

# Password Reset

Users can request a password reset link via email. The link expires after 1 hour and is single-use. Old links are invalidated when a new one is requested.

File: with\_custom\_blocks.feature  
Generated: May 23, 2026  
Scenarios: 6

SCENARIO 1 OF 6

## Request reset for known email sends link

Given	a user exists with email <code>&lt;code&gt;&amp;lt;code&amp;gt;&lt;/code&gt;"bob@example.com"&lt;code&gt;&amp;lt;/code&amp;gt;&lt;/code&gt;</code>
When	a password reset is requested for <code>&lt;code&gt;&amp;lt;code&amp;gt;&lt;/code&gt;"bob@example.com"&lt;code&gt;&amp;lt;/code&amp;gt;&lt;/code&gt;</code>
Then	the HTTP status is 200
And	a reset email is sent to <code>&lt;code&gt;&amp;lt;code&amp;gt;&lt;/code&gt;"bob@example.com"&lt;code&gt;&amp;lt;/code&amp;gt;&lt;/code&gt;</code>
And	the reset token expires in 3600 seconds

SCENARIO 2 OF 6

## Request reset for unknown email returns 200 silently

Given	no user exists with email <code>&lt;code&gt;&amp;lt;code&amp;gt;&lt;/code&gt;"ghost@example.com"&lt;code&gt;&amp;lt;/code&amp;gt;&lt;/code&gt;</code>
When	a password reset is requested for <code>&lt;code&gt;&amp;lt;code&amp;gt;&lt;/code&gt;"ghost@example.com"&lt;code&gt;&amp;lt;/code&amp;gt;&lt;/code&gt;</code>
Then	the HTTP status is 200
And	no email is sent

## SCENARIO 3 OF 6

## Using a valid token sets the new password

- Given** a valid reset token exists for `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>`
- When** the reset endpoint is called with the token and new password `<code>&lt;code&gt;</code>"newpassword99"<code>&lt;/code&gt;</code>`
- Then** the HTTP status is 200
- And** the user can log in with `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>` and `<code>&lt;code&gt;</code>"newpassword99"<code>&lt;/code&gt;</code>`
- And** the reset token is invalidated

## SCENARIO 4 OF 6

## Expired token is rejected

- Given** an expired reset token exists for `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>`
- When** the reset endpoint is called with the expired token and new password `<code>&lt;code&gt;</code>"xyz"<code>&lt;/code&gt;</code>`
- Then** the HTTP status is 400
- And** the response contains error `<code>&lt;code&gt;</code>"token_expired"<code>&lt;/code&gt;</code>`

## SCENARIO 5 OF 6

## Requesting a second reset invalidates the first token

- Given** a valid reset token exists for `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>`
- When** a password reset is requested again for `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>`
- And** the original token is used
- Then** the HTTP status is 400
- And** the response contains error `<code>&lt;code&gt;</code>"invalid_token"<code>&lt;/code&gt;</code>`

## Password strength validation on reset

- Given** a valid reset token exists for `<code>&lt;code&gt;</code>"bob@example.com"<code>&lt;/code&gt;</code>`
- When** the reset endpoint is called with the token and new password `<code>&lt;code&gt;</code>"<code>&lt;password&gt;</code>"<code>&lt;/code&gt;</code>`
- Then** the HTTP status is `<code>&lt;status&gt;</code>`
- And** the response contains `<code>&lt;code&gt;</code>"<code>&lt;message&gt;</code>"<code>&lt;/code&gt;</code>`

### EXAMPLES

password	status	message
short	400	password_too_short
validpass1	200	ok
12345678	400	password_too_simple