

Relatório Trabalho Prático – Rest Apps

Nomes: Caio Lelis e João Vítor Braga

O trabalho TP-02-REST-app da matéria Tecnologia de Programação foi feito por nossa dupla utilizando a api da Riot Games, desenvolvedora do game League of Legends, pois é um jogo que nós adoramos e teríamos mais interesse no desenvolvimento. A api nos disponibilizava a busca de um jogador pelo seu username e com essa busca poderia ser feito um detalhamento de suas partidas passadas, seus resultados e seu elo, a lista de todos os campeões e uma busca pelo id do campeão e, além disso, a disponibilização das maestrias de cada campeão para cada jogador.

Nosso primeiro planejamento contava com a ideia de fazer uma pesquisa pelo username e detalhar a conta do usuário ou uma pesquisa pelo campeão e detalhá-lo. Entretanto, optamos por fazer apenas uma listagem de todos os campeões e quando selecionados, recebíamos as informações disponíveis na api, como por exemplo, o nome, o título do campeão, um blurb (uma pequena história sobre o campeão), o id, e muitas outras informações. A imagem abaixo mostra o que a api (http://ddragon.leagueoflegends.com/cdn/8.17.1/data/en_US/champion.json) oferecia em relação à informação:

```
data:
  Aatrox:
    version: "8.11.1"
    id: "Aatrox"
    key: "266"
    name: "Aatrox"
    title: "the Darkin Blade"
    blurb: "One of the ancient darkin, Aatrox was once a peerless swordmaster who reveled in the bloody chaos of the battlefield. Trapped within his own blade by the magic of his foes, he waited out the millennia for a suitable host to wield him-this mortal warrior..."
    info:
      attack: 8
      defense: 4
      magic: 3
      difficulty: 4
    image:
      full: "Aatrox.png"
      sprite: "champion0.png"
      group: "champion"
      x: 0
      y: 0
      w: 48
      h: 48
    tags:
      0: "Fighter"
      1: "Tank"
      partype: "Blood Well"
  stats:
    hp: 580
    hpperlevel: 85
    mp: 100
    mpperlevel: 0
    movespeed: 345
    armor: 33
    armorperlevel: 3.8
    spellblock: 32.1
    spellblockperlevel: 1.25
    attackrange: 150
    hpregen: 6.5
    hpregenperlevel: 0.5
    mpregen: 0
    mpregenperlevel: 0
    crit: 0
    critperlevel: 0
    attackdamage: 70
    attackdamageperlevel: 3.2
    attackspeedoffset: -0.04
    attackspeedperlevel: 3
```

Tivemos alguns problemas durante o projeto no caching, no deserializer e no rate limit da api. O problema do caching foi o mal aproveitamento do tempo e não conseguimos aprender a tempo de realizá-lo, já o deserializer, foi muito complicado de dominar e por não ser uma matéria vista no curso, demoramos mais tempo para aprende-la, entretanto, após entender seu funcionamento conseguimos realizar suas funções de maneira tranquila. Por fim, o rate limit da api é baixíssimo nos forçando a esperar muito tempo para realizar novas requests, criar novas contas para testar o programa e ficarmos dependentes da boa vontade da api.

Tendo ciência disso, podemos concluir que nosso trabalho não foi um dos melhores, mas desconsiderando o má aproveitamento do tempo, fizemos o que estava em nosso alcance.