



Софийски университет "Св. Климент Охридски"
Факултет по Математика и Информатика

Приложно-програмни интерфейси за работа с
облачни архитектури с Амазон Уеб Услуги
(AWS)

Име на проекта
CodeSnip

Изготвили
Драгомир Ангелов 61959
Иво Яков 61891

1. Описание на проекта

CodeSnip е платформа, предоставяща възможност на потребителите да създават, редактират и споделят “парчета” текст/код. Всеки потребител може да създава и редактира парчета код след регистрация в системата. Впоследствие, потребителят може да генерира линк, към своя “Snip”, който да служи за споделяне.

CodeSnip е насочен по-скоро към съхранение на малки парчета код, подобно на gist(github), а не на цели проекти.

2. Използвани технологии

- Front-end

Front-end частта на платформата е реализирана с помощта на HTML, CSS, Javascript, както и с помощта на библиотеките JQuery, Bootstrap, HighlightJs и SweetAlert.

- Back-end

AWS Lambda е услуга, която позволява изпълнение на код безсървърно. Кодът е във вид на **Lambda функции** и не е нужно да се грижим за машината, за тяхното изпълнение и за скалиране при повече заявки. В нашето приложение те изпълняват потребителските заявки и обработват данните от базата данни за да реализират основната логика на системата. Езикът, който използваме за написването на lambda функциите е Node.js. Избрахме него заради по-бързия cold start на самите ламбда функции.

DynamoDB - Нерелационна база данни съхраняваща парчетата код и състоянията на потребителските акаунти.

Примерно представяне на базата от данни

Users	
Field	Type
username	string
email	string
first_name	string
last_name	string

password	string
----------	--------

Tokens	
Field	Type
token	string
username	string

Snippets	
Field	Type
username	string
id	string
title	string
content	string
lang	string

S3 Bucket - Играе роля на контейнер за статичния фронтенд на системата.

3 Реализация

Нашата система представлява статичен приложение, което се хоства като статичен сайт с помощта на услугата S3 на AWS. Всяка функционалност свързана с уебсайта е имплементирана чрез HTTP заявка към API Gateway, който от своя страна стартира Lambda функция. Тя използва входните данни (хедъри и тяло на заявката) за да модифицира стойностите в DynamoDB базата данни и след това връща отговор, със статус и JSON Body. По този начин имплементираме всички функционалности на системата и ги правим достъпни за уебсайта.

Най-ясно бизнес логиката може да покажем чрез схемата на крайните точки(**Endpoints**) на нашия **API**.

Адрес на CodeSnip API: <https://42tnl37ova.execute-api.eu-central-1.amazonaws.com/prod>

POST /login - при подадени username и password, аутентикира потребителя.

POST /register - при подадени необходими данни за потребителски акаунт, създава такъв.

GET /snippets - при подадени параметри username & id взима snippet-a с това id

GET /snippets - при подаден параметър username всички snippet-и на този user

POST /snippets - при подадени данни за snippet, създава нов запис.

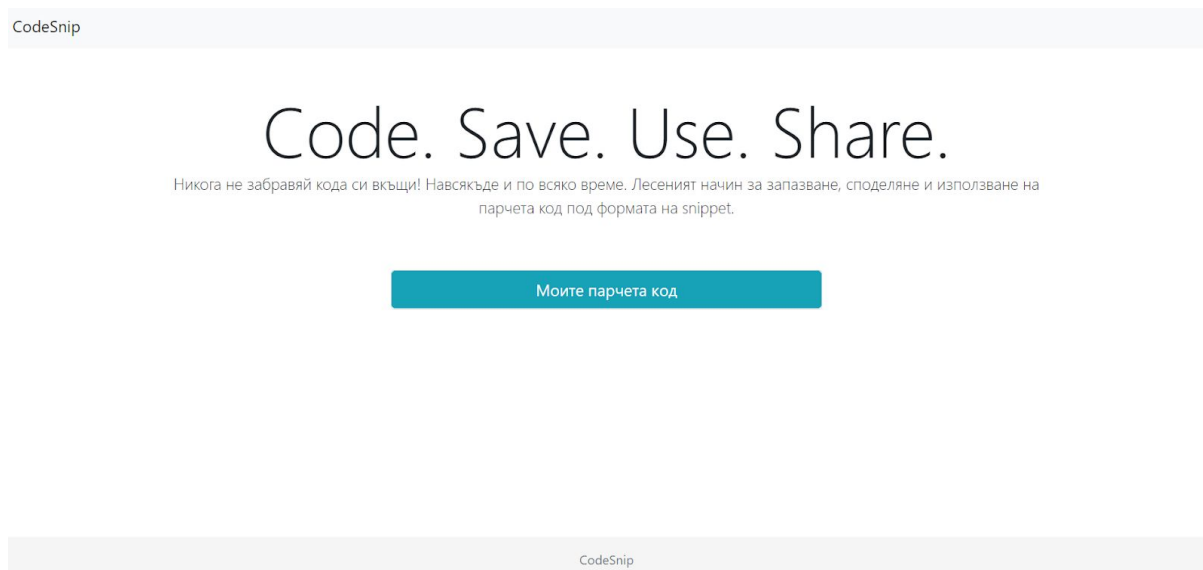
DELETE /snippets - при подаден username, token и id на snippet-a, изтрива дадения запис.

AWS Service - endpoints				
Route	Method	Headers	Parameters	Response
/login	POST	Content-type: application/json	body: { "username": <username>, "password": <password>, }	200 OK { "token": <token>, "username": <username> } 409 - Account not found, 500 - Server Error
/logout	POST	Content-type: application/json	body: { "token":<token> }	200 OK, 500 - Server Error
/register	POST	Content-type: application/json	body: { "first_name": <first_name>, "last_name": <last_name>, "email": <email>, "username": <username>, "password": <password>, "confirm_password": <password> }	200 OK { "id": <id>, "user_name": <username>, "first_name": <first_name>, "last_name": <last_name>, "email": <email> } - User Registered, 409 - Username taken, 500 - Server Error
/snippets	GET	N/D	?username=<username>&id=<snippet_id>	200 OK { "snippet" <snippet_obj>, "email" <creator_email>: } - Snippet fetch successful, 500 - Server Error
/snippets	GET	N/D	?username=<username>	200 OK { "snippets":<array_of_snippet_obj> }, 500 - Server Error
/snippets	POST	Content-type: application/json	body: { }	200 OK { "snippet" <snippet_obj> }

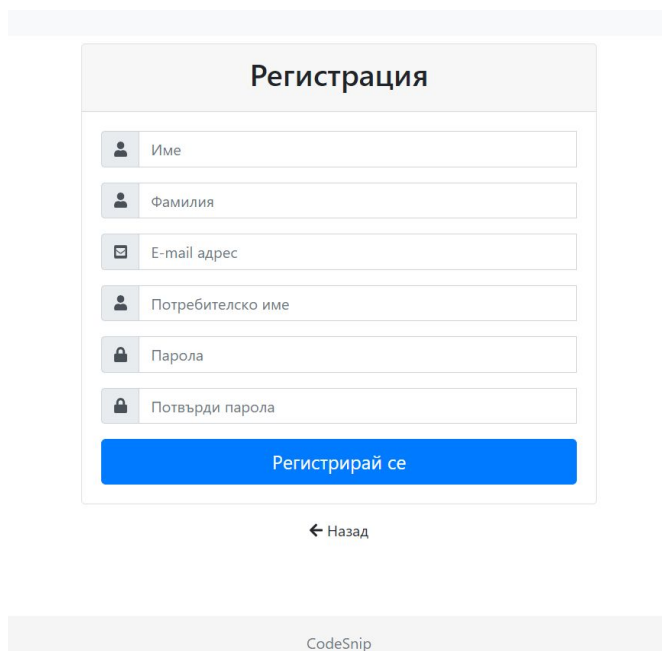
				} 400 - Not Authorised, 500 - Server Error
/snippets	DELETE	Content-type: application/json	body: { }	200 OK 400 - Not Authorised, 500 - Server Error

4 UI Screenshots

4.0 Начало



4.1 Регистрация



4.2 Вход

[Вход](#)


[Регистрация](#)

4.3 Dashboard


CodeSnip [Начало](#) [Моите парчета код](#) Здравей, Dragomir [Изход](#)

Моите парчета код [Добави парче код](#)


adadw
dada

Сподели: code-snip-bucket.s3-web 

new snippet
javascript

Сподели: code-snip-bucket.s3-web 

123
123

Сподели: code-snip-bucket.s3-web 

Task Manager

4.3.1 Добавяне на Snippet

Моите парчета код [Добави парче код](#)

Заглавие

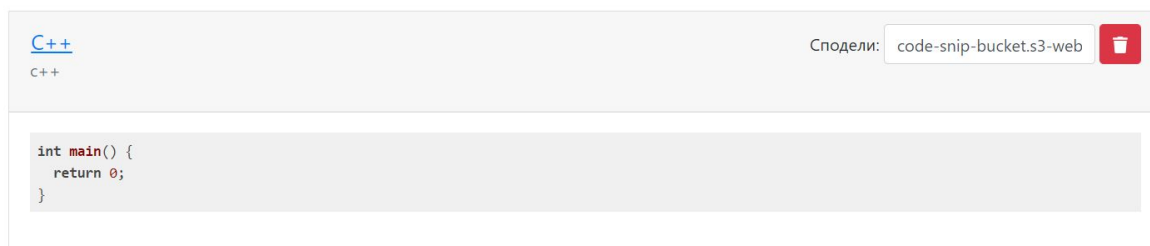
Език

Код

Поставете Вашето парче код тук

Запази

4.3 Преглед и премахване на Snippet



4.4 Преглед на публично споделен Snippet (от нерегистриран потребител)

CodeSnip



CodeSnip

4.5 Нотификации

