Web Security Outline

- WebSecurityConfiguration
- WebSecurity – I
- WebSecurity – II
- WebSecurity – III
- WebSecurityConfigurerAdapter
- WebSecurityConfigurer
- SecurityConfigurer

WebSecurityConfiguration

- Uses a WebSecurity to create the FilterChainProxy that performs the **web-based** security for Spring Security.
- **Customizations** can be made to WebSecurity
  - by implementing WebSecurityConfigurer and exposing it as a Configuration or
  - exposing a WebSecurityCustomizer bean.
- This configuration is imported when using EnableWebSecurity.

```
Creates the Spring Security Filter Chain

Returns: the Filter that represents the security filter chain

Throws: Exception

@Bean(name = AbstractSecurityWebApplicationInitializer.DEFAULT_FILTER_NAME)
public Filter springSecurityFilterChain() throws Exception {
    boolean hasFilterChain = !this.securityFilterChains.isEmpty();
    if (!hasFilterChain) {
        this.webSecurity.addSecurityFilterChainBuilder(() -> {
            this.httpSecurity.authorizeHttpRequests((authorize) -> authorize.anyRequest().authenticated());
            this.httpSecurity.formLogin(Customizer.withDefaults());
            this.httpSecurity.httpBasic(Customizer.withDefaults());
            return this.httpSecurity.build();
        });
    }
    for (SecurityFilterChain securityFilterChain : this.securityFilterChains) {
        this.webSecurity.addSecurityFilterChainBuilder(() -> securityFilterChain);
    }
    for (WebSecurityCustomizer customizer : this.webSecurityCustomizers) {
        customizer.customize(this.webSecurity);
    }
    return this.webSecurity.build();
}
```

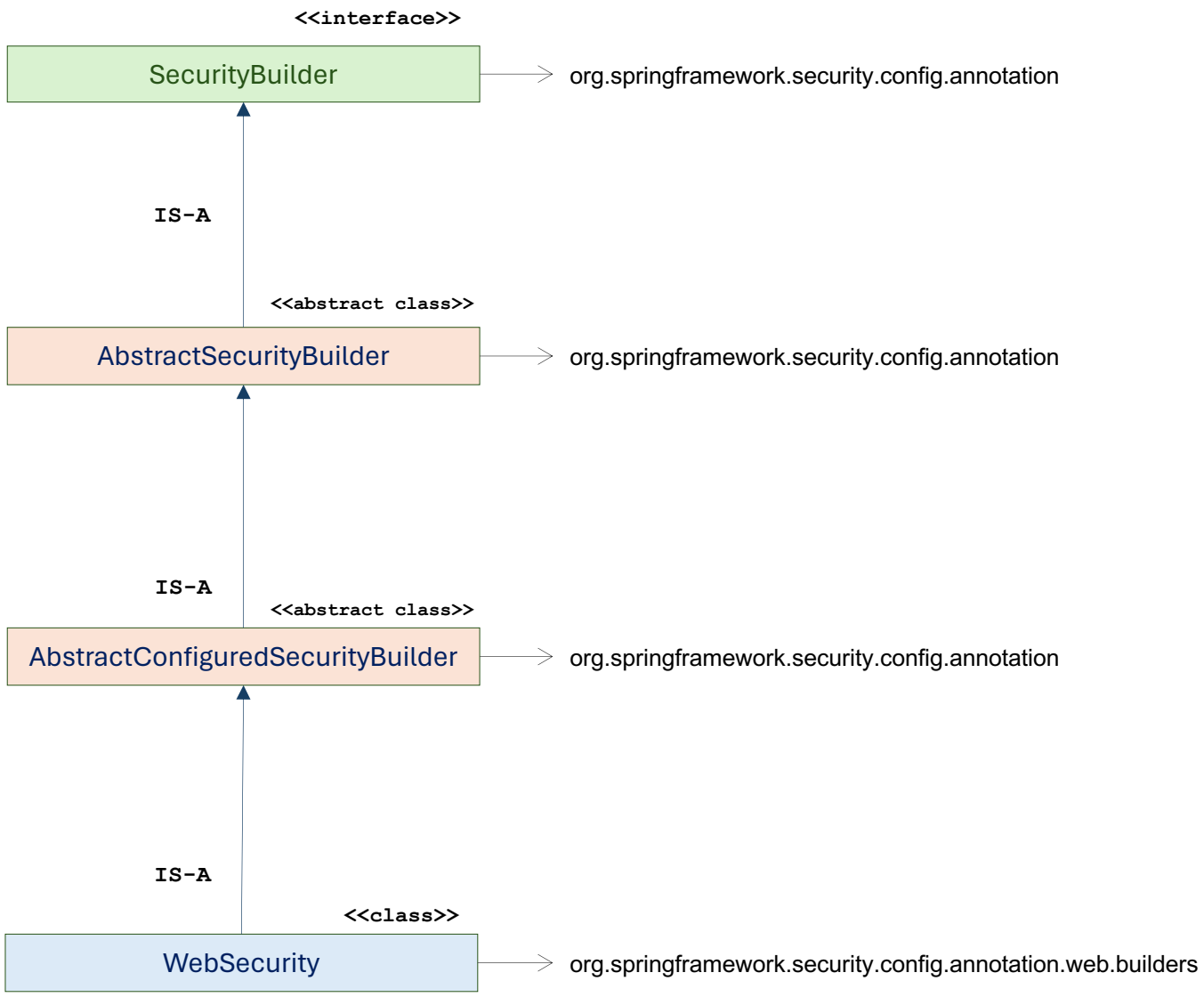
```
package org.springframework.security.config.annotation.web.configuration;

@Configuration(
    proxyBeanMethods = false
)
public interface WebSecurityConfiguration implements ImportAware, BeanClassLoaderAware {
    . . . // intentionally skipped

    private WebSecurity webSecurity;
    private HttpSecurity httpSecurity;

    private List<SecurityFilterChain> securityFilterChains = Collections.emptyList();

    . . . // intentionally skipped
}
```

WebSecurity - I

- The WebSecurity is created by WebSecurityConfiguration to create the FilterChainProxy known as the **Spring Security Filter Chain** (**springSecurityFilterChain**).
  - The **springSecurityFilterChain** is the Filter that the DelegatingFilterProxy delegates to.
  - Customizations to the WebSecurity can be made by
    - ***creating*** a WebSecurityConfigurer or
    - ***exposing*** a WebSecurityCustomizer bean.

```
package org.springframework.security.config.annotation.web.builders;

public final class WebSecurity extends AbstractConfiguredSecurityBuilder<Filter, WebSecurity> implements SecurityBuilder<Filter>, ApplicationContextAware, ServletContextAware {

    . . . // intentionally skipped

    private final List<RequestMatcher> ignoredRequests;
    private final List<SecurityBuilder<? extends SecurityFilterChain>> securityFilterChainBuilders;
    private HttpFirewall httpFirewall;
    private ServletContext servletContext;

    . . . // intentionally skipped
}
```

WebSecurity - II



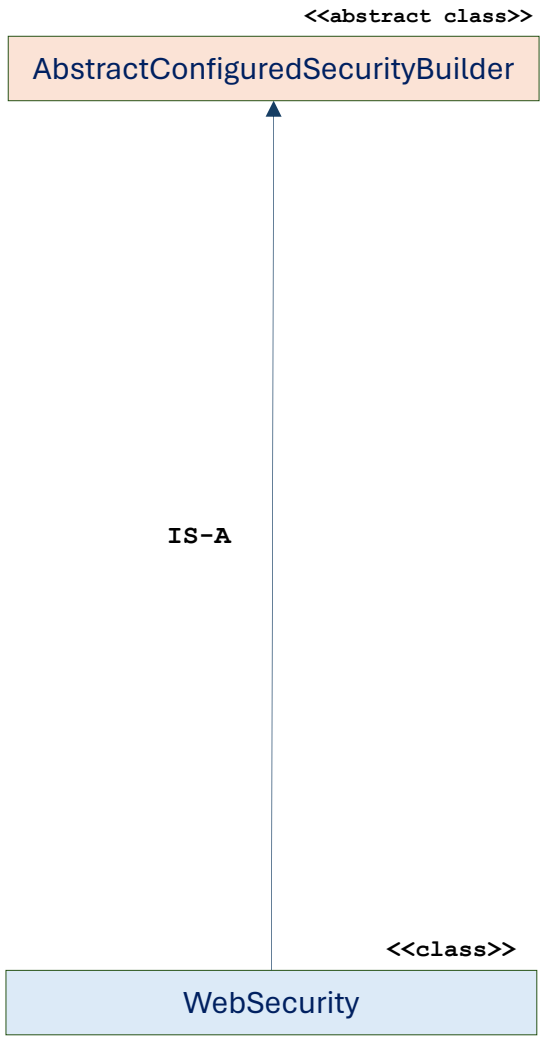

```java
package org.springframework.security.config.annotation;

public interface SecurityBuilder<O> {
    O build() throws Exception;
}
```

- Builds an Object.

```java
package org.springframework.security.config.annotation;

public abstract class AbstractSecurityBuilder<O> implements SecurityBuilder<O> {
    private AtomicBoolean building = new AtomicBoolean();
    private O object;

    @Override
    public final O build() throws Exception {
        if (this.building.compareAndSet(false, true)) {
            this.object = doBuild();
            return this.object;
        }
        throw new AlreadyBuiltException("This object has already been built");
    }

    public final O getObject() {
        if (!this.building.get()) {
            throw new IllegalStateException("This object has not been built");
        }
        return this.object;
    }

    protected abstract O doBuild() throws Exception;
}
```

- A base SecurityBuilder that ensures the object being built is only built one time.

WebSecurity - III

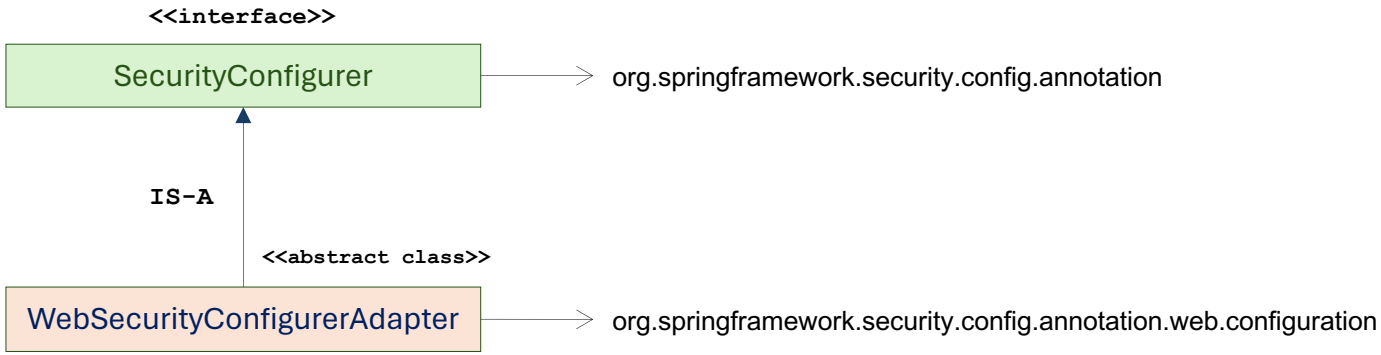



```java
package org.springframework.security.config.annotation;

public abstract class AbstractConfiguredSecurityBuilder<O, B extends SecurityBuilder<O>> extends
AbstractSecurityBuilder<O> {
    private final LinkedHashMap<Class<? extends SecurityConfigurer<O, B>>, List<SecurityConfigurer<O, B>>> configurers;
    private BuildState buildState = BuildState.UNBUILT;

    . . . // intentionally skipped

    @Override
    protected final O doBuild() throws Exception {
        synchronized (this.configurers) {
            this.buildState = BuildState.INITIALIZING;
            beforeInit();
            init();
            this.buildState = BuildState.CONFIGURING;
            beforeConfigure();
            configure();
            this.buildState = BuildState.BUILDING;
            O result = performBuild();
            this.buildState = BuildState.BUILT;
            return result;
        }
    }

    protected abstract O performBuild() throws Exception;

    . . . // intentionally skipped
}
```

- A base SecurityBuilder that allows SecurityConfigurer to be applied to it.
- This makes modifying the SecurityBuilder a strategy that can be customized and broken up into several SecurityConfigurer objects that have more specific goals than that of the SecurityBuilder.
    - For example, a SecurityBuilder may build an DelegatingFilterProxy, but a SecurityConfigurer might populate the SecurityBuilder with the filters necessary for **session management**, **form-based login**, **authorization**, etc.

WebSecurityConfigurerAdapter

- Provides a **convenient base class** for creating a WebSecurityConfigurer instance.
  - The implementation allows **customization** by overriding methods.




```
package org.springframework.security.config.annotation.web.configuration;

public abstract class WebSecurityConfigurerAdapter extends SecurityConfigurer<Filter, WebSecurity> {
    private final AuthenticationManagerBuilder authenticationBuilder;
    private final AuthenticationManagerBuilder parentAuthenticationBuilder;

    . . . // intentionally skipped

    private AuthenticationManager authenticationManager;
    private HttpSecurity http;

    . . . // intentionally skipped
}
```

- Configures the SecurityBuilder instance, *i.e.*, WebSecurity.

WebSecurityConfigurer

- Allows customization to the WebSecurity.
- In most instances users will use EnableWebSecurity and create a Configuration that exposes a SecurityFilterChain bean.
    - This will automatically be applied to the WebSecurity by the EnableWebSecurity annotation.

```
package org.springframework.security.config.annotation.web;

public interface WebSecurityConfigurer<T extends SecurityBuilder<Filter>> extends SecurityConfigurer<Filter, T> {
}
```

SecurityConfigurer

- Allows for configuring a SecurityBuilder.
    - All SecurityConfigurer first have their **init**(SecurityBuilder) method invoked.
    - After all **init**(SecurityBuilder) methods have been invoked, each **configure**(SecurityBuilder) method is invoked.

```
package org.springframework.security.config.annotation.web.configuration;

// Type parameters:
//     <O> – The object being built by the SecurityBuilder B
//     <B> – The SecurityBuilder that builds objects of type O.
//           This is also the SecurityBuilder that is being configured.
public interface SecurityConfigurer<O, B extends SecurityBuilder<O>> {

    void init(B builder) throws Exception;
    void configure(B builder) throws Exception;

}
```