



Упражнение: Консумиране на REST API

Messages

След като в последното занятие си създадохме Web API за съобщения, сега е време да имплементираме и Front-End частта.

Ще ни бъде предоставена обикновена HTML страница, оформена с Bootstrap. Страницата е конструирана като Front-End частта на приложението Messages. Съдържа проста форма за избор на текущото потребителско име и проста форма за изпращане на съобщения. Също така има списък с изпратените съобщения, които са всички съобщения, които в момента са в базата данни на приложението.

Welcome to Chat-Inc.

A simple messenger, to do you good.

Choose your username to begin chatting!

Username

Choose

Harry: Hello Mate!

Harry: Hello Bro!

Tony: Yo!

Tony: What is up!

Send

Не е необходимо да пишем какъвто и да е CSS. Въпреки това ще получим файл **app.js**, който трябва да допишем. Уеб API-то ни връща JSON обекти и нашата задача е да ги рендерираме на страницата с JavaScript.



Username Функционалност

След избора на потребителско име (натискане на бутона **[Choose]**) следва да се появи следният изглед

Welcome to Chat-Inc.
A simple messenger, to do you good.

Choose your username to begin chatting!

Username
Harry

Reset

Harry: Hello Mate!
Harry: Hello Bro!
Tony: Yo!
Tony: What is up!

Send

След като кликнем върху **[Reset]**, Потребителското име трябва да бъде нулирано и да можем да изберем друго Потребителско име.

Messaging Функционалност

След натискане на бутона **[Send]** съобщението трябва да бъде изпратено до уеб API-а и то да бъде създадено в базата данни. Всички съобщения трябва да бъдат обновени (изброени отново), така че новото съобщение да може да бъде прикачено.

Micro-Validations

Нека въведем микро-валидации както следва:

- Не трябва да можем да изпращаме съобщение, без първо да сме избрали потребителско име
- Не трябва да можем да изберем празно потребителско име
- Не трябва да можем да изпращаме празно съобщение