



TDDBUDDY.COM

# Password

---

## *The Kata*

Create class to store and validate passwords.

All passwords must be salted and hashed.<sup>1</sup>

Do not interact with an Email server.

Add an **AreValidUserCredentials** method that takes in a `userName` and `password`. The method salts and hashes the password to check its validity against what is stored. If it matched it returns true, else false.

Add a **SendResetEmail** method that take in an email address. If it matches what is on record for the user send an email with a validation link. The link must include a randomly generated token that will expire 1 hour after being created.

## Examples

```
AreValidUserCredentials("userName", "password")
```

```
SendResetEmail("emailAddress")
```

## *Hint*

- Pass in a mocked repository for password validation.
  - Or if you are feeling brave and using C# write an integration test using a real repo and TDD Buddy's LocalDb test wrapper <https://www.nuget.org/packages/TddBuddy.SpeedySqlLocalDb>
- Pass in a mocked email service for sending email.
  - You are not allowed to feel luck with this, it will be painful if you try.

## *Bonus*

- 1) Modify the reset logic to extend all previous request sent within the last hour to the expiry of the most recently issued token. E.g a link was issued 50 minutes ago. A user request a new link, now both links expire in one hour.
- 2) Passwords expire every 60 days and the user many not use the any of the previous 5 passwords.
- 3) If you did not write an integration test for DB interaction, please do so now. Remove your mock and make it use a real implementation.

---

<sup>1</sup> <https://crackstation.net/hashing-security.htm>