Spring Professional Exam Tutorial v5.0

Question 08

Question 08 - Does Spring Security support password hashing? What is salting?

Yes, Spring Security supports password hashing through `PasswordEncoder` interface and has built-in support for following encoders:

- bcrypt
- pbkdf2
- scrypt
- argon2
- sha256
- ...

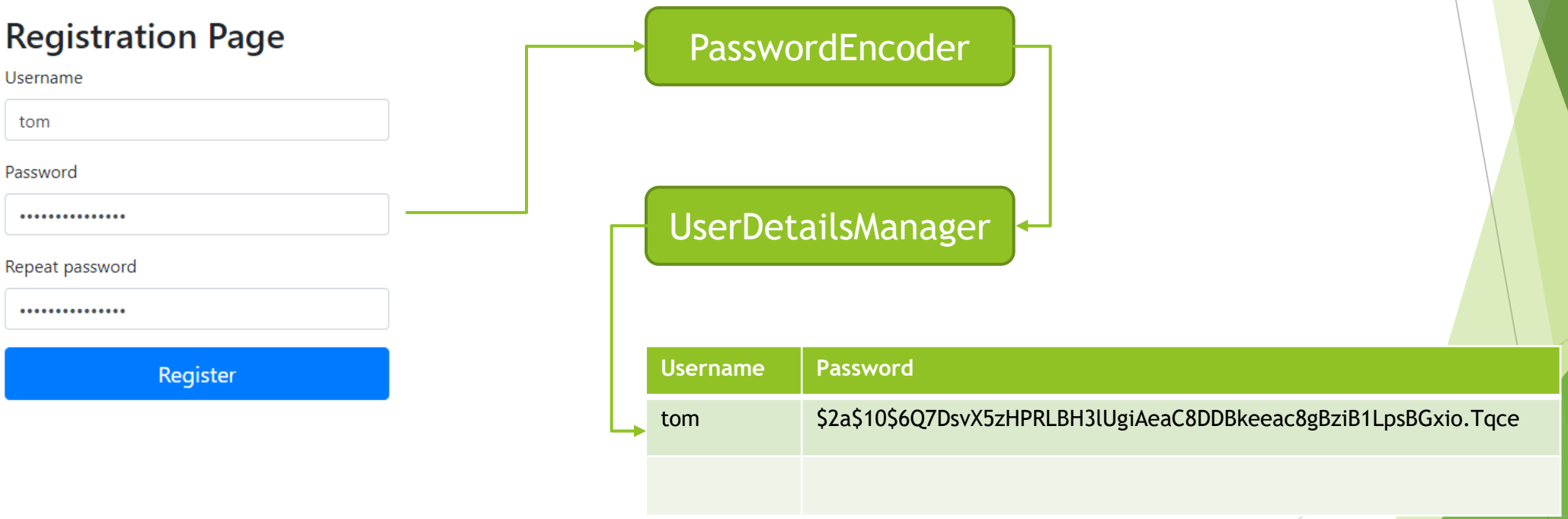
`PasswordEncoder` interface contains following methods:

- `encode` - encode the raw password
- `matches` – verifies if raw password provided as input matches encoded password, password is never decoded, one-way algorithms are used

Question 08 - Does Spring Security support password hashing? What is salting?

- Password hashing upon registration

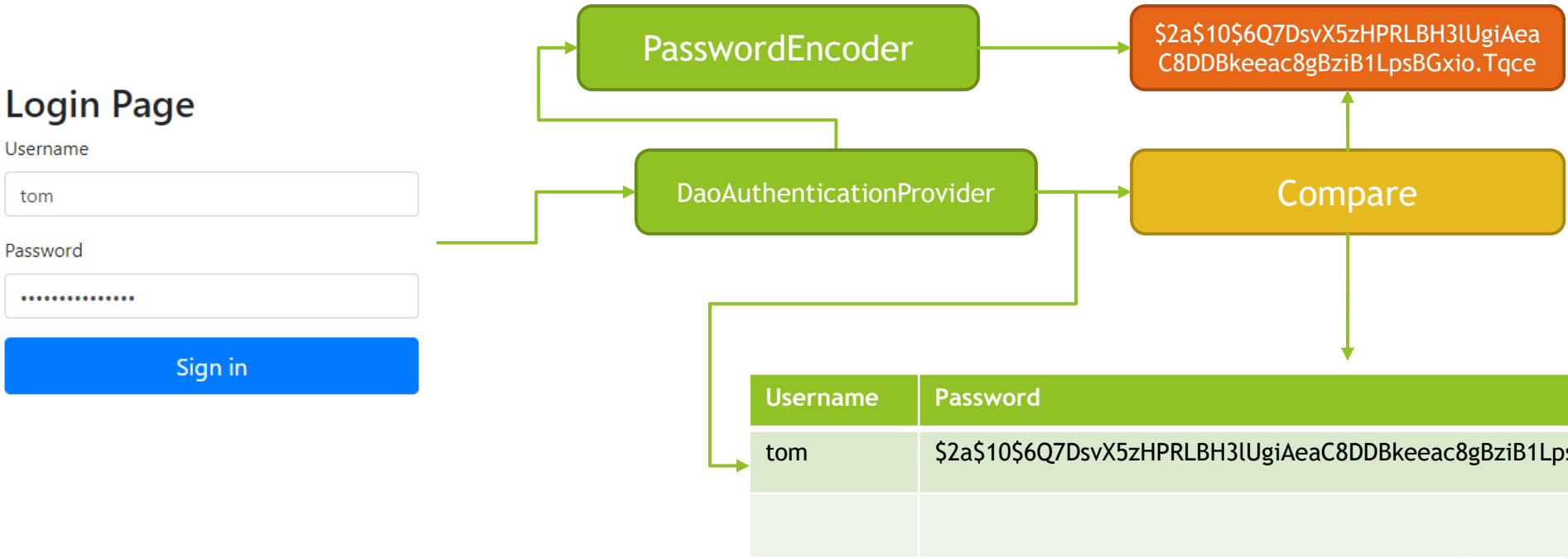
Upon registration password is encoded (hashed) and never stored in cleartext.



Question 08 - Does Spring Security support password hashing? What is salting?

- Password verification upon login

Upon login, provided password is encoded again and compared with one stored in database.



Question 08 - Does Spring Security support password hashing? What is salting?

Spring Security also provides `DelegatingPasswordEncoder`, which uses one of the selected `PasswordEncoder` to encode password, and list of provided passwords decoders to verify password upon login.

`DelegatingPasswordEncoder` is useful as it provides flexibility and ability to easily switch between `PasswordEncoders` while keeping backward compatibility, for already stored hash values of passwords.

`DelegatingPasswordEncoder` stores hash values for password as calculated by selected `PasswordEncoder` with identifier stored as prefix, for example:

```
{bcrypt}$2a$10$dXJ3SW6G7P50lGmMkkmwe.20cQQubK3.HZWzG3YB1tlRy.fqvM/BG
```

If storage contains other algorithms used as well, for example:

```
{bcrypt}$2a$10$dXJ3SW6G7P50lGmMkkmwe.20cQQubK3.HZWzG3YB1tlRy.fqvM/BG
{pbkdf2}5d923b44a6d129f3ddf3e3c8d29412723dcbde72445e8ef6bf3b508fbf17fa4ed4d6b99ca763d8dc
{sha256}97cde38028ad898ebc02e690819fa220e88c62e0699403e94fff291cfffaf8410849f27605abcbc0
```

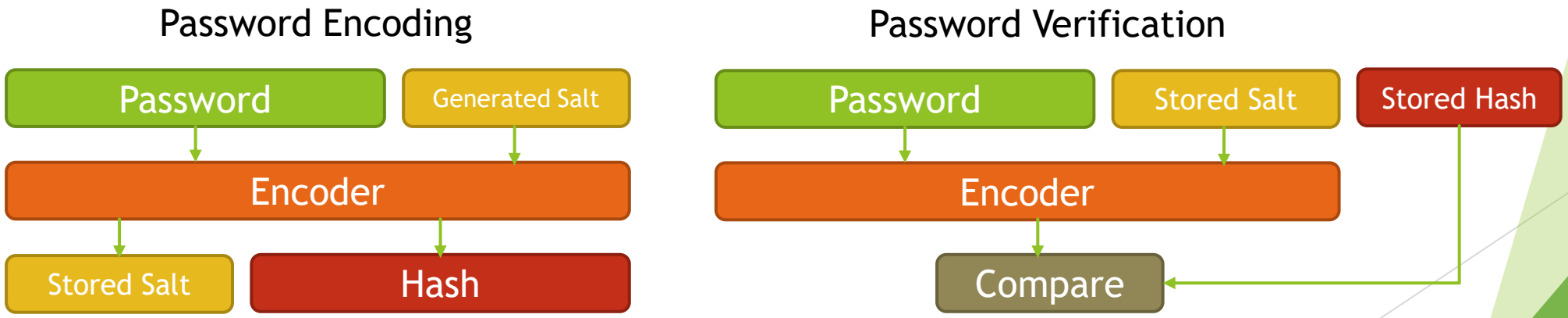
prefix is used to delegate password verification to correct `PasswordEncoder`.

Question 08 - Does Spring Security support password hashing? What is salting?

Password salting is a security mechanism invented to protect against reversing cryptographic hash functions, with usage of a precomputed tables like Rainbow Tables.

Password Salting assumes that whenever hash for password is computed, a sequence of random bytes, known as salt is added to cleartext password before calculating hash value. This way database will not contain same hash values for the same passwords.

When password is verified, salt that is stored in clear text alongside hash value for password is used again to verify if provided password matches hash value.



Question 08 - Does Spring Security support password hashing? What is salting?

As an example, let's look at how BCrypt will handle process of password encoding and verification:

- Password '`secretpassword`' is encoded and stored in database as following:

```
$2a$10$4Hw.ix095n8Hs3pPf6E5UOfJk/ym9R0WY6u58OIt9pzRhZPV3F1DS
```

- Password '`secretpassword`' is verified
  - Raw password '`secretpassword`' is being sent for comparison
  - Stored password is retrieved as - `$2a$10$4Hw.ix095n8Hs3pPf6E5UOfJk/ym9R0WY6u58OIt9pzRhZPV3F1DS`
  - Algorithm identified is checked - `$2a$`
  - Number of rounds is retrieved - `10`
  - Salt is retrieved - `4Hw.ix095n8Hs3pPf6E5UO`
  - Hash for provided password is computed
    - Hash('secretpassword', '4Hw.ix095n8Hs3pPf6E5UO', 10)
  - Newly generated hash is compared with stored hash