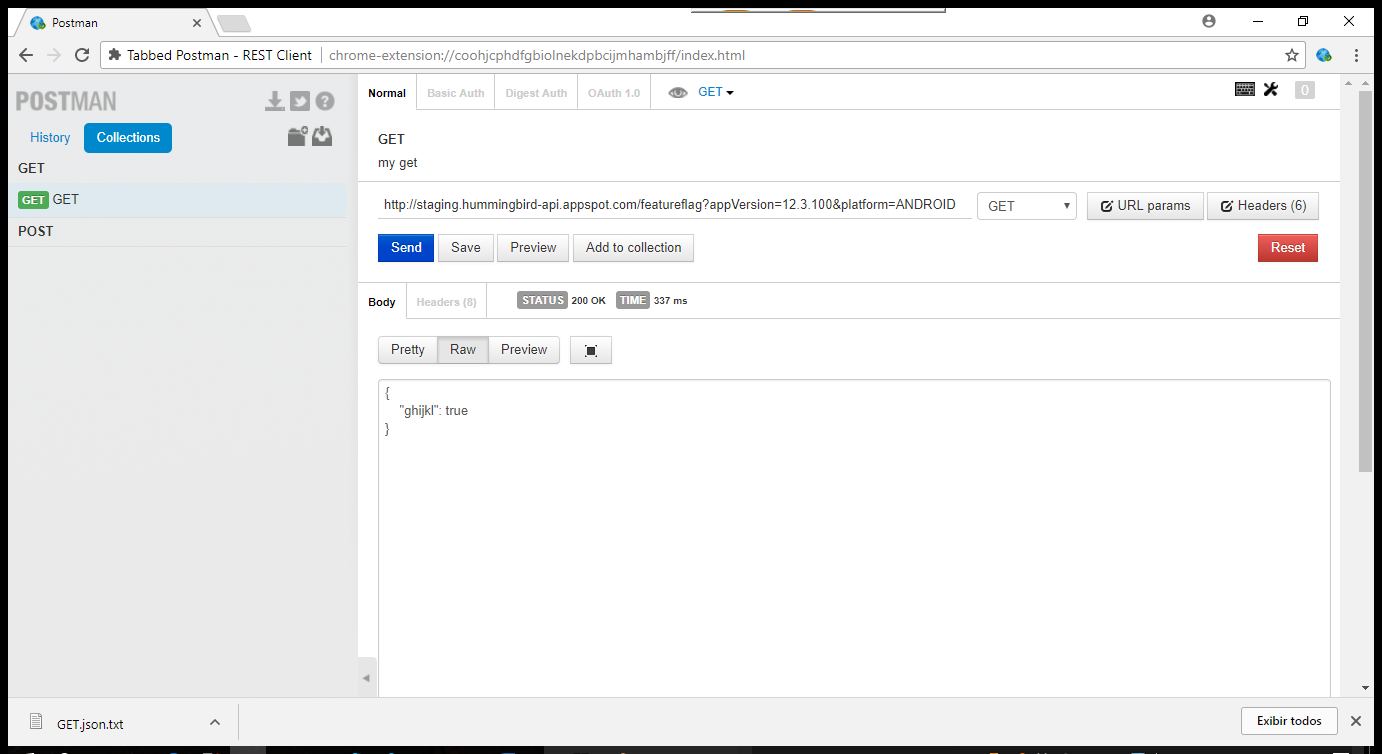
*Verificar a resposta positiva conforme abaixo.*

**

*Executar o script GET clicando em SEND*

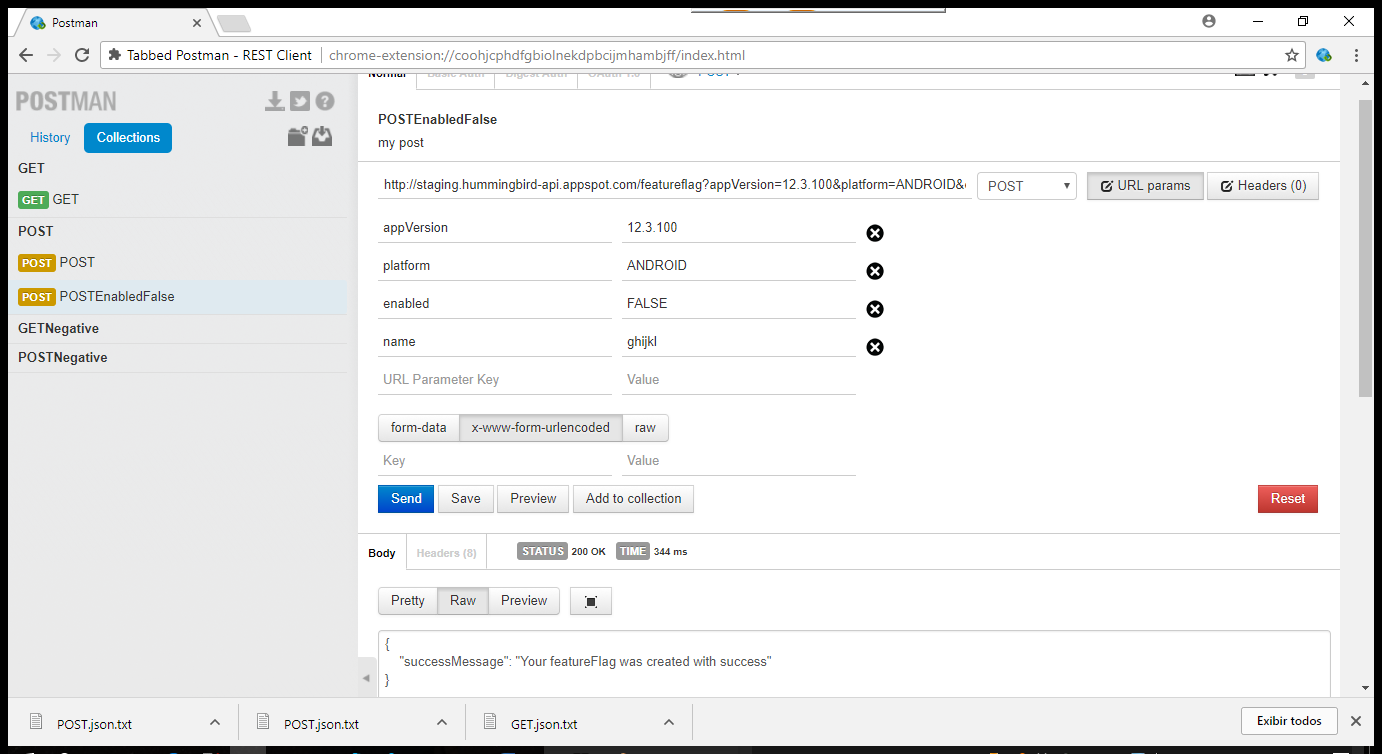


*Verificar a resposta positiva conforme abaixo.*

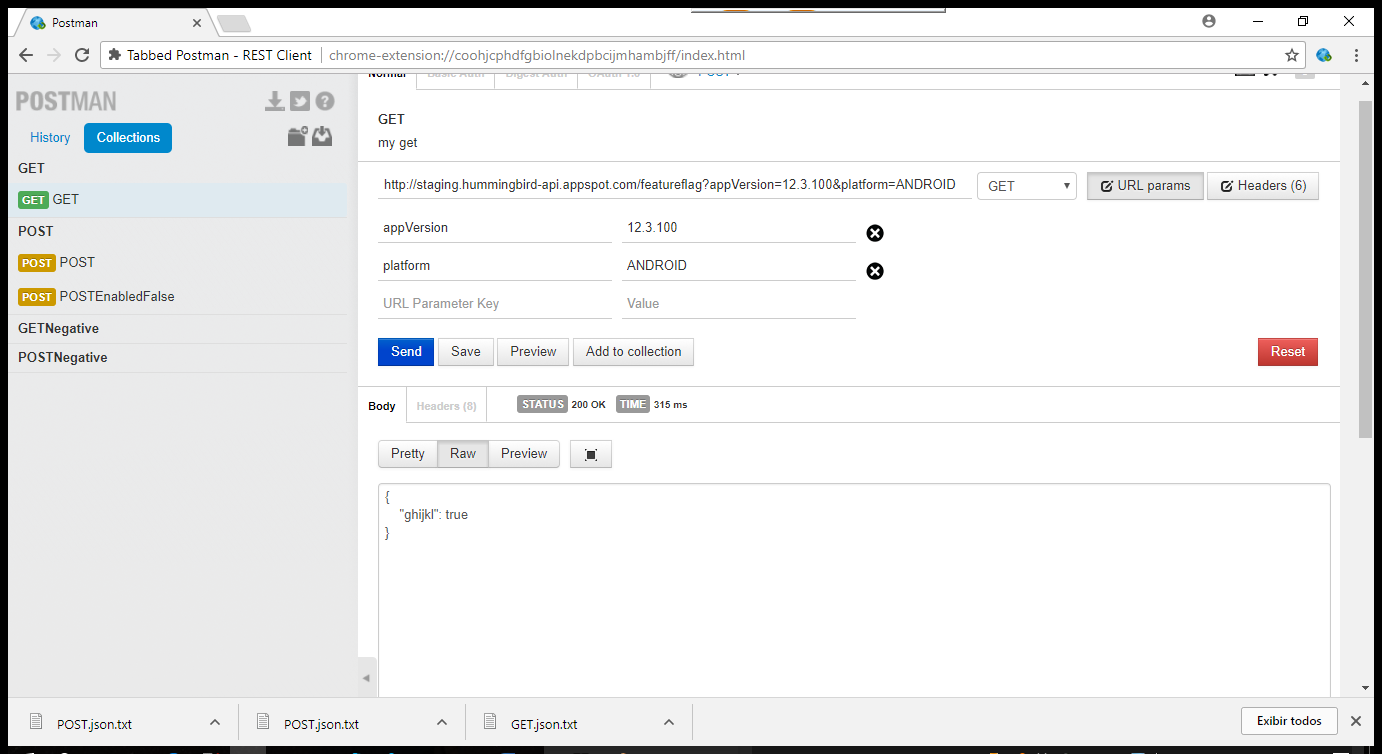


*Executar os demais scripts anexos na ordem desejada.*

*Executar o script POSTEnabledFalse*



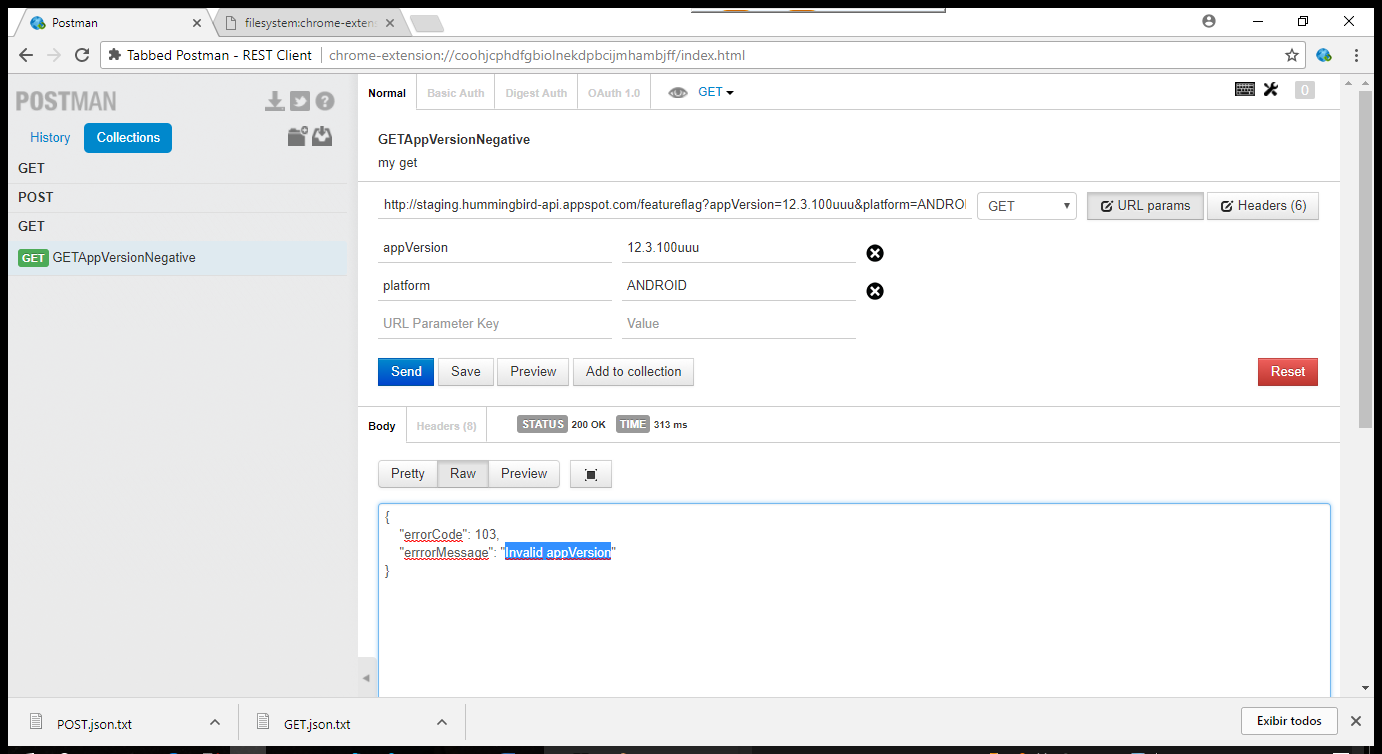
*Executar o script GET*



*A resposta deveria ser {“ghijkl”: false} (bug)*

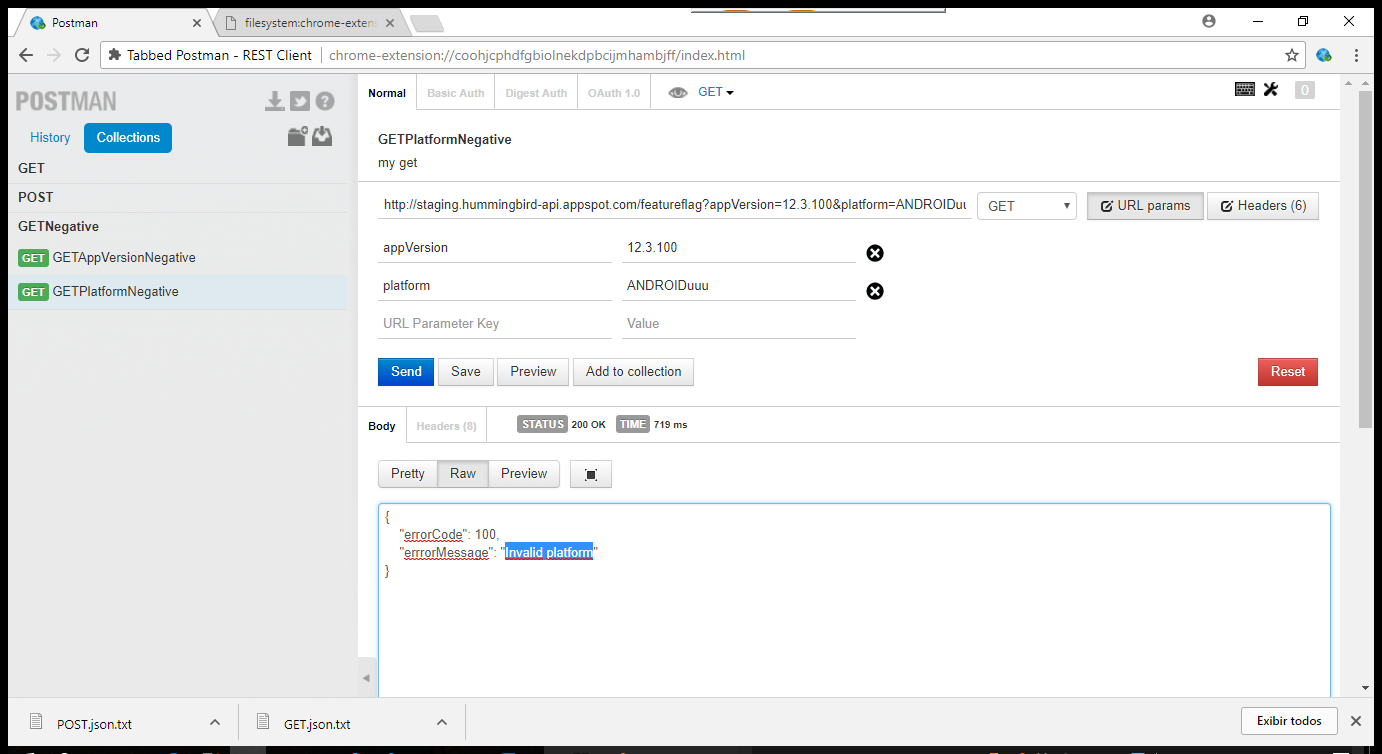
*Executar os script de teste negativo*

*GETAppVersionNegative*



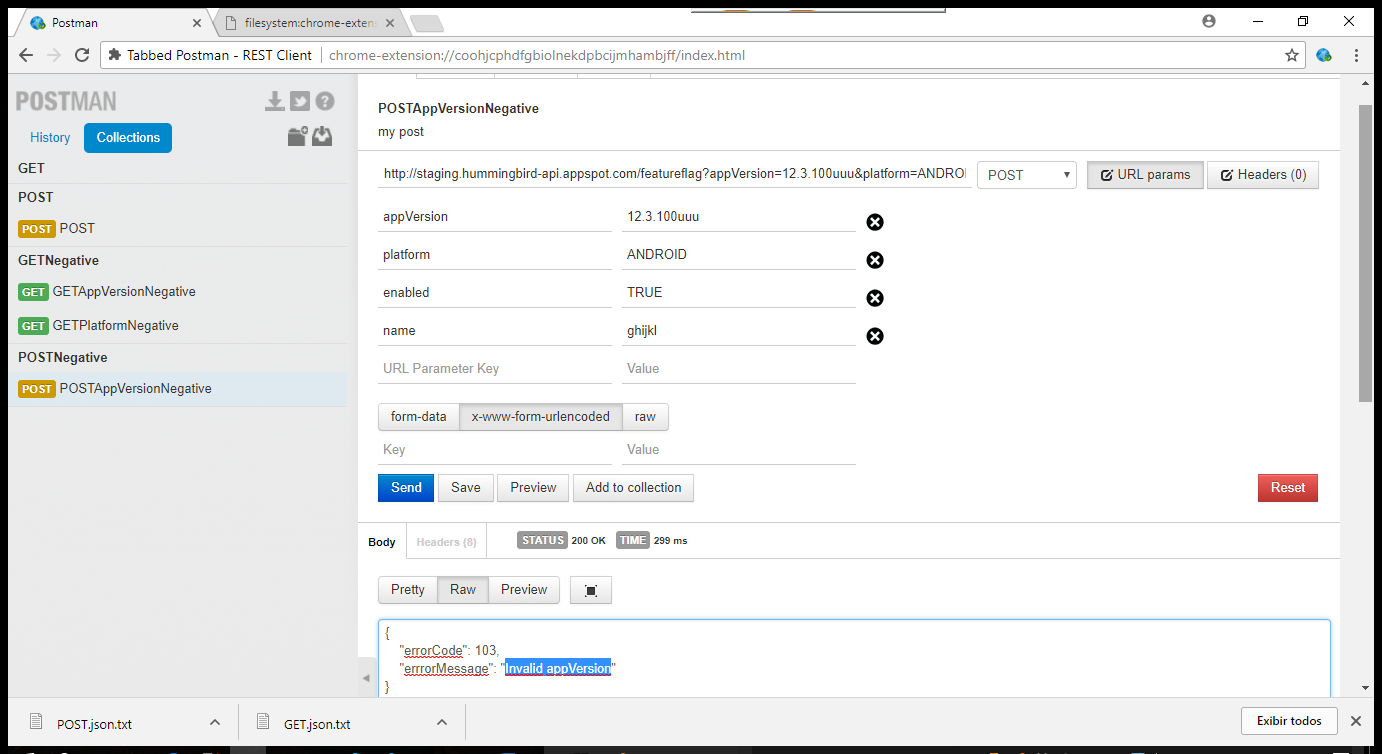
*Verificar a resposta. Este teste deve falhar, pois o parâmetro appVersion foi enviado fora do modelo.*

*GETPlatformNegative*



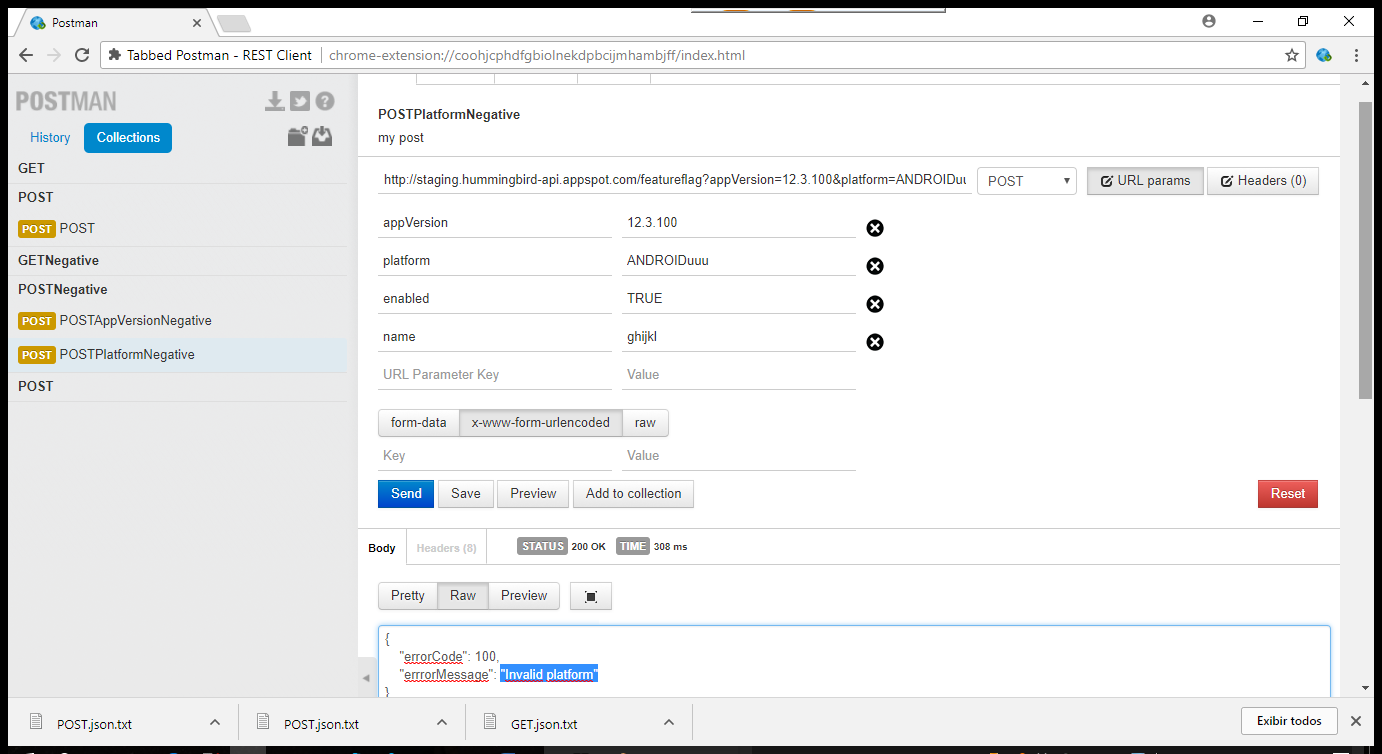
*Verificar a resposta. Este teste deve falhar, pois o parâmetro platform foi enviado fora do modelo.*

*POSTAppVersionNegative*



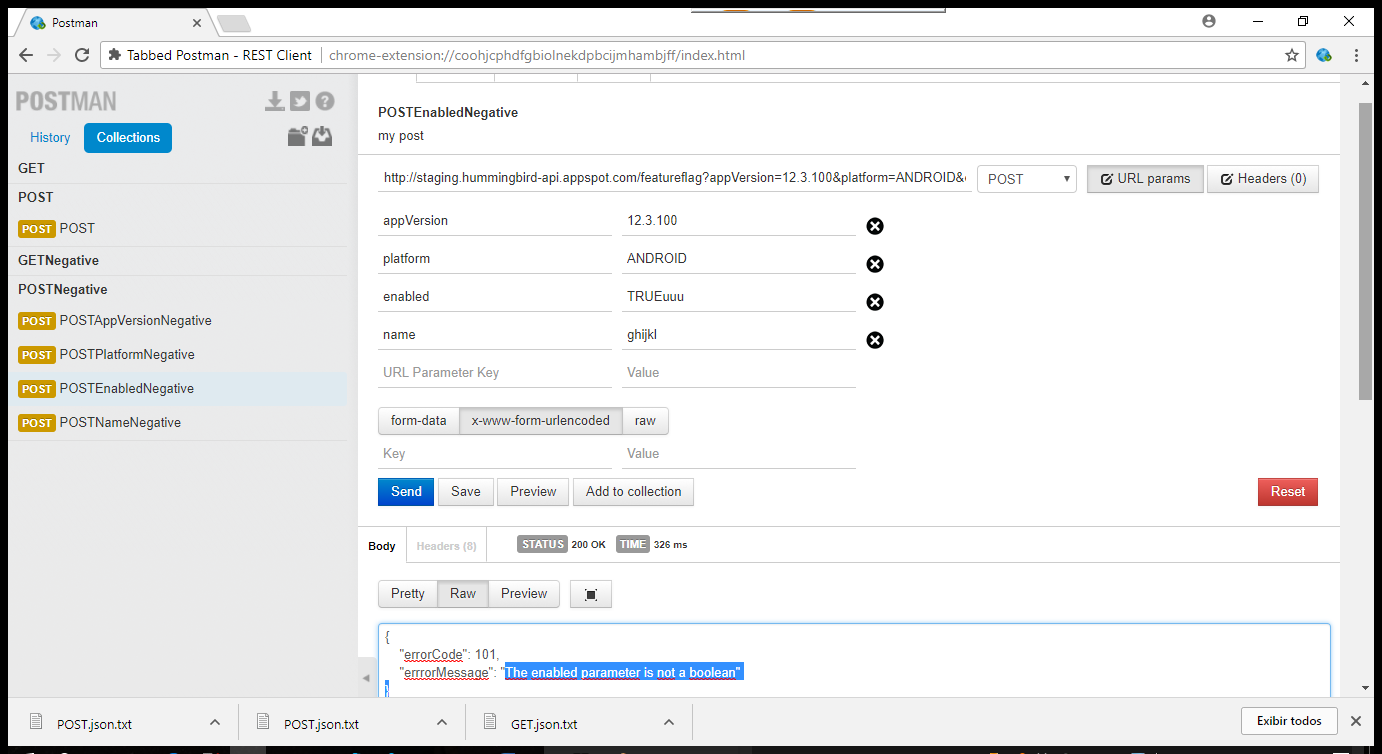
*Verificar a resposta. Este teste deve falhar, pois o parâmetro appVersion foi enviado fora do modelo.*

*POSTPlatformNegative*



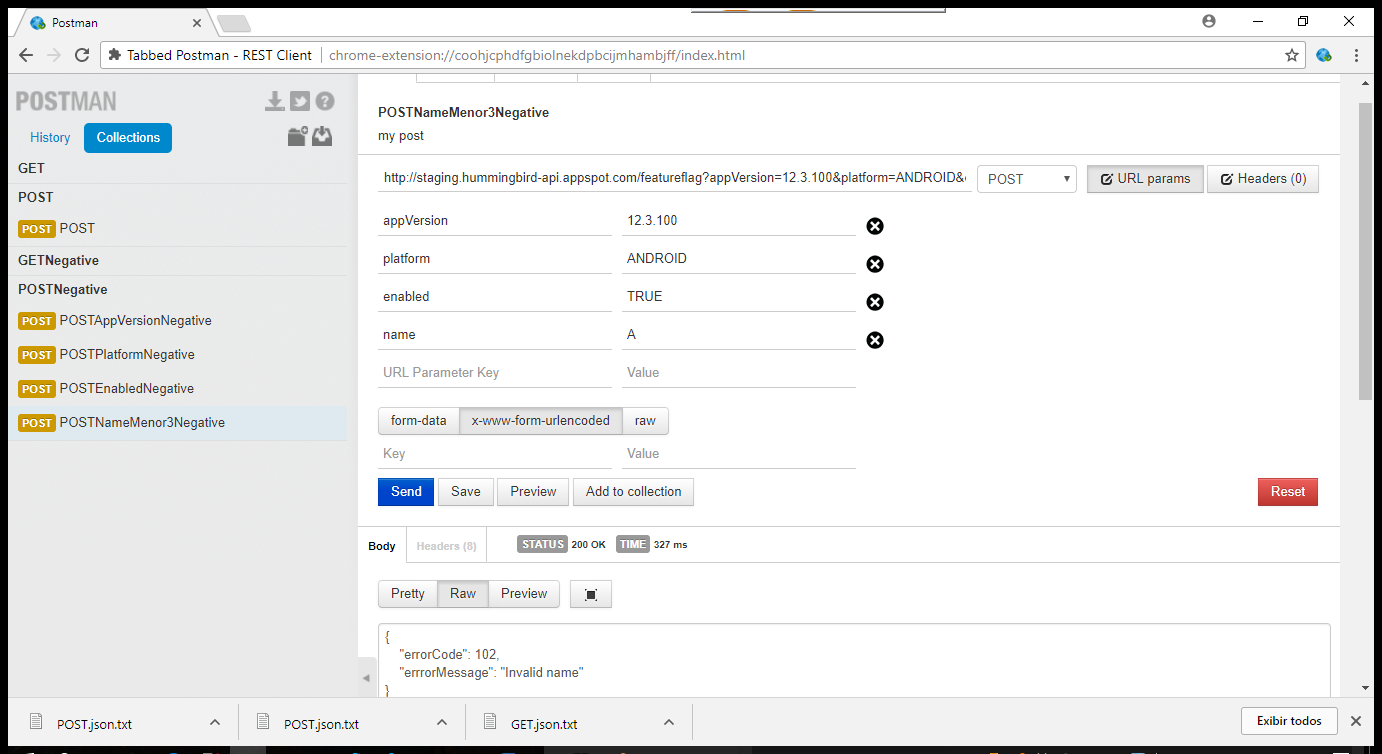
*Verificar a resposta. Este teste deve falhar, pois o parâmetro platform foi enviado fora do modelo.*

POSTEnabledNegative



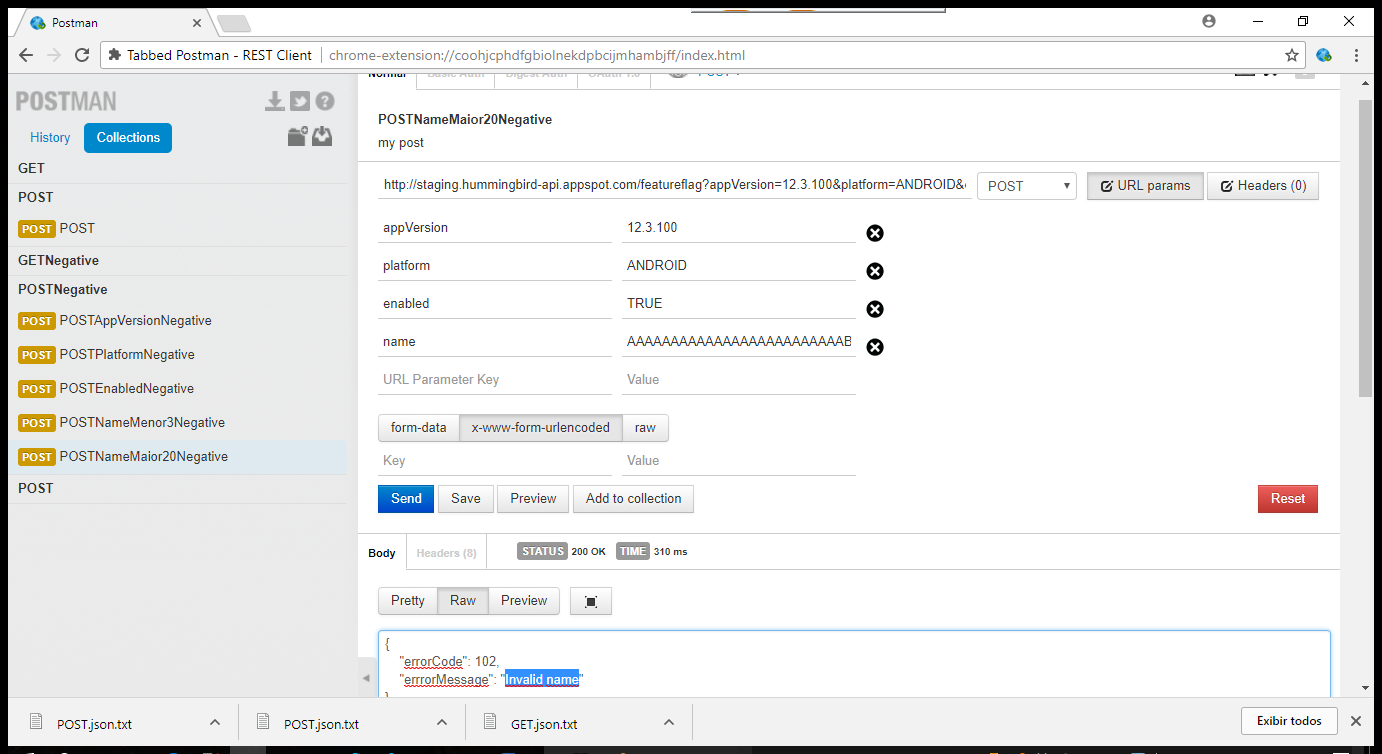
*Verificar a resposta. Este teste deve falhar, pois o parâmetro enabled foi enviado fora do modelo.*

POSTNameMenor3Negative



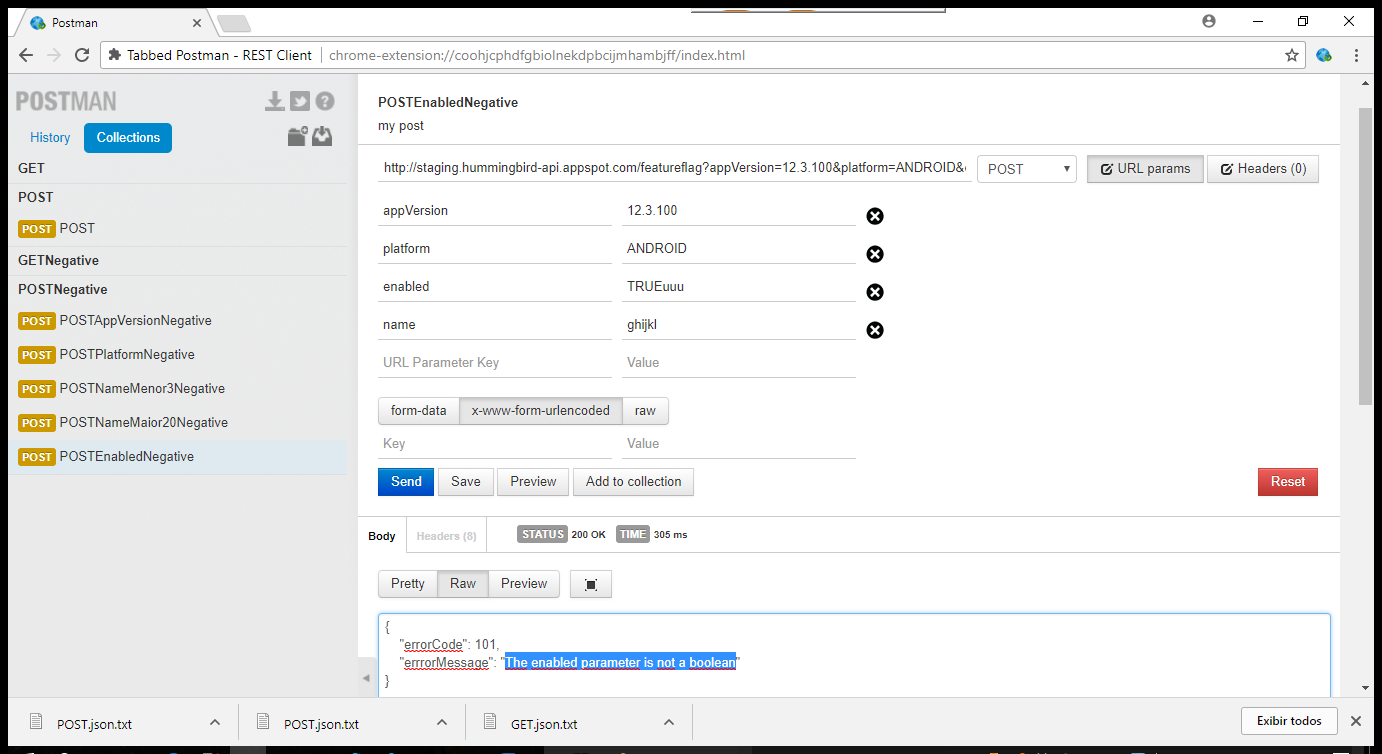
*Verificar a resposta. Este teste deve falhar, pois o parâmetro name foi enviado fora do modelo.*

POSTNameMaior20Negative



*Verificar a resposta. Este teste deve falhar, pois o parâmetro name foi enviado fora do modelo.*

*POSTEnabledNegative*



*Verificar a resposta. Este teste deve falhar, pois o parâmetro enabled foi enviado fora do modelo.*

***Bugs***

*Ao enviar o método POST com o parâmetro enabled FALSE o método GET está exibindo o featureflag como TRUE. Ver caso de teste POSTEnabledFalse neste documento.*