

## 36. Resolver user.ts / Mutation - changePassword()

#backend #graphql #resolver #authentication #mutation #mikroorm

### Add changePassword mutation

- Update the **user** resolver, adding the **changePassword()** mutation, as shown below:
- We will also log the users in when they change their password
- **token** is sent to the user in the email in **forgotPassword mutation**, as the **query** section of the **change password URL**:  
localhost:3000/change-password/05329837b4-deuh6-rben34-293874yt
- **token** was also saved in to the Redis store in the **forgotPassword mutation**
- this **token** will be retrieved by the front-end (**change-password page**) and sent to the backend (**changePassword mutation**), where we will use it to authenticate the password change attempt

/resolvers/user.ts

```
@Mutation(() => UserResponse)
async changePassword(
  @Arg("token") token: string,
  @Arg("newPassword") newPassword: string,
  @Ctx() { em, redis, req }: MyContext
): Promise<UserResponse> {
  if (newPassword.length <= 2) {
    return {
      errors: [
        {
          field: "newPassword", // must match the name of the field on front-
end
          message: "Length must be greater than 3",
        },
      ],
    };
  }

  const tokenKey = FORGOT_PASSWORD_PREFIX + token;
  const userId = await redis.get(tokenKey); // retrieve value for token from
redis
  if (!userId) {
    return {
```

```
      errors: [
        {
          field: "token",
          message: "Token expired",
        },
      ],
    };
  }
}
```

```
const user = await em.findOne(User, { id: parseInt(userId) });
```

```
if (!user) {
  return {
    errors: [
      {
        field: "token",
        message: "User no longer exists",
      },
    ],
  };
}
```

```
user.password = await argon2.hash(newPassword); // hash and set the pw in
user
```

```
await em.persistAndFlush(user); // change pw in db
await redis.del(tokenKey); // delete token so it can't be reused
req.session.userId = user.id; // log the user in
return { user };
}
```