ForgotPassword

Overview

This API initiates the password reset process. The user provides their email. If the email exists in the system, the system will:

Generate a secure, unique, 1-hour password reset token.

Save this token to the database (with type "PasswordReset").

Send an email to the user, containing a link (URL) to the frontend page, with this token attached.

API Specification

API	URL
POST	/api/auth/forgetPassword
Permission	N/A

Request sample

```
{
  "email": "user@example.com"
}
```

Field	Description	Data Type	Examples
email	Email address of user who forgot password.	string	user@example.com

Response sample

```
{
  "statusCode": 200,
  "message": "Password reset link has been sent to your email.",
  "data": null,
  "responsedAt": "2025-10-29T15:30:00.123Z"
}
```

Validation
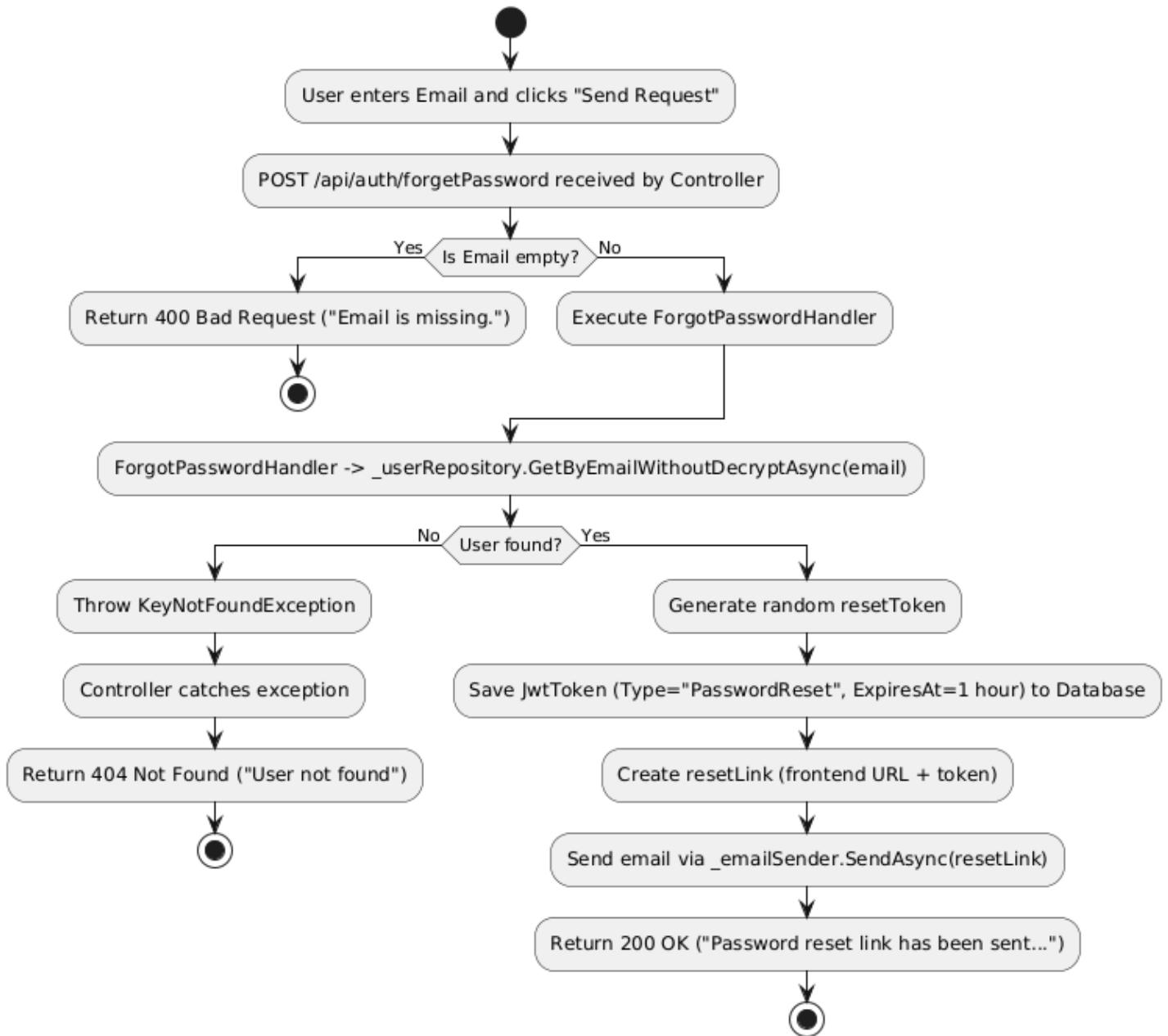
Status code	Description	Examples
400	Error: Email is missing or invalid.	```{ "statusCode": 400, "message": "Email is missing.", ...}```

404	Error: Email not found in system.	{ "statusCode": 404, "message": "Email not found in the system.", ...}
500	Server error (e.g. email failed to send).	{ "statusCode": 500, "message": "An error occurred while processing your request.",...}

Activity Diagram

Forgot Password - Activity Diagram

User enters Email and clicks "Send Request"

POST /api/auth/forgetPassword received by Controller

Is Email empty?

Yes → Return 400 Bad Request ("Email is missing.")

No → Execute ForgotPasswordHandler

ForgotPasswordHandler -> _userRepository.GetByEmailWithoutDecryptAsync(email)

User found?

No → Throw KeyNotFoundException

Controller catches exception

Return 404 Not Found ("User not found")

Yes → Generate random resetToken

Save JwtToken (Type="PasswordReset", ExpiresAt=1 hour) to Database

Create resetLink (frontend URL + token)

Send email via _emailSender.SendAsync(resetLink)

Return 200 OK ("Password reset link has been sent...")

Sequence Diagram

```mermaid
sequenceDiagram
    participant User
    participant AuthController
    participant IMediator
    participant ForgotPasswordHandler
    participant IUserRepository
    participant IJwtTokenRepository
    participant IEmailService
    participant Database

    User ->> AuthController: POST /forgetPassword (email)

    alt [Request invalid]
        AuthController -->> User: 400 Bad Request ("Email is missing.")
    end

    AuthController ->> IMediator: Send(ForgotPasswordCommand)
    IMediator ->> ForgotPasswordHandler: Handle(command)
    ForgotPasswordHandler ->> IUserRepository: GetByEmailWithoutDecryptAsync(email)
    IUserRepository ->> Database: SELECT * FROM Users WHERE Email = [encryptedEmail]
    Database -->> IUserRepository: User

    alt [User Not Found]
        IUserRepository -->> ForgotPasswordHandler: null
        ForgotPasswordHandler -->> IMediator: Throws KeyNotFoundException
        IMediator -->> AuthController: Throws KeyNotFoundException
        AuthController -->> User: 404 Not Found ("Email not found...")
    end

    IUserRepository -->> ForgotPasswordHandler: User
    ForgotPasswordHandler ->> ForgotPasswordHandler: GenerateResetToken()
    ForgotPasswordHandler ->> IJwtTokenRepository: SaveTokenAsync(PasswordResetToken)
    IJwtTokenRepository ->> Database: INSERT INTO JwtTokens...
    Database -->> IJwtTokenRepository: Success
    ForgotPasswordHandler ->> IEmailService: SendAsync(email, resetLink)
    IEmailService -->> ForgotPasswordHandler: Task
    ForgotPasswordHandler -->> IMediator: Unit.Value
    IMediator -->> AuthController: Unit.Value
    AuthController -->> User: 200 OK ("Password reset link sent...")
```