Apache Jena

- Home
- Download
- Learn
  - Tutorials
  - Overview
  - RDF core API tutorial
  - SPARQL tutorial
  - Manipulating SPARQL using ARQ
  - Using Jena with Eclipse
  - How-To's
  -
  - References
  - Overview
  - Javadoc
  - RDF API
  - RDF I/O
  - ARQ (SPARQL)
  - RDF Connection - SPARQL API
  - Elephas - tools for RDF on Hadoop
  - Text Search
  - TDB
  - SDB
  - SPARQL over JDBC
  - Fuseki
  - Permissions
  - Assembler
  - Ontology API
  - Inference API
  - Command-line tools
  - Extras
- Javadoc
  - Jena Core
  - ARQ
  - TDB
  - Fuseki
  - Elephas
  - Text Search
  - Spatial Search
  - Permissions
  - JDBC
  - All Javadoc
- Ask
- Get involved
  - Contribute
  - Report a bug
  -
  - Project
  - About Jena
  - Roadmap
  - Architecture
  - Project team

    - Related projects
    -
    - ASF
    - Apache Software Foundation
    - License
    - Thanks
    - Become a Sponsor
    - Security
- Improve this Page

1. DOCUMENTATION
2. FUSEKI2
3. FUSEKI SECURITY

Security in Fuseki2

Fuseki2 provides security by using Apache Shiro. This is controlled by the configuration file `shiro.ini` located at `$FUSEKI_BASE/shiro.ini`. If not found, the server initializes with a default configuration. This can then be replaced or edited as required. An existing file is never overwritten by the server.

In its default configuration, SPARQL endpoints are open to the public but administrative functions are limited to `localhost`. One can access it via `http://localhost:.../....` Or the according IPv4 or IPv6 address, for example `127.0.0.1` (IPv4), or `[::1]` (IPv6). Access from an external machine is not considered as localhost and thus restricted.

Once Shiro has been configured to perform user authentication it provides a good foundation on which the Jena Permissions layer can be configured. There is an example implementation documented in the Jena Permissions section. The Jena Permissions layer can be used to restrict access to specific graphs or triples within graphs.

A simple example to enable basic user/password authentication is shown in the default `shiro.ini` configuration. The default admin user is `admin` and the password is `pw`. This can be changed directly in the INI file. Note that this setup is not recommended for production for various reasons (no TLS, passwords in plain text etc.), consult the Shiro INI documentation for best practices.

As mentioned above, the default setup only restricts access to the admin pages of Fuseki. To avoid clashes with dataset names, the namespace of the admin interface starts with '/$/', consult the Fuseki HTTP Administration Protocol documentation for more details.

If access to SPARQL endpoints should be restricted, additional Shiro ACLs are necessary. This is done in the `[urls]` section of the configuration. As an example, restricting access to the `../query` SPARQL endpoint for all datasets on Fuseki could be done with this wildcard pattern:

```
/**/query = authcBasic,user[admin]
```

Anonymous SPARQL queries would no longer be possible in this example.

Again, please consult the Apache Shiro website for details and more sophisticated setups. The default configuration of Fuseki is kept simple but is *not* recommended for setups where sensitive data is provided.

Changing the security setup requires a server restart.

Contributions of more examples are very welcome.

Examples

The shipped `shiro.ini` has additional comments.

The default configuration.

This is a minimal configuration for the default configuration.

```
[main]
localhost=org.apache.jena.fuseki.authz.LocalhostFilter

[urls]
## Control functions open to anyone
/$/server = anon
/$/ping   = anon
## and the rest are restricted to localhost.
## See above for 'localhost'
/$/** = localhost
/**=anon
```

Simple user/password

This extract shows the simple user/password setup.

It adds a `[users]` section and changes the `/$/**` line in `[urls]`

```
[users]
admin=pw

[urls]
## Control functions open to anyone
/$/status = anon
/$/ping   = anon
/$/** = authcBasic,user[admin]
# Everything else
/**=anon
```