RIOT GAMES API CHALLENGE 2016

CAN I REACH CHALLENGER?

FABIO AUGUSTO ZUIN
LORRAN THIERRY DAVID

May 7, 2016

1. Introduction

As long-time players of League of Legends, we know that details are a big factor when it comes to winning ranked games. Every player of every region has yet to develop an understanding of the “metagame”, their strengths and weakness. That is the goal of “Can I Reach Challenger?”: provide enough information to help people get better and be successful on the Fields of Justice.

2. Our initial idea

We have always wondered how to make a cool application that could help players from our region to get better. At first we thought about comparing builds of the most played champions with the best build test based on the champions overall win rate but it wasn’t what we were expecting.

One week before the challenge deadline, we began to think again and surprisingly we had a good idea: create a website that compares you to the challenger league of your region. Even though it was a great idea, we did not expect how many calls we’d have to make to the API.

After wondering we decided to do something simple: compare a summoner to a random challenger player of its region, show some charts and give tips to help people reach their goals.

3. Data

To get all the statistics necessary we’re using Riot Games’ API that provides thousands of information for each player or game. We developed a few cURL requests to capture the data of the summoner, a ranked challenger player and five ranked games each. The result is a big JSON used to manipulate the data and show to the user just what he need to see.

Many people would think: why do we not use a database like MySQL or MongoDB?

The answer is short and direct, we wanted to make it easier and faster because we did not have enough time.

```php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://' . $summonerRegion . '.api.pvp.net/api/lol/' . $summonerRegion .
'/v1.4/summoner/by-name/' . $summonerName . '?api_key=                                    ');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
$response = curl_exec($ch);
```

Figure 1 - Using cURL to get summoner's information.

```php
$response_json = json_decode($response);

$summoner_id = $response_json->$summonerName->id;
$summoner_profileIcon = $response_json->$summonerName->profileIconId;
$summoner_level = $response_json->$summonerName->summonerLevel;
```

Figure 2 - Decoding the JSON response and setting up the data.

4. Back-end

We're using PHP and Slim Framework for the backend. We chose PHP because it can be used everywhere and is supported for most of the web servers, also it supports a library called cURL, or libcurl, which allow us to connect and request data from Riot Games' API. Slim Framework allow us to create a REST application, it's very lightweight and simple to use.

5. Front-end

For the front-end we're using AngularJS and the main reason behind this decision is: we want the app to be organized. Angular JS implements MVC by itself. AngularJS also uses a declarative interface and directives which help us to bring new functionalities, like our loader, to the app's user interface.

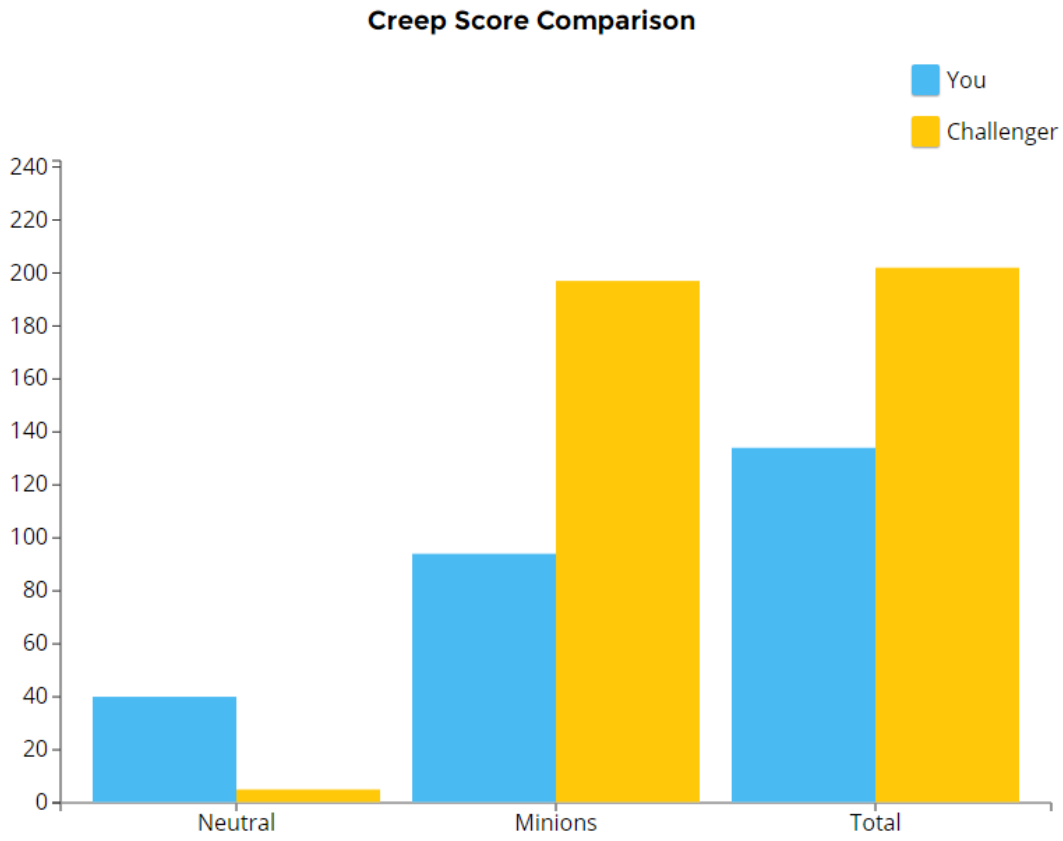

Figure 3 - Data as a chart using AngularJS.