

Creating a Security Check

Overview

A `SecurityCheck` is an object responsible for obtaining credentials from a client and validate them.

`securityCheckDefinition`

Security checks are defined inside adapters. Any adapter can theoretically define a `SecurityCheck`. An adapter can either be a *resource* adapter (meaning it serves resources/content to send to the client), a *SecurityCheck* adapter, or **both**. However it is recommended to define the `SecurityCheck` in a separate adapter.

In your **adapter XML** file add an XML element called `securityCheckDefinition`. For example:

```
<securityCheckDefinition name="sample" class="com.ibm.mfp.sampleSecurityCheck">
  <property name="successExpirationSec" defaultValue="60"/>
  <property name="failureExpirationSec" defaultValue="60"/>
  <property name="maxAttempts" defaultValue="3"/>
</securityCheckDefinition>
```

- The `name` attribute will be the name of your `SecurityCheck`
- The `class` attribute specifies the implementation of the `SecurityCheck`
- Some `SecurityChecks` can be configured with a list of `property` elements.

SecurityCheck implementation

The class file of your `SecurityCheck` is where all of the logic happens. Your implementation should extend one of the provided base classes, below.

The parent class you choose will determine the balance between customization and simplicity.

`SecurityCheckWithUserAuthentication`

TODO

`SecurityCheckWithAttempts`

TODO

`SecurityCheckWithExternalization`

TODO

`SecurityCheck`

TODO

SecurityCheckConfiguration

Each `SecurityCheck` implementation class can use a `SecurityCheckConfiguration` that defines properties available for that `SecurityCheck`. Each base `SecurityCheck` class comes with a matching `SecurityCheckConfiguration` class. You can create your own implementation that extends one of the base `SecurityCheckConfiguration` classes and use it for your custom `SecurityCheck`.

For example, `SecurityCheckWithUserAuthentication`'s `createConfiguration` method returns an instance of `SecurityCheckWithAuthenticationConfig`.

```

public abstract class SecurityCheckWithUserAuthentication extends SecurityCheckWithAttempts {
    @Override
    public SecurityCheckConfiguration createConfiguration(Properties properties) {
        return new SecurityCheckWithAuthenticationConfig(properties);
    }
}

```

SecurityCheckWithAuthenticationConfig enables a property called rememberMeDurationSec.

```

public class SecurityCheckWithAuthenticationConfig extends SecurityCheckWithAttemptsConfig {

    public int rememberMeDurationSec;

    public SecurityCheckWithAuthenticationConfig(Properties properties) {
        super(properties);
        rememberMeDurationSec = getIntProperty("rememberMeDurationSec", properties, 0);
    }

}

```

Those properties can be configured at several levels:

adapter.xml

TODO

application xml?

TODO

console?

TODO

Built-in Security Checks

Also available are these out-of-the-box security checks:

- Application Authenticity (../application-authenticity/)
- Direct Update (../using-the-mfpf-sdk/direct-update)
- LTPA (../websphere-ltpa-based-authentication/)