

SendMail Service

SendMail wants to bring its newest product to the market: Email Marketing Web service. The prototype will be able send an email to the registered recipient.

Workflow instructions

1. create a project
2. implement a code
3. zip the project
4. send it to tom@dough.com

Your Task

1. create an end point to register a user
 - name
 - email address: used as username and should be valid and unique
 - password
 - email verified
2. create an end point to get access token using username and password
 - Once correct username and password is provided, the end point returns JWT token for authorization
3. create end point for recipients
 - register recipients
 - name
 - email: should be unique and valid email address
 - retrieve a list of recipients with filters
 - by name: exact match, contains input
 - by email: exact match, contains input
 - delete a recipient
 - update a recipient
4. create an end point for send emails to the user
 - you can send an email to multiple recipients with CC or BCC option
 - the content will be identical for every recipient except greetings of user
 - ' Dear {recipient name}
 - {content body}
 - ,
5. create an end point to get list of emails sent
 - pagination
 - filter by
 - title: exact, contains
 - sent date range
 - * the response should contains
 - the number of emails
 - and the list of emails
 - each email list shows
 - email id
 - email title
 - email date
 - total number of recipients
6. create an end point to see the details of email,
 - email title
 - email date

- total number of recipients
- recipients lists

Requirements

1. use Python/Java to write a back-end API. Choice of framework is yours, but Python with Django might be a good candidate for something simple like this.
2. How to store the user and recipient information in the backend persistently is up to you. A simple solution is preferred.
3. Validation function is required
4. add automated tests where you deem necessary, document steps how to run those tests
5. document steps how to run your application in a local environment
6. the time spent on the task is not relevant, but the quality of the result is.
7. If you are unfamiliar with the tools and need time to learn them to some degree then please spend the time. (An experienced developer would probably spend between 4 to 8 hours to finish the task in good quality, but this is just a guideline and NOT a requirement)
8. Explicit is better than implicit. Simple is better than complex.