



# Spring Security

---

Beginner to Guru

Google Authenticator with Spring Security



## Google Authenticator with Spring Security

- Goal - Configure Spring Security to use Google Authenticator for 2FA
- Google Authenticator Registration:
  - Scan QR Code to register application
- Provide User Ability to opt-in to 2FA
- Users which have opted into 2FA required to enter 2FA code after login
- Users not opted in, do not need to enter code



## Spring Security Configuration

- Update User Entity to hold:
  - Use 2Fa? - Is user registered for 2FA
  - 2FA Secret - Shared Secret with Google Authenticator
  - 2FA Required? - Transient Property to require entering of 2FA code
- Spring Security 2FA Filter
  - On Login, if User is 2FA enabled, forward to 2FA Authentication URL





## 2FA Registration

- Add Menu Option and page to Enable 2FA
- Controller will update User with 2FA secret and show QR code
- Accept Form Post of TOTP code from Google Authenticator
  - If valid, enable 2FA for user
  - Redirect to index



## 2FA Login

- Accept username password as normal
- If user is 2FA enabled forward to 2FA verification page
- Use Spring Security Filter for forward
  - Filter will restrict user to verification page until proper code is entered



## Other

- Example intentionally for Spring MVC, will work for most typical web applications
- Not following TDD - well covered, typical unit testing
- Not covering user registration - little Spring Security needed
  - Traditional Spring MVC application
- Single Page Javascript could be adopted to this example
  - ie Angular, ReactJS, etc





