



Gobierno de Santa Fe
Ministerio de Gobierno y Reforma del Estado
Secretaría de Tecnologías para la Gestión

Estándar de desarrollo

Bundle de Tema Symphony 2: ThemeAplicativoBundle

BORRADOR

Índice de contenido

1 -Introducción.....	3
2 -Ámbito de aplicación.....	3
3 -Instalación.....	3
4 - Características.....	4
4.1 -Vistas principales.....	4
4.2 -Componentes.....	6
4.2.1 -Excepciones.....	7
4.2.2 -Librerías.....	7
4.3 -Parametrización.....	8
4.4 -Filtros.....	10
Anexo A -Control de Documentación.....	13

BORRADOR

1 - Introducción

El objetivo del presente documento es brindar los lineamientos de la Secretaría de Tecnologías para la Gestión, para el diseño web de aplicaciones/trámites, basados en el framework symfony 2.3.x. Para esto se han definido una serie de bundles estándar para su utilización en desarrollo de sistemas web. En el presente documento vamos a especificar el bundle "ThemeAplicativoBundle".

2 - Ámbito de aplicación

El ámbito de aplicación serán los organismos y sus respectivas sectoriales, que requieran publicar aplicaciones/trámites.

3 - Instalación

Este componente se desplegará via composer para poder anexar a cualquier proyecto symfony 2.

Dirigirse a la url del repositorio de paquetes de la provincia para buscar el componente:

<https://app.santafe.gov.ar/archivist>

Como se indica en el sitio, agregar al archivo *composer.json* las siguientes entradas:

```
{
  "repositories": [
    {
      "type": "composer",
      "url": "https://app.santafe.gov.ar/archivist"
    }
  ]
}

require: "stg/theme-bundle": "dev-trunk"
```

Luego, es necesario realizar un *"php composer.phar update"*

Una vez realizado este paso, se debe activar el bundle en *app/AppKernel.php*:

```
<?php
// app/AppKernel.php

public function registerBundles()
{
    $bundles = array(
        // ...
        new STG\DEIM\Themes\Bundles\AplicativoBundle\ThemeAplicativoBundle(),
    );
}
```

También se debe importar el archivo de ruteo del Theme en el archivo routing de nuestra aplicación, de la siguiente forma :

```
# app/config/routing.yml

theme_aplicativo:
    resource: "@ThemeAplicativoBundle/Resources/config/routing.yml"
    prefix:   /
```

4 - Características

Este bundle brinda los componentes de interface para el usuario. El mismo puede ser utilizado con la funcionalidad base y extenderlo de ser necesario, con el objetivo de normalizar la capa de presentación de aplicaciones web.

La librería está definida bajo el siguiente espacio de nombres:

```
<?php

namespace STG\DEIM\Themes\Bundles\AplicativoBundle;
```

Las características principales son:

- Plantea un diseño adaptativo (responsive design)
- Soporte para validaciones de html5.
- Define plantillas de estilos CSS3 para su utilización
- Define estilos css de compatibilidad para navegadores no estandar. (Internet explorer 7, 8 y 9) utilizando: ie.css
- Brinda todos los componentes de JQuery UI para la interface de usuario:
 - <http://jqueryui.com/demos/>
- Define una serie de vistas extensibles (twig) en la carpeta "Default" y "Components"

4.1 - Vistas principales

Para visualizar una vista previa de los assets y de algunos de los elementos que provee el componente ThemeAplicativoBundle, luego de instalarlo acceder a la siguiente url de su proyecto:

```
En desarrollo:
{nombre_de_proyecto_web}/web/app_dev.php/theme-demo/

En producción:
{nombre_de_proyecto_web}/web/theme-demo/
```

Dentro de la carpeta "Default" se encuentran las plantillas de las vistas principales

- base.html.twig: esta es la vista base de donde debemos extender nuestras pantallas de la siguiente manera:

```
{# ejemplo.html.twig #}  
  
{# De esta manera podemos extender la vista base definida para  
nuestra vista. #}  
{% extends 'ThemeAplicativoBundle:Default:base.html.twig' %}
```

A continuación indicamos en la Fig.1, las vistas incluídas y los bloques que se pueden extender o sobre-escribir.

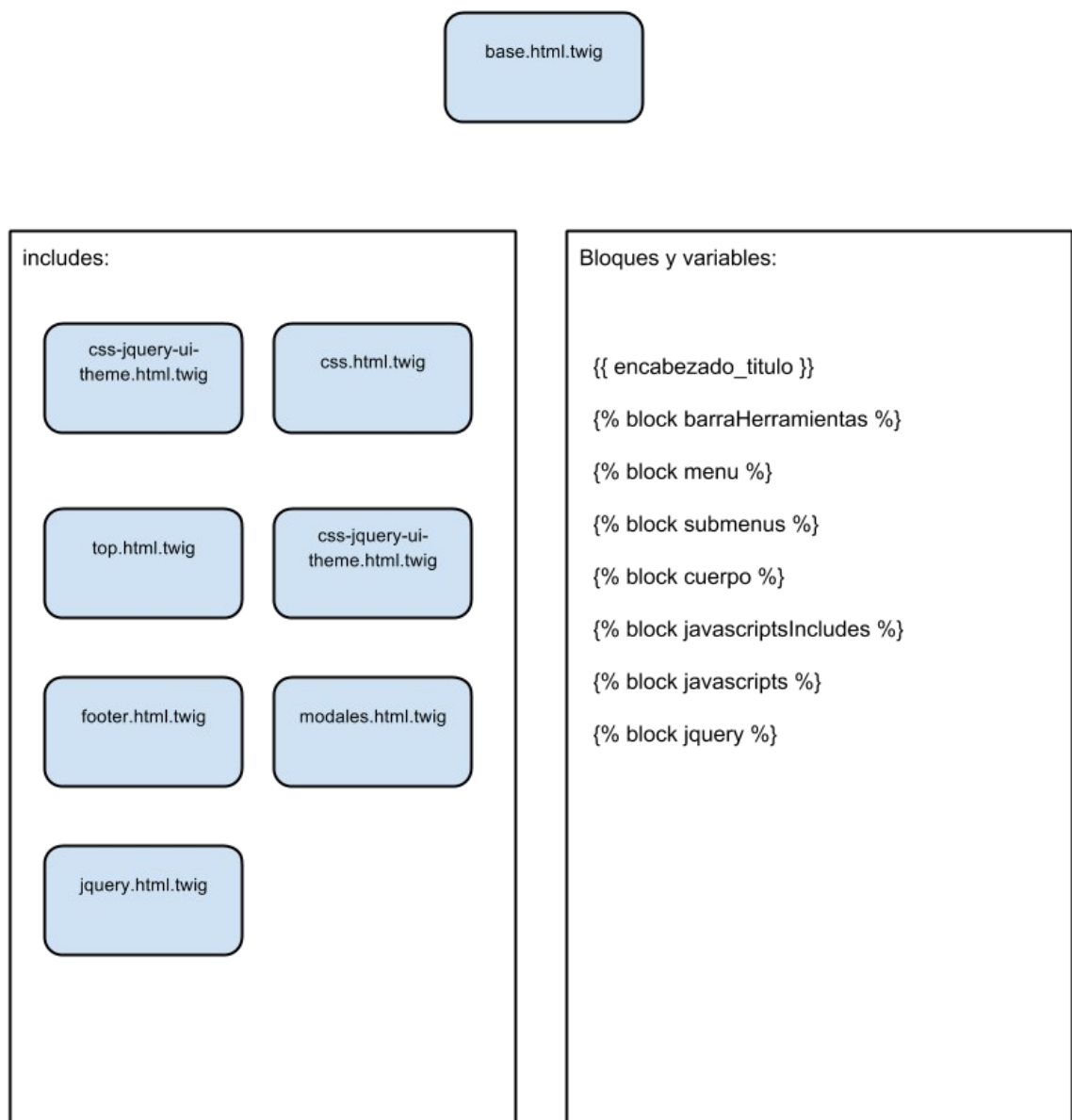


Fig.1: Estructura del template base.html

- top.html.twig
- footer.html.twig
- modales.html.twig: Esta vista es para colocar los div que se van a utilizar para las ventanas modales del sistema, sobre todo las que utilizan contenido que retorna via ajax.
- css-jquery-ui-theme.html.twig: esta vista es la que brinda el soporte del theme de jqueryui.
- css.html.twig: define los estilos base normalizados para los diferentes navegadores, define los estilos de plantilla generales del layout principal, da soporte para tablas responsives y estilos de paginación.
- jquery.html.twig: brinda el soporte de jquery 1.10.x

4.2 - Componentes

A continuación se definen una serie de vistas correspondientes a componentes (html) listos para su reutilización:

- tabla.html.twig:

Se podrá incluir de la siguiente manera:

```
{# ejemplo.html.twig #}

{% include 'ThemeAplicativoBundle:Default:tabla.html.twig' %}

{# Se muestra a continuación como utilizar el formato de tabla establecido por el themeAplicativoBundle, para
poder utilizarlo con nuestra aplicación #}

{% embed 'ThemeAplicativoBundle:Default:tabla.html.twig' %}

    {% block TituloTabla %}Titulo de tabla{% endblock %}

    {% block headerTabla %}

    {% endblock %}

    {% block cuerpoTabla %}

    {% endblock %}

{% endembed %}

//se podrán utilizar todas las funciones válidas que define el component twig
```

- breadcrumb.html.twig: define la ruta de navegación del sistema.
- selectTable.html.twig: define una vista de selección múltiple.
- ayuda.html.twig: define una vista para ayuda general del sistema de tipo acordeones.
- acordion.html.twig: define una vista estilo acordion para su reutilización.
- tabs.html.twig: define una vista de pestañas para su reutilización.
- mensajeFlash.html.twig: define la vista para mensajes de usuario.

4.2.1 - Excepciones

Para manejo de vistas de las excepciones lanzadas por el sistema:

- error.html.twig
- error404.html.twig
- exception_full.html.twig
- exception.html.twig
- layout.html.twig
- logs.html.twig
- trace.html.twig
- traces_text.html.twig
- traces.html.twig

4.2.2 - Librerías

Esta librería, "interface.js", define una serie de métodos que brindan soporte a las vistas, encapsulando las siguientes funcionalidades:

- Ventanas modales:

```
//Este método permite instanciar una ventana modal.  
ventana_modal(url, div, titulo, ancho, alto, funcion_close,  
funcion_abrir, forzarAncho)
```

- Ventana "wait":

```
//Este método permite instanciar una ventana de espera.  
waitModal(mostrar)
```

- Mensajes de usuario permanente:

```
//Brinda soporte para mensajes persistentes en pantalla.  
mensaje_usuario_permanente(titulo, mensaje, tipo)
```

- Mensaje de usuario para confirmación:

```
//Permite instanciar una ventana de dialogo de confirmación
mensaje_usuario_confirma(titulo, mensaje, callback)
```

- Inicialización de componentes de JQuery UI

```
- $("input[type=submit],
  input[type=button],button,.btn").button();
- $(".campo_fecha").datepicker();
- $("#tabs, .tabs").tabs();
- $(".[title]").tooltip({track:true});
```

4.3 - Parametrización

Los elementos que muestran la información básica de la aplicación para los encabezados y pie de página son definidos mediante parámetros de la aplicación. Así se evita incluir estas cadenas de texto en el propio código HTML, permitiendo modificar estos datos de manera sencilla y transparente en los archivos de configuración de la aplicación.

Se deberá incluir en el archivo YAML de configuración un apartado "theme_aplicativo" con las configuraciones propias del bundle.

Cabe aclarar que se implementa validación de los parámetros de esta sección utilizando el mecanismo descrito en <http://symfony.com/doc/current/components/config/definition.html>.

El valor "datos_aplicativo" debe estar definido por la aplicación que desee usar el theme bundle. Dentro de este campo, son obligatorios encabezado y pie_pagina. Dentro de los valores contenidos, el único obligatorio es 'titulo'. Estos parámetros deben cargarse en el archivo de configuración que corresponda (pueden cambiar según el entorno a ejecutar).

```
# app/config/config.yml
theme_aplicativo:
  datos_aplicativo:
    encabezado:
      titulo: "%datos_aplicativo_encabezado_titulo%"
      nombre_dependencia: "%datos_aplicativo_encabezado_nombre_dependencia%"
      nombre_organismo: "%datos_aplicativo_encabezado_nombre_organismo%"
      path_ayuda: "%datos_aplicativo_path_ayuda%"
      path_logout: "%datos_aplicativo_path_logout%"
      path_home: "%datos_aplicativo_path_home%"
      webmaster_correo: "%datos_aplicativo_webmaster_correo%"
    pie_pagina:
      titulo: "%datos_aplicativo_pie_pagina_titulo%"
      domicilio: "%datos_aplicativo_pie_pagina_domicilio%"
      telefono: "%datos_aplicativo_pie_pagina_telefono%"
      copyright: "%datos_aplicativo_pie_pagina_copyright%"
```

Cuando un valor está dentro de %% significa que es un parámetro definido de la aplicación

(parameters.yml). Se muestra un caso de ejemplo en "parameters.yml" incluyendo las siguientes líneas.

```
# app/config/parameters.yml
```

```
    datos_aplicativo_encabezado_titulo: 'Administración de la Intranet
Provincial'
    datos_aplicativo_encabezado_nombre_dependencia: 'Secretaría de tecnologías
para la gestión'
    datos_aplicativo_encabezado_nombre_organismo: 'Ministerio de Gobierno y
Reforma del Estado'
    datos_aplicativo_pie_pagina_titulo: 'GOBIERNO DE SANTA FE'
    datos_aplicativo_pie_pagina_domicilio: '3 de Febrero 2649 (S3000DEE) Santa
Fe'
    datos_aplicativo_pie_pagina_telefono: '54 342 4506600 | 4506700 | 4506800'
    datos_aplicativo_pie_pagina_copyright: '© 2013 - Todos los derechos
reservados. Créditos | Términos y condiciones'
    datos_aplicativo_webmaster_correo: webmaster@santafe.gov.ar
    datos_aplicativo_path_ayuda: ruta_ayuda
    datos_aplicativo_path_logout: ruta_logout
    datos_aplicativo_webmaster_home: ruta_home
```

Las rutas de ayuda, logout y home debe ser definidas para los elementos del encabezado que llevarán a esas secciones. Se pueden completar con valor vacío, en cuyo caso estos links no llevarán a otra página.

Estos valores definidos luego son usados en todas nuestras plantillas twig mediante los nombres cortos correspondientes:

```
# ejemplo.html.twig
```

```
{{ pie_pagina_titulo }}
{{ pie_pagina_domicilio }}
{{ pie_pagina_telefono }}
{{ pie_pagina_copyright' }}
{{ encabezado_titulo }}
{{ encabezado_nombre_dependencia }}
{{ encabezado_nombre_organismo }}
{{ webmaster_correo }}
{{ path_ayuda }}
{{ path_logout }}
{{ path_home }}
```

BORRADOR

4.4 - Filtros

Los listados incluyen la funcionalidad de filtros para visualizar elementos que cumplen con criterios ingresados por el usuario.

Los filtros se generan automáticamente al usar la anotación `@Filter` en un campo de una entidad que se desea filtrar. Para esto se tiene que utilizar el namespace correspondiente en el archivo donde se define la entidad, de la siguiente forma:

```
//Entidad.php  
  
use STG\DEIM\Themes\Bundles\AplicativoBundle\Annotation\Filter;
```

La anotación tiene la propiedad obligatoria `type`, que define que tipo de filtro se va a mostrar. Actualmente están definidos los tipos `text`, `si_no` y `fecha`.

En el caso del tipo `text`, se va a visualizar un campo de texto libre para filtrar el campo. En el caso del tipo `si_no` se va a mostrar un campo de selección simple con los campos "sí" o "no".

Los campos de tipo "fecha" se van a visualizar con 2 input del tipo fecha para establecer el rango de fechas.

Un ejemplo de configuración de un campo "nombreUsuario" de una entidad con filtrado de tipo `text` es el siguiente:

```
//Entidad.php  
  
/**  
 * @var string  
 *  
 * @ORM\Column(name="nombre_usuario", type="string", length=45,  
 * nullable=false)  
 * @Filter(type="text")  
 */  
private $nombreUsuario;
```

También existen 2 propiedades opcionales para la anotación `@Filter`.

label: En esta propiedad se puede definir el label del campo input que se mostrará en la sección de filtros del listado. En caso de no estar definida en el filtro se mostrará el label con el mismo valor que la propiedad de la entidad.

source: En esta propiedad se puede definir el namespace de la entidad en un campo de relación con una entidad (no implementado aún). Este campo es necesario cuando `type` es "entidad".

Ejemplos de implementación

```

/**
 * @var string
 *
 * @ORM\Column(name="nombre_usuario", type="string", length=45,
nullable=false)
 * @Filter(type="text", label="Nombre de Usuario")
 */
/* Esta sentencia permitirá filtrar por nombre de usuario como campo de
texto, con el label "Nombre de Usuario". */
private $nombreUsuario;

/**
 * @var string
 *
 * @ORM\Column(name="aprobado", type="boolean")
 * @Filter(type="si_no", label="Estado de aprobación")
 */
private $aprobado;

```

Esta sentencia permitirá filtrar por aprobado si o no en un campo del tipo booleano. El label mostrado será "Estado de aprobación". En la plantilla twig está definido un bloque filtros que automáticamente va a mostrar nuestros filtros.

Actualmente para que funcione el filtrado en un controlador específico, se deben ejecutar ciertas sentencias en el controlador previamente al renderizado del template.

```

//Controlador.php
$filtros = $this->get('stg.deim.themes.aplicativo.filtro')->
    getFiltros(new Entidad());

$consulta = $this->get('stg.deim.themes.aplicativo.filtro')
    ->generarConsultaMultiplesFiltros($this->nombreEntidad,
        $filtros);

```

De este modo la variable `$filtros` va a contener un arreglo con el `type`, la `columna`, `source`, `propiedad` y `label` de cada propiedad que va a ser usada para filtrar. Esta variable va a ser utilizada en el template para mostrar los campos de ingreso de datos correspondientes.

La variable `$consulta` va a contener la consulta a realizar (es un objeto del tipo `Doctrine\ORM\Query`). Esta va a ser utilizada por el componente de paginación para mostrar los resultados.

Se puede agregar un tercer parámetro opcional al método *generarConsultaMultiplesFiltros*, para determinar el orden predeterminado del paginado. Esto se agregará al orden determinado por la paginación de manera visual (haciendo click en la columna correspondiente).

Este parámetro es la cadena del nombre de la propiedad como está descripta. Por defecto, se ordena de manera ascendente por este valor, si se agrega ' DESC' o ' ASC' podemos especificar el orden del agrupamiento(descendente o ascendente).

```
//Obtener la consulta ordenada por defecto por fecha ascendente
$consulta = $this->get('stg.deim.themes.aplicativo.filtro')
->generarConsultaMultiplesFiltros($this->nombreEntidad,
    $filtros, 'fecha');

//Obtener la consulta ordenada por defecto por fecha descendente
$consulta = $this->get('stg.deim.themes.aplicativo.filtro')
->generarConsultaMultiplesFiltros($this->nombreEntidad,
    $filtros, 'fecha DESC');
```

BORRADOR

Anexo A - Control de Documentación

Título		Bundle de Tema Symfony 2: ThemeAplicativoBundle	
Código de Documento		STG-ED-1	
Elaborado por		Lucas Aguilar, Federico Baroni, Matías Godoy	Fecha Elaboración 05/05/2014
Revisado por			Fecha Revisión
Aprobado por			Fecha Publicación
Nombre de Archivo		STG-ED-1_Bundle_de_Tema_Symfony2_ThemeAplicativoBundle	
Versión	Revisor	Fecha Publicación	Resumen de Cambios

BORRADOR