

Grouper ESB Provisioner

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

The Grouper ESB Connector is designed to enable Grouper to interface with an ESB in order to send and receive individual events as changes occur (see [architectural diagram](#)). It was included in Grouper 1.6 and is currently experimental. This means that, although it is in use in a production environment, it has not seen sufficient use to be considered stable. It hooks into the [changelog consumer](#) and is not a general import/export tool, supporting a limited number of operations. It is deliberately lightweight and if functionality you require is not available you may wish to consider using the Web Service instead.

The connector is intended to interface with an ESB, but anything that can receive and process events packages as JSON strings send over HTTP(S) or XMPP can be used.

Example HTTP ESB outgoing

Configure in grouper-loader.properties

```
changeLog.consumer.esb.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.EsbConsumer
#run every minute
changeLog.consumer.esb.quartzCron = 0 * * * * ?
changeLog.consumer.esb.filter = event.eventType eq 'GROUP_DELETE' || event.eventType eq 'GROUP_ADD' || event.
event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType eq 'MEMBERSHIP_ADD'
changeLog.consumer.esb.publisher.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.
EsbHttpPublisher
changeLog.consumer.esb.publisher.url = http://localhost:8100/whatever/something
```

Copy the grouper/lib/jetty/jetty.jar and jetty-lib.jar to grouper/lib/grouper

Run the loader: gsh -loader

See this event at the HTTP endpoint

```
POST /whatever/something HTTP/1.1
Content-Type: application/json; charset=utf-8
User-Agent: Jakarta Commons-HttpClient/3.0
Host: localhost:8100
Content-Length: 218
{"esbEvent":{"displayName":"etc:rulesActAsGroup","eventType":"GROUP_ADD","id":"c0660833739147d996edf35ec0418398","name":"etc:rulesActAsGroup","parentStemId":"453bc7000061473aa9a0f00d0f67c62e","sequenceNumber":"319"}}
[]}
```

Outgoing

All events which can be notified to an event log consumer are supported. Single events are loaded into an instance of the EsbEvent bean, which is added to an array in an instance of the EsbEvents bean. The EsbEvents class is transformed into an JSON string and despatched. The transformation includes the transformation of the EsbEvent beans, and as this occurs properties with null values are ignored so that they do not appear in the JSON string. More information on the [outgoing beans](#) is available.

In a future release it will be possible to send multiple events as a single JSON string, but only one is supported for now as it makes integration debugging easier.

Incoming

Membership add/deletes for a subject are supported. A listener can be set up to receive events over HTTP(S) or XMPP. The event must be a JSON string, which will be transformed into an EsbListenerEvents bean. This contains an array of one or more EsbListenerEvent beans. More information on the [incoming beans](#) is available.

Configuration

Examples of how to [configure Grouper to use the ESB connector](#) are available. Detailed configuration examples for various uses cases linked to an ESB can be found with the documentation for the (open source) [ESB/Rules engine hybrid](#) in use at Cardiff University.

Grouper client and example

The grouper client can consume ESB XMPP notifications and handle them. An [example is here](#). In this sense, the grouper client can run as a daemon to listen for grouper changes of certain things (filter is configured on server and client), and do perform certain actions (implement an interface or use a built in one for example to update a text file). The grouper client will also allow a quartz (cron like) schedule of a full refresh to prevent data corruption.