GÉANT
Networks • Services • People

Identity Federation Training

Jagger Federation management tool :

Installation guide

www.geant.org

Jagger (ResourceRegistry3) is a web application developed by HEAnet to manage the Edugate multiparty SAML federation. Also Jagger can be used to manage federation, web-of-trust for a single entity or as GUI for the Shibboleth SAML Identity Provider, offering proper Attribute Filter functionality for it. Jagger also offer possibility to enrich IdPs metadata by adding missing attributes requested by target SPs.

Jagger requirements, description, and other useful information can be found at
https://jagger.heanet.ie

GitHub with latest changes is located at
https://github.com/Edugate/Jagger

Federation management using Jagger

Jagger can be installed manually or by using Docker swarm.
Manual installation guide contains installation and configuration of:

1. CentOS 7 permissions and file structure
2. Apache Web server and PHP
3. MariaDB Database
4. Codeigniter Framework
5. Jagger tool

Commands and instructions for Jagger manual installation are listed on next slides.
All commands are supposed to be applied using privileged user account.

Jagger has no special requirements regarding hardware.
Software requirements are limited to:

- Linux OS
- MySQL > 5.1
- PHP >= 5.6 (recommended >=7.1)
- Apache >= 2.4

Install Jagger - Environment

Used operational system :	Operating System: CentOS Linux 7 (Core)
Software to be installed :	Apache 2.4.6
	MariaDB 10.5.2
	PHP 7.3 + additional packages
	Docker
	Composer
	Codeigniter 3.11
	Git, mc, wget, unzip, memcached, gearmand

Install Jagger - Environment setup

Apache

- *yum install httpd*
- *systemctl start httpd.service*
- *systemctl enable httpd.service*
- *firewall-cmd --list-ports*
- *firewall-cmd --permanent --add-port=80/tcp*
  - *firewall-cmd --reload*

This set of commands will install, enable autostart and launch Apache. Also in firewall rules, port 80 will be opened.

MariaDB

- *mcedit /etc/yum.repos.d/MariaDB.repo*

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.5/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

- *yum install MariaDB-server*
- *systemctl start mariadb*
- *systemctl enable mariadb.service*
- *mysql_secure_installation*

This set of commands will add new package repo source, install, enable autostart and launch MariaDB. Last command is used to secure current installation by setting root account for MariaDB.

Install Jagger - Environment setup

Jagger installation require PHP 5.6 or higher. But for future compatibility with any other applications, PHP 7.3 and some of its modules will be installed.

In order to do this, first, custom repo is added, then PHP itself is installed. Finally to get changes applied web server is restarted.

- *yum install epel-release yum-utils*
- *yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm*
- *yum-config-manager --enable remi-php73*
- *yum install php php-common php-opcache php-mcrypt php-gd php-curl php-mysqlnd php-intl php-xml php-mbstring php-xmlrpc php-soap php-bcmath php-cli php-zip php-gearman python-pip*
- *systemctl restart httpd.service*

Install Jagger - Environment setup

Jagger Installation and configuration process will require some additional packets like:

wget - used to download data from provided URL
unzip - used to extract Codeigniter from archive
git - github manager, used to clone Jagger to local storage
gearmand - used by Jagger to perform periodic jobs

- *yum install mc git wget unzip memcached gearmand java*
- *php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"*
- *php composer-setup.php --install-dir=/usr/local/bin --filename=composer*

Install Jagger Resource Registry

In order to get Jagger installed, first of all it will be cloned from github. In order to adjust used packages to current requirements *composer.json* will be edited and runned. Then *Codeigniter* framework installed and adjusted. For best compatibility in this installation is used latest stable *Codeigniter 3-rd* version. Because of specifics of archive format, *unzip* package will be used.

- *git clone https://github.com/Edugate/Jagger /opt/rr3*
- edit /opt/rr3/application/composer.json : *add line* "symfony/console": "*",
  edit line "doctrine/orm": "*",
- */usr/local/bin/composer install*
- *cd /opt*
- *wget https://codeload.github.com/bcit-ci/CodeIgniter/zip/3.1.11*
- *unzip 3.1.11*
- *mv CodeIgniter-3.1.11 codeigniter*
- *cp /opt/codeigniter/index.php /opt/rr3/*
- edit /opt/rr3/index.php : *$system_path = '/opt/codeigniter/system';*

Install Jagger Resource Registry

Next step is to create database user and database itself. Provided commands should be applied from *mariadb cli*. Just use your own credentials instead of highlighted text.

- create database : *create database rr3 CHARACTER SET utf8 COLLATE utf8_general_ci;*
- create user: *grant all on rr3.* to **rr3user**@'localhost' identified by '**rr3pass**';*
- apply changes : *flush privileges;*

Codeigniter database configuration should be edited in order to use provided database type.

- edit /opt/rr3/application/config/database.php :

$db['default']['dbdriver'] = 'mysqli';

Install Jagger Resource Registry

Apache configuration file should be edited to work with Jagger and Codeigniter

- edit apache config (*mcedit /etc/httpd/conf/httpd.conf*)

```
Alias /rr3 /opt/rr3
  <Directory /opt/rr3>
      #  you may need to uncomment next line
      Require all granted
        RewriteEngine On
        RewriteBase /rr3
        RewriteCond $1
!^(Shibboleth\.sso|index\.php|logos|signedmetadata|flags|images|app|schemas|fonts|styles|images|js|robots\.t
xt|pub|includes)
        RewriteRule  ^(.*)$ /rr3/index.php?/$1 [L]
  </Directory>
  <Directory /opt/rr3/application>
        Order allow,deny
        Deny from all
  </Directory>
```

- *systemctl restart httpd*

Install Jagger Resource Registry

Populate configuration files.

- *cd /opt/rr3/*
- *./install.sh*
- *cd application/config*
- *cp config-default.php config.php*
- *cp config_rr-default.php config_rr.php*
- *cp database-default.php database.php*
- *cp email-default.php email.php*
- *cp memcached-default.php memcached.php*

Use default configuration already available in configuration files or follow recommendations from: https://jagger.heanet.ie/jaggerdocadmin/configfile.html

There can be found detailed explanation for main settings and their values.

For database configuration use credentials applied during **rr3** database creation.

Install Jagger Resource Registry

Centos 7 use SELinux kernel security module which has three modes:

- Enforcing: SELinux allows access based on SELinux policy rules.
- Permissive: SELinux only logs actions that would have been denied if running in enforcing mode.
- Disabled: No SELinux policy is loaded.

SELinux default is in enforcing mode. So it will not allow access to required files.

In order to grant access, following commands should be applied:

- *chown apache:apache -R /opt/rr3/application/models/Proxies*
- *chcon -t httpd_sys_rw_content_t /opt/rr3/application/models/Proxies –R*
- or simply disable Enforced mode using: *setenforce 0*

Next step is to populate database with required tables.

- *cd /opt/rr3/application*
- *./doctrine orm:schema-tool:create*
- *./doctrine orm:generate-proxies*

Install Jagger Resource Registry

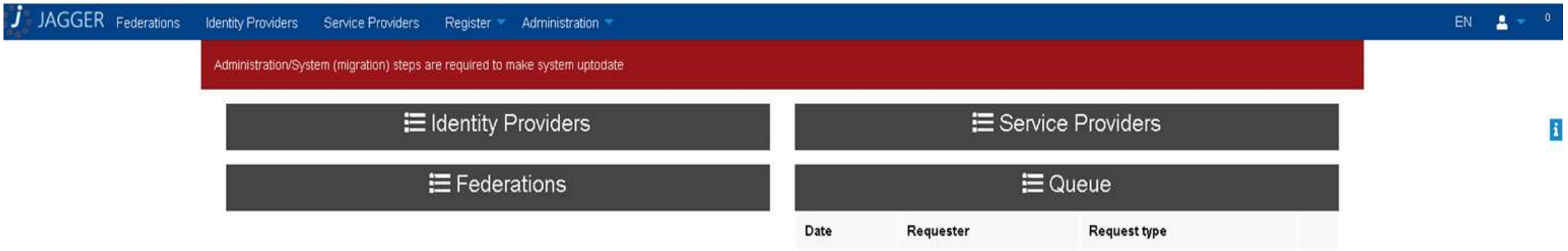
- edit config-rr.php : *$config['rr_setup_allowed'] = TRUE;*
- open: https://yourhost.example.com/rr3/setup

If your connection is not secured, then edit config.php : *$config['cookie_secure'] = FALSE;*

- fill data, thus create administrative account

Jagger Resource Registry Interface

- edit config_rr.php : *$config['rr_setup_allowed'] = FALSE;*
- open: https://yourhost.example.com/rr3

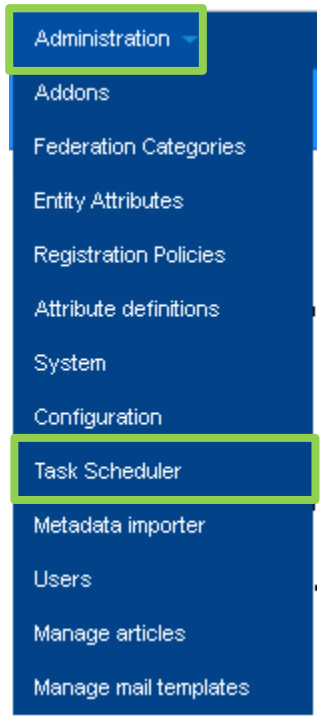


That's it. Now Jagger instance is installed and is near to be ready to use. Next step is to test installation and make final configuration for secure current installation.

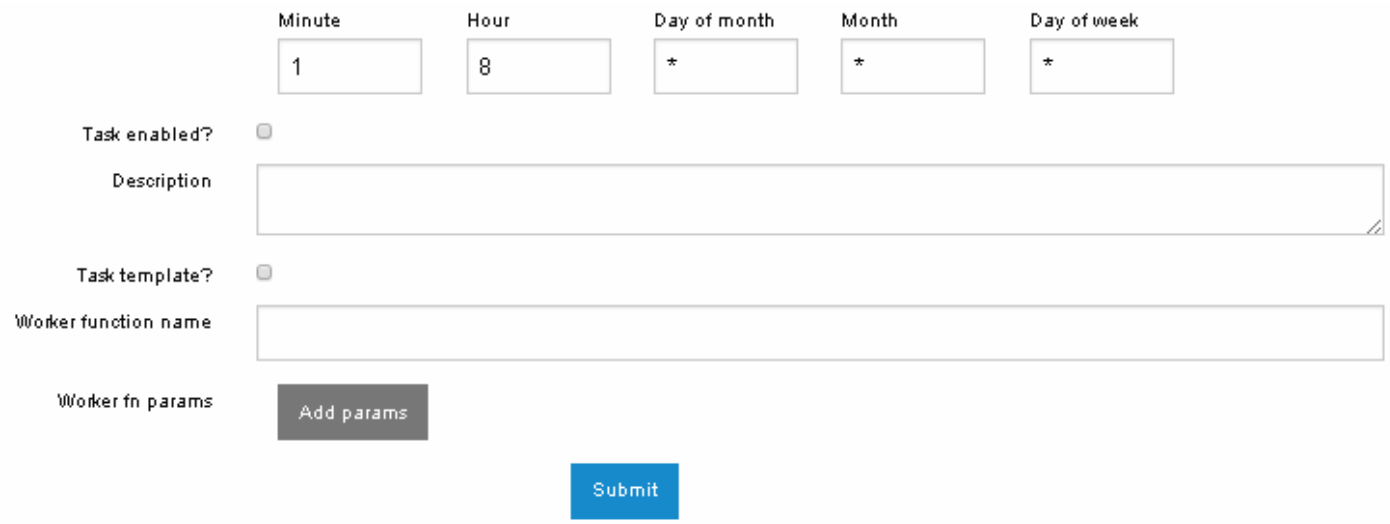
Jagger Resource Registry signing tool

- edit config_rr.php : *$config['featenable']['tasks'] = TRUE;*

Task Scheduler menu should be available in *Administration* menu. Metadata signing task can be added. Time format to run task is similar to *Linux crontab*. Worker name and params will be adjusted after instances will be added to registry.
Signing tool **should be configured**.



Task scheduler interface. New job.



Jagger Resource Registry signing tool

Current version of Jagger supports two worker types to run signing metadata job:

1. RabbitMQ
2. Gearman

Used option can be configured in *config_rr.php* by setting *$config['mq']* equal to *rabbitmq* or *gearman* accordingly and enable selected module within same file. Metadata itself is signed using *xmlsectool* from Shibboleth. It can be added using:

- *cd /opt*
- *wget http://shibboleth.net/downloads/tools/xmlsectool/2.0.0/xmlsectool-2.0.0-bin.zip*
- *unzip xmlsectool-2.0.0-bin.zip*
- *mv xmlsectool-2.0.0 xmlsectool*

Jagger Resource Registry signing tool

SAML flow require metadata to be signed. Certificate used to sign metadata should be generated. It can be done by using *openssl*. Self-signed certificate will be generated into *xmlsectool* folder.

- *cd /opt/xmlsectool*
- *openssl req -x509 -newkey rsa:4096 -keyout **key.**key -out **cert.**crt -days 3650 -subj "/C=**MY**/L=**City**/O=**NREN**/OU=**Federation**/CN=**www.federation.my**"*
- provide password of 4 -1024 symbols length

Setup metadata signing tool and its environment. *gearman* extension will be registered in *python* and signed metadata output folder will be created.

- *pip install gearman*
- *mkdir /opt/rr3/signedmetadata*
- create and insert data to */opt/gearman-worker-metasigner.py*, using provided with presentation template

Now metadata can be signed *manually* by pressing dedicated button.

Jagger Resource Registry signing tool

Metadata can be signed periodically by using internal Jagger *cron* tool. In order to run added in Jagger Task Scheduler jobs, *jcron monitor* script should be started.

- *php /opt/rr3/index.php gworkers jcronmonitor*

Script will check jobs to run every 30 seconds. Current version of Jagger supports three job types:

1. *metadatasigner* – sign federation metadata,
2. *statcollector* – collect statistics on entities flow over federation lifetime,
3. *syncentity* – synchronize entities.

Some job types can accept input parameters, paired as *key*<>*value*, witch extend their functionality. Exist two levels of parameters: which define action and which define required parameter.

Jagger Resource Registry signing tool

Simplest available solution to get metadata signed periodically is to use gearmand + xmlsectool + python + bash script. Follow checklist to enable periodically metadata signing process:

1. Check **.crt* and **.key* files in */opt/xmlsectool/*
2. Edit */opt/gearman-worker-metasigner.py*
   - *Check certificate and key names*
   - *Chech password*
3. Start gearmand process: *gearmand –d*
4. Run script : *sh /opt/runner*

   This script will start two processes, if they wasn't started yet, and store their PIDs in files. It can be installed in linux *crontab* and executed periodically in order to ensure that processes are running.
5. Check *gearmand* workers or status:
   - *gearadmin --workers* - output should contain: *127.0.0.1 metadata : metadatasigner*
   - *gearadmin --status* - output should contain: *metadatasigner 0 0 1*
6. Setup task in Jagger UI and check output file after task will run

Jagger Resource Registry signing tool

Job type *metadatasigner* – metadata sing.

*type** <> *federation* > *sysname* <> *short name of federation* (value)
provider > *entityid* <> *id of local managed entity* (value)
bulk > *name* <> *providers* – sign all entities metadata one by one
federations – sign all federations metadata
all – sign all entities and federations metadata

Shown configuration will apply *metadatasigner* job at 7:55, 11:55, 15:55, 19:55 every day. Job will sign all entities and federations metadata using provided certificate.

Minute	Hour	Day of month	Month	Day of week
55	7,11,15,19	*	*	*

Task enabled? ☑

Description: bulk metadata sign

Task template? ☑

Worker function name: metadatasigner

Worker fn params

arg name	arg value
type	bulk
name	all

Thank you

Any questions?

www.geant.org

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).