Activedirectory Spring Boot Starter

Starter for configuring activedirectory authentication, optimized for service to service
communication via caching of credentials.

Version 2.3.3-SNAPSHOT

Table of Contents

Runtime Dependencies

- Java >= `1.8`
- Spring Boot >= `2.0.0.RELEASE`

The module was compiled against :

- Spring Boot Actuator = `2.3.3.RELEASE`

Installation

Using the starter

```
    <dependency>
        <groupId>de.dm.auth</groupId>
        <artifactId>activedirectory-spring-boot-starter</artifactId>
        <version>2.3.3-SNAPSHOT</version>
    </dependency>
```

The starter contains all required dependencies, including `spring-boot-starter-cache` and `ehcache`. A default `ehcache.xml` is delivered with the module, which should suffice for most use cases.

Usage

The module uses spring boot's auto-configuration mechanism to register an `AuthenticationProvider`.

The auto-configuration will also provide a spring boot health indicator for active directory connection (ldapHealthIndicator).

If you're using spring security >= 4.1.0 the `AuthenticationProvider` will be registered by default, unless you're already configuring an `AuthenticationManagerBuilder`. Consult the spring security documentation for more details.

If you're using spring security < 4.1.0 or want to control the registration, then use the following configuration as a baseline:

Adding the AuthenticationProvider

```
@Configuration
public class ActiveDirectorySampleConfiguration {

    @Autowired
    @ActiveDirectoryProvider ①
    private AuthenticationProvider provider;

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) {
        auth.authenticationProvider(provider);
    }

}
```

① `@Qualifier` annotation to easily inject the auto-configured `AuthenticationProvider`

Role Mapping

All ActiveDirectory groups will be converted to spring roles. The role name will be the group name converted to uppercase, prefixed with `ROLE_`. E.g. `sample-admins` will be mapped to `ROLE_SAMPLE-ADMINS`. Additionally, the user will be assigned the role `ADMIN` which is the default used by actuator endpoints.

To use the role in your app take the following configuration as an example:

```
@Configuration
public class RoleSampleConfiguration extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.antMatcher("/**")
                .authorizeRequests()
                .anyRequest().hasRole("SAMPLE-ADMINS"); ①
    }
}
```

① Note that when using `hasRole` the `ROLE_` prefix has to be omitted.

Configuration

No special configuration is necessary. If you're already using ehcache in your application, you'll need to configure a cache named `authCache`. Consult the sample `ehcache.xml` below for default values.

The default ehcache.xml

```
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="http://ehcache.org/ehcache.xsd"
         updateCheck="false" monitoring="autodetect"
         dynamicConfig="true">
    <cache name="authCache" timeToLiveSeconds="30" maxEntriesLocalHeap="100"/>
</ehcache>
```

Table 1. The following properties can be configured

Name	Type	Default value	Description
security.activedirectory.enabled	java.lang.Boolean	true	Enable or disable the auto configuration.
security.activedirectory.cache-enabled	java.lang.Boolean	true	Enable caching of authentication.
security.activedirectory.domain	java.lang.String	SAMPLE.INC	The AD domain that users authenticate against.
security.activedirectory.urls	java.lang.String[]	ldaps://sample01:636,ldaps://sample02:636	URLs that point to one or more ActiveDirectory instances. Multiple URLs need to be separated by commas.
security.activedirectory.connect-timeout	java.lang.String	1000	The timeout in milliseconds establishing a connection against an ActiveDirectory instance.
security.activedirectory.read-timeout	java.lang.String	5000	The timeout in milliseconds waiting for a response from an ActiveDirectory instance.

License

portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.