ResetPassword

Overview

This API completes the password reset process. The user provides a token (received from email) and a newPassword (new password).

The system will:

1. Validate the token (exists? is the correct type? is still valid? hashed?).
2. Validate the newPassword (minimum length).
3. If valid, the system will hash the new password.
4. Update the new password for the user in the database.
5. Revoke the used token so it cannot be reused.

API Specification

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

Request sample


```
{
  "token": "a1b2c3d4e5f6...",
  "newPassword": "MyNewSecurePassword123"
}
```


Field	Description	Data Type	Examples
token	Token code received from email.	string	a1b2c3d4e5f6...
newPassword	New password.	string	MyNewSecurePassword123

Response sample


```
{
  "statusCode": 200,
  "message": "Password has been reset successfully.",
  "data": null,
  "responsedAt": "2025-10-29T15:35:00.456Z"
}
```


Validation

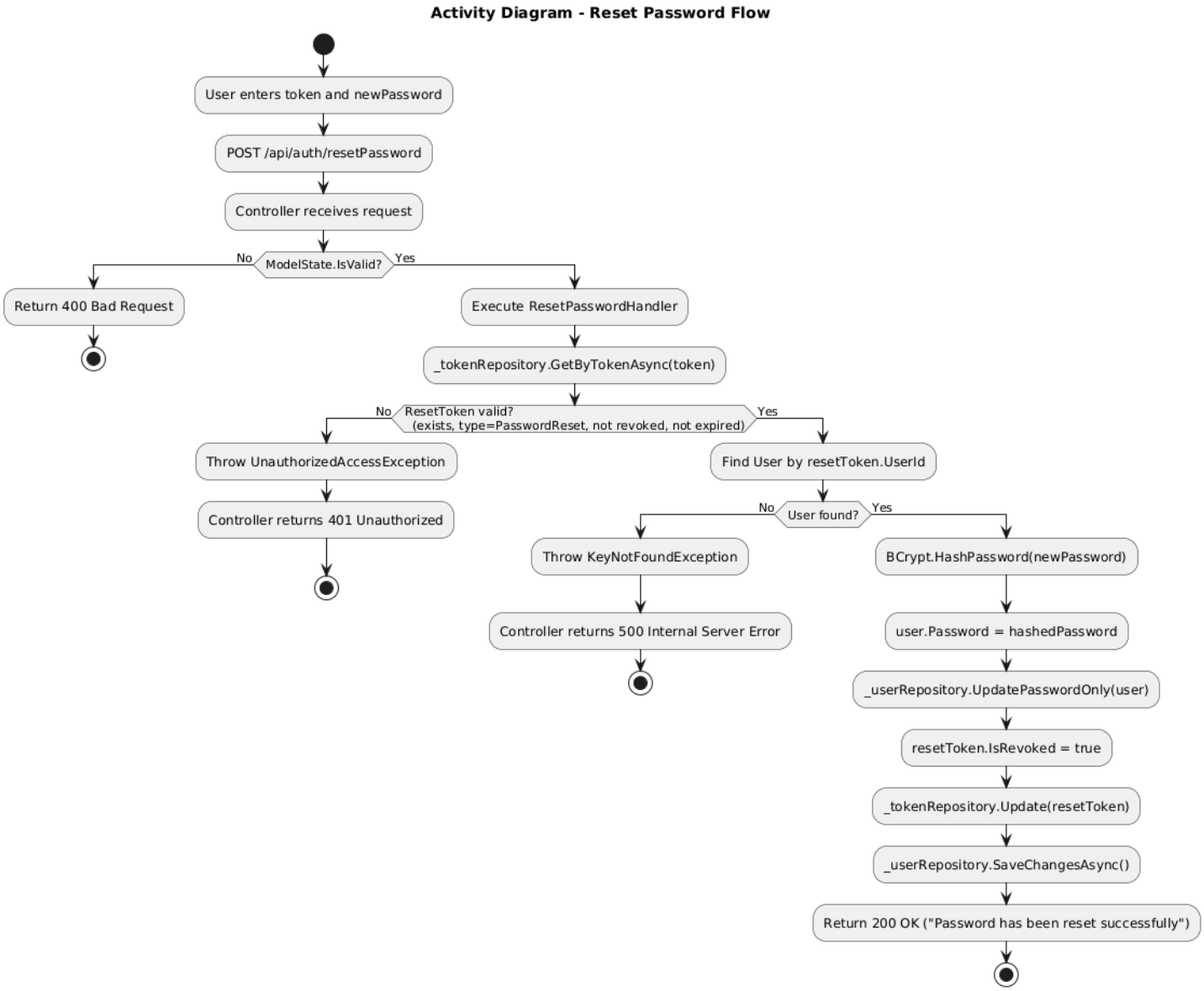
Status code	Description	Examples

400	Error: Invalid data (from [Required], [StringLength]).	{ "statusCode": 400, "message": "Invalid request data.", ...}
401	Error: Invalid token (not found, wrong type).	{ "statusCode": 401, "message": "Invalid reset token.", ...}

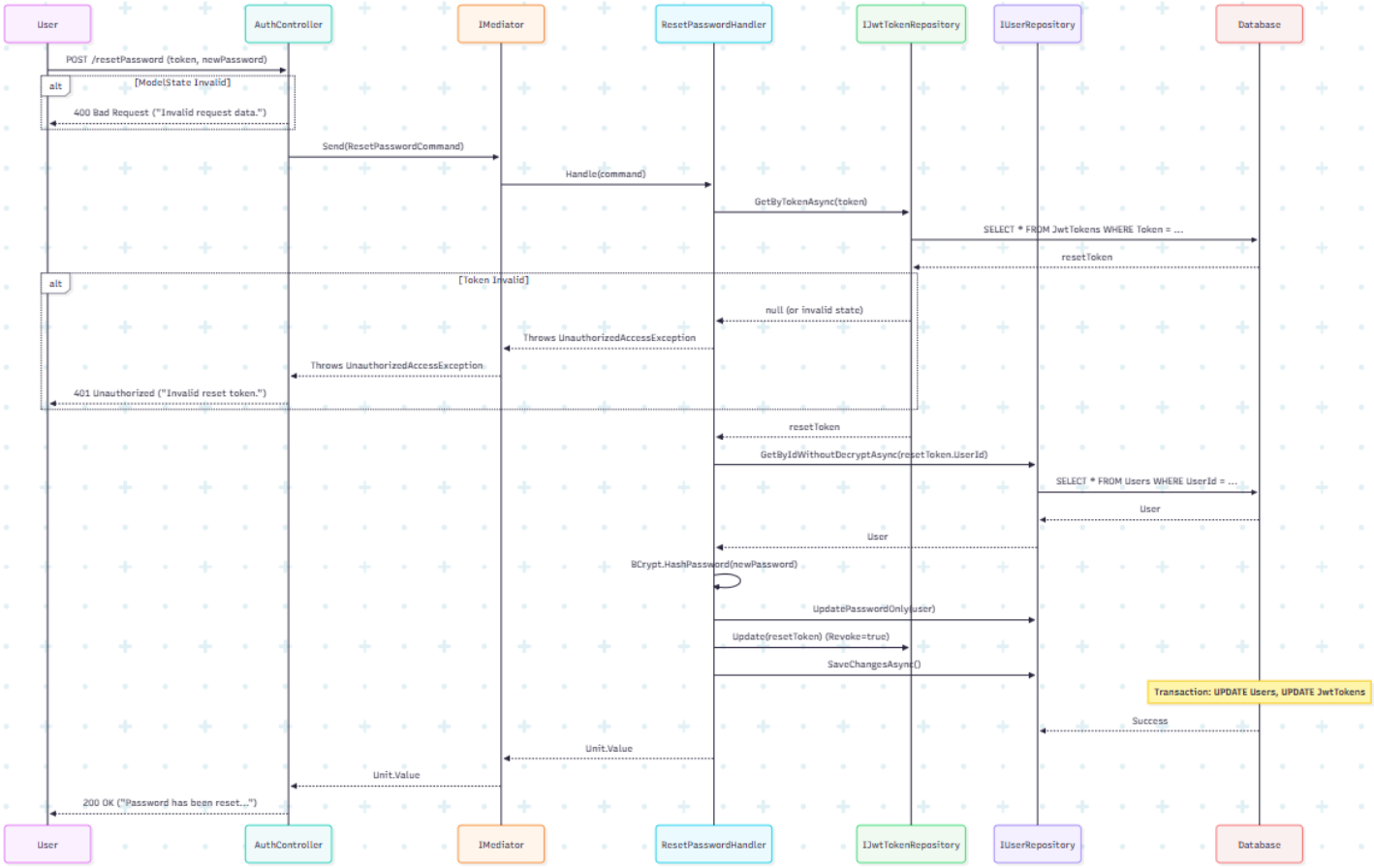
401	Error: Token has been revoked.	{ "statusCode": 401, "message": "Reset token has been revoked.", ...}
401	Error: Token has expired..	{ "statusCode": 401, "message": "Reset token has expired.", ...}

500	Server error (e.g., user associated with token not found).	{ statusCode": 500, "message": "An error occurred...", ...}

Activity Diagram



Sequence Diagram


User
AuthController
IMediator
ResetPasswordHandler
IJwtTokenRepository
IUserRepository
Database
POST /resetPassword (token, newPassword)
alt
[ModelState Invalid]
400 Bad Request ("Invalid request data.")
Send(ResetPasswordCommand)
Handle(command)
GetByTokenAsync(token)
SELECT * FROM JwtTokens WHERE Token = ...
resetToken
alt
[Token Invalid]
null (or invalid state)
Throws UnauthorizedAccessException
Throws UnauthorizedAccessException
401 Unauthorized ("Invalid reset token.")
resetToken
GetByIdWithoutDecryptAsync(resetToken.UserId)
SELECT * FROM Users WHERE UserId = ...
User
User
BCrypt.HashPassword(newPassword)
UpdatePasswordOnly(user)
Update(resetToken) (Revoke=true)
SaveChangesAsync()
Transaction: UPDATE Users, UPDATE JwtTokens
Success
Unit.Value
Unit.Value
200 OK ("Password has been reset...")
User
AuthController
IMediator
ResetPasswordHandler
IJwtTokenRepository
IUserRepository
Database