Version 0.9.x

Version 0.9.1 (previous stable release)

Release date: 25th April 2016

This release adds a single new feature:

- MDA-163: add stage to detect CR characters in metadata

This adds a `CRDetectionStage` for use in detecting metadata that can trigger the SSPCPP-684 issue in the Shibboleth SP.

Version 0.9.0

Release date: 18th December 2015.

For a complete list of issues addressed in this release, see https://issues.shibboleth.net/jira/issues/?filter=10873

This is a major pre-1.0 feature release.

Highlights

Now using Spring `Resources` instead of (now deprecated) Shibboleth `Resources`.

The factory bean classes `PrivateKeyFactoryBean`, `PublicKeyFactoryBean`, `X509CertificateFactoryBean` and `X509CertificateChainFactoryBean` bundled from the `spring-extensions` package have significant API improvements. Each factory now takes a "resource" property which is a Spring `Resource` rather than a Java `File`. This allows these factories to be used with any kind of Spring resource, including `ClassPathResource`. Existing configurations will need to change to compensate for this.

Before

```
<bean class="...X509CertificateFactoryBean">
    <property name="certificateFile">
        <bean class="java.io.File">
            <constructor-arg value="..."/>
        </bean>
    </property>
</bean>
```

After

```
<bean class="...X509CertificateFactoryBean">
    <property name="resource">
        <bean class="org.springframework.core.io.FileSystemResource">
            <constructor-arg value="..."/>
        </bean>
    </property>
</bean>
```

If you were previously setting the input property of one of these factories to a string value representing the path, and relying on the Spring resource loader to convert that into a `File` object, you may need to change your configuration to explicitly create a `FileSystemResource` if that is not the default used by the Spring context type in use in your application.

Now uses the JAXP implementation supplied by the JRE, rather than a much older "endorsed" version. This will affect any configurations which depended on Xerces or Xalan specific extensions; re-endorse the implementation of your choice if this is an issue.

All provided stages now implement a new `collectionPredicate` property. This can be set to a `Predicate<Collection<Item<T>>>` which will be applied to each collection passed to the stage. If the `collectionPredicate` returns `true`, the stage is executed as normal; this is the default. If the `collectionPredicate` returns `false`, the stage is skipped. This can be used used to perform lightweight conditional operations such as forming an `EntitiesDescriptor` from a collection only if the collection contains at least two items. The `AtLeastCollectionPredicate` class has been added to address this specific use case. Conditional evaluation of a series of stages with the same `collectionPredicate` can be simplified by use of a `CompositeStage`

This release bundles a new version of the Shibboleth `spring-extensions` package, which provides a new `IdentifiableBeanPostProcessor` class. If you include an instance of this class in your Spring configuration, you can now default the "id" property on all Shibboleth components from the bean's "id" attribute, simplifying your configuration by removing the usual duplication between these values.

Before

```
<bean class="..." id="theBean">
    <property name="id" value="theBean"/>
    ...
</bean>
```

After

```
<bean class="net.shibboleth.ext.spring.config.IdentifiableBeanPostProcessor"/>

<bean class="..." id="theBean">
    ...
</bean>
```

The `ItemSerializer` interface is no longer defined over a collection of items, but now (less surprisingly) operates on a single item. A new `ItemCollectionSerializer` interface (with a `serializeCollection` method) takes its place in operating on collections of items. In addition, `ItemSerializer` and `ItemCollectionSerializer` implementations are no longer responsible for closing the `OutputStream` they write the serialized form of their input to. These changes allow reuse of serializer implementations in cases other than the current `SerializationStage`. The `SerializationStage` implementation now accepts an `ItemCollectionSerializer` rather than an `ItemSerializer`, but `DOMElementSerializer` has been changed to support both interfaces so that no changes to configurations should be required.

The `SetValidUntilStage` and `SetCacheDurationStage` duration setters are now marked using an annotation to indicate that they take non-negative duration values. If you provide an appropriate converter in your Spring configuration, this means that configurations can now use ISO duration values (e.g., "PT6H") rather than a literal number of milliseconds (e.g., "21600000"). For example:

```
<!-- This bean MUST be called "conversionService" to work properly. -->
<bean id="conversionService" class="org.springframework.context.support.ConversionServiceFactoryBean">
    <property name="converters">
        <set>
            <bean class="net.shibboleth.ext.spring.config.DurationToLongConverter" />
            <bean class="net.shibboleth.ext.spring.config.StringToIPRangeConverter" />
            <bean class="net.shibboleth.ext.spring.config.BooleanToPredicateConverter" />
            <bean class="net.shibboleth.ext.spring.config.StringBooleanToPredicateConverter" />
            <bean class="net.shibboleth.ext.spring.config.StringToResourceConverter" />
        </set>
    </property>
</bean>

<bean id="stage" class="net.shibboleth.metadata.dom.saml.SetValidUntilStage"
    p:id="stage"
    p:validityDuration="PT6H"
    init-method="initialize"
    destroy-method="destroy"/>
```

API Additions

- MDA-55: added `EntityAttributeFilteringStage` and associated matchers: `EntityCategoryMatcher`, `EntityCategorySupportMatcher`, `MultiPredicateMatcher`, `RegistrationAuthorityMatcher`. Additional support classes: `SAMLSupport`, `MDAttrSupport`.
    - `EntityAttributeFilteringStage` evaluates a list of matching rules for each entity attribute present in a SAML `EntityDescriptor`. The list of rules is logically ORed to determine (along with a whitelisting/blacklisting property) whether each attribute value is retained or filtered out.
    - Each matching rule is in the form of a `Predicate` over an `EntityAttributeContext` containing the attribute's value, `Name`, `NameFormat` and the entity's registration authority.
    - The registration authority value in the `EntityAttributeContext` is taken from a `RegistrationAuthority` object in the entity's item metadata. This would normally be extracted from the entity beforehand using the `RegistrationAuthorityPopulationStage`.
    - The `EntityCategoryMatcher` and `EntityCategorySupportMatcher` classes match a given attribute value with appropriate attribute `Name` and `NameFormat` values as defined in the entity category specification.
    - `RegistrationAuthorityMatcher` can match against a specific registrar authority, or against the absence of any authority.
    - `MultiPredicateMatcher` can be used with arbitrary `Predicate<CharSequence>` objects evaluated against the four components of the `EntityAttributeContext`. Suitable `Predicate` objects can be obtained, for example, from Guava's `Predicates.containsPattern` method. Unset component predicates are evaluated as `true`.
    - If the filtering out of an `AttributeValue` results in an empty `Attribute` container, that container is removed.
    - If the removal of an empty `Attribute` container results in an empty `EntityAttributes` container, that container is removed.
- MDA-109: added `ElementWhitespaceTrimmingStage` to trim whitespace from start and end of text contents of selected elements
- MDA-132: new property `collectionPredicate` added on all stages; new `AtLeastCollectionPredicate` class added

- MDA-139: new classes supporting the Metadata Query Protocol:
  - `ItemIdTransformStage`
  - `MDQueryMD5ItemIdTransformer`
  - `MDQuerySAML1ItemIdTransformer`
- MDA-141: New `ItemMetadataAddingStage` adds a collection of `ItemMetadata` objects to each `Item`'s item metadata
- MDA-150: added `NamespacesStrippingStage` to whitelist/blacklist multiple namespaces
- MDA-154: added `X509ValidationStage` to allow validation of X.509 certificates in XML metadata. This is supplied with a list of `Validator<X509Certificate>` instances to determine the validation performed.
  - MDA-69: `X509RSAOpenSSLBlacklistValidator` checks for RSA modulus values from blacklist set. A `blacklistResource` property is used to set a Spring `Resource` from which the blacklist set is read in OpenSSL blacklist format. The following resources are made available in the classpath for common use cases such as Debian weak keys and popular known-compromised keys such as those improperly shipped with SAML software releases:
    - `net/shibboleth/metadata/validate/x509/debian-512.txt`
    - `net/shibboleth/metadata/validate/x509/debian-1024.txt`
    - `net/shibboleth/metadata/validate/x509/debian-2048.txt`
    - `net/shibboleth/metadata/validate/x509/debian-4096.txt`
    - `net/shibboleth/metadata/validate/x509/compromised-1024.txt`
    - `net/shibboleth/metadata/validate/x509/compromised-2048.txt`
    - Multiple `X509RSAOpenSSLBlacklistValidator` instances should be configured to test for multiple blacklist sets, as only one `Resource` can be consumed by each instance. Note, however, that if RSA key length is also constrained to, say, 2048 bits, blacklists corresponding to shorter keys can be ignored.
  - MDA-74: `X509RSAKeyLengthValidator` checks for RSA modulus sizes smaller than a given number of bits. Properties allow setting a warning and error threshold; by default, modulus values less than 2048 bits in length are regarded as errors.
  - MDA-155: `X509RSAExponentValidator` checks for invalid (negative or odd) or insecurely small RSA exponent values. Properties allow setting a warning and error threshold; by default, values of e smaller than 5 are regarded as errors.
- MDA-156: added `RegistrationAuthorityItemIdentificationStrategy` for interfederation use cases. This extends the basic identifier produced by `FirstItemIdItemIdentificationStrategy` by adding a component corresponding to `RegistrationAuthority` item metadata, if present. This would normally be extracted from the entity beforehand using the `RegistrationAuthorityPopulationStage`.
  - A set of registration authorities can be ignored by setting the `ignoredRegistrationAuthorities` property. For example, you may wish to provide only basic identifiers for entities from your own registration authority.
  - Registration authority names (URIs) can be mapped to more convenient display names (such as country codes or federation proper names) by setting a `Map<String, String>` as the `registrationAuthoritiesDisplayNames` property.

API Changes

- MDA-131: the `identifierStrategy` property of `ItemMetadataFilterStage`, `ItemMetadataTerminationStage` and `StatusMetadataLoggingStage` has been renamed to `identificationStrategy` for consistency with other parts of the API.
- `PrivateKeyFactoryBean`, `PublicKeyFactoryBean`, `X509CertificateFactoryBean`and `X509CertificateChainFactoryBean` input properties are all now called "resource" and are all Spring `Resource` objects rather than Java `File` objects.
- `ItemSerializer#serialize` now takes `Item<T>` instead of `Collection<Item<T>>`
- `DomDocumentFactoryBean` is now `DOMDocumentFactoryBean`
- `DOMDocumentFactoryBean`'s `documentResource` property is now `resource`
- The `SetValidUntilStage` and `SetCacheDurationStage` duration setters now throw `ConstraintViolationException` if a value less than or equal to zero is provided, rather than leaving this to be detected at initialization time.
- The `connectionDisregardSslCertificate` property of the `net.shibboleth.utilities.java.support.httpclient.HttpClientBuilder` has been renamed to be `connectionDisregardTLSCertificate`.
- MDA-123: `EntityRegistrationAuthorityFilterStage` has moved from the `net.shibboleth.metadata.dom.saml` package to `net.shibboleth.metadata.dom.saml.mdrpi`

API Removals

- MDA-129: `ElementFormattingStage` removed
- MDA-122: `EntityPublisherPathFilterStage` removed
- MDA-122: `PushDownCacheDurationStage` removed
- MDA-122: `PushDownValidUntilStage` removed
- MDA-122: `SetPublicationInfo` removed
- MDA-122: `XMLSignatureSigningStage`'s `deriveKeyNames` property removed
- MDA-123: `SAMLMetadataSupport.RPI_NS` removed (use `MDRPIMetadataSupport.MDRPI_NS`)