# IdP Reloading the Configuration

New options with IdPv3

SWITCH

SWITCHaai Team
aai@switch.ch

## Reloading the configuration with v2

- only supported in a limited way – by setting the **`configurationResourcePollingFrequency`** attribute of one these services to a short value:
  - attribute resolver (`attribute-resolver.xml`)
  - attribute filtering engine (`attribute-filter.xml`)
  - profile handler manager (`handler.xml`)
  - relying party configuration manager (`relying-party.xml`)
- potentially dangerous when repeated reload attempts fail (by default, **`configurationResourcePollingRetryAttempts`** is only set to 3, after which reloading stops)
- no option to explicitly trigger a reload, so only achieved by a relatively awkward constantly-watch-for-file-changes check

# New reloading options with v3

- reload is explicitly triggered by calling two special-purpose **admin flows,** which are configured under
```
https://aai-login.example.org/idp/profile/admin/reload-service?id=bean-id
https://aai-login.example.org/idp/profile/admin/reload-metadata?id=md-id
```
- available bean IDs for service reloads: see
```
$ grep "bean id=.*class" /opt/shibboleth-idp/system/conf/services-system.xml
```
  and the corresponding resource lists in **`services.xml`**
- reloading metadata: find the IDs with
```
$ grep Provider.*id /opt/shibboleth-idp/conf/metadata-provider-*xml
```
- by default, access to the `reload-*` URLs is restricted to localhost (and if **`access-control.xml`** is configured as suggested in the SWITCH installation guide, to the AAI Resource Registry)
- two reload scripts get installed under **`/opt/shibboleth-idp/bin`**, and serve the same purpose

  depend on **`JAVA_HOME`** being set, and a proper **`-u`** argument being specified… requesting the respective URL with **`curl`** seems more straightforward

# Available bean IDs for service reloads

**`shibboleth.LoggingService`**: logging configuration reload (`logback.xml`)

**`shibboleth.AttributeFilterService`**: attribute filter reload

**`shibboleth.AttributeResolverService`**: reloads attribute and data connector definitions (`attribute-resolver-*.xml` files)

**`shibboleth.NameIdentifierGenerationService`**: reloads the configuration in the `saml-nameid.xml` file

**`shibboleth.RelyingPartyResolverService`**: reloads `relying-party.xml` and `credentials.xml`

**`shibboleth.MetadataResolverService`**: reloads the metadata list specified in `services.xml`

**`shibboleth.ReloadableAccessControlService`**: reloads the configuration in the `access-control.xml` file

Missing from this list: an ID for reloading the **`shibboleth.MessageSourceResources`** list, i.e. the message text files under **`/opt/shibboleth-idp/messages/`**.
By default, the IdP only caches these for five minutes, however, so they are reloaded automatically (see also `idp.message.cacheSeconds` in `services.properties`).

# And restartless login page editing, too

- the IdP v3 has switched to Velocity templates as the new default mechanism for rendering the login (and error) pages
  - edit the **`.vm`** files under **`/opt/shibboleth-idp/views/`**, and the changes become effective immediately
  - say goodbye to container restarts (Tomcat), which was required when JSP files were changed with the IdP v2

# Still requiring a restart with v3

- changes to the contents of **`services.xml`**

  i.e., changes to the **`<util:list>`** elements themselves (such as adding an additional **`attribute-resolver-*.xml`** file)
- changes to **`global.xml`** (SQL data source, HTTP client settings)
- changes to the authentication configuration, such as LDAP parameters etc.
- changes to **`/opt/shibboleth-idp/edit-webapp/...`** files (need **`build.sh`** to be run first, followed by a container restart)
- and a few more, of course… but under normal operating conditions, such reconfigurations relatively rarely occur

# (Ideas for) hands-on exercises

- try reloading a couple of the services listed on slide 4
  ```
  curl https://aai-login.example.org/idp/profile/admin/reload-service?id=…
  ```
- check what happens when specifying invalid bean IDs
- insert a syntax error into a configuration file, and try reloading the corresponding service
- entries in `idp-audit.log` just record reload events with
  ```
  …||||http://shibboleth.net/ns/profiles/reload-metadata||||||||
  …||||http://shibboleth.net/ns/profiles/reload-service-configuration||||||||
  ```
  How can you determine what `id=` argument was supplied?
- what is an easy method to quickly print the currently running IdP and Java version details to `idp-process.log`?