

Hello,

Thank you very much for your interest in lolguru.gg. Before starting, we would like to have a test for your skills.

## DESCRIPTION

We would like to have a small web-based application to fetch ongoing game matches from a server and showing detailed information of a game.



Figure 1: Homepage

This is the homepage when you first access to the application. It shows all predefined servers of LoL. When you click on a server, figure 2 is shown.

BR	
<b>CLASSIC/MATCHED_GAME</b>	06/05/2016 20:30
SkHaseo Nemozeiro Khâ Zik Bill Cipher MeuKokoroTaZuado	Skep'sTar Bastia0 Heidenan fightingforlove DAT BF
<b>CLASSIC/MATCHED_GAME</b>	06/05/2016 20:30
Swordslord Ziggs BOT PIGS LostSoul Jingg 1 Pato	kkkkkkkk VAI DRAGON LORD Boss MonkeyKing1 Leo César
<b>ARAM/MATCHED_GAME</b>	06/05/2016 20:30
NoT.Xit x0rthecranzyx O Thresh xNietz Shadow KnightH	Vyacheslavskaya ThePlaymaker j/m Escherichia Dethl
<b>CLASSIC/MATCHED_GAME</b>	06/05/2016 20:30

Figure 2: Ongoing games in server BR

Every game show its mode and type, starting time and a list of participants from two teams. When you click on any team member, the page changes to show detailed information of a game, as in figure 3.

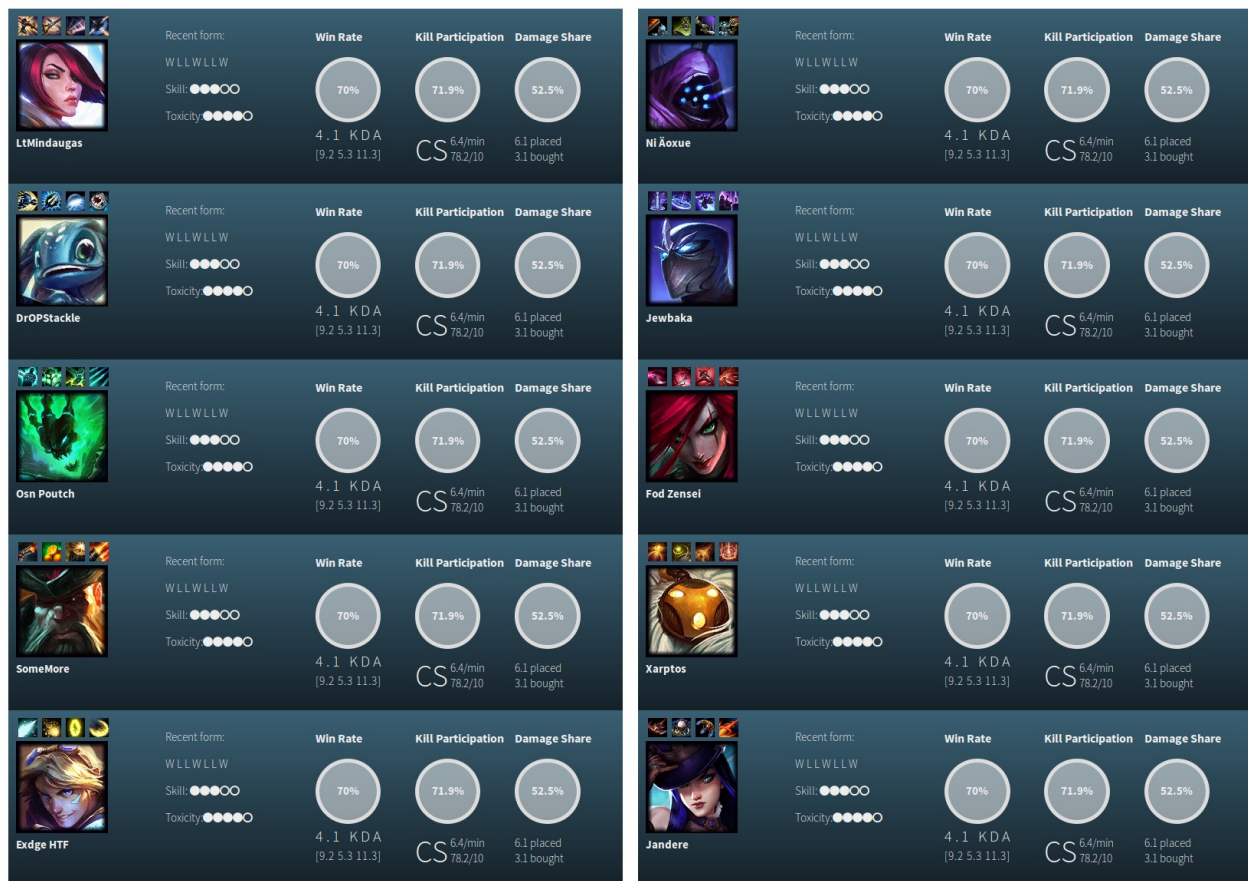


Figure 3: Game information

In figure 3, we would like to have those data included:

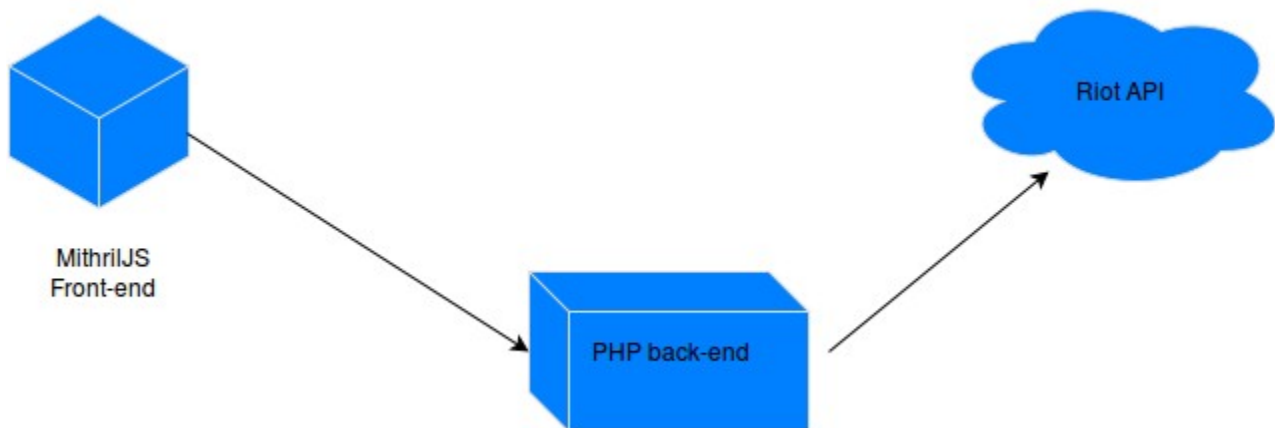
- Player name (summoner name)
- Champion avatar
- Champion's spells and its icon
- Recent form: showing 7 last records of the player (how many wins/loses)

All other values are optional and it would be nice if you manage to calculate and show them (see **Strictly optional values**, page 6 of this document ).

## REQUIREMENTS

Obviously, implement a website as described in the **Description**. We want you to study

and use MithrilJS <<http://mithril.js.org/>> in the front-end. The back-end will use PHP to interact with Riot API <<https://developer.riotgames.com>>.



The front-end page SHOULD have 2 routes:

- / the homepage
- /match/:summonerId as in figure 3

Notes:

- MithrilJS MUST be used in front-end. The website is a single-page application, i.e. everything should be AJAX-powered and no refresh.
- Minimum PHP version is 5.5. You are NOT allowed to use any PHP framework. Just vanilla PHP with Composer <<http://getcomposer.org/>> and libraries from Packagist <<http://packagist.org/>>.

## INSTRUCTIONS

- Send an email to An <[kcj.pop.com@gmail.com](mailto:kcj.pop.com@gmail.com)> /cc Ákos <[akos.podolszki@lolguru.gg](mailto:akos.podolszki@lolguru.gg)> and include your Gitlab <<http://gitlab.com/>> username or email. We'll add you in the test repo.
- There you'll find the HTML layouts, along with this instruction.
- Fork the repo and create a new branch in format `firstname-lastname-

<randomstring>`.e.g. `an-ca0-theloser`

- Work on that branch and push it back to Gitlab **before 23:59 5<sup>th</sup> June 2016 GMT**.  
Don't worry if you finish it before. We can check the history to see who was fast.
- You can use this credential to interact with Riot API

Key: 27bec553-bfba-470e-a5c6-902fb6f0b0ff

Username: lolgurugg

Pasword: lolgurutester1

Server: EU West

Rate limits are:

10 requests every 10 seconds

500 requests every 10 minutes

## EVALUATION CRITERIA

- **IT DOES NOT NEED TO WORK**. It's nice if you can complete the test, but we want to see how you approach new tools, your attitude and skills. You are free to make some extra functions if you think it's cool.
- We **DO** look at your coding style. PHP code **MUST** confront PSR-2  
<<http://www.php-fig.org/psr/psr-2/>>. Have a look at PHP The Right Way  
<<http://www.phptherightway.com>>.
- You can write JavaScript in ES5/6. Just to make sure that we can run it for evaluation. You can follow `standard` <<http://standardjs.com/>> but this is optional.
- We **DO** have a look at how you use git & your style of writing commit messages.

If you have any question, feel free to ask by email or on Gitter <<https://gitter.im>>. Ping An <<https://gitter.im/kcipop>> to add yourself into the chat room.

## **STRICTLY OPTIONAL VALUES**

All these stats are for only the champion the player currently playing, they are Not total values for all the players games. For example if the player is playing Fizz and he so far played 4 games with Fizz then you need to check what happened in only that 4 games (the other games with other champions does not matter). Round numbers to one decimal (e.g. 4,6).

KDA: Average Kill / Death / Assist ratio with the champion he is currently playing.

$KDA = (Kill + Assist) / Death$  If the player never died we still divide with 1 not 0. For example 3/0/5 and 3/1/5 results in the same KDA score 8. Round the numbers to 1 decimal.

Win Rate=Wins/Total games with the selected champion only.

CS /min: Creep Score per minute. How many minions the player kills on average with the selected champion per minute. Find the minion kills and divide by the length of the game.

CS @10 (Creep Score at 10 minutes): Similar to above, but with this we want to show much minions the player has on average with the given champion at the 10 minute mark of the game. For example he kills average 82,5 minions with Fizz in 10 minutes.

Wards bought: Wards are an item in the game that grants vision in an area of the map. It have different kind. In the shop you can buy Vision Wards. The question is how many vision wards the player buys with the chosen champion on average.

Wards placed: How many wards he put down on the map during his games (average with the selected champ). This include all kind of wards not just the above mentioned ones.

Damage Share: How much percentage the player deals of the total team's damage

against enemy champions. For example his team's total damage is 10000 against enemy champions and he dealt 27000 damage against enemy champions his damage share will be 27%.

Kill Participation: How much kills the guy participated in from the team's total kills. For example the team have total 50 kills and the player has 10 kills and 15 assists than he participated in 50% of the kills.