What is the main goal of the feature?

The main purpose of this feature is to generate dynamically the Acted Home page and to implement the registration (sign‐up) process to allow Acted customers and artists to create their accounts.

What are the feature inputs?

* http://www.acted.co/ URL.
* The design mockup of the home page, under the “index.html” file, in the Mockup repo.
* Mockup repo. Acted database.

What are the feature outputs?

A dynamically generated HTML document, respecting the “index.html” file design, filled with the requested information (stored in the database).  
An operational registration process allowing to create Acted users (clients and artists).

What are the main business rules implemented?

Now users can create own accounts.

Define a quick “How it works” schema.

Class FormLoginAuthenticator implements Guard interface. Guard is the new authentication subsystem. FormLoginAuthenticator extract email/password params from Request and checks user’s credentials. Class ArtistVoter is Voter. It is provides method for checking permissions of authenticated user. RegisterType is used during the sign-up procedure. If we must add a new Artist we have to use three different Entity: User, Profile, Artist. So we can not link RegisterType to target Entity. For this purpose, we use POPO class RegisterUser and UserManager. Also confirmationToken field is added to User Entity, because we have to verify email address.

We use the bcrypt for passwords storing. After registration, the system generates random string, sends it inside verification email and sets confirmation\_token field. When user clicks the link, the system checks the token and sets activation flag for this user. We use PHPSESSID cookie to identify authenticated user. And the system uses files as session storage engine.

Which bundle handles the feature?

ActedLegalDocsBundle contains all application logic.

What are the main classes?

Acted\LegalDocsBundle\Security\FormLoginAuthenticator - the main class for authentication subsystem.

Acted\LegalDocsBundle\Model\UserManager - contains methods for user creating procedure and sends verifications emails.

Acted\LegalDocsBundle\Form\RegisterType - the main Form for registration procedure.

Acted\LegalDocsBundle\Controller\SecurityController - contains actions for registration, authentication, resetting passwords and verification.

Acted\LegalDocsBundle\FormErrors\FormErrorsSerializer - provides method for serialization form errors to human readable format.

What are the services?

app.form\_login\_authenticator - implements Guard Authentication for login form.

app.user.manager - provides methods for user management.

app.form\_errors\_serializer - is used for simple errors output.

app.artist\_voter - is Voter. It is used for checking user’s permissions.

What are the Twig templates?

ActedLegalDocsBundle:Default:confirmation.txt.twig - verification email template;

ActedLegalDocsBundle:Default:resetting.txt.twig - email template that send when user requests password resetting;

ActedLegalDocsBundle:Default:cardSection.html.twig - categories output for homepage;

ActedLegalDocsBundle:Default:home.html.twig - the main template for homepage;

ActedLegalDocsBundle:Default:registration.html.twig - sign up form;

ActedLegalDocsBundle:Default:spotlights.html.twig - homespotlights.

What are the add-on bundle you installed?

NelmioApiDocBundle provides methods for API documentation creation.

What are the commands to run to install the feature?

$ composer update

$ php app/console doctrine:schema:drop --force

$ php app/console doctrine:schema:create

$ php app/console doctrine:fixtures:load

$ php app/console assets:install

Also you should check parameters.yml.dist and add new parameters to parameters.yml:

mailer\_transport: smtp # or gmail

mailer\_host: 127.0.0.1 # host

mailer\_user: ~ # username

mailer\_password: ~ # password

mailer\_delivery\_address: ~ # if you add this param, all messages will be send to this email

mailer\_from\_address: ~ # sender address

resetting.token\_ttl: 86400 # lifetime of the password resetting token

image\_upload\_dir: images

Example:

mailer\_transport: gmail

mailer\_host: null

mailer\_user: email@gmail.com

mailer\_password: pa33word

mailer\_delivery\_address: email@gmail.com

mailer\_from\_address: [no-reply-email@gmail.com](mailto:no-reply-email@gmail.com)

resetting.token\_ttl: 86400

image\_upload\_dir: images

What are the commands to run to test the feature?

We do not have tests for this feature.

How to test the feature?  
Go to homepage - **http://[SITENAME]/**. You will see home spotlights and categories. Click Sign Up and you will see the popup with registration form. If you fill up form with valid data you will get a verification email with instructions. Then you can go to homepage and click Log In.