

RIOTGAMES API

1º TP - Programação Web



2º Ano Licenciatura em Informática – ISMAI

Ricardo Sousa NºMec.: A037322
Hugo Castro NºMec.: A036537
Guilherme Silva NºMec.: A036390

ÍNDICE



















- Descrição dos Objetivos da API Web;
- Descrição de Funções;
 - Obter dados do Jogador;
 - Obter histórico de jogos recentes;
 - Consultar dados de um jogo específico;
 - Obter dados de um Champion;
 - Criar Torneios;
- Avaliação e Análise Crítica;
 - Prós;
 - Contras;

PLAY HOME TFT CLASH

OVERVIEW MATCH HISTORY RANKED HIGHLIGHTS STATS

RECENT GAMES (LAST 20 PLAYED)

GENERAL TFT

 16	DEFEAT AR Ultra Rapid Fire		Summoner's Rift 20:46 • 08/03/21
		10 / 10 / 2 71 ⬇ 15,743 ⬇	
 9	VICTORY AR Ultra Rapid Fire		Summoner's Rift 10:40 • 08/03/21
		10 / 2 / 6 62 ⬇ 9,570 ⬇	
 19	DEFEAT AR Ultra Rapid Fire		Summoner's Rift 25:36 • 08/03/21
		5 / 12 / 4 139 ⬇ 17,028 ⬇	
 17	VICTORY AR Ultra Rapid Fire		Summoner's Rift 20:43 • 08/03/21
		14 / 4 / 9 68 ⬇ 16,911 ⬇	
 9	VICTORY AR Ultra Rapid Fire		Summoner's Rift 10:31 • 08/03/21
		8 / 4 / 5 50 ⬇ 9,047 ⬇	
 17	DEFEAT AR Ultra Rapid Fire		Summoner's Rift 24:03 • 08/03/2021
			

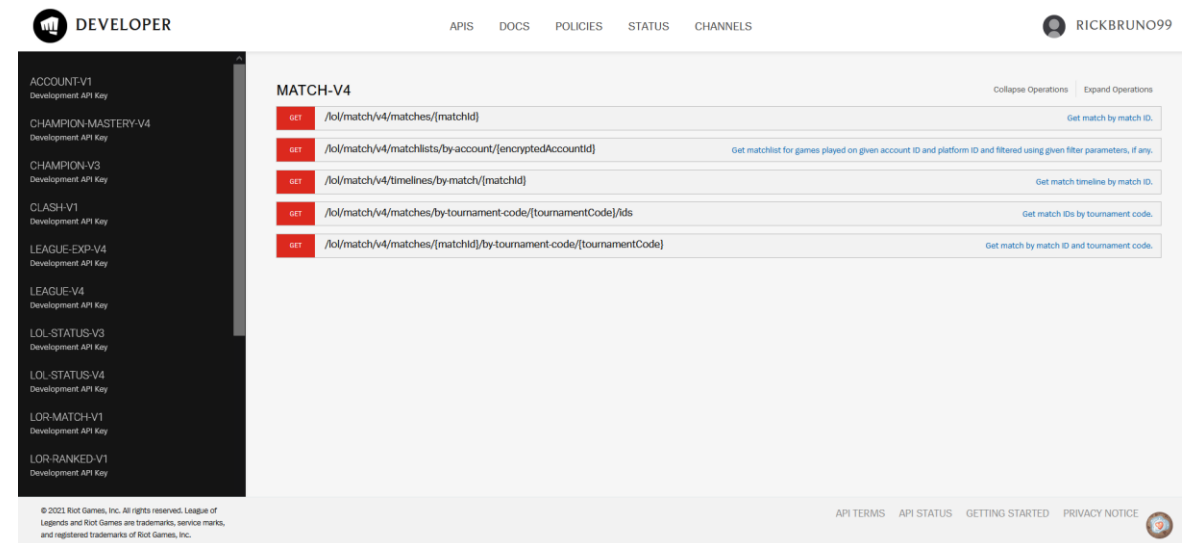
DESCRIÇÃO DOS OBJETIVOS DA API WEB

- Obter informações da base de dados da Riot Games tais como:
 - Informações do Summoner(Jogador);
 - Histórico de Jogos;
 - Jogadores com os maiores Rankings da região;
 - Consultar estatísticas de Champions(personagens);
 - Consultar dados de um Jogo específico;
 - Etc.
- Criação de Torneios

DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

Podemos testar a API de duas formas:

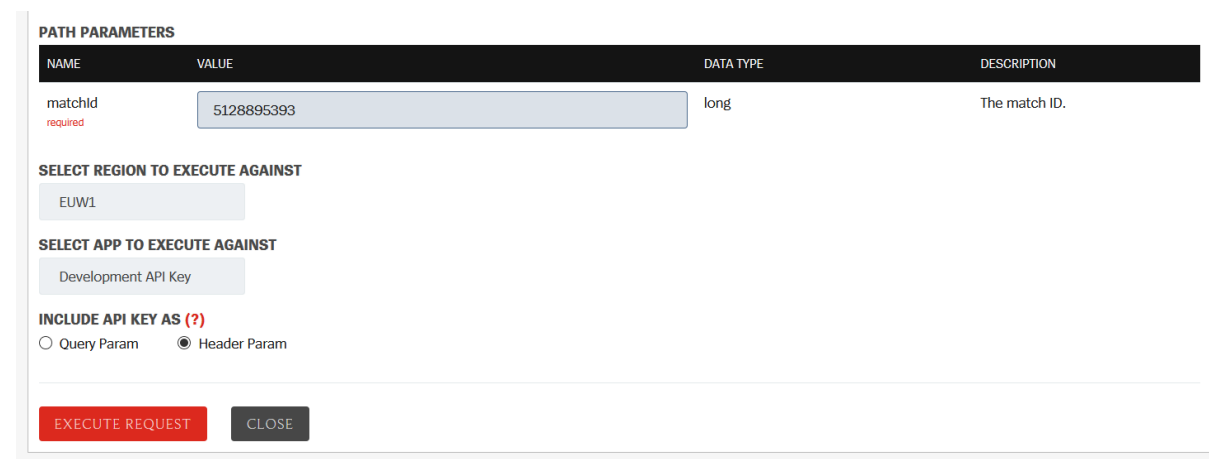
- Através do site de Desenvolvimento da Riot Games



The screenshot shows the Riot Games Developer Portal interface. On the left, a sidebar lists various API endpoints under the 'MATCH' category, including ACCOUNT-V1, CHAMPION-MASTERY-V4, CHAMPION-V3, CLASH-V1, LEAGUE-EXP-V4, LEAGUE-V4, LOL-STATUS-V3, LOL-STATUS-V4, LOR-MATCH-V1, and LOR-RANKED-V1. The main content area displays the 'MATCH-V4' section with a list of endpoints and their descriptions:

Method	Endpoint	Description
GET	/lol/match/v4/matches/{matchId}	Get match by match ID.
GET	/lol/match/v4/matchlists/by-account/{encryptedAccountId}	Get matchlist for games played on given account ID and platform ID and filtered using given filter parameters, if any.
GET	/lol/match/v4/timelines/by-match/{matchId}	Get match timeline by match ID.
GET	/lol/match/v4/matches/by-tournament-code/{tournamentCode}/ads	Get match IDs by tournament code.
GET	/lol/match/v4/matches/{matchId}/by-tournament-code/{tournamentCode}	Get match by match ID and tournament code.

At the bottom of the page, there is a footer with copyright information and links to API TERMS, API STATUS, GETTING STARTED, and PRIVACY NOTICE.



The screenshot shows the 'PATH PARAMETERS' section for the MATCH-V4 API endpoint. It includes a table with columns for NAME, VALUE, DATA TYPE, and DESCRIPTION. The 'matchId' parameter is highlighted, and its value is shown as '5128895393'. Below the table, there are sections for 'SELECT REGION TO EXECUTE AGAINST' (EUW1) and 'SELECT APP TO EXECUTE AGAINST' (Development API Key). There is also a section for 'INCLUDE API KEY AS (?)' with radio buttons for 'Query Param' and 'Header Param'. At the bottom, there are buttons for 'EXECUTE REQUEST' and 'CLOSE'.

NAME	VALUE	DATA TYPE	DESCRIPTION
matchId <small>required</small>	5128895393	long	The match ID.

SELECT REGION TO EXECUTE AGAINST
EUW1

SELECT APP TO EXECUTE AGAINST
Development API Key

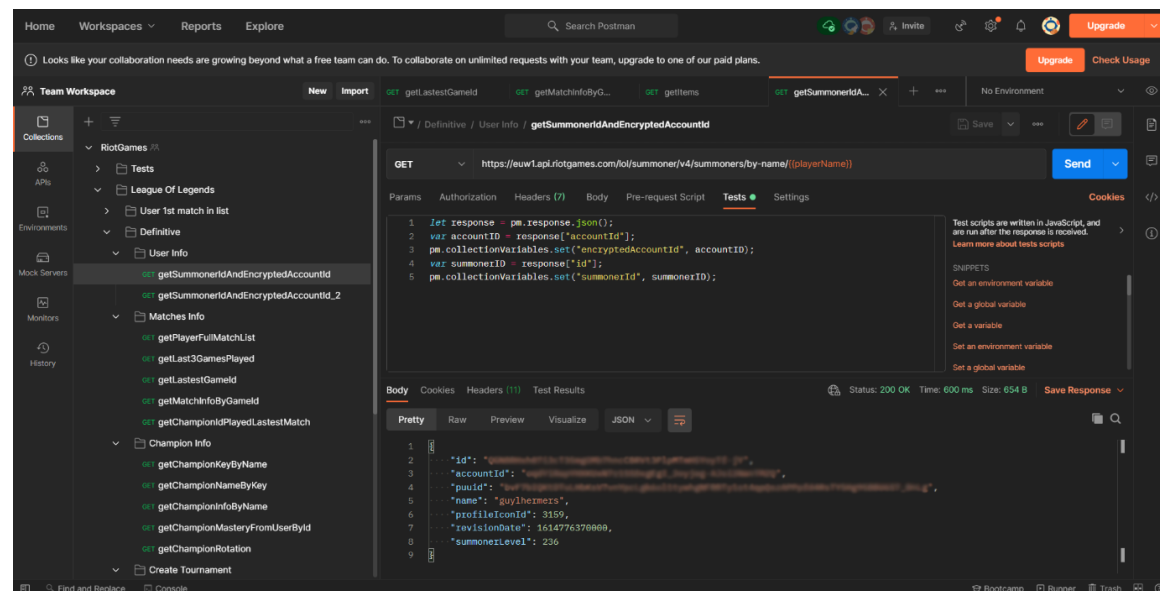
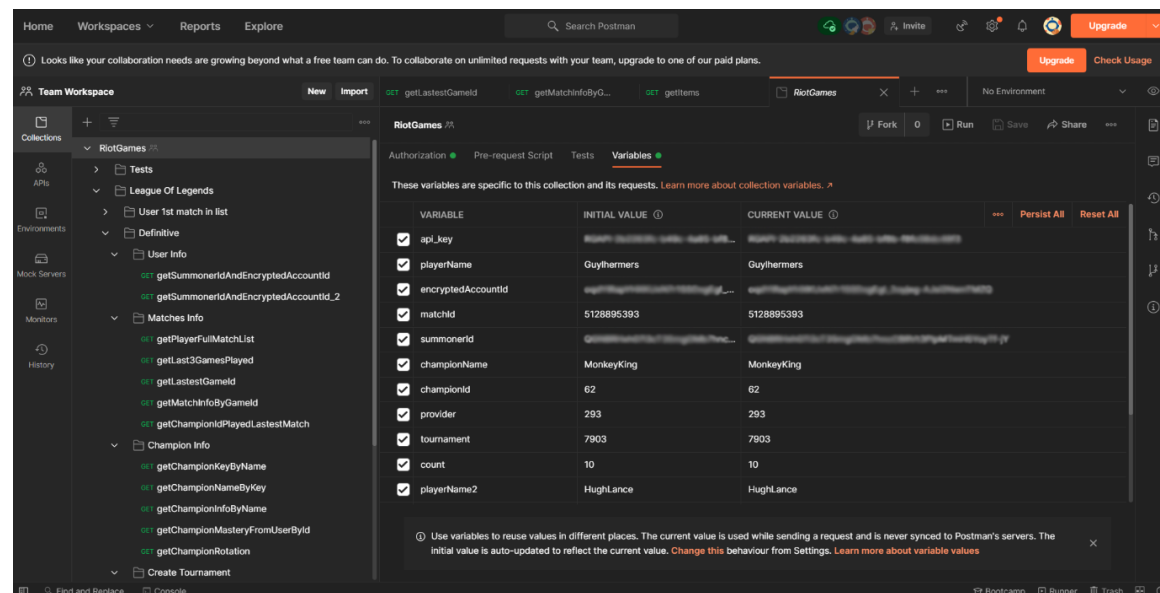
INCLUDE API KEY AS (?)
☐ Query Param ☒ Header Param

EXECUTE REQUEST **CLOSE**

DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

Podemos testar a API de duas formas:

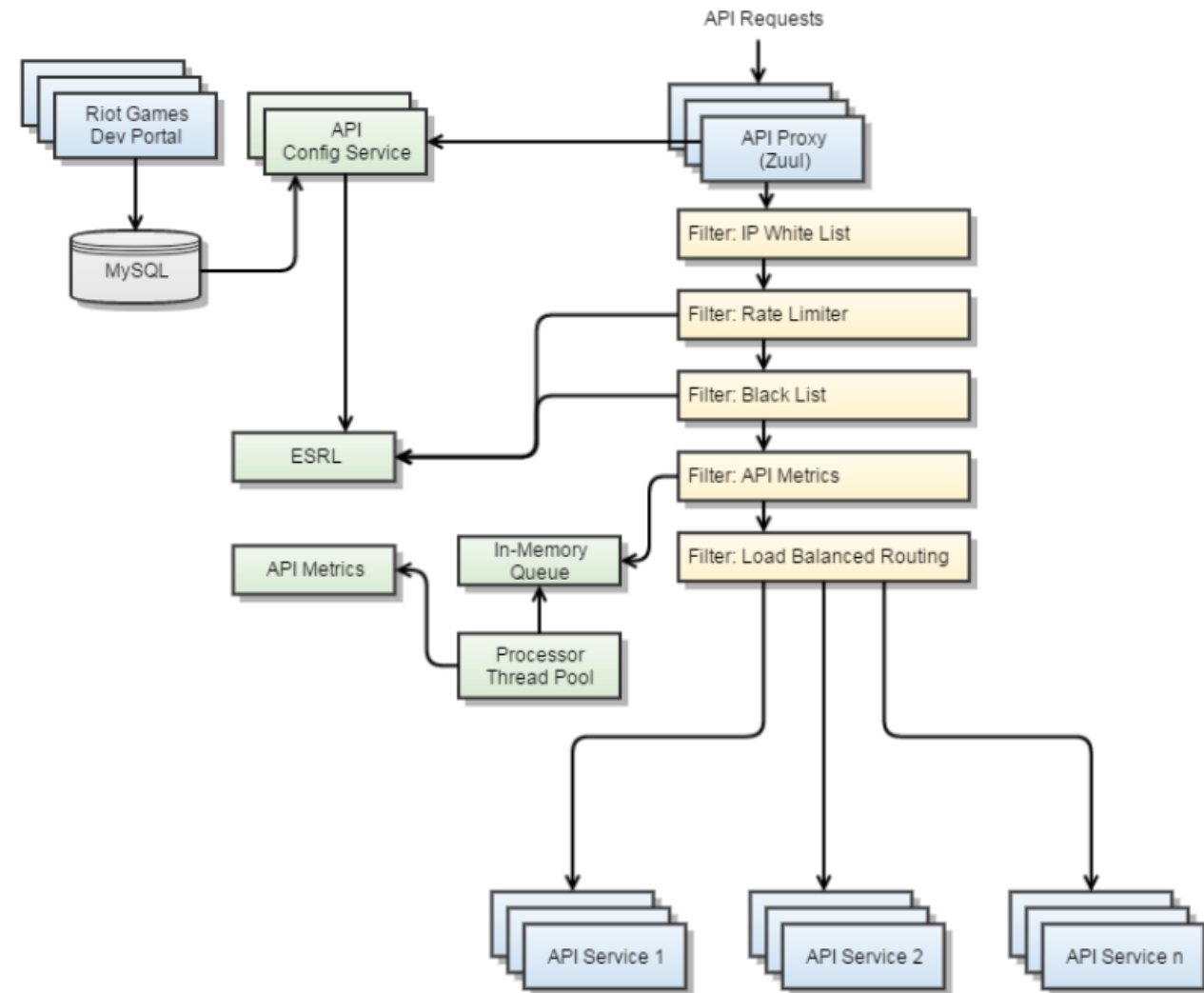
- Através do API cliente do Postman



DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

A Riot Games API é uma Web API do tipo REST (REpresentation State Transfer) que através de pedidos HTTPS do tipo GET, POST, PUT, e DELETE, devolve uma resposta em formato JSON (JavaScript Object Notation).

Estes pedidos necessitam de uma autenticação, neste caso do tipo API Key contida no header do pedido.

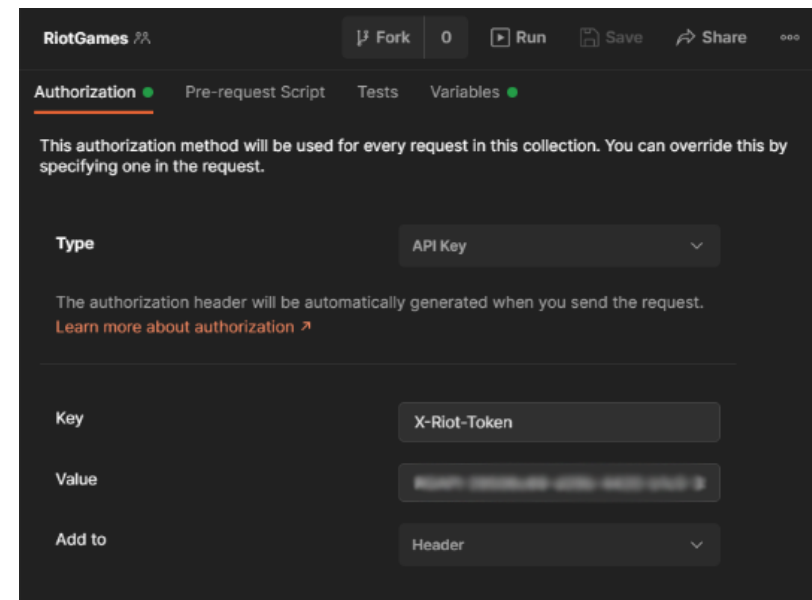
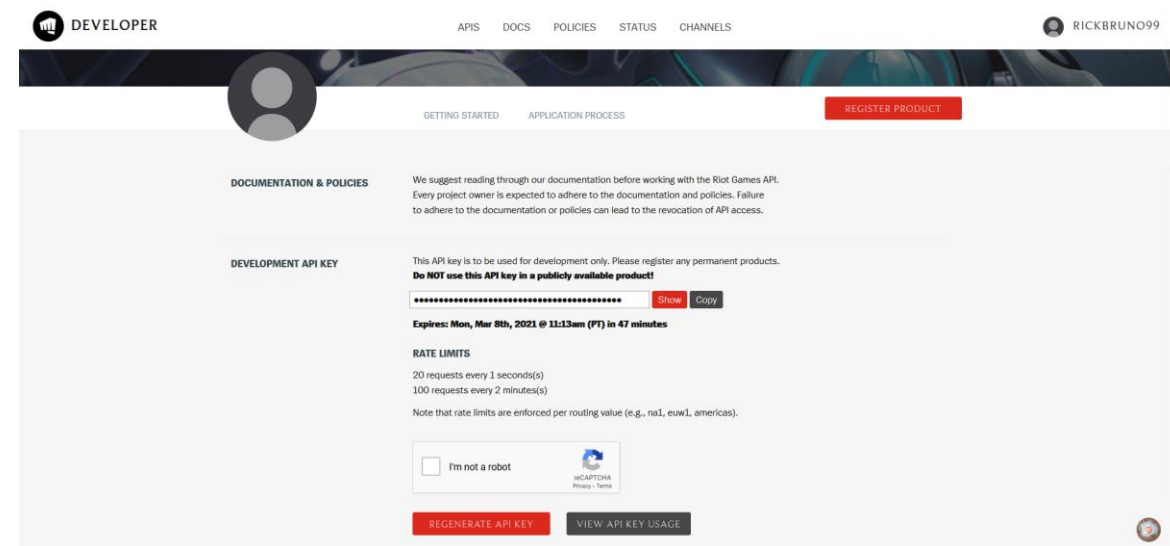


DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

De forma a termos acesso à API tivemos de gerar uma chave de desenvolvimento que permitisse a autenticação em cada um dos pedidos efetuados à API.

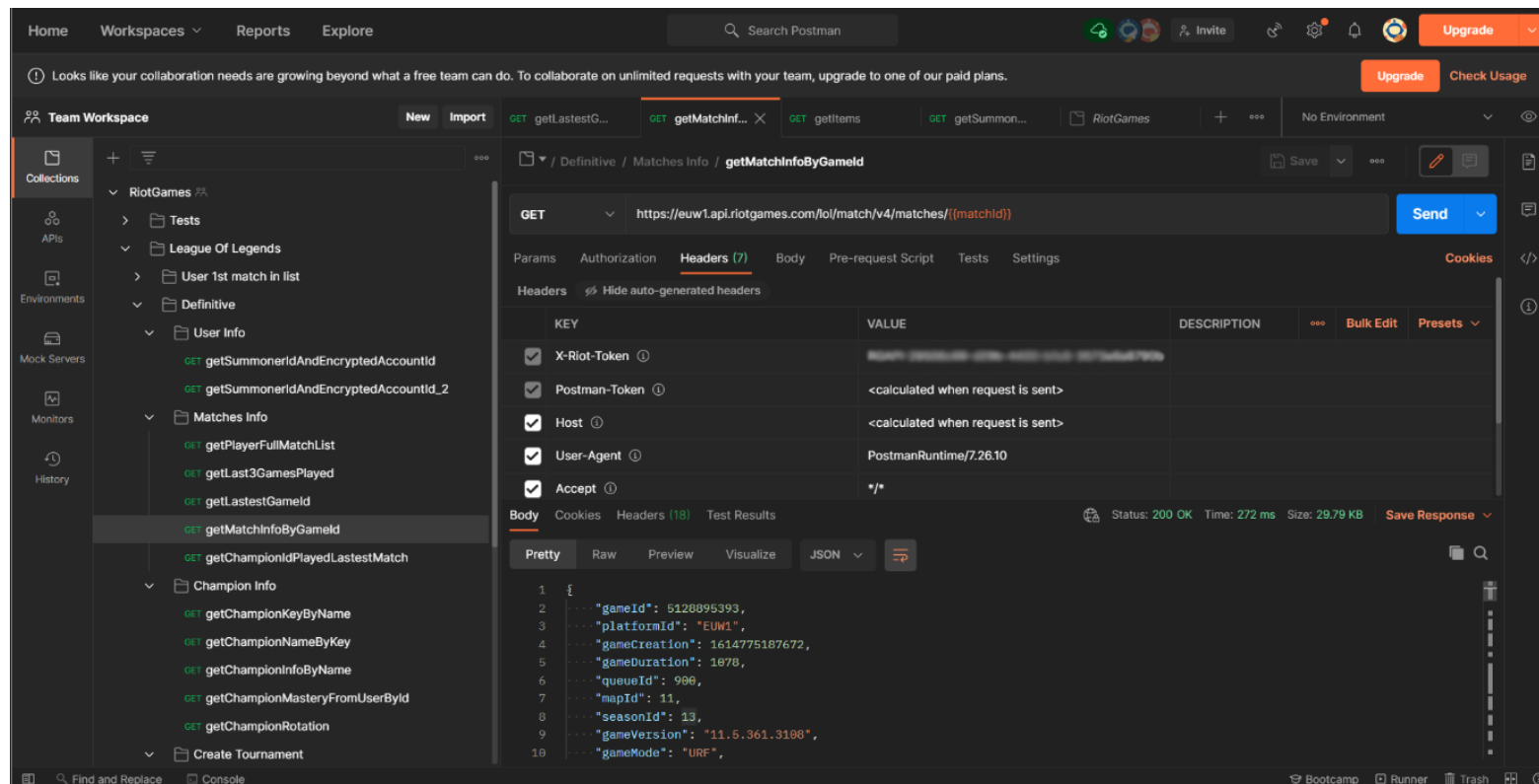
Devido à falta de uma licença de desenvolvimento da Riot com privilégios mais elevados e também por questões de segurança, a API key apenas dura 24h até ser invalidada.

Existe ainda uma limitação de 20 pedidos por segundo, ou 100 a cada 2 minutos.



DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

No Postman, um pedido HTTPS do tipo GET para a API seria algo deste género:



DESCRIÇÃO DO MODO DE FUNCIONAMENTO DA API

Quanto à comunicação com a API é necessário executar um pedido (GET,POST,DELETE, etc.) e inserir o "Routing Value" correto, de forma a obter a informação pretendida. Nesta API em específico esse "Routing Value" divide-se em 3 partes: a base (api.riotgames.com), a região/plataforma e o caminho específico para a função pretendida.

GET



<https://euw1.api.riotgames.com/lol/match/v4/matches/{{matchId}}>

Região/Plataforma

Base

Caminho

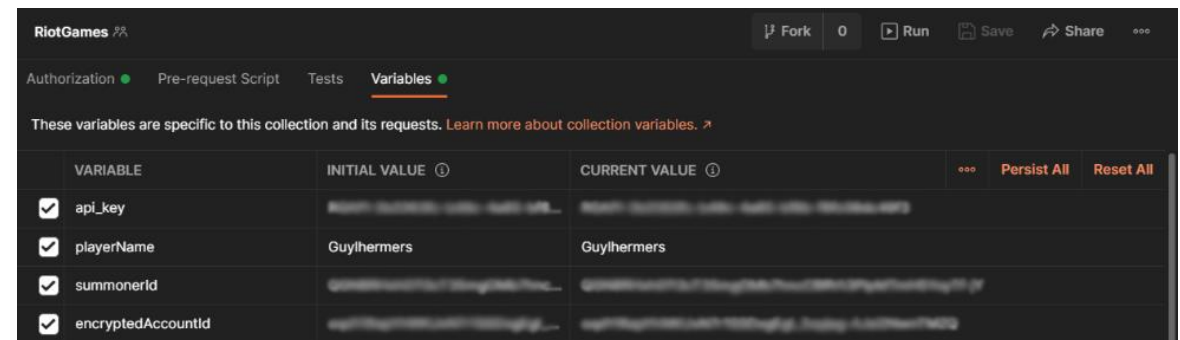
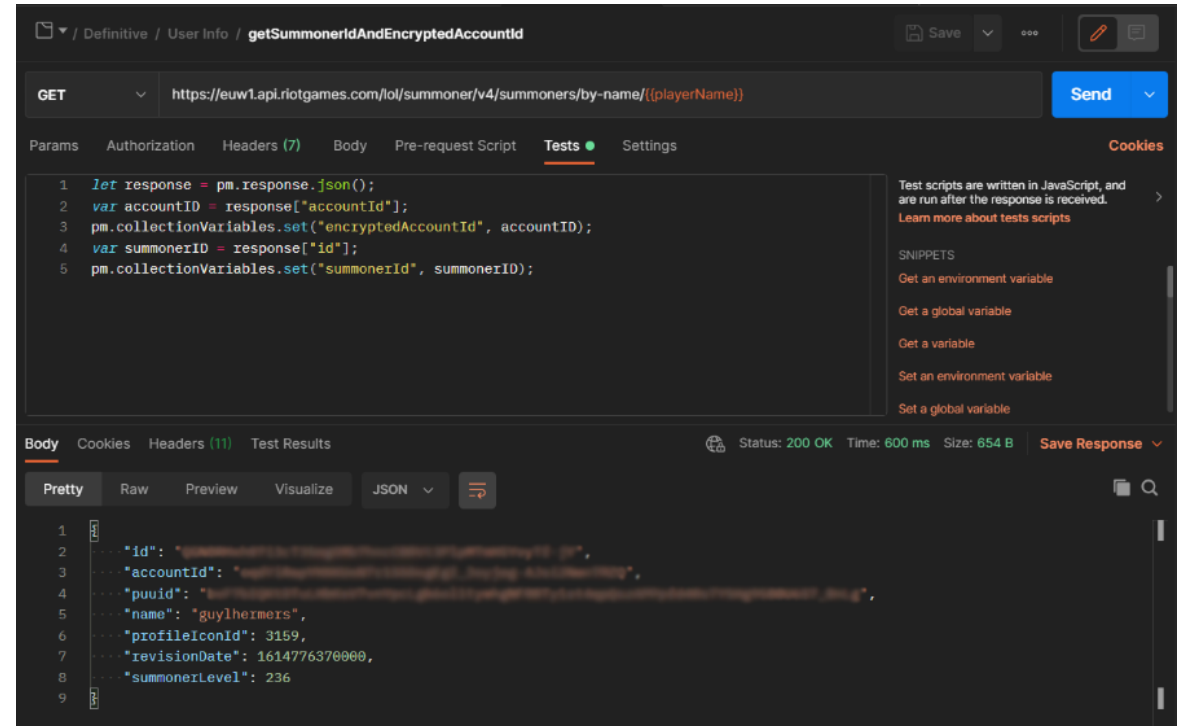
PLATFORM	HOST
BRI	br1.api.riotgames.com
EUN1	eun1.api.riotgames.com
EUW1	euw1.api.riotgames.com
JPI	jpl.api.riotgames.com
KR	kr.api.riotgames.com
LAI	la1.api.riotgames.com
LA2	la2.api.riotgames.com
NA1	na1.api.riotgames.com
OC1	oc1.api.riotgames.com
TRI	tr1.api.riotgames.com
RU	ru.api.riotgames.com

REGION	HOST
AMERICAS	americas.api.riotgames.com
ASIA	asia.api.riotgames.com
EUROPE	europe.api.riotgames.com

DESCRIÇÃO DE FUNÇÕES

Como primeira função iremos obter o Summoner ID e o Account ID de um jogador através do seu nome e a região em que joga, devido a ser um dado necessário para os pedidos seguintes.

Este pedido será do tipo GET e após o pedido, através de um script iremos guardar os dados pretendidos em variáveis da coleção.

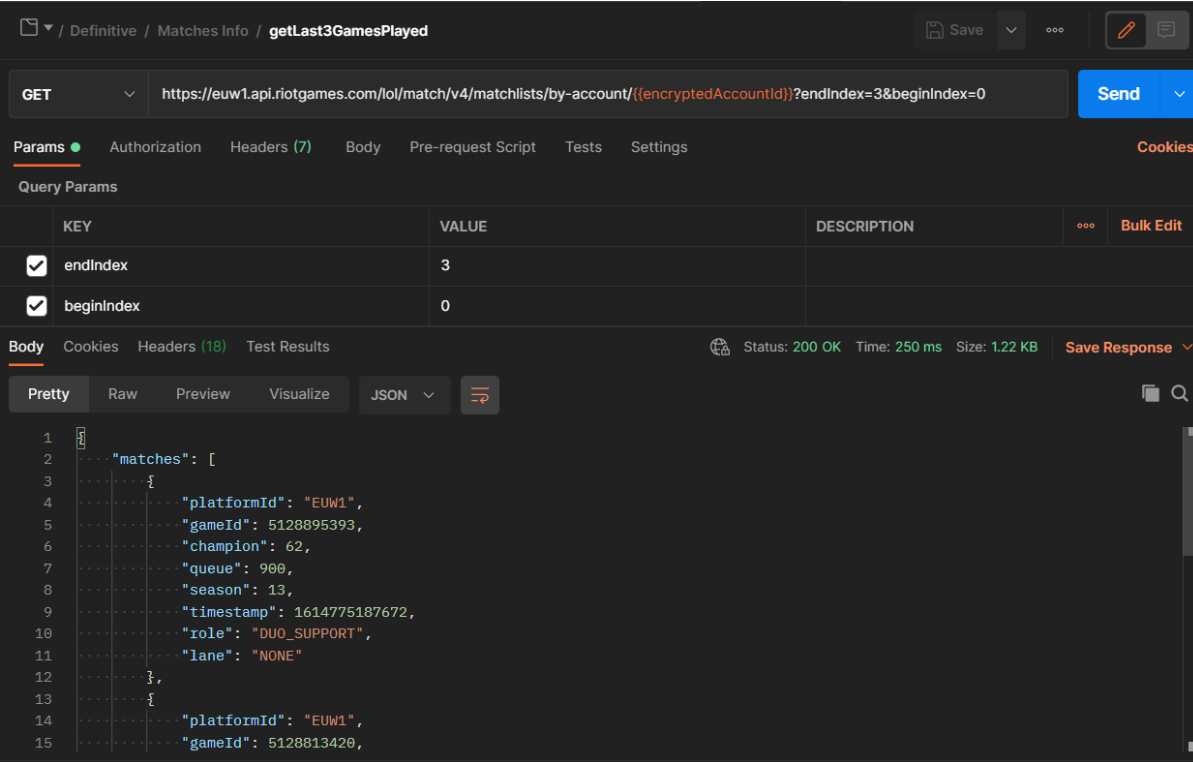


DESCRIÇÃO DE FUNÇÕES

De seguida com o Account ID obtido anteriormente e com uma outra função iremos obter os últimos 3 jogos do jogador em questão.

Para a realização desta operação iremos utilizar a variável mencionada acima, e ainda passaremos como parâmetro o índice final e o índice inicial dos jogos que pretendemos.

NOTA: Além disso, existe um conjunto de parâmetros opcionais para cada função da API que não usamos.



The screenshot shows a REST client interface with the following details:

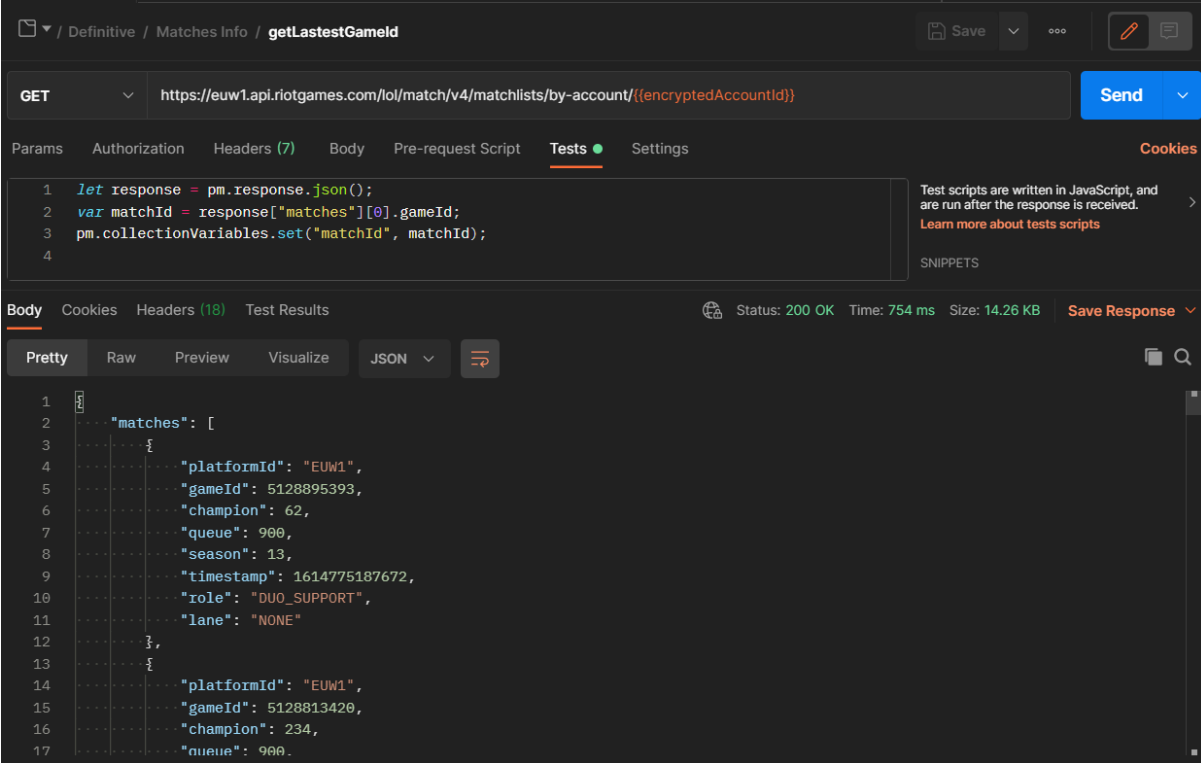
- Method:** GET
- URL:** `https://euw1.api.riotgames.com/lol/match/v4/matchlists/by-account/{encryptedAccountId}?startIndex=3&beginIndex=0`
- Params:** Authorization, Headers (7), Body, Pre-request Script, Tests, Settings
- Query Params:**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> endIndex	3	
<input checked="" type="checkbox"/> beginIndex	0	
- Body:** Cookies, Headers (18), Test Results
- Status:** 200 OK, Time: 250 ms, Size: 1.22 KB
- Response:** Pretty, Raw, Preview, Visualize, JSON
- JSON Response:**

```
1 {
2   "matches": [
3     {
4       "platformId": "EUW1",
5       "gameId": 5128895393,
6       "champion": 62,
7       "queue": 900,
8       "season": 13,
9       "timestamp": 1614775187672,
10      "role": "DUO_SUPPORT",
11      "lane": "NONE"
12    },
13    {
14      "platformId": "EUW1",
15      "gameId": 5128813420,
```

DESCRIÇÃO DE FUNÇÕES

Através da função usada anteriormente vamos guardar numa variável o game ID do jogo mais recente.



The screenshot shows a REST client interface with a GET request to `https://euw1.api.riotgames.com/lol/match/v4/matchlists/by-account/{{encryptedAccountId}}`. The request is successful, returning a 200 OK status. The response body is a JSON array of match objects. The first match object is expanded, showing details like platformId, gameId, champion, queue, season, timestamp, role, and lane.

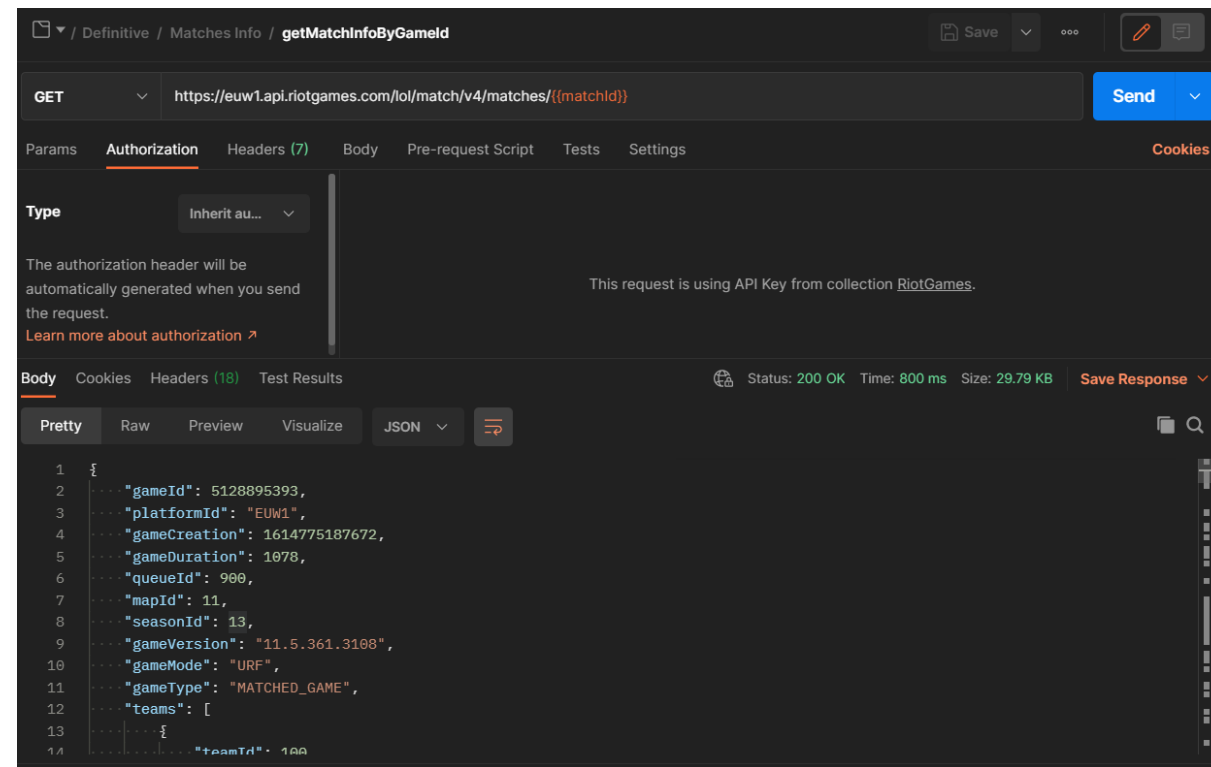
```
1 let response = pm.response.json();
2 var matchId = response["matches"][0].gameId;
3 pm.collectionVariables.set("matchId", matchId);
4
```

```
1 {
2   "matches": [
3     {
4       "platformId": "EUW1",
5       "gameId": 5128895393,
6       "champion": 62,
7       "queue": 900,
8       "season": 13,
9       "timestamp": 1614775187672,
10      "role": "DUO_SUPPORT",
11      "lane": "NONE"
12    },
13    {
14      "platformId": "EUW1",
15      "gameId": 5128813420,
16      "champion": 234,
17      "queue": 900
18    }
19  ]
20 }
```

DESCRIÇÃO DE FUNÇÕES

Com o Game ID obtido anteriormente na função destinada à obtenção da informação relativa a um jogo em específico iremos obter toda a informação relativa a esse jogo.

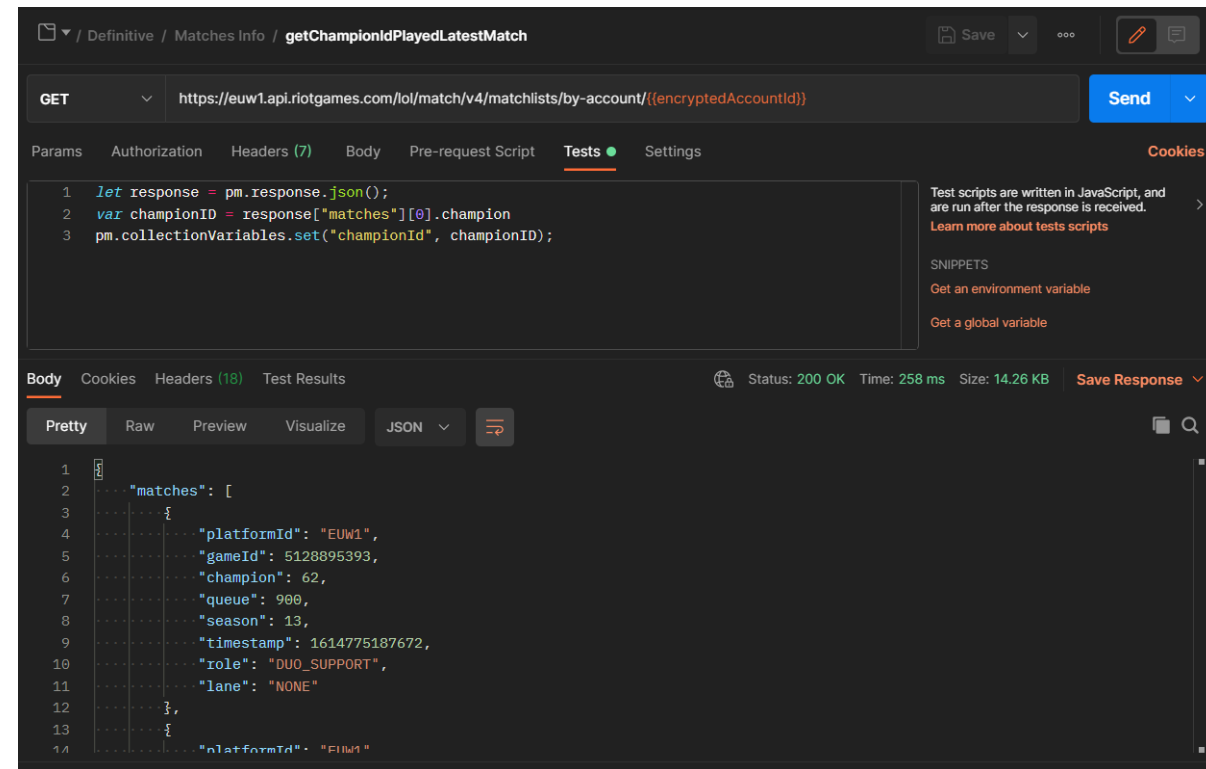
Algumas dessas informações são relativas aos Champions (personagens) utilizados, KDA (Kill, Death, Assist) de cada jogador participante, resultado final do jogo, etc..



DESCRIÇÃO DE FUNÇÕES

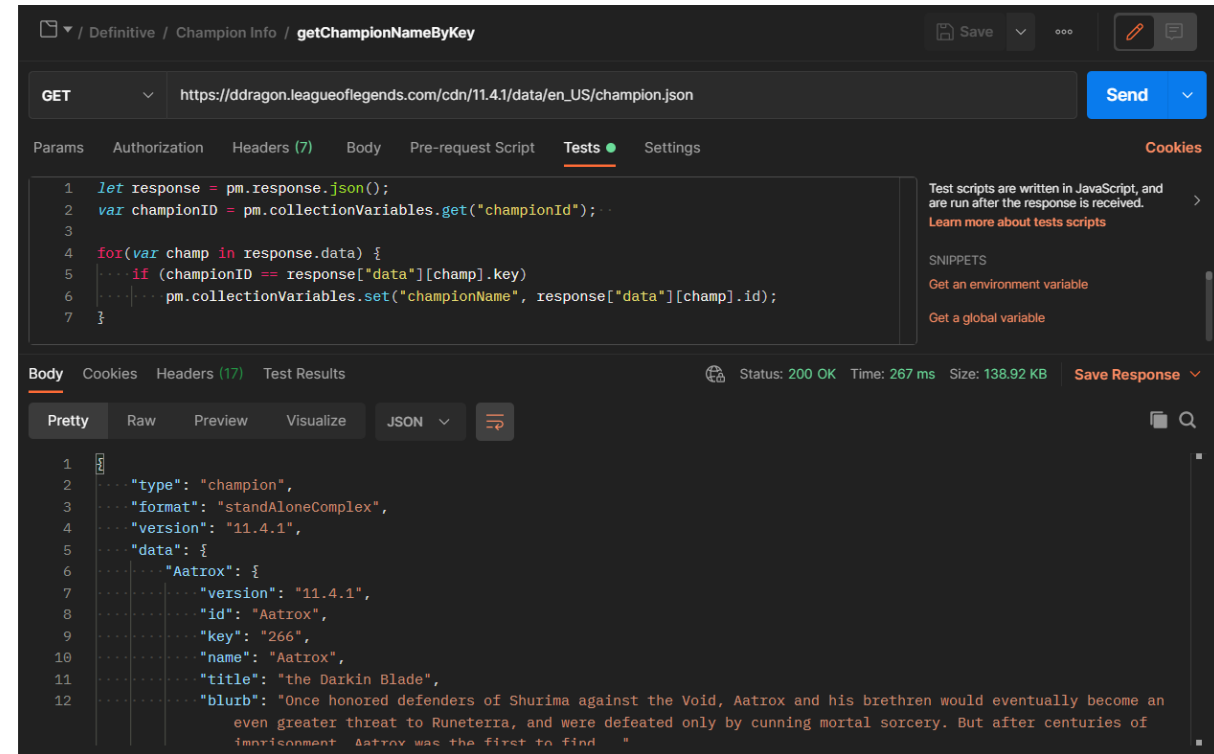
Com a função utilizada para obter a match list iremos guardar numa variável o ID (key) do Champion utilizado no último jogo.

Com este ID poderemos consultar informações mais detalhadas sobre o mesmo.



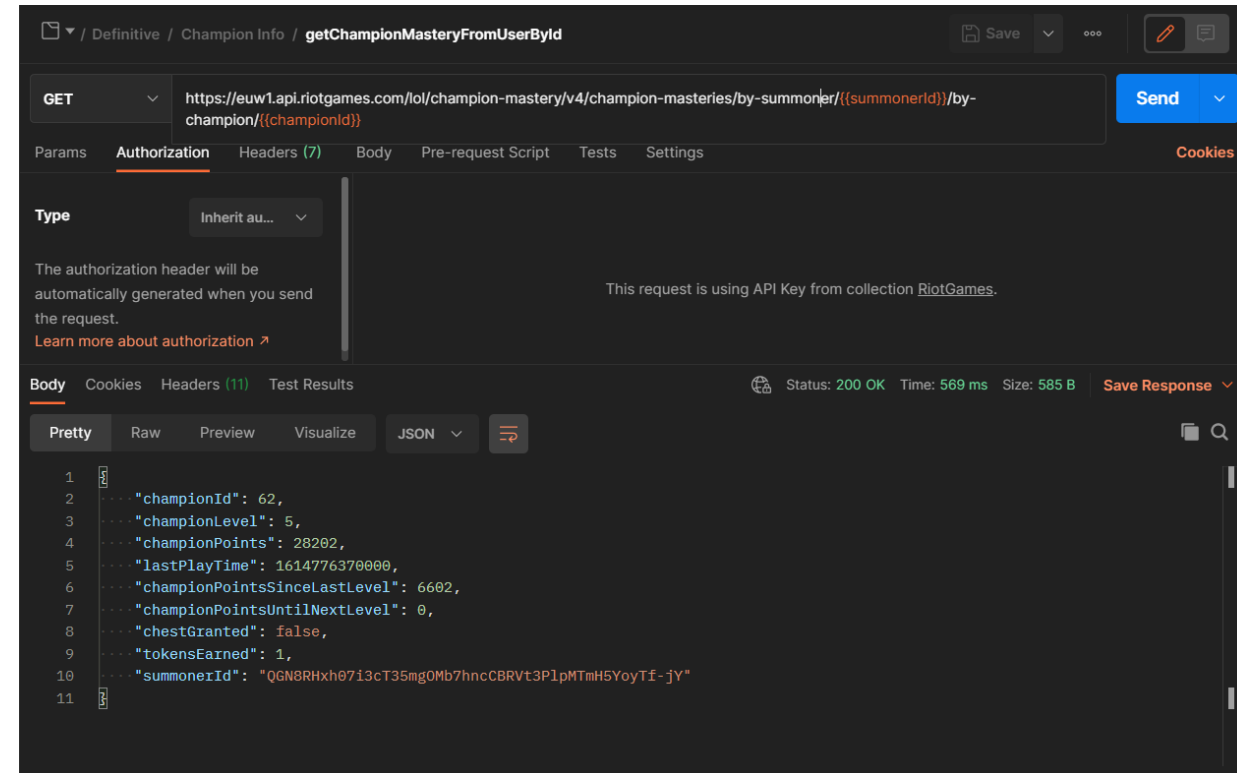
DESCRIÇÃO DE FUNÇÕES

Com esta função, podemos utilizar a key do Champion utilizado para podermos obter o seu nome, tal como mais informações: o seu título, versão e descrição.



DESCRIÇÃO DE FUNÇÕES

Com esta função, vamos obter a pontuação individual de um jogador num determinado Champion, bem como o seu último jogo com o mesmo e o seu nível de mestria.

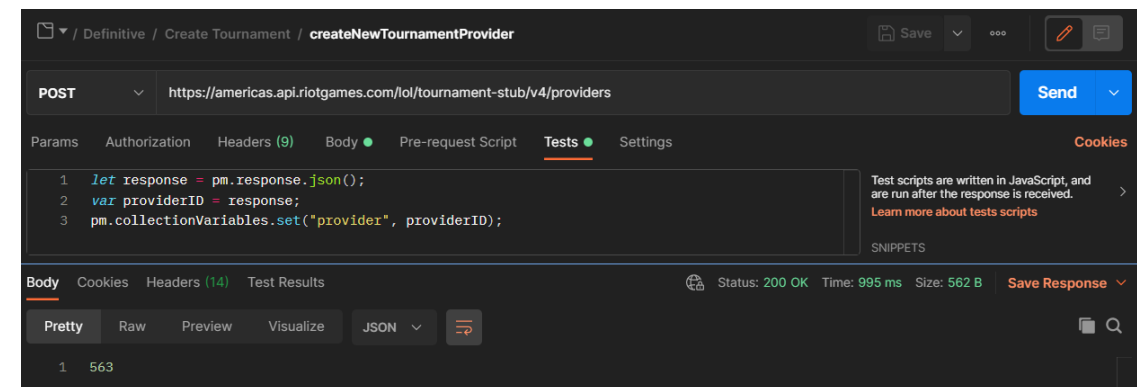
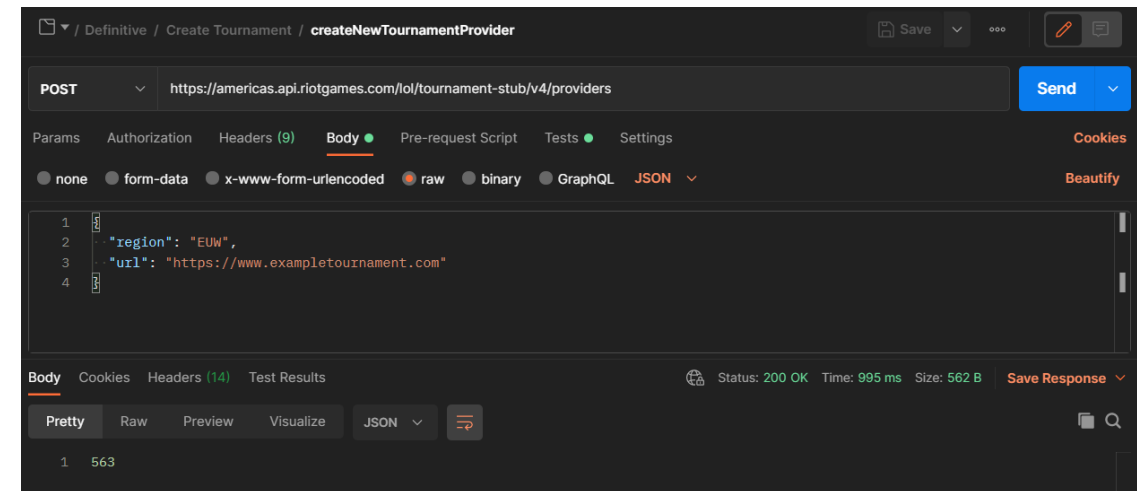


DESCRIÇÃO DE FUNÇÕES

Com esta função procede-se à criação de um organizador de torneios, sendo o output da mesma um número de identificação (Provider ID) necessário para a criação de torneios.

Para isso no corpo deste pedido estão contidos dois campos necessários: a região para a qual será criado o torneio e o link que levará ao site do organizador do evento.

Ainda após efetuado o pedido será preciso guardar numa variável o ID que identifica o organizador do evento (providerId).

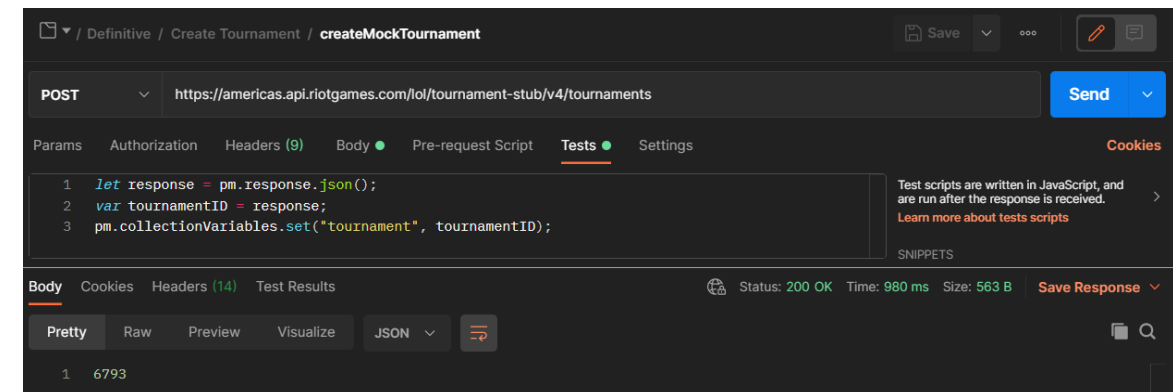
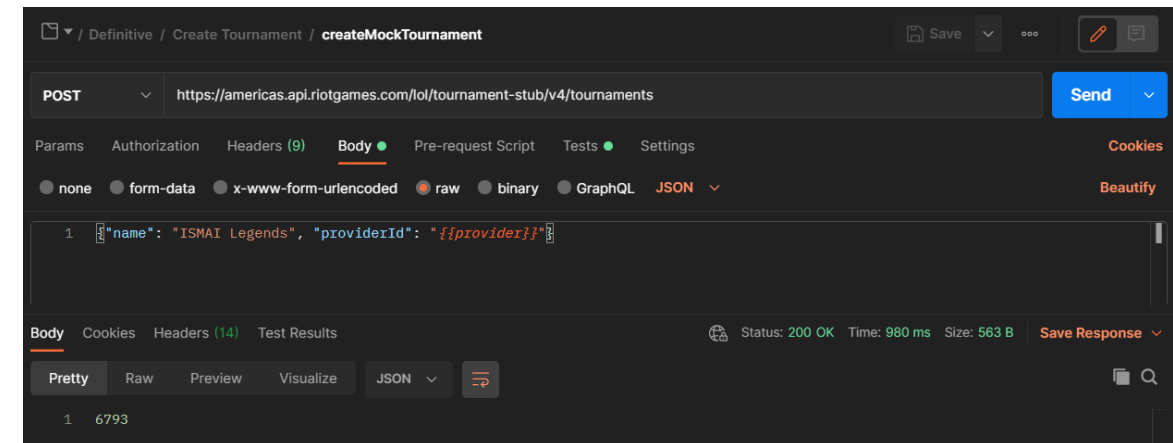


DESCRIÇÃO DE FUNÇÕES

Esta função ocorre em seguimento à anterior, utilizando a API Key e o Provider ID para proceder à criação de um torneio, sendo o output desta função o Tournament ID, utilizado posteriormente para a criação de códigos e para a alteração de detalhes do mesmo.

Para isso é necessário no corpo do pedido feito à API que constem: o nome do evento e o ID do organizador do mesmo.

Após o pedido é necessário guardar o ID do torneio numa variável para uso futuro.

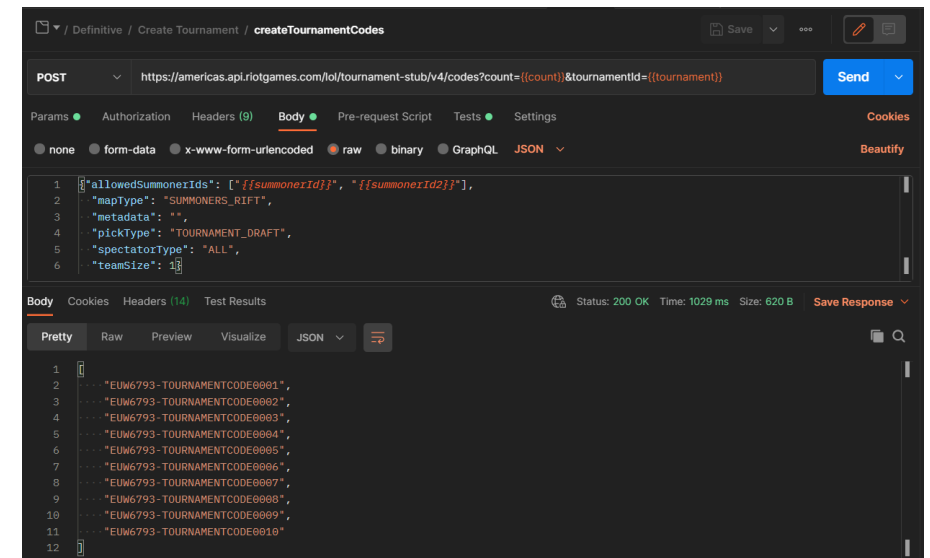


DESCRIÇÃO DE FUNÇÕES

Após a criação de um torneio e a obtenção de um Tournament ID, podemos usar esse número de identificação para criar códigos relativos ao torneio. Estes códigos serão utilizados pelos jogadores para poderem participar no mesmo.

De forma a isso ser possível no corpo do pedido é necessário que constem os seguintes campos: os summonerId dos jogadores participantes, tipo de mapa, tamanho de cada equipa, etc. .

Finalmente guardar os códigos gerados pela API numa variável de forma a enviar aos jogadores para puderem participar no torneio.

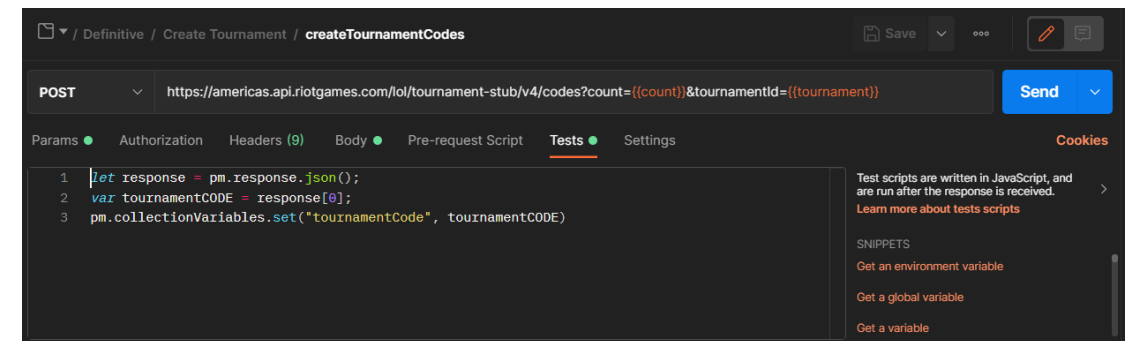


The screenshot shows a REST client interface with a POST request to `https://americas.api.riotgames.com/lol/tournament-stub/v4/codes?count={{count}}&tournamentId={{tournamentId}}`. The request body is a JSON object with the following fields:

```
1 {"allowedSummonerIds": ["{{summonerId1}}", "{{summonerId2}}"],
2  "mapType": "SUMMONERS_RIFT",
3  "metadata": "",
4  "pickType": "TOURNAMENT_DRAFT",
5  "spectatorType": "ALL",
6  "teamSize": 1}
```

The response is shown in the 'Body' tab, displaying a list of 10 tournament codes:

```
1 "EUM6793-TOURNAMENTCODE0001",
2 "EUM6793-TOURNAMENTCODE0002",
3 "EUM6793-TOURNAMENTCODE0003",
4 "EUM6793-TOURNAMENTCODE0004",
5 "EUM6793-TOURNAMENTCODE0005",
6 "EUM6793-TOURNAMENTCODE0006",
7 "EUM6793-TOURNAMENTCODE0007",
8 "EUM6793-TOURNAMENTCODE0008",
9 "EUM6793-TOURNAMENTCODE0009",
10 "EUM6793-TOURNAMENTCODE0010"
```



The screenshot shows the same REST client interface, but with the 'Tests' tab selected. A test script is written in JavaScript:

```
1 let response = pm.response.json();
2 var tournamentCODE = response[0];
3 pm.collectionVariables.set("tournamentCode", tournamentCODE)
```

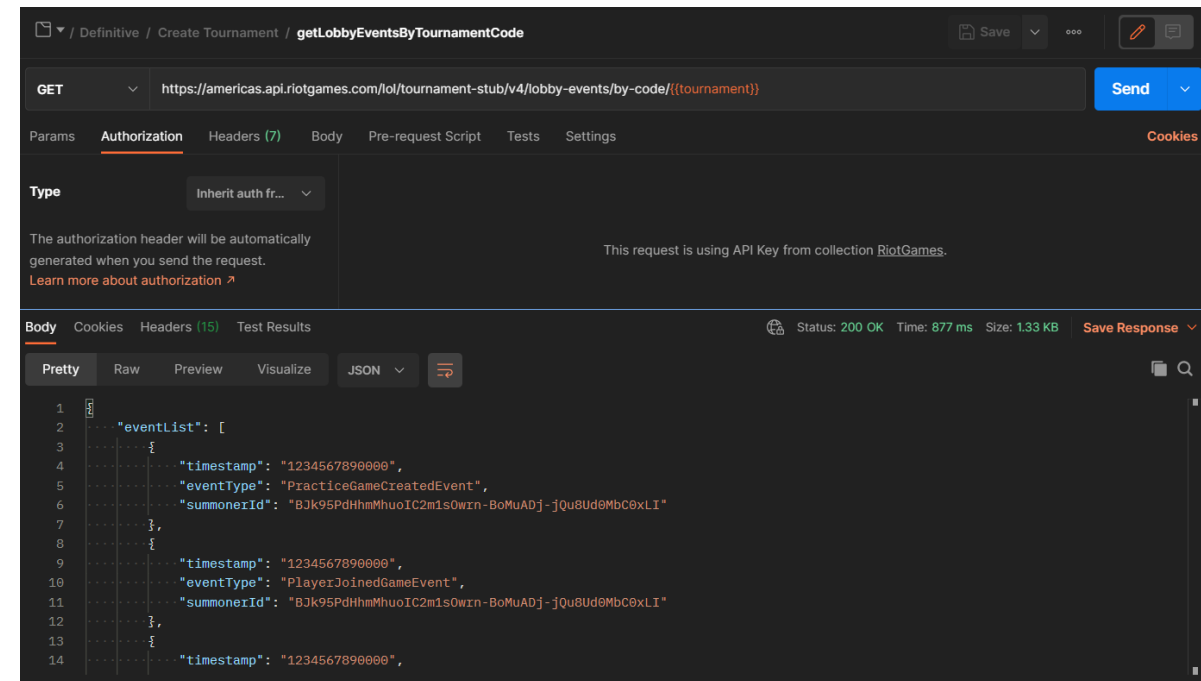
On the right side, there is a note: "Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts". Below this, there are snippets for setting environment variables, global variables, and local variables.

DESCRIÇÃO DE FUNÇÕES

Com intuito de verificar as ações que decorrem no torneio é possível usar a esta função.

Esta irá retornar um histórico de ações realizadas pelos jogadores.

Neste função apenas é necessário a introdução do TournamentId obtido anteriormente.

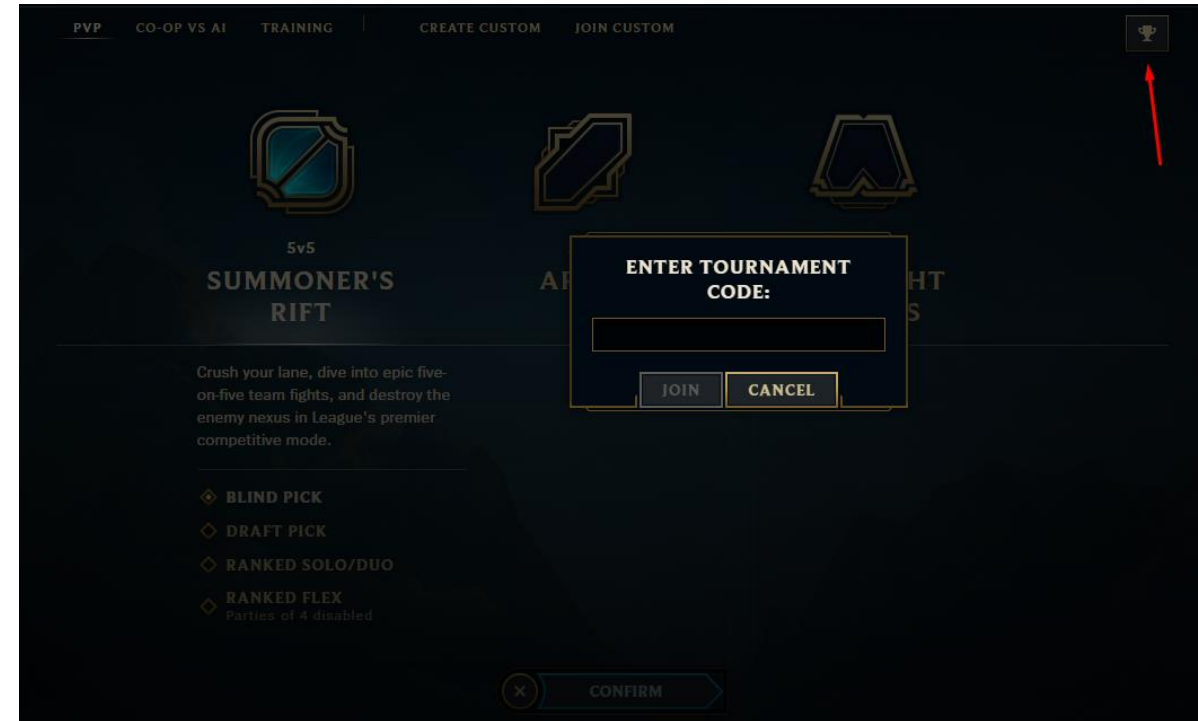


DESCRIÇÃO DE FUNÇÕES

As funções referidas anteriormente são utilizadas para criar um "Mock Tournament", ou seja, para testar as funções de criação de um torneio.

Para proceder à criação de um torneio é necessário obter permissão especificamente para esse fim por parte da Riot Games, sendo esse processo mais complexo.

Por ser um torneio teste não foi possível testar os códigos no Client do League of Legends, mas estes códigos seriam introduzidos numa área destinada à inserção dos mesmos.



AVALIAÇÃO E ANÁLISE CRÍTICA - PRÓS

No geral é uma API bem documentada e estruturada, as funções são intuitivas e o output é claro, sendo simples de navegar. Apesar da simplicidade superficial da API, esta possui diferentes níveis de complexidade sendo possível utilizar as funções de maneira a obter informações muito específicas.

É também uma boa prática a fácil acessibilidade e utilização da API, permitindo a qualquer pessoa começar a desenvolver aplicações com bastante utilidade para a comunidade.

A simplicidade e complexidade desta API fazem com que as aplicações provenientes dela sejam, se devidamente desenvolvidas, ferramentas com diversas funcionalidades, que apresentam as informações de forma mais organizada e simples para utilizadores casuais, mas tal como a API, apresentam diferentes níveis de complexidade destinados a diferentes níveis de competitividade.

AVALIAÇÃO E ANÁLISE CRÍTICA - CONTRAS

Durante a exploração da API encontramos algumas dificuldades relativas ao uso da mesma nas quais achamos que deveríamos dar a devida atenção e a nossa opinião sobre elas.

Uma das dificuldades que encontramos foi através do Champion ID podermos obter a informação do mesmo, tais como nome, descrição, estatísticas, etc.. De acordo com a documentação da API disponível no site a única forma de consultar essas informações é recorrendo a documento JSON disponível noutro domínio pertencente à Riot Games e utilizando um ciclo "for" comparando a key de cada objecto contido até chegar ao objecto que a key corresponde e daí retirar a informação relativa a esse Champion.

AVALIAÇÃO E ANÁLISE CRÍTICA - CONTRAS

Outro dos problemas que nos deparamos foi o nome de variáveis que continham o mesmo valor serem trocados de função para função da API, o que por vezes se tornou um pouco confuso na utilização dessas variáveis noutras funções.

Alguns dos casos são:

- A key “accountId” do ficheiro JSON que recebemos do método GET em SUMMONER-V4 e em MATCH-V4 ([GET] /lol/match/v4/matches/{matchId}), que em MATCH-v4 ([GET] /lol/match/v4/matchlists/by-account/{encryptedAccountId}) passa a ter o nome de “encryptedAccountId”;
- A key “champion” que contém um valor do tipo INT no ficheiro JSON que recebemos do método [GET] /lol/match/v4/matchlists/by-account/{encryptedAccountId}, que depois corresponde à key “key” do ficheiro JSON que contém a informação de todos os champions do jogo. Ainda no mesmo ficheiro existe uma key “id” que é do tipo STRING que contém o nome do champion, quando seria de esperar que "id" se referisse a um número único que o distingue.

AVALIAÇÃO E ANÁLISE CRÍTICA - CONTRAS

Mais uma das críticas que achamos relevantes e o facto do número de funções que nos são disponibilizadas é um pouco limitado.

Embora seja possível submeter um projeto de aplicação mais concreto, que após verificação por parte da Riot será possível usar uma licença de desenvolvimento mais completa.