TodoList app

Documentation

Summary :

I - What is TodoList and how it works?

TodoList is a web application that allows users to manage their daily tasks .

The users can add multiples new tasks to do with a title and a content.

When the tasks is done , they can mark it as done , if the tasks change , the user can edit the task , and if the user dont want to see the task anymore , they can delete it .

Currently , there is two types of users : the classic user and the admin.

The user can only access the task management section of the application.

The admin can create , edit , and acces to the list of users , but he can also use the app like a classic user and manage his tasks .

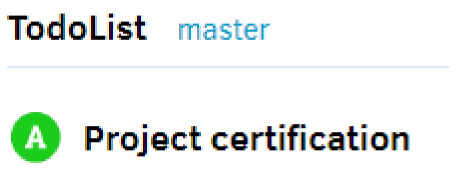
II - Code quality

Todolist uses the framework symfony , it just been updated and now run on Symfony 3.4

In therms of code quality , the project respect the PSR1 and PSR2 rules .

The project also stay as close as possible from the Symfony good practices.

The code quality has been analysed with Codacy and obtain the A grade :



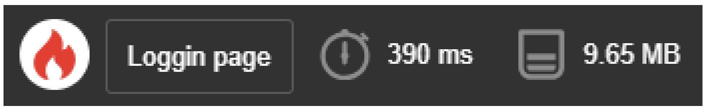
The project need to keep this grade in the future, every pull request have to respect PSR1/2 and symfony best practices.

III - Code performance

The code performace of the app has been analysed with Blackfire.

The actual pages load in less than 1/2 second, the app will try to keep this 2 second limit for the future.

The average loading time is between 300 and 1600 ms :



Here are the links of the main pages performances analysis :

Main page :

https://blackfire.io/profiles/a1db4d83-dbb7-461a-b309-00f7aaf56143/graph

Loggin page :

https://blackfire.io/profiles/1988d640-b3ae-42b7-84fc-4a3140609597/graph

Tasks list page :

https://blackfire.io/profiles/e04c8978-1cd2-46b0-8a43-753c5fd745d0/graph

Task edit page :

https://blackfire.io/profiles/dce1ac54-ca6c-482d-8239-af4e92f11a54/graph

As the graphics shows, Composer/Autoload /includeFiles takes up most of the loading time. This function need to be worked on to improve the quality of the application

IV - Areas for improvement

Here are some improvement who will need to be done in the future for short medium and long term.

Short term :

The user creation problematic :

For now , only admins can create users , so admins know the password of the users they create . This is a problem who cant stay like this , thats why there is a short term decision to make .

There is 2 main solutions who are possible :

Solution 1 :

Admins keep create the users , they generate a random password for each user. When the user is created , an email with the random password is sent to him , and inviting him to change it on the first connexion .

With this solution, the user have to edit the random password on the first connexion, and then his account is safe and he is the only one knowing his password .

Solution 2 :

Admins can't create users anymore . Each annonymous visitor can create his account , and enter his own password . So the password is immediatly safe .

With this solution, the amount of active user will vary according to the app popularity.

Solution 2,5 : If Todoapp want to keep the control of his users , this solution need to implement an admin valitation for each account . So the user create his account , the account is waiting for validation , the admin validate the account , and then the user can access to the app.

Medium term :

For medium term improvements, the loading time could be decreased by implement CDN loading for bootstrap . To consider for a faster app.

It could also be relevant to create an Api for TodoList , and then a mobile app to consume it , having access to Todolist everywhere could be usefull for the users and can increase the popularity of the Todolist . But this improvement will need a lot off ressources, this choice has to be considered in a mid/long term developpement of the app.

Long term :

For the long term improvements , we will consider that the Todolist app has a lot of success and a lot of users . With this huge amount of request to be processed simultaneously , a load balancing between multiples servers will be necessary .