arcasolutions

eDirectory 12

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.

Arca Solutions
7138 Little River Turnpike
#1825
Annandale, VA 22003
www.arcasolutions.com

arcasolutions

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 7.1
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 *
12. Node *
13. Gulp *

* requirements mandatory for development environment only.

2. Installing eDirectory on a develop environment

1. Extract the file 12.0.02.tar.gz on the main folder of your eDirectory installation.

2. After extracting the file, you’ll see the following file structure:

```
.
..
composer.lock
ElasticConfigs
app
test
bin
web
gulpfile.hosts.conf.json.sample
gulpfile.sitemgr.conf.json
12.0.02.tar.gz
package.json
gulpfile.sample.js
gulpfile.frontend.conf.json
gulpfile.js
composer.json
vendor
src
```

3. Change the permissions on the following files/folders:
   - /edirectory/web/custom
   - /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.

```
chmod -r 755 web/custom/
```

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

```
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.

```
        # ------------------------------------------------------------------------------------------
        # DATABASE CONNECTION PARAMETERS
        # ------------------------------------------------------------------------------------------
        define("DEFAULT_DB",         "DIRECTORYDB");
        define("_DIRECTORYDB_HOST",  "localhost");
        define("_DIRECTORYDB_USER",  "[redacted]");
        define("_DIRECTORYDB_PASS",  "[redacted]");
        define("_DIRECTORYDB_NAME",  "installedir_main");
        define("_DIRECTORYDB_EMAIL", "f[redacted]");
        if ((defined("DEMO_DEV_MODE") && DEMO_DEV_MODE) || !isset($_SERVER["HTTP_HOST"])) {
                define("_DIRECTORYDB_DEBUG", "display");
        } else {
                define("_DIRECTORYDB_DEBUG", "hide");
        }

        define("DB_NAME_PREFIX", "edirinstall_domain");

        define("MYSQL_TIMEOUT", 10); // Seconds information to each connection of Connection Pool
}
```

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 *automation.arcasolutions.com* or any other value in this file with your current domain.

   Please note that the ID (= 1) must match the same ID in the 'Domain' table located inside the 'main' database (Insert done on step 6).

```
<?
$domainInfo["installedir.arcasolutions.com"] = 1;
~
```

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.

```
# MultiDomain Configuration
parameters:
    api_tokens:
    api_pin:

multi_domain:
    hosts:
        installedir.arcasolutions.com:
            id: 1
            path: custom/domain_1/
            template: default
            locale: en-us
            database: installedir_domain
            elastic: installedir_domain
            title: eDirectory
            branded: on
~
```

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. Usually 9200.

   h) Line 20 (secret): You can use this tool to generate a secure 256-bit encryption key.

```
# This file is auto-generated during the composer install
parameters:
    database_driver: pdo_mysql
    database_host: localhost
    database_port: 3306
    database_name: installedir_main
    database_user: [redacted]
    database_password: [redacted]
    database_charset: UTF8
    database_version: 5.6
    elastic_host: elastic.edirectory
    elastic_port: 9200
    mailer_transport: smtp
    mailer_host: 127.0.0.1
    mailer_user: null
    mailer_password: null
    mailer_port: 465
    mailer_encryption: ssl
    mailer_sender: null
    mailer_auth_mode: login
    locale: en
    charset: UTF-8
    secret: UwdJ7Rg9jpN4sXZsnOOKmOcceVax58DN
    alias_members_module: sponsors
    alias_sitemgr_module: sitemgr
    mixpanel_token: 3d4e8fa7cb2851ed6354668616c0bf92
    unsplash_access_key: e1e311eb661a94cc85e3973d69567f38b7a453997517b79e3b3d87a6cf3fd91b
```

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

   Please note to replace by the exactly domain you'll run this install, including the domain.

```
domain.payment.yml.sample
domain.route.yml.sample
installedir.arcasolutions.com.configs.yml
.gitkeep
installedir.arcasolutions.com.payment.yml
installedir.arcasolutions.com.route.yml
domain.configs.yml.sample
```

6. 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 be used. 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.

Before executing the commands to create the database, open the file *app/config/database.yml* and change the line 23 replacing "~" by the name of the **domain** database of your installation. This step is temporary and will be reverted by the end of this tutorial.

```
# Doctrine Configuration
doctrine:
    dbal:
        default_connection: domain
        connections:
            main:
                dbname:   "%database_name%"
                driver:   "%database_driver%"
                host:     "%database_host%"
                port:     "%database_port%"
                user:     "%database_user%"
                password: "%database_password%"
                charset:  "%database_charset%"
                logging: %kernel.debug%
                profiling: %kernel.debug%
                persistent: false
                server_version: "%database_version%"
                mapping_types:
                    enum: string
                    set: string
                    tinyint: boolean
            domain:
                dbname:   installedir_domain
                driver:   "%database_driver%"
                host:     "%database_host%"
                port:     "%database_port%"
                user:     "%database_user%"
                password: "%database_password%"
                charset:  "%database_charset%"
                logging: %kernel.debug%
                profiling: %kernel.debug%
                persistent: false
                server_version: "%database_version%"
                mapping_types:
                    enum: string
                    set: string
                    tinyint: boolean
```

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

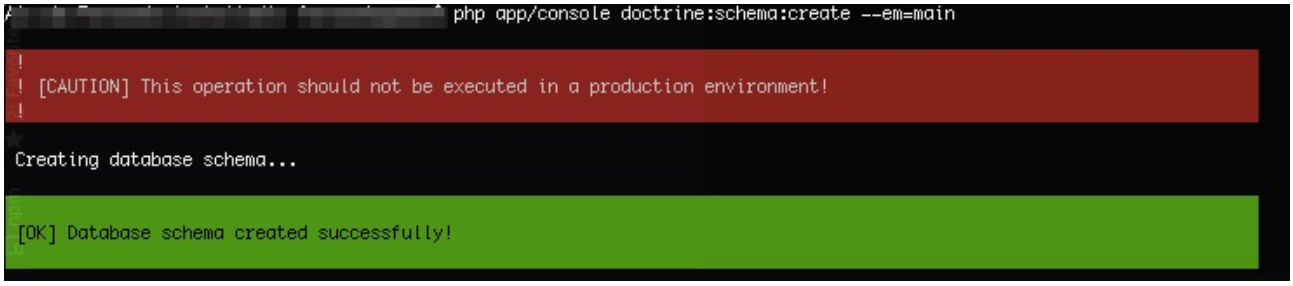
```
$ php app/console doctrine:database:create --connection=main
Created database `installedir_main` for connection named main
```

```
php app/console doctrine:database:create --connection=domain
Created database `installedir_domain` for connection named domain
```

Don't worry about the warning messages, they're just alerting to be careful to not run the commands in production environment, in case the project is already live.

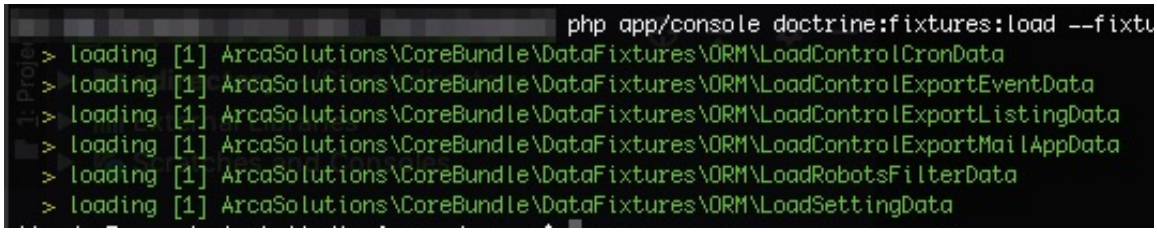
7. Create the database structure
    1. Create the database structure of the databases by running the following commands:

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



    2. 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



    3. Populate the domain database with the default data required for a fresh install of eDirectory. Run the following commands, separately and in this exactly order:

        *php app/console doctrine:fixtures:load --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***

        *php app/console doctrine:fixtures:load --fixtures=src/ArcaSolutions/WysiwygBundle/DataFixtures/ORM/ --append --domain=**DOMAIN_URL***

        *php app/console doctrine:fixtures:load --fixtures=src/ArcaSolutions/WebBundle/DataFixtures/ORM/Common --fixtures=src/ArcaSolutions/WebBundle/DataFixtures/ORM/**LANGUAGE** --append --domain=**DOMAIN_URL***

Note: Replaces **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.

```
[redacted] php app/console doctrine:fixtures:load --fixtures=src/ArcaSo
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadControlCronData
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadControlExportEventData
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadControlExportListingData
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadControlExportMailAppData
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadRobotsFilterData
  > loading [1] ArcaSolutions\CoreBundle\DataFixtures\ORM\LoadSettingData
[redacted] php -d memory_limit=-1 app/console doctrine:fixtures:load --f
Solutions/ClassifiedBundle/DataFixtures/ORM --fixtures=src/ArcaSolutions/ArticleBundle/DataFixtures/ORM
  > loading [1] ArcaSolutions\ListingBundle\DataFixtures\ORM\LoadListingLevelData
  > loading [1] ArcaSolutions\ListingBundle\DataFixtures\ORM\LoadListingLevelFieldData
  > loading [1] ArcaSolutions\ListingBundle\DataFixtures\ORM\LoadListingTemplateData
  > loading [1] ArcaSolutions\EventBundle\DataFixtures\ORM\LoadEventLevelData
  > loading [1] ArcaSolutions\EventBundle\DataFixtures\ORM\LoadEventLevelFieldData
  > loading [1] ArcaSolutions\ClassifiedBundle\DataFixtures\ORM\LoadClassifiedLevelData
  > loading [1] ArcaSolutions\ClassifiedBundle\DataFixtures\ORM\LoadClassifiedLevelFieldData
  > loading [1] ArcaSolutions\ArticleBundle\DataFixtures\ORM\LoadArticleLevelData
  > loading [1] ArcaSolutions\BannersBundle\DataFixtures\ORM\LoadBannerLevelData
[redacted] php -d memory_limit=-1 app/console doctrine:fixtures:load --f
  > loading [1] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadPageTypeData
  > loading [1] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadThemeData
  > loading [1] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadWidgetData
  > loading [2] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadPageData
  > loading [2] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadWidgetThemeData
  > loading [3] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\ThemeDefault\LoadPageWidgetData
  > loading [3] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\ThemeDoctor\LoadPageWidgetData
  > loading [3] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\ThemeRestaurant\LoadPageWidgetData
  > loading [3] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\ThemeWedding\LoadPageWidgetData
  > loading [4] ArcaSolutions\WysiwygBundle\DataFixtures\ORM\Common\LoadWidgetPageTypeData
[redacted] php -d memory_limit=-1 app/console doctrine:fixtures:load --f
ain=installedir.arcasolutions.com
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\Common\LoadFaqData
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\Common\LoadLangData
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\Common\LoadSettingData
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\Common\LoadSettingLocationData
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\EN\LoadEmailNotificationData
  > loading [1] ArcaSolutions\WebBundle\DataFixtures\ORM\EN\LoadEmailNotificationDefaultData
  > loading [5] ArcaSolutions\WebBundle\DataFixtures\ORM\Common\LoadSettingNavigationData
```

8. 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]', '[databasehost]', '[databaseport]', '[databaseuser]', '[databasepass]', '[databasename]','[url_without_http://]', 'A', 'P', '[date-YYYY-MM-DD]', '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

9. 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.

```
[redacted]: php -d memory_limit=-1 app/console edirectory:synchronize --domai
Synchronization Initiated

Synchronizing domain 'eDirectory Install':

Recreating Index...
 10/10 [============================] 100%
Index Recreated.

############### Modules ###############

========= Article =========
    0 [>---------------------------]

Operation took 0 seconds

========= Blog =========
    0 [>---------------------------]

Operation took 0 seconds

========= Classified =========
    0 [>---------------------------]

Operation took 0 seconds

========= Deal =========
    0 [>---------------------------]

Operation took 0 seconds

========= Event =========
    0 [>---------------------------]

Operation took 0 seconds

========= Listing =========
    0 [>---------------------------]
```

10. 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

11. So far your installation is on **production** mode. If you want to run it on development mode, please change your edirectory/web/.htaccess file as following:
    1. On the first line, change ‘prod’ to ‘dev’
       setEnv SYMFONY_ENV ‘dev’

```
1 setEnv SYMFONY_ENV 'dev'
2
3 # eDirectory Rules Start
4 ## EXPIRES CACHING ##
5 <IfModule mod_expires.c>
6
7 # Enable expirations
8 ExpiresActive On
9
```

2. Replace all references for “*app.php*” to “*app_dev.php*”

```
38 # Use the front controller as index file. It serves as a fallback solution when
39 # every other rewrite/redirect fails (e.g. in an aliased environment without
40 # mod_rewrite). Additionally, this reduces the matching process for the
41 # start page (path "/") because otherwise Apache will apply the rewriting rules
42 # to each configured DirectoryIndex file (e.g. index.php, index.html, index.pl).
43 DirectoryIndex app_dev.php
44
45 # Disabling MultiViews prevents unwanted negotiation, e.g. "/app" should not resolve
46 # to the front controller "/app.php" but be rewritten to "/app.php/app".
47 <IfModule mod_negotiation.c>
48     Options -MultiViews
49 </IfModule>
50
```

```
114
115     # Rewrite all other queries to the front controller.
116     RewriteRule .? %{ENV:BASE}/app_dev.php [L]
117 </IfModule>
118
119 <IfModule !mod_rewrite.c>
120     <IfModule mod_alias.c>
121         # When mod_rewrite is not available, we instruct a temporary redirect of
122         # the start page to the front controller explicitly so that the website
123         # and the generated links can still be used.
124         RedirectMatch 302 ^/$ /app_dev.php/
125         # RedirectTemp cannot be used instead
126     </IfModule>
127 </IfModule>
```

12. Revert the change on the file app/config/database.yml done on step 6. On line 23, revert the database name back to “~”

```
18                  mapping_types:
19                      enum: string
20                      set: string
21                      tinyint: boolean
22              domain:
23                  dbname:   ~
24                  driver:   "%database_driver%"
25                  host:     "%database_host%"
26                  port:     "%database_port%"
27                  user:     "%database_user%"
28                  password: "%database_password%"
```

13. Update composer

    Run the following command on your installation folder:

    composer install

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

app/console assetic:dump

15. 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***".