

Features

Table of contents

| | |
|---|---|
| 1 Features in version 0.6-SNAPSHOT..... | 2 |
| 2 Next Release..... | 2 |
| 3 Future features..... | 2 |

1. Features in version 0.6-SNAPSHOT

- Support for any database model for managing users and groups.
- Storage of passwords as plain text or using any message digest as supported by Java's `MessageDigest.getInstance(String)` can be used.
- Configuration of the digest algorithm. Any digest algorithm supported by `MessageDigest.getEncoding(String)`. Also, it is possible to avoid use of a digest algorithm altogether. Also the character set used to convert the password to its byte array representation as input to the digest algorithm can be configured. The default character set is that returned by `Charset.defaultCharset().name()`.
- Optional configuration of the encoding of the password digest. Hexadecimal and base64 encoding are supported. In addition, text encoding is supported whereby the byte array as output from the digest algorithm is converted back to a string using the character set. The default is hexadecimal encoding.
- Support for a seed in the encoding where the seed for a specific user is retrieved from the database. When a seed is used the password is encoded together with the user-specific seed to prevent dictionary attacks. The way the seed is used together with the password is configured using a format string.
- When Hex encoding is used, the hex output is always padded with leading zeroes. By default 32 characters are used but this can be configured to an arbitrary number.
- Logging: Use FINEST level logging to see details.

2. Next Release

-

3. Future features

Configuration of the caching of groups. At the moment no caching is done.