Controller

_______________________ Load User Data ___________________________________

```
<?php

namespace AppBundle\DataFixtures\ORM;
use Doctrine\Common\DataFixtures\FixtureInterface;
use Doctrine\Common\Persistence\ObjectManager;
use AppBundle\Entity\User;
use Symfony\Component\DependencyInjection\ContainerAwareInterface;
use Symfony\Component\DependencyInjection\ContainerInterface;

class LoadUserData implements FixtureInterface, ContainerAwareInterface
{
    /**
     * @var ContainerInterface */
    private $container;

    public function setContainer(ContainerInterface $container = null)
    {
        $this->container = $container;
    }

    public function load(ObjectManager $manager)
    {
        // create objects
        $userSuperAdmin = $this->createActiveUser('super', 'super',
'super@admin.com', ['ROLE_SUPER_ADMIN']);
        $userAdmin = $this->createActiveUser('admin', 'admin', 'admin@admin.com',
['ROLE_ADMIN']);
        $userFlo = $this->createActiveUser('flo', 'coste', 'cosefloric a@yahoo.com',
['ROLE_FLO']);
        // store to DB
        $manager->persist($userSuperAdmin);
        $manager->persist($userAdmin);
        $manager->persist($userFlo);
        $manager->flush();
    }

    // default role s ROLE_USER
    private function createActiveUser($username, $plainPassword, $email, $roles =
['ROLE_USER']):user
    {
        $users = new user();
        $users->setUsername($username);
        $encodedPassword = $this->encodePassword($users, $plainPassword);
        $users->setPassword($encodedPassword);
        $users->setEmail($email);
        $users->setRoles($roles);
        $users->setIsActive(true);
        // password - and encoding

        return $users;
    }

    private function encodePassword($users, $plainPassword):string
    {

        $encoder = $this->container->get('security.password_encoder');
        $encodedPassword = $encoder->encodePassword($users, $plainPassword);
        return $encodedPassword;
    }
}
```

_______________________ Security Controller ___________________________________

```
<?php
namespace AppBundle\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Request;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;

class SecurityController extends Controller
{
    /**
     * @Route("/login", name="login")
     */
    public function loginAction(Request $request)
    {
        $authenticationUtils = $this->get('security.authentication_utils'); // get
the login error if there is one
        $error = $authenticationUtils->getLastAuthenticationError(); // last
username entered by the user
        $lastUsername = $authenticationUtils->getLastUsername();
// Twig stuff
        $templateName = 'security/login';
        $argsArray = [
            'last_username' => $lastUsername,
            'error' => $error,];
        return $this->render($templateName . '.html.twig', $argsArray);
    }
}
```

__________________ Load User Data ____________________________________

```
<?php

namespace AppBundle\Controller;

use AppBundle\Entity\product_details;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;use
Symfony\Component\HttpFoundation\Request;

/**
 * Product_detail controller.
 *
 * @Route("product_details")
 */
class product_detailsController extends Controller
{
    /**
     * Lists all product_detail entities.
     *
     * @Route("/", name="product_details_index")
     * @Method("GET")
     */
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();

        $product_details = $em->getRepository('AppBundle:product_details')-
>findAll();

        return $this->render('product_details/index.html.twig', array(
            'product_details' => $product_details,
        ));
    }

    /**
```

```
     * Creates a new product_detail entity.
     *
     * @Route("/new", name="product_details_new")
     * @Method({"GET", "POST"})
     */
    public function newAction(Request $request)
    {
        $product_detail = new Product_details();
        $form = $this->createForm('AppBundle\Form\product_detailsType',
$product_detail);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->persist($product_detail);
            $em->flush($product_detail);

            return $this->redirectToRoute('product_details_show', array('id' =>
$product_detail->getId()));
        }

        return $this->render('product_details/new.html.twig', array(
            'product_detail' => $product_detail,
            'form' => $form->createView(),
        ));
    }

    /**
     * Finds and displays a product_detail entity.
     *
     * @Route("/{id}", name="product_details_show")
     * @Method("GET")
     */
    public function showAction(product_details $product_detail)
    {
        $deleteForm = $this->createDeleteForm($product_detail);

        return $this->render('product_details/show.html.twig', array(
            'product_detail' => $product_detail,
            'delete_form' => $deleteForm->createView(),
        ));
    }

    /**
     * Displays a form to edit an existing product_detail entity.
     *
     * @Route("/{id}/edit", name="product_details_edit")
     * @Method({"GET", "POST"})
     */
    public function editAction(Request $request, product_details $product_detail)
    {
        $deleteForm = $this->createDeleteForm($product_detail);
        $editForm = $this->createForm('AppBundle\Form\product_detailsType',
$product_detail);
        $editForm->handleRequest($request);

        if ($editForm->isSubmitted() && $editForm->isValid()) {
            $this->getDoctrine()->getManager()->flush();

            return $this->redirectToRoute('product_details_edit', array('id' =>
$product_detail->getId()));
        }

        return $this->render('product_details/edit.html.twig', array(
            'product_detail' => $product_detail,
            'edit_form' => $editForm->createView(),
            'delete_form' => $deleteForm->createView(),
        ));
    }

    /**
     * Deletes a product_detail entity.
     *
     * @Route("/{id}", name="product_details_delete")
     * @Method("DELETE")
     */
    public function deleteAction(Request $request, product_details $product_detail)
    {
        $form = $this->createDeleteForm($product_detail);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->remove($product_detail);
            $em->flush();
        }

        return $this->redirectToRoute('product_details_index');
    }

    /**
     * Creates a form to delete a product_detail entity.
     *
     * @param product_details $product_detail The product_detail entity
     *
     * @return \Symfony\Component\Form\Form The form
     */
    private function createDeleteForm(product_details $product_detail)
    {
        return $this->createFormBuilder()
            ->setAction($this->generateUrl('product_details_delete', array('id' =>
$product_detail->getId())))
            ->setMethod('DELETE')
            ->getForm()
        ;
    }
}
```

_______________________ Load User Data ___________________________________

```
<?php

namespace AppBundle\Controller;

use AppBundle\Entity\suppliers_quotation;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;use
Symfony\Component\HttpFoundation\Request;

/**
 * Suppliers_quotation controller.
 *
 * @Route("suppliers_quotation")
 */
class suppliers_quotationController extends Controller
{
    /**
     * Lists all suppliers_quotation entities.
     *
     * @Route("/", name="suppliers_quotation_index")
     * @Method("GET")
     */
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();

        $suppliers_quotations = $em-
>getRepository('AppBundle:suppliers_quotation')->findAll();

        return $this->render('suppliers_quotation/index.html.twig', array(
            'suppliers_quotations' => $suppliers_quotations,
        ));
    }

    /**
     * Creates a new suppliers_quotation entity.
     *
     * @Route("/new", name="suppliers_quotation_new")
     * @Method({"GET", "POST"})
     */
    public function newAction(Request $request)
    {
        $suppliers_quotation = new Suppliers_quotation();
        $form = $this->createForm('AppBundle\Form\suppliers_quotationType',
$suppliers_quotation);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->persist($suppliers_quotation);
            $em->flush($suppliers_quotation);

            return $this->redirectToRoute('suppliers_quotation_show', array('id' =>
$suppliers_quotation->getId()));
        }

        return $this->render('suppliers_quotation/new.html.twig', array(
            'suppliers_quotation' => $suppliers_quotation,
            'form' => $form->createView(),
        ));
    }

    /**
     * Finds and displays a suppliers_quotation entity.
     *
     * @Route("/{id}", name="suppliers_quotation_show")
     * @Method("GET")
     */
    public function showAction(suppliers_quotation $suppliers_quotation)
    {
        $deleteForm = $this->createDeleteForm($suppliers_quotation);

        return $this->render('suppliers_quotation/show.html.twig', array(
            'suppliers_quotation' => $suppliers_quotation,
            'delete_form' => $deleteForm->createView(),
        ));
    }

    /**
     * Displays a form to edit an existing suppliers_quotation entity.
     *
     * @Route("/{id}/edit", name="suppliers_quotation_edit")
     * @Method({"GET", "POST"})
     */
    public function editAction(Request $request, suppliers_quotation
$suppliers_quotation)
    {
        $deleteForm = $this->createDeleteForm($suppliers_quotation);
        $editForm = $this->createForm('AppBundle\Form\suppliers_quotationType',
$suppliers_quotation);
        $editForm->handleRequest($request);
        if ($editForm->isSubmitted() && $editForm->isValid()) {
            $this->getDoctrine()->getManager()->flush();

            return $this->redirectToRoute('suppliers_quotation_edit', array('id' =>
$suppliers_quotation->getId()));
        }

        return $this->render('suppliers_quotation/edit.html.twig', array(
            'suppliers_quotation' => $suppliers_quotation,
            'edit_form' => $editForm->createView(),
            'delete_form' => $deleteForm->createView(),
        ));
    }

    /**
     * Deletes a suppliers_quotation entity.
     *
     * @Route("/{id}", name="suppliers_quotation_delete")
     * @Method("DELETE")
     */
    public function deleteAction(Request $request, suppliers_quotation
$suppliers_quotation)
    {
        $form = $this->createDeleteForm($suppliers_quotation);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->remove($suppliers_quotation);
            $em->flush();
        }

        return $this->redirectToRoute('suppliers_quotation_index');
    }

    /**
     * Creates a form to delete a suppliers_quotation entity.
     *
     * @param suppliers_quotation $suppliers_quotation The suppliers_quotation
entity
     *
     * @return \Symfony\Component\Form\Form The form
     */
    private function createDeleteForm(suppliers_quotation $suppliers_quotation)
    {
        return $this->createFormBuilder()
            ->setAction($this->generateUrl('suppliers_quotation_delete', array('id'
=> $suppliers_quotation->getId())))
            ->setMethod('DELETE')
            ->getForm()
        ;
    }
}
```