

eDirectory 11

Install Guide

Interested in our professional installation service? Our engineers can perform the installation on your environment, saving you time to focus on running your business. [Contact our sales team.](#)

1. System Requirements

1. Apache (2.2 or 2.4) with mod-rewrite enabled
2. Mysql 5.6 or MariaDB 10.0.27
3. PHP 5.6
4. PHP standard libraries + additional libraries: mcrypt, gd, intl, mbstring, pdo, pdo_mysql, exif, apcu/apc.
5. cURL enabled
6. JSON needs to be enabled
7. Set on php.ini:
 1. date.timezone
 2. Safemode Disabled
 3. Openbase_dir Disabled
 4. Open_short_tag must be "ON"
8. Ability to setup at least 11 cron jobs
9. Elasticsearch 2.3.4
 1. Script files should be enabled to install the scripts searchDistance.groovy and updateView.groovy
 2. Recurring events plugin must be installed
10. OpenSSL
11. Composer (for development environments)

2. Installing eDirectory on a develop environment

1. Unzip the file live.tar.gz
2. Move the content from the extracted folder "live" to your installation folder (in this guide, the folder will be called "edirectory")
3. Change the permissions on the following files/folders:
 - /edirectory/web/custom
 - /edirectory/web/bin
 - /app/config/domains
 - /app/config/domain.yml

For folders, permissions should be **755**. Files, should be **644**.

If these permissions are not set properly, the activation of eDirectory and other core functionalities will NOT work.
4. Make sure the document root of your local server is pointing to "*edirectory/web*". Alternatively, you can create a symbolic link from the folder public_html to the folder "*edirectory/web*".

Use the following command to create a symbolic link: `ln -s edirectory/web public_html`

5. Edit the configuration files so the system can connect to the databases and include all necessary files:

1. `/web/conf/config.inc.php`

Modify the following lines:

- a) Line 32: Host of your MySQL server. It can be "localhost" if it's running on the same server used by the application, or the server IP.
- b) Line 33: Username to connect to the MySQL server.
- c) Line 34: Password to connect to the MySQL Server.
- d) Line 35: Name of the main database, i.e, "yourdomain_main"
- e) Line 43: The prefix is the database name used before the suffix. For instance: if the databases were named 'directory_main' and 'directory_domain', the prefix would be 'directory'. This is used when creating new sites through the site manager interface. New databases will be created using this prefix followed by "domain_[id]"
- f) Lines 25 and 26: If you need to receive database debug email messages, feel free to change this email.

2. `/web/custom/domain/domain.inc.php`

Change the value on the array `$domainInfo` with the current domain of your installation in order to make it available (the domain should not include the protocol or www). Replace domain.com or any other value in this file with your current domain. Please notice that the ID (= 1) must match the same ID in the 'Domain' table located inside the 'main' database (Insert done on step 6).

3. `/app/config/domain.yml`

Modify the following lines

- a) Line 8: Change this line for the domain that your eDirectory will work on (without the protocol and www).
- b) Line 9 (id): Change the ID to the same value insert on the table Domain (step 6).
- c) Line 10 (path): Change the path to match the ID of your domain.
- d) Line 11 (theme): Theme of your installation.
- e) Line 12 (locale): Language that your eDirectory will work, for instance, "en-us".
- f) Line 13 (database): Name of the domain database, i.e, "yourdomain_domain1".
- g) Line 14 (elastic): Name of the Elasticsearch index for this domain.
- h) Line 15 (title): Your eDirectory title. This can be changed later through the site manager interface.
- i) Line 16 (branded): on/off to show eDirectory brand.

4. `/app/config/parameters.yml`

Modify the following lines:

- a) Line 4 (database_host): Host of your MySQL server. It can be “localhost” if it’s running on the same server used by the application, or the server IP.
- b) Line 5 (database_port): Port to connect to your MySQL server.
- c) Line 6 (database_name): Name of the main database, i.e, “yourdomain_main”
- d) Line 7 (database_user): Username to connect to the MySQL server.
- e) Line 8 (database_pass): Password to connect to the MySQL Server
- f) Line 10 (elastic_host): Host of your Elasticsearch server. It can be “localhost” if it’s running on the same server used by the application, or the server IP.
- g) Line 11 (elastic_port): Port to connect to your Elasticsearch server.
- h) Line 20 (secret): You can use [this tool](#) to generate a secure 256-bit encryption key.

5. Copy the samples files from the folder /app/config/domains/ and rename them to the following:

- [yourdomain.com].configs.yml
- [yourdomain.com].payment.yml
- [yourdomain.com].route.yml

6. Update composer

Run the following command on your installation folder:

```
composer install
```

7. Create two new databases with “utf8_general_ci” collation in your MySQL server.

eDirectory needs at least two databases to work, one for the main administration and general settings and a secondary one for each domain intended to use. In order to avoid confusion, we suggest the use of specific suffixes for each database name (_main and _domain1 for example). We will refer to these databases as 'main' and 'domain' respectively.

Note: for versions 11.3 or higher, you can create the databases by running the following commands:

```
php app/console doctrine:database:create --connection=main
```

```
php app/console doctrine:database:create --connection=domain
```

8. Create the database structure

1. For versions previous to 11.3: Load the empty_main.dump.sql into the 'main' database and empty_domain.dump.sql into the 'domain' database. All the additional sql files for locations must be loaded into the main database as well.

2. For version 11.3 or higher:

a) Create the database structure of the databases by running the following commands:

```
php app/console doctrine:schema:create --em=main
```

```
php app/console doctrine:schema:create --em=domain
```

- b) Populate the main database with the default data required for a fresh install of eDirectory:

```
php app/console doctrine:fixtures:load --fixtures=src/ArcaSolutions/CoreBundle/DataFixtures/ORM/ --append --em=main
```

- c) Populate the domain database with the default data required for a fresh install of eDirectory.

```
php app/console doctrine:fixtures:load --fixtures=src/ArcaSolutions/WysiwygBundle/DataFixtures/ORM/ --fixtures=src/ArcaSolutions/WebBundle/DataFixtures/ORM/Common --fixtures=src/ArcaSolutions/WebBundle/DataFixtures/ORM/DOMAIN LANGUAGE --fixtures=src/ArcaSolutions/ListingBundle/DataFixtures/ORM --fixtures=src/ArcaSolutions/EventBundle/DataFixtures/ORM --fixtures=src/ArcaSolutions/ClassifiedBundle/DataFixtures/ORM --fixtures=src/ArcaSolutions/ArticleBundle/DataFixtures/ORM --fixtures=src/ArcaSolutions/BannersBundle/DataFixtures/ORM --append --domain=DOMAIN URL
```

Note: Replaces **DOMAIN LANGUAGE** for your domain primary language. Supported values: EN (english), PT (portuguese / Brazil) or ES (spanish); and **DOMAIN URL** should be your domain URL, without protocol.

9. Run the following query on the main database to populate the 'Domain' table after replacing the bold text values to match the database information:

```
INSERT INTO `Domain` (`id`, `smaccount_id`, `name`, `database_host`, `database_port`, `database_username`, `database_password`, `database_name`, `url`, `status`, `activation_status`, `created`, `deleted_date`, `article_feature`, `banner_feature`, `classified_feature`, `event_feature`) VALUES ('1', '1', '[sitename]', '[host]', '[port]', '[user]', '[pass]', '[databasename]', '[url_without_http://]', 'A', 'P', '[date-YYYY-MMDD]', '0000-00-00', 'on', 'on', 'on', 'on');
```

Column	Description	Example
[sitename]	Site title used for quick UI	Domain.com
[host]	The MySQL server address	Localhost
[port]	MySQL db port. (leave empty for default)	3306
[user]	MySQL db user	root
[pass]	MySQL db password	Pass
[databasename]	MySQL secondary 'domain' database name	mydirectory_domain1
[url_without_http]	The url eDirectory will work on	mydirectory.com
[date-YYYY-MMDD]	The date of the setup	2017-01-01

10. Create your Elasticsearch index

Run the following command on your installation folder:

```
php -d memory_limit=-1 app/console edirectory:synchronize --domain=[YOURDOMAIN]
--recreate-index
```

Replace [YOURDOMAIN] for your site domain, without the protocol.

11. Configure the cron jobs (not mandatory for development environments):

Use 'crontab -e' on your shell or set the cronjob options inside your Control Panel.

Example of our cronjob command list:

Versions previous to 11.3:

```
*/5 * * * * php -f [PATH]/web/cron/cron_manager.php 1>&2 >> [PATH]/web/cron/cron.log
*/5 * * * * php -f [PATH]/web/cron/import.php 1>&2 >> [PATH]/web/cron/cron.log
*/5 * * * * php -f [PATH]/web/cron/import_events.php 1>&2 >> [PATH]/web/cron/cron.log
*/10 * * * * php -f [PATH]/web/cron/prepare_import.php 1>&2 >> [PATH]/web/cron/cron.log
*/10 * * * * php -f [PATH]/web/cron/prepare_import_events.php 1>&2 >> [PATH]/web/cron/
cron.log
```

Versions 11.3 or higher:

```
*/5 * * * * php -f /Users/fernandogomes/Sites/edirectory/web/cron/cron_manager.php 1>&2
>> /Users/fernandogomes/Sites/edirectory/web/cron/cron.log
*/5 * * * * php /Users/fernandogomes/Sites/edirectory/app/console edirectory:import --all-
domains 1>&2 >> /Users/fernandogomes/Sites/edirectory/app/logs/import.log
*/10 * * * * php /Users/fernandogomes/Sites/edirectory/app/console edirectory:sync -i --all-
domains 1>&2 >> /Users/fernandogomes/Sites/edirectory/app/logs/import-sync.log
```

Replace [PATH] with the absolute path for your directory, for instance: /home/edirectory/ You may want to change cron.log to match your site name, i.e: cron.log → directory_cron.log

12. Change your edirectory/web/.htaccess file to dev mode:

1. On the first line, change 'prod' to 'dev'

```
setEnv SYMFONY_ENV 'dev'
```

2. Replace all references for "*app.php*" to "*app_dev.php*"

13. Run the following command to generate the assets files (javascripts and stylesheets) for the frontend pages

```
app/console assetic:dump
```

14. Log in the Administration Area (Site Manager):

URL: <http://domain.com/sitemgr>

User: sitemgr@demodirectory.com

Password: abc123

3. Versioning your project

If you want, you can add your project under a version control system (like Git). You can use free version control systems like Github, Gitlab or Bitbucket.

Once you have create your project on your version control system, you just need to initialize Git in your existing project and push the files.

1. Create a .gitignore file

This file is extremely important to avoid versioning files that should not be added on your version control system.

Example of .gitignore for eDirectory:

```
### Symfony ###  
/web/bundles/  
/app/bootstrap.php.cache  
app/cache/  
/app/config/parameters.yml  
/app/logs/*  
!app/cache/.gitkeep  
!app/logs/.gitkeep  
/app/check.php  
/app/SymfonyRequirements.php  
/build/  
/vendor/  
/bin/
```

```
.travis.yml
# eDirectory Ignores
#####
/web/.htaccess
config.inc.php
domain.inc.php
reg.key
web/custom/reg.key
web/custom/*/tmp/*
web/custom/domain_*/
web/assets/css/style_theme_1.css
web/assets/css/style.css
web/assets/css/demobar_part_1_demobar_1.css
web/assets/css/demobar.css
web/assets/*/css/*.css
web/js/
public_html/
!web/dist/
app/Resources/upgrades/*
!app/Resources/upgrades/.gitkeep
```

```
# eDirectory Ignores - Symfony Files
```

```
#####
/app/config/config_app.yml
/app/config/domain.yml
!/app/config/*.yml.dist
/app/config/domains/*
!/app/config/domains/.gitkeep
!/app/config/domains/*.yml.sample
/app/Resources/translations/*com.*.xlf
```

```
# Custom Ignores
```

```
#####
web/media
web/custom/tmp/*
!web/custom/tmp/.gitkeep
!web/custom/tmp/index.*
```

```
web/custom/profile/*
!web/custom/profile/.gitkeep
!web/custom/profile/index.*
web/custom/log/*
!web/custom/log/.gitkeep
!web/custom/log/index.*
web/custom/domain/*
!web/custom/domain/domain.inc.php.sample
!web/custom/domain/index.*
```

2. Run the following commands:

```
git init
git remote add origin [YOUR REPOSITORY URL]
git add .
git commit -m "Initial commit"
git push -u origin master
```

4. Deploying your project

There are several ways to deploy your project to a live server. One of the possibilities is to use the web hooks on push events offered by the source control systems.

This allow you to execute a script on your live server every time a change is pushed to your repository.

Click [here](#) to read more about working with web hooks with Gitlab.

You can find a simple php script to manage gitlab push web hooks [here](#).

After deploying your project for the first time, repeat the following steps from this guide on the production server: Steps 4, 6, 7, 8, 9, 10.

Do not forget to create the file `web/.htaccess` (simply copy and remove the extension from `edirectory/web/.htaccess.sample`. **This step should be done BEFORE running installing composer (step 8).**

You'll notice that some configuration files won't exist, such as `edirectory/app/conf/config.yml` You'll find the following file instead: `edirectory/app/conf/config.yml.sample`. Simply copy this file removing the extension "**.sample**".