



CEIS
CENTRO DE ESTUDIOS DE
INGENIERÍA DE SOFTWARE

Facultad de Ingeniería y Ciencias Centro de Estudios de Ingeniería de Software



UNIVERSIDAD
DE LA FRONTERA



CEIS
CENTRO DE ESTUDIOS DE
INGENIERÍA DE SOFTWARE

EXTENSION BUILDER – WEBSERVICES



Este instructivo tiene como finalidad implementar un sistema intranet utilizando una clase de extensión builder

CONTENIDOS

Webservice

- Paso 1: Utilización de Webservice
- Paso 2: Construcción de webservice personalizado

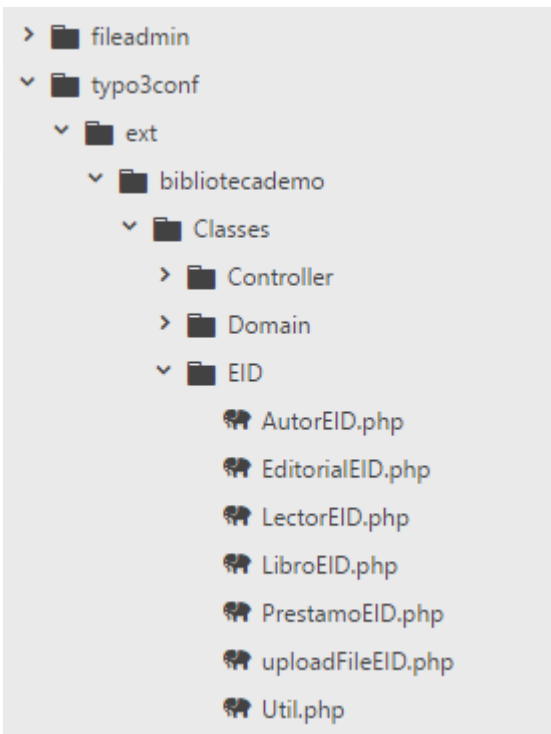
Bloquear Modificaciones de Modelo



Webservices Disponibles

Nota: Recuerda que es necesario tener una extensión creada con extensión builder para continuar con este tutorial.

`/var/www/html/rutaproyecto/typo3conf/ext`



Ruta hacia los archivos de configuración de los webservices

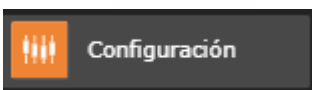
Archivos de configuración de los webservices a partir de los objetos de extensión builder

Archivos de configuración de archivos por defecto.

Webservices Habilitados

Los web services habilitados por el modelo de extensión builder, se encuentran en la sección de configuración dentro de las configuraciones de frontend (FE), en la subsección eID_include

```
... eID_include
... bibliotecademo_autor = EXT:bibliotecademo/Classes/EID/AutorEID.php
... bibliotecademo_editorial = EXT:bibliotecademo/Classes/EID/EditorialEID.php
... bibliotecademo_lector = EXT:bibliotecademo/Classes/EID/LectorEID.php
... bibliotecademo_libro = EXT:bibliotecademo/Classes/EID/LibroEID.php
... bibliotecademo_prestamo = EXT:bibliotecademo/Classes/EID/PrestamoEID.php
... bibliotecademo_upload_file = EXT:bibliotecademo/Classes/EID/uploadFileEID.php
```



Configuración



Paso 1: Utilizar un web service

Para probar los web services habilitados, necesitaras de: postman y datos de acceso

Postman



POSTMAN

Mediante este programa serás capaz de testear los webservice habilitados.

Datos de acceso

Ruta de Webservice

http://moscu.ceisufro.cl/typo37/index.php?eID=bibliotecademo_autor

Ruta de pagina de inicio de proyecto

Nombre de webservice

Indicador de webservice

Parámetros de webservice

Los parámetros se encuentran descritos dentro de los archivos de configuración de cada web service habilitado

```
JSON params: {"usr":"","pass":"","action":"list"}
```

Usuario frontend

Contraseña frontend

Acción habilitada



Acciones de webservice

Las acciones de los webservices se encuentran en el archivo de configuración.

Acciones por defecto y datos adicionales:

1. list: no requiere datos adicionales
2. show: uid de objeto
3. create: datos de propiedades
4. update: uid de objeto, datos de propiedades
5. remove: uid de objeto

Acción "list": test con postman

Link de webservice

Funcion rest

POST `http://moscu.ceisufro.cl/typo37/index.php?eID=bibliotecademo_autor`

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

1 `{"usr": "pedro", "pass": "pedro.2017", "action": "list"}`

Cuerpo de mensaje

Sintaxis de mensaje

Salida de datos

Formato de visualización de JSON

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "data": [
4     {
5       "crdate": {
6         "date": "2017-10-02 18:31:17.000000",
7         "timezone_type": 1,
8         "timezone": "-03:00"
9       },
10      "tstamp": {
11        "date": "2017-10-02 18:31:17.000000",
12        "timezone_type": 1,
13        "timezone": "-03:00"
14      },
15      "nombre": "Leonardo",
16      "uid": 1,
17      "pid": 0
18    },
19    {
20      "crdate": {
21        "date": "2017-10-02 18:33:51.000000"
```

Cuerpo de json



Acción "show": test con postman

POST ▼ http://moscu.ceisufro.cl/typo37/index.php?eID=bibliotecademo_autor

Authorization Headers (1) **Body** ● Pre-request Script Tests

form-data x-www-form-urlencoded raw binary **JSON (application/json)** ▼

```
1 [{"usr": "pedro", "pass": "pedro.2017", "action": "show", "uid": "1"}]
```

uid de objeto

Salida de datos

Formato de visualización de JSON

Cuerpo de json

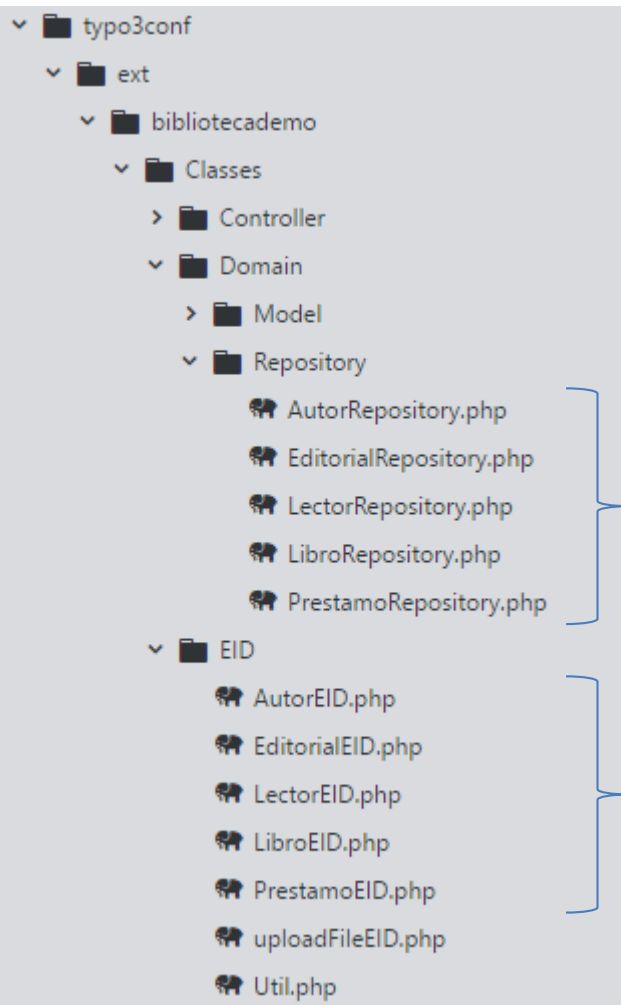
Pretty Raw Preview **JSON** ▼

```
1 {
2   "status": "success",
3   "data": {
4     "crdate": {
5       "date": "2017-10-02 18:31:17.000000",
6       "timezone_type": 1,
7       "timezone": "-03:00"
8     },
9     "tstamp": {
10      "date": "2017-10-02 18:31:17.000000",
11      "timezone_type": 1,
12      "timezone": "-03:00"
13    },
14     "nombre": "Leonardo",
15     "uid": 1,
16     "pid": 0
17   }
18 }
```



Paso 2: Construcción webservice adicional

Necesitaras acceso a los siguientes archivos para poder crear de forma correcta un nuevo webservice personalizado.



Estos archivos contiene las consultas especificas acceder a los objetos de una clase especifica.
Esos archivos se encuentran programados en PHP.

La configuración de los webservice se encuentran en estos archivos, por lo que dentro de cada uno de estos se agregarán los nuevos webservice para cada clase.
Estos archivos se encuentran programados en PHP



Consulta personalizada

 LibroRepository.php Archivo al que se agregará la consulta personalizada

Declaración de parámetro de entrada {
Función de consulta personalizada
Se realizará la búsqueda por el nombre del libro

```
/**  
 * @param $searchTerm  
 */  
public function findbyNombre($searchTerm){  
    $query = $this->createQuery();  
    $query->matching(  
        $query->like("nombre", "%".$searchTerm."%"));  
    return $query->execute();  
}
```



Una vez realizadas las modificaciones, será necesario desactivar y activar nuevamente la extensión modificada para que se recarguen los cambios.



Añadir acción a webservice

LibroEID.php

Archivo al que se agregará la acción personalizada

Condición de objeto
no encontrado

Condición de Libro
no encontrado

Ciclo para almacenar
coincidencias

```
elseif ($datos->action == 'prestamo')
{
    if( !isset($datos->nombre) ) {
        $eidUtil->JSendResponse('fail','no hay nombre');
        die;
    }
    /** @var \CEISUFRO\Bibliotecademo\Domain\Model\Libro $libro */
    $libros = $libroRepository->findbyNombre($datos->nombre)->toArray();
    error_log(count($libros));

    if($libros == NULL){
        $eidUtil->JSendResponse('fail',$datos->nombre.' no encontrado');
        die;
    }
    $dataResponse = array();
    $total = count($libros);
    for( $i = 0; $i < $total; $i++) {
        /** @var \CEISUFRO\Bibliotecademo\Domain\Model\Libro $libro */
        $libro = $libros[$i];
        $dataResponse[] = $libro->_getProperties();
    }
    $eidUtil->JSendResponse('success',$dataResponse);
    die;
}
```



Test

Acción "prestamo": test con postman

POST ▼ http://moscu.ceisufro.cl/typo37/index.php?eID=bibliotecademo_libro Params Send ▼

Authorization Headers (1) **Body** ● Pre-request Script Tests

form-data x-www-form-urlencoded raw binary **JSON (application/json)** ▼

```
1 [{"usr": "pedro", "pass": "pedro.2017", "action": "prestamo", "nombre": "Notas"}]
```

-----● Nueva acción personalizado

Salida de datos

Pretty Raw Preview **JSON** ▼

```
1 {
2   "status": "success",
3   "data": [
4     {
5       "nombre": "Notas de vida",
6       "prestamo": true,
7       "editorial": {},
8       "autor": {},
9       "crdate": {
10        "date": "2017-10-02 18:50:43.000000",
11        "timezone_type": 1,
12        "timezone": "-03:00"
13      },
14       "tstamp": {
15        "date": "2017-10-03 17:59:55.000000",
16        "timezone_type": 1,
17        "timezone": "-03:00"
18      },
19       "uid": 1,
20       "pid": 0
21     }
22   ]
23 }
```

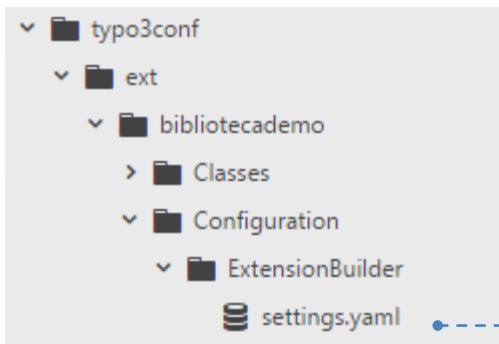
● Nombre de libro

● Estado de Prestado



Bloquear Modificaciones de Modelo

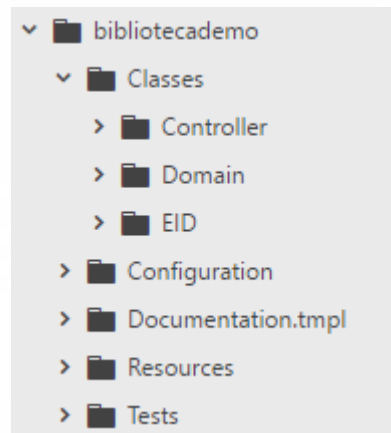
Cada modificación que se realiza al modelo en extensión builder, reescribe todos los archivos que necesitan ajustarse para la implementación del nuevo modelo, lo que puede resultar en el borrado de ciertas configuraciones personalizadas.



Archivo de configuración que contiene las acciones que se ejecutarán cada vez que se modifique el modelo de extension builder

```
overwriteSettings:  
  Classes:  
#   Controller: merge  
  Domain:  
#   Model: merge  
#   Repository: merge  
  
Configuration:  
  #TCA: merge  
  #TypoScript: keep  
  
Resources:  
  Private:  
    #Language: merge  
    Templates: keep  
    Partial: keep  
  
ext_icon.gif: keep
```

La estructura de los archivos e indentación asociada corresponde a las carpetas dentro de la misma extensión.



Las opciones son:

1. merge.: añadir las nuevas configuraciones a las existentes
2. Keep: no modificar, incluso si hay modificaciones que realizar
3. #comentario o no listado: todo aquel archivo que no aparezca en este archivo de configuración será reescrito desde cero.