

# SEE Identity Federation Training

Jagger Federation management tool :  
Installation guide

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.x

MariaDB 10.5.x

PHP 7.3 + additional packages

Docker (optional)

Composer

Codeigniter 3.11

git, mc, wget, unzip, memcached, gearmand

**Please update your system and install base tools:**

```
yum -y update
```

```
yum -y install git wget unzip mc telnet
```

# Install/update firewall package

## Firewall package

- `yum -y install firewalld`
- `systemctl enable firewalld.service`
- `systemctl start firewalld.service`
- `firewall-cmd --permanent --add-port=22/tcp`
- `firewall-cmd --reload`

This set of commands will install, enable autostart and launch Firewall.

# Install Jagger - Environment setup

## Apache

- `yum -y install httpd mod_ssl mod_rewrite`
- `systemctl enable httpd.service`
- `systemctl start httpd.service`
- `firewall-cmd --list-ports`
- `firewall-cmd --permanent --add-port=80/tcp`
- `firewall-cmd --permanent --add-port=443/tcp`
- `firewall-cmd --reload`

This set of commands will install, enable autostart and launch Apache. Also in firewall rules, port 80 and 443 will be opened.

## MariaDB

- `yum -y install mariadb-server`
- `systemctl enable mariadb.service`
- `systemctl start mariadb.service`
- `mysql_secure_installation`

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

# Configure apache (SSL)

## Apache

- *Update /etc/httpd/conf.d/ssl.conf file*
- *Set proper path HTTPS cert: SSLCertificateFile*
- *Set proper path for HTTPS key: SSLCertificateKeyFile*
- *systemctl restart httpd.service*

This set of commands will update SSL configuration for Apache and restart it.

# Install Jagger - Environment setup

Jagger installation require PHP 5.6 or higher. But for future compatibility with any other applications, PHP 7.3.x 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 -y install epel-release yum-utils`
- `yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm`
- `yum-config-manager --enable remi-php73`
- `yum -y update`
- `yum -y 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:

*gearmand* - used by Jagger to perform periodic jobs

*composer* – dependency manager for PHP

- *yum -y install memcached gearmand java openssl java-11-openjdk-devel*
- *systemctl enable gearmand.service*
- *systemctl start gearmand*
- *yum -y install 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 mcedit /opt/rr3/application/composer.json :
- add line in "require" section "symfony/console": "\*", edit line "doctrine/orm": "\*",
- `cd /opt`
- `wget -O 3.1.11.zip https://codeload.github.com/bcit-ci/CodeIgniter/zip/3.1.11`
- `unzip 3.1.11.zip && rm 3.1.11.zip`
- `mv CodeIgniter-3.1.11 codeigniter`
- `cp /opt/codeigniter/index.php /opt/rr3/`
- edit mcedit /opt/rr3/index.php : `$system_path = '/opt/codeigniter/system';`

# Install Jagger Resource Registry

Apache configuration file should be edited to work with Jagger and Codeigniter edit apache config: `mcedit /etc/httpd/conf.d/z-01-jagger.conf`

```
Alias /rr3 /opt/rr3
<Directory /opt/rr3>
  # you may need to uncomment next line
  Require all granted

  RewriteEngine On

  RewriteCond %{HTTPS} !=on
  RewriteRule ^/?(.*) https://%{SERVER_NAME}/rr3/$1 [R,L]

  RewriteBase /rr3
  RewriteCond $1 !^(Shibboleth\.sso|index\.php|logos|signedmetadata|flags|images|app|schemas|fonts|styles|images|js|robots\.txt|pub|includes)
  RewriteRule ^(.*)$ /rr3/index.php?/$1 [L]
</Directory>
<Directory /opt/rr3/application>
  Order allow,deny
  Deny from all
</Directory>
```

- `systemctl restart httpd.service`

# Install Jagger Resource Registry

Populate configuration files.

- `cd /opt/rr3/`
- `./install.sh`
- `cd /opt/rr3/application/; composer update; composer install`
- `cd /opt/rr3/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.

# 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.

```
mysql -u root -p
```

- 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 mcedit /opt/rr3/application/config/database.php :
```

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

```
$db['default']['username'] → set db username (eg. rr3user)
```

```
$db['default']['password'] → set db password (eg. rr3pass)
```

```
$db['default']['database'] → set db name (eg. rr3)
```

```
$db['default']['dsn'] → update/change 'dbname=CHANGEME' (eg. dbname=rr3)
```

# Install Jagger Resource Registry

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

1. Enforcing: SELinux allows access based on SELinux policy rules.
2. Permissive: SELinux only logs actions that would have been denied if running in enforcing mode.
3. 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:

- `setsebool httpd_can_network_connect_db 1`
- `chcon -t httpd_sys_rw_content_t /opt/rr3/ -R;`
- `chcon -t httpd_sys_rw_content_t /opt/rr3/application/models/Proxies -R`
- or simply disable it using: *setenforce 0*
  
- *chown apache:apache -R /opt/rr3/ /opt/codeigniter*
- `chown apache:apache -R /opt/rr3/application/models/Proxies`

Next step is to populate database with required tables.

- `cd /opt/rr3/application`
- `./doctrine orm:schema-tool:create`

# Install Jagger Resource Registry

- edit /opt/rr3/application/config/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;`

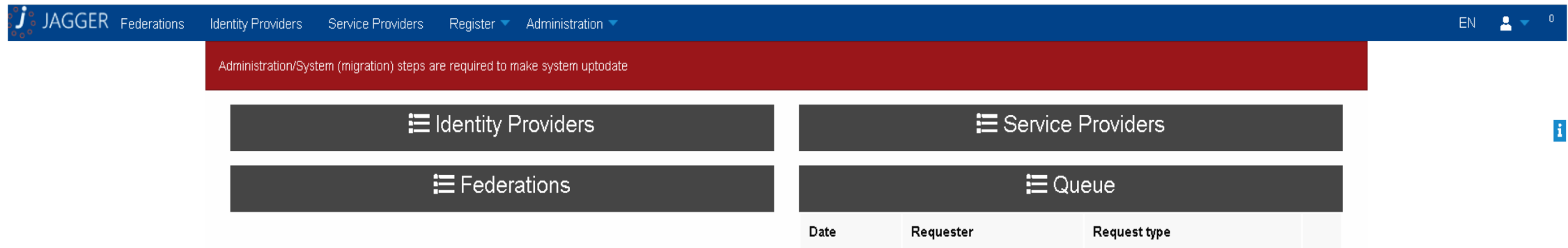
Administrator details

Username	<input type="text"/>
email	<input type="text"/>
Password	<input type="password"/>
Confirm password	<input type="password"/>
First name	<input type="text"/>
Surname	<input type="text"/>

- 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>



The screenshot shows the Jagger Resource Registry Interface. The top navigation bar includes the JAGGER logo and menu items: Federations, Identity Providers, Service Providers, Register, and Administration. A red notification banner at the top states: "Administration/System (migration) steps are required to make system up to date". Below the navigation bar, there are four main menu items: Identity Providers, Service Providers, Federations, and Queue. The Queue menu item is active, displaying a table with the following columns: Date, Requester, and Request type.

Date	Requester	Request type
------	-----------	--------------

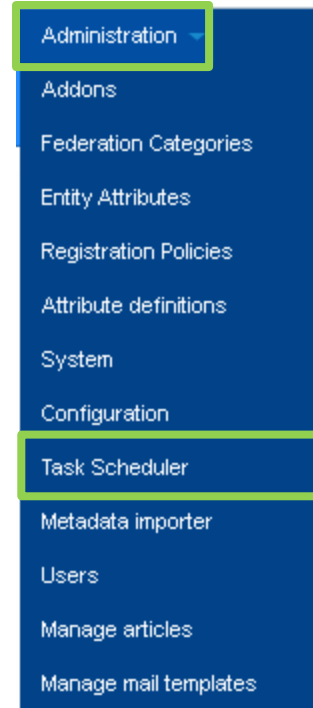
That's it. Now Jagger instance is installed and is near to be ready to use. Next step is to test installation and make final configuration for secure current installation.



# 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.

Minute	Hour	Day of month	Month	Day of week
<input type="text" value="1"/>	<input type="text" value="8"/>	<input type="text" value="*"/>	<input type="text" value="*"/>	<input type="text" value="*"/>
Task enabled?	<input type="checkbox"/>			
Description	<input type="text"/>			
Task template?	<input type="checkbox"/>			
Worker function name	<input type="text"/>			
Worker fn params	<input type="button" value="Add params"/>			
<input type="button" value="Submit"/>				

# 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 *gearman* accordingly and enable selected module within same file. Set following options:

```
$config['mq'] = 'gearman';  
$config['gearman'] = TRUE;
```

Metadata itself is signed using *xmlsectool* from Shibboleth. It can be added using:

- *cd /opt*
- *wget <http://shibboleth.net/downloads/tools/xmlsectool/3.0.0/xmlsectool-3.0.0-bin.zip>*
- *unzip xmlsectool-3.0.0-bin.zip; rm xmlsectool-3.0.0-bin.zip*
- *mv xmlsectool-3.0.0 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 --upgrade "pip < 21.0"; pip install gearman`
- `mkdir /opt/rr3/signedmetadata`
- create and insert data to `/opt/gearman-worker-metasigner.py`, using provided with presentation template

# Jagger Resource Registry signing tool

- Prepare gearman worker service

```
mcedit /etc/systemd/system/gearman-workers.service
```

```
[Unit]
```

```
Description=gearman-workers service
```

```
After=gearmand.service
```

```
[Service]
```

```
Type=simple
```

```
Restart=always
```

```
RestartSec=1
```

```
User=apache
```

```
ExecStart=/usr/bin/env python /opt/gearman-worker-metasigner.py
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
systemctl enable gearman-workers;
```

```
systemctl start gearman-workers
```

```
systemctl status gearman-workers
```

```
# error can occurs – user/group for apache have to be updated in /opt/gearman-worker-metasigner.py file
```

# 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. Before that email.php file have to be configured.

```
mcedit /etc/systemd/system/rr3gworker-jcronmonitor.service
```

```
[Unit]
```

```
Description=RR3 gworkers jcronmonitor service
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
Restart=always
```

```
RestartSec=1
```

```
User=apache
```

```
ExecStartPre=/usr/bin/env php /opt/rr3/index.php gworkers mailqueuesender
```

```
ExecStart=/usr/bin/env php /opt/rr3/index.php gworkers jcronmonitor
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
systemctl enable rr3gworker-jcronmonitor.service
```

```
systemctl start rr3gworkers-jcronmonitor.service
```

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.

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

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
<input type="text" value="55"/>	<input type="text" value="7,11,15,19"/>	<input type="text" value="*"/>	<input type="text" value="*"/>	<input type="text" value="*"/>
Task enabled?	<input checked="" type="checkbox"/>			
Description	<input type="text" value="bulk metadata sign"/>			
Task template?	<input checked="" type="checkbox"/>			
Worker function name	<input type="text" value="metadatasigner"/>			
Worker fn params	arg name	arg value		
	<input type="text" value="type"/>	<input type="text" value="bulk"/>		
	<input type="text" value="name"/>	<input type="text" value="all"/>		

# Jagger Resource Registry mail tool

Mail are send periodically by using internal Jagger *cron* tool. In order to run added in Jagger Task Scheduler jobs, *jcron mailsript* should be started.

```
mcedit /etc/systemd/system/rr3gworker-mailqueue.service
```

```
[Unit]
```

```
Description=RR3 gworkers mailqueuesender service
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
Restart=always
```

```
RestartSec=1
```

```
User=apache
```

```
ExecStart=/usr/bin/env php /opt/rr3/index.php gworkers mailqueuesender
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
systemctl enable rr3gworker-mailqueue.service
```

```
systemctl start rr3gworker-mailqueue.service
```



# Thank you

Any questions?

[www.geant.org](http://www.geant.org)

