Mailing Service

Path: com.opus.backend.features.MailingService.class

This class provides a utility method for sending email using the SendGrid API.

1. **public class MailingService**: Defines the **MailingService** class.
2. **public static void SendMessage(String key, String smtp\_from, String subject, String body, String mailTo) throws IOException**: This static method sends an email using the SendGrid API. It takes the following parameters:
   * **key**: The SendGrid API key.
   * **smtp\_from**: The email address from which the message should be sent.
   * **subject**: The subject line of the email.
   * **body**: The content of the email.
   * **mailTo**: The email address to which the message should be sent.

Here's a breakdown of the method:

1. **Email from = new Email(smtp\_from)**: Creates a **from** **Email** object using the provided **smtp\_from** email address.
2. **Email to = new Email(mailTo)**: Creates a **to** **Email** object using the provided **mailTo** email address.
3. **Content content = new Content("text/plain", body)**: Creates a **Content** object with the mime type "text/plain" and the provided email **body**.
4. **Mail mail = new Mail(from, subject, to, content)**: Creates a **Mail** object with the **from**, **subject**, **to**, and **content** objects.
5. **SendGrid sg = new SendGrid(key)**: Initializes a **SendGrid** object with the provided API **key**.
6. **Request request = new Request()**: Creates a new **Request** object for the SendGrid API call.

The following steps handle the request building and the API call:

1. **request.setMethod(Method.POST)**: Sets the HTTP request method to POST.
2. **request.setEndpoint("mail/send")**: Sets the API endpoint to "mail/send".
3. **request.setBody(mail.build())**: Sets the request body to the JSON representation of the **mail** object.
4. **Response response = sg.api(request)**: Sends the request to the SendGrid API and stores the response.

Finally, the method prints the response status code, body, and headers:

1. **System.out.println(response.getStatusCode())**: Prints the response status code.
2. **System.out.println(response.getBody())**: Prints the response body.
3. **System.out.println(response.getHeaders())**: Prints the response headers.

The method also handles any **IOException** that may occur during the API call by re-throwing the exception.