_sf_ Symfony Exception   🖹 Docs   🛟 Support

InvalidArgumentException   HTTP 500 Internal Server Error

Could not load type "submit": class does not exist.

| **Exception** | Logs **1** | Stack Trace |

Symfony\Component\Form\Exception   ⊟
InvalidArgumentException

⊟ in vendor\symfony\symfony\src\Symfony\Component\Form**FormRegistry.php** (line 86)

```
81.                 }
82.
83.             if (!$type) {
84.                 // Support fully-qualified class names
85.                 if (!class_exists($name)) {
86.                     throw new InvalidArgumentException(sprintf('Could not load type "
87.                 }
88.                 if (!is_subclass_of($name, 'Symfony\Component\Form\FormTypeInterface'
89.                     throw new InvalidArgumentException(sprintf('Could not load type "
90.                 }
91.
```

⊟ FormRegistry -> **getType** ('submit')
    in vendor\symfony\symfony\src\Symfony\Component\Form**FormFactory.php** (line 74)

```
69.
70.             if (!is_string($type)) {
71.                 throw new UnexpectedTypeException($type, 'string');
72.             }
73.
74.             $type = $this->registry->getType($type);
75.
76.             $builder = $type->createBuilder($this, $name, $options);
77.
78.             // Explicitly call buildForm() in order to be able to override either
79.             // createBuilder() or buildForm() in the resolved form type
```

⊟ FormFactory -> **createNamedBuilder** ('save', 'submit', _null_, _array_('label' => 'Guardar'))
    in vendor\symfony\symfony\src\Symfony\Component\Form**FormBuilder.php** (line 106)

500

```
101.        if (null === $type && null === $this->getDataClass()) {
102.            $type = 'Symfony\Component\Form\Extension\Core\Type\TextType';
103.        }
104.
105.        if (null !== $type) {
106.            return $this->getFormFactory()->createNamedBuilder($name, $type, null, $c
107.        }
108.
109.        return $this->getFormFactory()->createBuilderForProperty($this->getDataClass(
110.    }
111.
```

◀   ▶

⊟ FormBuilder -> **create** ('save', 'submit', *array*('label' => 'Guardar'))
in vendor\symfony\symfony\src\Symfony\Component\Form**FormBuilder.php** (line 269)

```
264.         * Converts all unresolved children into {@link FormBuilder} instances.
265.         */
266.        private function resolveChildren()
267.        {
268.            foreach ($this->unresolvedChildren as $name => $info) {
269.                $this->children[$name] = $this->create($name, $info['type'], $info['optio
270.            }
271.
272.            $this->unresolvedChildren = array();
273.        }
274.    }
```

◀   ▶

⊟ FormBuilder -> **resolveChildren** ()
in vendor\symfony\symfony\src\Symfony\Component\Form**FormBuilder.php** (line 215)

```
210.        {
211.            if ($this->locked) {
212.                throw new BadMethodCallException('FormBuilder methods cannot be accessed
213.            }
214.
215.            $this->resolveChildren();
216.
217.            $form = new Form($this->getFormConfig());
218.
219.            foreach ($this->children as $child) {
220.                // Automatic initialization is only supported on root forms
```

◀   ▶

500

in vendor\symfony\symfony\src\Symfony\Component\Form**FormFactory.php** (line 30)

```
25.        /**
26.         * {@inheritdoc}
27.         */
28.        public function create($type = 'Symfony\Component\Form\Extension\Core\Type\FormTy
29.        {
30.            return $this->createBuilder($type, $data, $options)->getForm();
31.        }
32.
33.        /**
34.         * {@inheritdoc}
35.         */
```

FormFactory -> **create** ('FormularioBundle\\Form\\SolicitudesType', *object*(Solicitudes), *array*())
  in vendor\symfony\symfony\src\Symfony\Bundle\FrameworkBundle\Controller**ControllerTrait.php** (line 395)

```
390.        *
391.        * @final since version 3.4
392.        */
393.       protected function createForm($type, $data = null, array $options = array())
394.       {
395.           return $this->container->get('form.factory')->create($type, $data, $options);
396.       }
397.
398.       /**
399.        * Creates and returns a form builder instance.
400.        *
```

Controller -> **createForm** ('FormularioBundle\\Form\\SolicitudesType', *object*(Solicitudes))
  in src\FormularioBundle\Controller**FormController.php** (line 86)

```
81.                  https://symfony.com/doc/3.4/form/direct_submit.html#form-call-submit-dire
82.                  If you need more control over exactly when your form is submitted or whic
83.           */
84.           $solicitudes = new Solicitudes();
85.
86.           $form = $this->createForm(SolicitudesType::class, $solicitudes);
87.
88.           return $this->render('AppBundle:Pruebas:add.html.twig', array('form' => $fo
89.       }
90.
91.  /*
```

500

FormController -> **addAction** ()

[–] **call_user_func_array** (*array*(*object*(FormController), 'addAction'), *array*())
  in var\cache\dev**classes.php** (line 4359)

```
4354.   $arguments = $this->argumentResolver->getArguments($request, $controller);
4355.   $event = new FilterControllerArgumentsEvent($this, $controller, $arguments, $request,
4356.   $this->dispatcher->dispatch(KernelEvents::CONTROLLER_ARGUMENTS, $event);
4357.   $controller = $event->getController();
4358.   $arguments = $event->getArguments();
4359.   $response = \call_user_func_array($controller, $arguments);
4360.   if (!$response instanceof Response) {
4361.   $event = new GetResponseForControllerResultEvent($this, $request, $type, $response);
4362.   $this->dispatcher->dispatch(KernelEvents::VIEW, $event);
4363.   if ($event->hasResponse()) {
4364.   $response = $event->getResponse();
```

[–] HttpKernel -> **handleRaw** (*object*(Request), 1)
  in var\cache\dev**classes.php** (line 4314)

```
4309.   }
4310.   public function handle(Request $request, $type = HttpKernelInterface::MASTER_REQUEST,
4311.   {
4312.   $request->headers->set('X-Php-Ob-Level', ob_get_level());
4313.   try {
4314.   return $this->handleRaw($request, $type);
4315.   } catch (\Exception $e) {
4316.   if ($e instanceof RequestExceptionInterface) {
4317.   $e = new BadRequestHttpException($e->getMessage(), $e);
4318.   }
4319.   if (false === $catch) {
```

[–] HttpKernel -> **handle** (*object*(Request), 1, *true*)
  in vendor\symfony\symfony\src\Symfony\Component\HttpKernel**Kernel.php** (line 202)

```
197.         $this->boot();
198.         ++$this->requestStackSize;
199.         $this->resetServices = true;
200.
201.         try {
202.             return $this->getHttpKernel()->handle($request, $type, $catch);
203.         } finally {
204.             --$this->requestStackSize;
```

500

```
206.        }
207.
```

Kernel -> **handle** (*object*(Request))
in web**app_dev.php** (line 29)

```
24.  $kernel = new AppKernel('dev', true);
25.  if (PHP_VERSION_ID < 70000) {
26.      $kernel->loadClassCache();
27.  }
28.  $request = Request::createFromGlobals();
29.  $response = $kernel->handle($request);
30.  $response->send();
31.  $kernel->terminate($request, $response);
32.
```

require ('C:\\\\Users\\\\m.fernandezsu\\\\Audiovisiales\\\\Formulario_Prueba\\\\web\\\\app_dev.php')
in vendor\symfony\symfony\src\Symfony\Bundle\WebServerBundle\Resources**router.php** (line 42)

```
37.
38.  // Since we are rewriting to app_dev.php, adjust SCRIPT_NAME and PHP_SELF accordingly
39.  $_SERVER['SCRIPT_NAME'] = DIRECTORY_SEPARATOR.$script;
40.  $_SERVER['PHP_SELF'] = DIRECTORY_SEPARATOR.$script;
41.
42.  require $script;
43.
44.  error_log(sprintf('%s:%d [%d]: %s', $_SERVER['REMOTE_ADDR'], $_SERVER['REMOTE_PORT'],
45.
```

500