# AdminBundle/AdminBundle.php

Show:

## \PFC\AdminBundle\AdminBundle

**Package:** global

**Parent(s)**

\Symfony\Component\HttpKernel\Bundle\Bundle

# AdminBundle/Controller/CategoryController.php

## \PFC\AdminBundle\Controller\Ca

**Package:** PFC\AdminBundle\Controller;
Clase controlador que implementa el CRUD de las categorías
**Parent(s)**
    \Symfony\Bundle\FrameworkBundle\Controller\Controller
**Version**
    1.0

## Methods

### createAction() : \PFC\AdminBundle\Controller\Function

» Crea una categoría, asociándole la asignatura actual, y guardando solo el nombre introducido

**Returns**

| Type | Description |
|---|---|
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

### deleteAction(Integer $category_id) : \PFC\AdminBundle\Controller\Function

» Elimina la categoría seleccionada, todas las subcategorías relacionadas y sus cuestiones

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$category_id** | Integer | *Id de la categoría a eliminar* |

**Returns**

| Type | Description |
|---|---|
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

### deleteTestAction(String $subject, Integer $category_id, Integer $question_id) :

» Elimina una pregunta especifica seleccionada

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subject** | String | *Slug de la asignatura seleccionada* |
| **$category_id** | Integer | *Id de la categoría seleccionada* |
| **$question_id** | Integer | *Id de la pregunta a eliminar* |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $this->testListAction($subject, $category_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**exportAction**(Integer $category_id) : \PFC\AdminBundle\Controller\Function

» Método para que exporta a un fichero moodelXml, una batería de preguntas pertenecientes a la categoría pasada como parámetro

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$category_id** | Integer | *Id de la categoría seleccionada* |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $fa->exportTest($em, $path, 'Category', $category_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**listAction**() : \PFC\AdminBundle\Controller\Render

» Método para que muestra un listado de las categorías

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Render | Devuelve la vista AdminBundle:Subject:list.html.twig. |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**testListAction**(string $subject, integer $category_id) : \PFC\AdminBundle\Controller\Render

» Método para que muestra un listado de preguntas de una categoría

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subject** | string | *slug único de la asignatura* |
| **$category_id** | integer | *id de la categoría seleccionada* |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Render | Devuelve la vista index.html.twig |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**updateAction**(string $field) : \Symfony\Component\HttpFoundation\Response

» Actualiza los campos de la categoría, guardándolos en la bd

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$field** | string | *Campo a actualizar de la categoría actual* |

**Returns**

| Type | Description |
|---|---|
| \Symfony\Component\HttpFoundation\Response | Vuelve a la vista actual |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/Controller/QuestionController.php

## \PFC\AdminBundle\Controller\Q

**Package:** PFC\AdminBundle\Controller;
Clase controlador que implementa las operaciones de importación de preguntas
**Parent(s)**
   \Symfony\Bundle\FrameworkBundle\Controller\Controller
**Version**
   1.0

## Methods

---

**indexAction**(integer $subcategory_id) : **\PFC\AdminBundle\Controller\Render**

» Método para que muestra el formulario de importación de preguntas

### Parameters

| Name | Type | Description |
|------|------|-------------|
| **$subcategory_id** | integer | *id de la subcategoría* |

### Returns

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Controller\Render | Devuelve la vista AdminBundle:Question:index.html.twig |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**uploadAction**(integer $subcategory_id) : **\PFC\AdminBundle\Controller\Render**

Método para la importación de Questions. Landing page para la importación. Selecciona el método dependiendo del tipo de importación a realizar.

### Parameters

| Name | Type | Description |
|------|------|-------------|
| **$subcategory_id** | integer | *Identificador de la subcategoría a la que pertenecerá las Quiestions* |

### Returns

| Type | Description |
|------|-------------|
|  |  |

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Render | Devuelve la vista AdminBundle:Question:index.html.twig con el resultado de la operación de importación |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

# AdminBundle/Controller/SubcategoryController.php

## \PFC\AdminBundle\Controller\Sub

**Package:** PFC\AdminBundle\Controller;
Clase controlador que implementa el CRUD de las subcategorías
**Parent(s)**
    \Symfony\Bundle\FrameworkBundle\Controller\Controller
**Version**
    1.0

## Methods

**createAction() : \PFC\AdminBundle\Controller\Function**

» Crea una subcategoría, asociándole la asignatura y categoría actual, y guardando solo el nombre introducido

### Returns

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**deleteAction(Integer $subcategory_id) : \PFC\AdminBundle\Controller\Function**

» Elimina la subcategoría seleccionada, y todas sus cuestiones asociadas

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$subcategory_id** | Integer | *Id de la subcategoría a eliminar* |

### Returns

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**deleteTestAction(String $subject, Integer $subcategory_id, Integer $question_id) : \PFC\AdminBundle\Controller\Function**

» Elimina una pregunta especifica seleccionada

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subject** | String | *Slug de la asignatura seleccionada* |
| **$subcategory_id** | Integer | *Id de la subcategoría seleccionada* |
| **$question_id** | Integer | *Id de la pregunta a eliminar* |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $this->testListAction($subject, $subcategory_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**exportAction(Integer $subcategory_id) : \PFC\AdminBundle\Controller\Function**

» Método para que exporta a un fichero moodelXml, una batería de preguntas pertenecientes a la subcategoría pasada como parámetro

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subcategory_id** | Integer | *Id de la subcategoría seleccionada* |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Function | $fa->exportTest($em, $path, 'Subcategory', $subcategory_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**listAction() : \PFC\AdminBundle\Controller\Render**

» Método para que muestra un listado de las subcategorías

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Controller\Render | Devuelve la vista AdminBundle:Subject:list.html.twig, que se encuentra factorizada en la clase factoryActions->subjectComponentsList |

**Details**

**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**testListAction**(string $subject, integer $subcategory_id) : \PFC\AdminBundle\Controller\Render

» Método para que muestra un listado de cuestiones de la subcategoría

### Parameters

| Name | Type | Description |
|---|---|---|
| **$subject** | string | *slug único de la asignatura* |
| **$subcategory_id** | integer | *id de la subcategoría seleccionada* |

### Returns

| Type | Description |
|---|---|
| \PFC\AdminBundle\Controller\Render | Devuelve la vista Subject:subcategory-test-list.html.twig, que se encuentra factorizada en la clase factoryActions->testList |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

---

**updateAction**(string $field) : \Symfony\Component\HttpFoundation\Response

» Actualiza los datos de las subcategorías

### Parameters

| Name | Type | Description |
|---|---|---|
| **$field** | string | *Campo de la subcategoría a actualizar* |

### Returns

| Type | Description |
|---|---|
| \Symfony\Component\HttpFoundation\Response | Devuelve el control a la vista actual |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

# AdminBundle/Controller/SubjectController.php

## \PFC\AdminBundle\Controller\S

**Package:** PFC\AdminBundle\Controller;
Clase controlador que implementa el CRUD de las asignaturas
**Parent(s)**
    \Symfony\Bundle\FrameworkBundle\Controller\Controller
**Version**
    1.0

## Methods

### createAction() : \PFC\AdminBundle\Controller\Function

» Crea una asignatura vacía y guardando solo el titulo introducido

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

### deleteAction(Integer $subject_id) : \PFC\AdminBundle\Controller\Function

» Elimina la asignatura seleccionada, las categorías, subcategorías y preguntas asociadas

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$subject_id** | Integer | *Id de la asignatura a eliminar* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Controller\Function | $this->listAction() |

**Details**
**Package**
    PFC\AdminBundle\Controller;
**Version**
    1.0

**deleteTestAction**(String $subject, Integer $question_id) : \PFC\AdminBundle\Controller\Function

» Elimina una pregunta especifica seleccionada

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$subject** | String | *Slug de la asignatura seleccionada* |
| **$question_id** | Integer | *Id de la pregunta a eliminar* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Controller\Function | $this->testListAction($subject, $subcategory_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**exportAction**(Integer $subject_id) : \PFC\AdminBundle\Controller\Function

» Método para que exporta a un fichero moodelXml, una batería de preguntas pertenecientes a la asignatura pasada como parámetro

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$subject_id** | Integer | *Id de la asignatura seleccionada* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Controller\Function | $fa->exportTest($em, $path, 'Subject', $subject_id) |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**listAction**() : \PFC\AdminBundle\Controller\Render

» Método para que muestra un listado de las asignaturas

**Returns**

| Type | Description |
|---|---|
| \PFC\AdminBundle\Controller\Render | Devuelve la vista AdminBundle:Subject:list.html.twig, que se encuentra factorizada en la clase factoryActions->subjectComponentsList |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**testListAction(string $subject) : \PFC\AdminBundle\Controller\Render**

» Método para que muestra un listado de cuestiones de la asignatura

**Parameters**

| Name | Type | Description |
|---|---|---|
| $subject | string | *slug único de la asignatura* |

**Returns**

| Type | Description |
|---|---|
| \PFC\AdminBundle\Controller\Render | Devuelve la vista Subject:subcategory-test-list.html.twig, que se encuentra factorizada en la clase factoryActions->testList |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**updateAction(string $field) : \Symfony\Component\HttpFoundation\Response**

» Actualiza los datos de las asignaturas

**Parameters**

| Name | Type | Description |
|---|---|---|
| $field | string | *Campo de la asignatura a actualizar* |

**Returns**

| Type | Description |
|---|---|
| | |

| Type | Description |
| --- | --- |
| \Symfony\Component\HttpFoundation\Response | Devuelve el control a la vista actual |

**Details**
**Package**
　PFC\AdminBundle\Controller;
**Version**
　1.0

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/DependencyInjection/AdminExtension.php

## \PFC\AdminBundle\DependencyInjecti...

**Package:** global

This is the class that loads and manages your bundle configuration

To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html}

**Parent(s)**

    \Symfony\Component\HttpKernel\DependencyInjection\Extension

### Methods

**load**( $configs, $container) : void

{@inheritDoc}

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$configs** | | |
| **$container** | | |

Documentation was generated by phpDocumentor 2.5.0 .

# AdminBundle/DependencyInjection/Configuration.php

Show: **inherited**

## \PFC\AdminBundle\DependencyInjec

**Package:** global

This is the class that validates and merges configuration from your app/config files

To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html#cookbook-bundles-extension-config-class}

**Implements**

## Methods

**getConfigTreeBuilder**() : void

{@inheritDoc}

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/Entity/AssessmentItem.php

» Entidad AssessmentItem

1.0 @date 2014-7-22

## \PFC\AdminBundle\Entity\As

**Package:** global
**JMS\XmlRoot**
    ("assessmentItem")

## Constants

**TAGS** = '<assessmentItem><responseDeclaration><correctResponse><value><mapping><mapEntry>
<itemBody><choiceInteraction><extendedTextInteraction><prompt><simpleChoice><feedbackInline>'

## Properties

### $adaptive

**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("adaptive")
**JMS\Type**
    ("string")
**JMS\XmlAttribute**

### $itemBody

**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("itemBody")
**JMS\Type**
    ("PFC\AdminBundle\Entity\ItemBody")

### $ref

Referencia al fichero XML que es serializado en esta clase
**Details**
**Type**

n/a
**JMS\Type**
  ("string")

---

### $responseDeclaration

**Details**
**Type**
  n/a
**JMS\Type**
  ("ArrayCollection")
**JMS\XmlList**
  (inline = true, entry="responseDeclaration") //Inline, solo categoriza, no incluye items como
  nodo

---

### $title

**Details**
**Type**
  n/a
**JMS\SerializedName**
  ("title")
**JMS\Type**
  ("string")
**JMS\XmlAttribute**

## Methods

**__construct**() : void

# \PFC\AdminBundle\Entity\ChoiceInteraction

**Package:** global

## Properties

### $prompt

Texto de la pregunta
**Details**
**Type**
  n/a
**JMS\SerializedName**
  ("prompt")

**JMS\Type**
    ("string")

### $responseIdentifier

**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("responseIdentifier")
**JMS\Type**
    ("string")
**JMS\XmlAttribute**

### $simpleChoice

**Details**
**Type**
    n/a
**JMS\Type**
    ("ArrayCollection")
**JMS\XmlList**
    (inline = true, entry="simpleChoice")

## Methods

**__construct() : void**

# \PFC\AdminBundle\Entity\CorrectResponse

**Package:** global

## Properties

### $value

**Details**
**Type**
    n/a
**JMS\Type**
    ("array")
**JMS\XmlList**
    (inline = true, entry="value")

# \PFC\AdminBundle\Entity\ExtendedTextInteraction

**Package:** global

## Properties

### $prompt

Texto de la pregunta
**Details**
**Type**
  n/a
**JMS\SerializedName**
  ("prompt")
**JMS\Type**
  ("string")

### $responseIdentifier

**Details**
**Type**
  n/a
**JMS\SerializedName**
  ("responseIdentifier")
**JMS\Type**
  ("string")
**JMS\XmlAttribute**

# \PFC\AdminBundle\Entity\ItemBody

**Package:** global

## Properties

### $choiceInteraction

**Details**
**Type**
  n/a
**JMS\Type**
  ("ArrayCollection")
**JMS\XmlList**
  (inline = true, entry="choiceInteraction")

### $extendedTextInteraction

**Details**
**Type**
  n/a
**JMS\Type**
  ("ArrayCollection")
**JMS\XmlList**
  (inline = true, entry="extendedTextInteraction")

### $text

**Details**
**Type**
  n/a
**JMS\Type**
  ("string")
**JMS\XmlValue**

## Methods

**__construct() : void**

# \PFC\AdminBundle\Entity\MapEntry

**Package:** global

## Properties

### $mapKey

**Details**
**Type**
  n/a
**JMS\SerializedName**
  ("mapKey")
**JMS\Type**
  ("string")
**JMS\XmlAttribute**

### $mappedValue

**Details**

**Type**
    n/a
**JMS\SerializedName**
    ("mappedValue")
**JMS\Type**
    ("string")
**JMS\XmlAttribute**

# \PFC\AdminBundle\Entity\Mapping

**Package:** global

## Properties

### $defaultValue

**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("defaultValue")
**JMS\Type**
    ("string")
**JMS\XmlAttribute**

### $mapEntry

**Details**
**Type**
    n/a
**JMS\Type**
    ("ArrayCollection")
**JMS\XmlList**
    (inline = true, entry="mapEntry")

## Methods

**__construct() : void**

# \PFC\AdminBundle\Entity\ResponseDeclaration

**Package:** global

## Properties

### $cardinality

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("cardinality")
**JMS\Type**
   ("string")
**JMS\XmlAttribute**


### $correctResponse

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("correctResponse")
**JMS\Type**
   ("PFC\AdminBundle\Entity\CorrectResponse")


### $identifier

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("identifier")
**JMS\Type**
   ("string")
**JMS\XmlAttribute**


### $mapping

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("mapping")
**JMS\Type**
   ("PFC\AdminBundle\Entity\Mapping")

## Methods

**__construct**() : void

# \PFC\AdminBundle\Entity\SimpleChoice

**Package:** global

## Properties

**$identifier**

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("identifier")
**JMS\Type**
   ("string")
**JMS\XmlAttribute**

**$value**

**Details**
**Type**
   n/a
**JMS\Type**
   ("string")
**JMS\XmlValue**

Documentation was generated by phpDocumentor 2.5.0 .

# AdminBundle/Entity/XmlManifest.php

» Entidad XmlmManifest, que es utilizado como objeto int encarga de serializar el manifiesto.

## \PFC\AdminBundle\Entity\XmlMani

**Package:** global
**JMS\XmlRoot**
    ("manifest")

### Properties

> **$resources**
>
> **Details**
> **Type**
>     n/a
> **JMS\Type**
>     ("ArrayCollection")
> **JMS\XmlList**
>     (inline = true, entry="resources")

### Methods

> **__construct() : void**

## \PFC\AdminBundle\Entity\XmlResource

**jump to top**

**Package:** global

### Properties

> **$href**
>
> **Details**
> **Type**
>     n/a
> **JMS\Type**
>     ("string")
> **JMS\XmlAttribute**

> **$identifier**

**Details**
**Type**
    n/a
**JMS\Type**
    ("string")
**JMS\XmlAttribute**



**$scormTipe**

**Details**
**Type**
    n/a
**JMS\Type**
    ("string")
**JMS\XmlAttribute**



**$type**

**Details**
**Type**
    n/a
**JMS\Type**
    ("string")
**JMS\XmlAttribute**



# \PFC\AdminBundle\Entity\XmlResources

**Package:** global

## Properties

**$resource**

**Details**
**Type**
    n/a
**JMS\Type**
    ("ArrayCollection")
**JMS\XmlList**
    (inline = true, entry="resource")

## Methods

**__construct**() : void

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/Entity/XmlQuiz.php

Show: **inherited**

» Entidad XmlQuiz, que es utilizado como objeto intermedio ficheros, en la importación y exportación

**Package**
   PFC\AdminBundle\Entity;

## \PFC\AdminBundle\Entity\StringClass

**Package:** PFC\AdminBundle\Entity;

### Properties

#### $format

**Details**
**Type**
   n/a
**JMS\Type**
   ("string")
**JMS\XmlAttribute**

#### $text

**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("text")
**JMS\Type**
   ("string")

## \PFC\AdminBundle\Entity\XmlAnswer

**jump to top**

**Package:** PFC\AdminBundle\Entity;

### Properties

#### $feedback

Texto de la retroalimentación
**Details**
**Type**
   n/a
**JMS\SerializedName**

("feedback")
**JMS\Type**
   ("PFC\AdminBundle\Entity\StringClass")

### $format

**Details**
**Type**
   n/a
**JMS\Type**
   ("string")
**JMS\XmlAttribute**

### $fraction

**Details**
**Type**
   n/a
**JMS\Type**
   ("integer")
**JMS\XmlAttribute**

### $text

Texto redacción de la respuestas
**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("text")
**JMS\Type**
   ("string")

### $tolerance

Tolerancia
**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("tolerance")
**JMS\Type**
   ("double")

**Methods**

**__construct**() : void

# \PFC\AdminBundle\Entity\XmlQuestion

**Package:** PFC\AdminBundle\Entity;

## Properties

**$answers**

ArrayCollection de las respuestas
**Details**
**Type**
    n/a
**JMS\Type**
    ("ArrayCollection")
**JMS\XmlList**
    (inline = true, entry="answer") //Inline, solo categoriza, no incluye answers como nodo

**$category**

Categoría de la pregunta si la tuviera
**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("category")
**JMS\Type**
    ("PFC\AdminBundle\Entity\StringClass")

**$defaultgrade**

Puntuación por defecto
**Details**
**Type**
    n/a
**JMS\SerializedName**
    ("defaultgrade")
**JMS\Type**
    ("double")

## $generalfeedback

Texto de la retroalimentacion
**Details**
**Type**
n/a
**JMS\SerializedName**
("generalfeedback")
**JMS\Type**
("PFC\AdminBundle\Entity\StringClass")

## $hidden

Oculta
**Details**
**Type**
n/a
**JMS\SerializedName**
("hidden")
**JMS\Type**
("integer")

## $name

**Details**
**Type**
n/a
**JMS\SerializedName**
("name")
**JMS\Type**
("PFC\AdminBundle\Entity\StringClass")

## $penalty

Penalizacion por respuesta erronea
**Details**
**Type**
n/a
**JMS\SerializedName**
("penalty")
**JMS\Type**
("double")

## $questiontext

Texto descriptivo de la pregunta
**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("questiontext")
**JMS\Type**
   ("PFC\AdminBundle\Entity\StringClass")

### $single

Tipo de respuesta, sencilla o múltiple
**Details**
**Type**
   n/a
**JMS\SerializedName**
   ("single")
**JMS\Type**
   ("boolean")

### $type

Tipo de la prengunta
**Details**
**Type**
   n/a
**JMS\Type**
   ("string")
**JMS\XmlAttribute**

## Methods

**__construct**() : void

# \PFC\AdminBundle\Entity\XmlQuiz

**Package:** PFC\AdminBundle\Entity;
**JMS\XmlRoot**
   ("quiz")

## Properties

### $questions

ArrayCollection de preguntas

**Details**

**Type**

  n/a

**JMS\Type**

  ("ArrayCollection")

**JMS\XmlList**

  (inline = true, entry="question") //Inline, solo categoriza, no incluye questions como nodo

**$text**

**Details**

**Type**

  n/a

**JMS\SerializedName**

  ("text")

**JMS\Type**

  ("string")

## Methods

**__construct() : void**

# AdminBundle/Tests/Controller/DefaultControllerTest.php

## \PFC\AdminBundle\Tests\Controller\De

**Package:** global
**Parent(s)**
  \Symfony\Bundle\FrameworkBundle\Test\WebTestCase

### Methods

**testIndex() : void**

Documentation was generated by phpDocumentor 2.5.0 .

# AdminBundle/Utils/exportXml.php

Show: **inherited**

## \PFC\AdminBundle\Utils\exportXml

**Package:** global

» Clase que realiza el proceso de exportación a moodleXML.

## Properties

### $pathDir

**Details**
**Type**
   n/a

## Methods

### __construct() : void

### getFileName() : string

Get fileName

#### Returns

| Type | Description |
| --- | --- |
| string | |

### getPathDir() : string

Get pathDir

#### Returns

| Type | Description |
| --- | --- |
| string | |

### loadFromBd(\PFC\AdminBundle\Utils\manager $em, array $questions) : void

» Inserta en el objeto xmlQuiz con las preguntas recibidas como parámetro.

#### Parameters

| Name | Type | Description |
| --- | --- | --- |
| $em | \PFC\AdminBundle\Utils\manager | Doctrine Manager |

| Name | Type | Description |
|------|------|-------------|
| **$questions** | array | *array de preguntas para exportar.* |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**saveFile() : boolean**

» Recoge el contenido del objeto xmlQuiz y lo serializa. El resultado lo inserta el fichero que será exportado.

**Returns**

| Type | Description |
|------|-------------|
| boolean | Devuelve verdadero si la operación se ha realizado correctamente. |

**Details**
**Package**
   PFC\AdminBundle\Controller;
**Version**
   1.0

---

**setPathDir(string $pathDir) : \PFC\AdminBundle\Utils\importXml**

Set pathDir

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$pathDir** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\importXml | |

Documentation was generated by phpDocumentor 2.5.0 .

# AdminBundle/Utils/factoryActions.php

## \PFC\AdminBundle\Utils\factoryActi

**Package:** global

» Clase que factoriza varias funciones usadas en el proyecto.

## Methods

**deleteQuestions**(\Doctrine\Common\Persistence\ObjectManager $em, array $questions) : void

» Método para que elimina un array de preguntas. Evita su repetición en los controladores.

### Parameters

| Name | Type | Description |
|------|------|-------------|
| $em | \Doctrine\Common\Persistence\ObjectManager | Doctine manager. |
| $questions | array | Array de preguntas a eliminar. |

### Details
**Package**
   PFC\AdminBundle\Utils;
**Version**
   1.0

**downloadfileAction**(string $path, string $filename) : \PFC\AdminBundle\Utils\response

» Devuelve al controlador la respuesta para la descarga de un fichero

### Parameters

| Name | Type | Description |
|------|------|-------------|
| $path | string | Directorio donde se encuentra el fichero a descargar en el navegador del cliente. |
| $filename | string | Nombre del fichero a descargar en el navegador del cliente. |

### Returns

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\response | objeto response compuesto con el fichero adjuntado. |

### Details
**Package**

PFC\AdminBundle\Utils;
**Version**
   1.0

---

**exportTest**(\Doctrine\Common\Persistence\ObjectManager $em, string $path, string $selector = '',  $id = '', string $level = 0, string $numQuestions = '') : \PFC\AdminBundle\Utils\render

» Método que gestiona la exportación de un cuestionario a un fichero.

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$em** | \Doctrine\Common\Persistence\ObjectManager | *Doctine manager.* |
| **$path** | string | *Directorio donde se encuentra el fichero a descargar en el navegador del cliente.* |
| **$selector** | string | *Indica el tipo objeto Subject, Category o Subcategory, para el que se exporta la lista.* |
| **$id** | | |
| **$level** | string | *Parámetro opcional con el nivel de las preguntas a exportar. 0 implica cualquier nivel* |
| **$numQuestions** | string | *Parámetro opcional con el número de preguntas máximo a exportar.* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\render | Renderización de un objeto response con el fichero adjunto. Provoca la descarga en el navegador del cliente |

**Details**
**Package**
   PFC\AdminBundle\Utils;
**Version**
   1.0

---

**subjectComponentsList**(\Doctrine\Common\Persistence\ObjectManager $em) : array

» Método para que muestra un listado de las asignaturas, y sus componentes, categorías y

subcategorías

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$em** | \Doctrine\Common\Persistence\ObjectManager | *Doctine manager* |

**Returns**

| Type | Description |
|------|-------------|
| array | array con los parámetros para la renderización de la vista. |

**Details**
**Package**
   PFC\AdminBundle\Utils;
**Version**
   1.0

---

**testList**(\Doctrine\Common\Persistence\ObjectManager $em, \PFC\AdminBundle\Utils\subject $subject, string $selector = '', $id = '') : \PFC\AdminBundle\Utils\array()

» Método para que muestra un listado de preguntas, del tipo seleccionado en el parámetro $selector

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$em** | \Doctrine\Common\Persistence\ObjectManager | *Doctine manager.* |
| **$subject** | \PFC\AdminBundle\Utils\subject | *Objeto subject sobre el que se realiza la consulta.* |
| **$selector** | string | *Indica el tipo objeto Subject, Category o Subcategory, para el que se exporta la lista.* |
| **$id** | | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\array() | Parametros necesarios para la renderización |

**Details**
**Package**
   PFC\AdminBundle\Utils;
**Version**

1.0

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/Utils/importGift.php

## \PFC\AdminBundle\Utils\importGift

**Package:** global

## Properties

### $pathDir

**Details**
**Type**
  n/a

## Methods

### getFileName() : string

Get fileName

**Returns**

| Type | Description |
| --- | --- |
| string | |

### getPathDir() : string

Get pathDir

**Returns**

| Type | Description |
| --- | --- |
| string | |

### importFile() : integer

Importa el archivo a procesar y obtiene los datos necesarios para construir las preguntas y sus respuestas

**Returns**

| Type | Description |
| --- | --- |
| integer | |

### saveFileToBd(object $em, object $subcategory) : void

Construye las entidades Question y Answer y las persiste en la base de datos

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$em** | object | *Objeto que representa un Entity Managers de Doctrine* |
| **$subcategory** | object | *Objeto de la entidad Subcategory* |

---

**setFileName**(string $fileName) : \PFC\AdminBundle\Utils\importXml

Set fileName

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$fileName** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\importXml | |

---

**setPathDir**(string $pathDir) : \PFC\AdminBundle\Utils\importXml

Set pathDir

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$pathDir** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\importXml | |

# AdminBundle/Utils/importQTI.php

## \PFC\AdminBundle\Utils\importQTI

**Package:** PFC\AdminBundle\Controller;
Clase para el manejo de paquetes IMS QTI durante el procesos de importación
**Version**
  1.0

## Properties

### $pathDir

Atributo que contiene la ruta al paquete de IMS QTI subido al servidor
**Details**
**Type**
  n/a

## Methods

### __construct() : \PFC\AdminBundle\Utils\PFC\AdminBundle\Entity\XmlManifest

Constructor de la clase

#### Returns

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Utils\PFC\AdminBundle\Entity\XmlManifest | |

### assessmentItemValidator(array $assessmentItems) : array

Valida que un objeto PFC\AdminBundle\Entity\AssessmentItem sea compatible con al aplicación. Su extructura es la siguiente:

AssessmentItem (title*, adactative) \_ ResponseDeclaration (identifier, cardinality) | \_ CorrectResponse | | \_Value | \_ Mapping (defaultValue)* | \_ MapEntry (mapKey, mappedValue) \_ ItemBody (text*) \_ ChoiceInteraction (responseIdentifier) | \_ Prompt | \_ SimpleChoice (identifier, value) \_ ExtendedTextInteraction (responseIdentifier) \_ Prompt * No son obligatorios

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| $assessmentItems | array | Array de objetos de la clase 'PFC\AdminBundle\Entity\AssessmentItem', seralizados de los XML de en un paquete QTI |

**Returns**

| Type | Description |
|---|---|
| array | Devuelve el array de entrada con los elementos no compatibles eliminados |

**getPathDir**() : string

Get pathDir

### Returns

| Type | Description |
|---|---|
| string | |

**importFile**() : integer

Importa el archivo a procesar y obtiene los datos necesarios para construir las preguntas y sus respuestas

### Returns

| Type | Description |
|---|---|
| integer | |

**saveFileToBd**(object $em, object $subcategory) : void

Construye las entidades Question y Answer y las persiste en la base de datos

### Parameters

| Name | Type | Description |
|---|---|---|
| $em | object | *Objeto que representa un Entity Managers de Doctrine* |
| $subcategory | object | *Objeto de la entidad Subcategory* |

**setPathDir**(string $pathDir) : \PFC\AdminBundle\Utils\importQTI

Set pathDir

### Parameters

| Name | Type | Description |
|---|---|---|
| $pathDir | string | |

### Returns

| Type | Description |
|---|---|
| \PFC\AdminBundle\Utils\importQTI | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# AdminBundle/Utils/importScorm.php

## \PFC\AdminBundle\Utils\importScorm

**Package:** PFC\AdminBundle\Controller;

Clase mara el manejo de paquetes Scorm durante el procesos de importación

**Version**
   1.0

### Properties

#### $pathDir

**Details**
**Type**
   n/a

### Methods

#### __construct() : \PFC\AdminBundle\Utils\PFC\AdminBundle\Entity\XmlManifest

Constructor de la clase

**Returns**

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Utils\PFC\AdminBundle\Entity\XmlManifest | |

#### getFileName() : string

Get fileName

**Returns**

| Type | Description |
| --- | --- |
| string | |

#### getPathDir() : string

Get pathDir

**Returns**

| Type | Description |
| --- | --- |
| string | |

#### importFile() : integer

Importa el archivo a procesar y obtiene los datos necesarios para construir las preguntas y sus respuestas

### Returns

| Type | Description |
| --- | --- |
| integer | |

---

**saveFileToBd( $em, $subcategory) :**

Guarda

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$em** | | |
| **$subcategory** | | |

---

**setFileName(string $fileName) : \PFC\AdminBundle\Utils\imporScorm**

Set fileName

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$fileName** | string | |

### Returns

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Utils\imporScorm | |

---

**setPathDir(string $pathDir) : \PFC\AdminBundle\Utils\importXml**

Set pathDir

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$pathDir** | string | |

### Returns

| Type | Description |
| --- | --- |
| \PFC\AdminBundle\Utils\importXml | |

# AdminBundle/Utils/importXml.php

Show:

## \PFC\AdminBundle\Utils\importXml

**Package:** global

## Properties

### $pathDir

**Details**
**Type**
  n/a

## Methods

### __construct() : void

### getFileName() : string

Get fileName

#### Returns

| Type | Description |
| --- | --- |
| string | |

### getPathDir() : string

Get pathDir

#### Returns

| Type | Description |
| --- | --- |
| string | |

### importFile() : integer

Importa el archivo a procesar y obtiene los datos necesarios para construir las preguntas y sus respuestas

#### Returns

| Type | Description |
| --- | --- |
| integer | |

**saveFileToBd**(object $em, object $subcategory) : void

Construye las entidades Question y Answer y las persiste en la base de datos

### Parameters

| Name | Type | Description |
|------|------|-------------|
| **$em** | object | *Objeto que representa un Entity Managers de Doctrine* |
| **$subcategory** | object | *Objeto de la entidad Subcategory* |

**setFileName**(string $fileName) : \PFC\AdminBundle\Utils\importXml

Set fileName

### Parameters

| Name | Type | Description |
|------|------|-------------|
| **$fileName** | string | |

### Returns

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\importXml | |

**setPathDir**(string $pathDir) : \PFC\AdminBundle\Utils\importXml

Set pathDir

### Parameters

| Name | Type | Description |
|------|------|-------------|
| **$pathDir** | string | |

### Returns

| Type | Description |
|------|-------------|
| \PFC\AdminBundle\Utils\importXml | |

# ModelBundle/Controller/DefaultController.php

## \PFC\ModelBundle\Controller\D

**Package:** global

**Parent(s)**

  \Symfony\Bundle\FrameworkBundle\Controller\Controller

### Methods

**indexAction**( $name) : void

#### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$name** | | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/DataFixtures/ORM/LoadPageData.php

Show: **inherited**

## \PFC\ModelBundle\DataFixtures\O

**Package:** global

» Data Fixtures que crea las páginas y las guarda en la base de datos

**Implements**

**Parent(s)**

  \Doctrine\Common\DataFixtures\AbstractFixture

## Properties

## Methods

### getOrder() : void

### load( $manager) : void

» Método que establece el orden de la carga todos los datos de prueba y de las páginas.

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$manager** | | |

**Details**

**Package**

  PFC\ModelBundle\DataFixtures\ORM;

**Version**

  1.0

### setContainer( $container = null) : void

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$container** | | |

Documentation was generated by phpDocumentor 2.5.0 .

# ModelBundle/DataFixtures/ORM/LoadSubjectsData.php

## \PFC\ModelBundle\DataFixtures\ORM

**Package:** global
» Data Fixtures que crea las asignaturas, categorías, y subcategorías de prueba, y relación entre ellas
**Implements**
**Parent(s)**
    \Doctrine\Common\DataFixtures\AbstractFixture

## Properties
## Methods

---

**getOrder**() : void

---

**load**( $manager) : void

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$manager** | | |

---

**setContainer**( $container = null) : void

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$container** | | |

---

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/DependencyInjection/Configuration.php

## \PFC\ModelBundle\DependencyInjec

**Package:** global

This is the class that validates and merges configuration from your app/config files

To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html#cookbook-bundles-extension-config-class}

**Implements**

### Methods

**getConfigTreeBuilder**() : void

{@inheritDoc}

# ModelBundle/DependencyInjection/ModelExtension.php

## \PFC\ModelBundle\DependencyInjection

**Package:** global

This is the class that loads and manages your bundle configuration

To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html}

**Parent(s)**

   \Symfony\Component\HttpKernel\DependencyInjection\Extension

## Methods

---

**load**( $configs, $container) : void

{@inheritDoc}

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$configs** | | |
| **$container** | | |

---

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/Entity/Answer.php

Show:

## \PFC\ModelBundle\Entity\Answer

**Package:** global
» Entidad Answers
**ORM\Entity**

**ORM\Table**
  (name="Answers")
**UniqueEntity**
  ("id")

## Properties
## Methods

### getContent() : string

Get content

#### Returns

| Type | Description |
| --- | --- |
| string | |

### getId() : integer

Get id

#### Returns

| Type | Description |
| --- | --- |
| integer | |

### getQuestion() : \PFC\ModelBundle\Entity\Question

Get question

#### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

### getTolerance() : float

Get tolerance

**Returns**

| Type | Description |
|------|-------------|
| float | |

**getValue() : float**

Get value

**Returns**

| Type | Description |
|------|-------------|
| float | |

**setContent(string $content) : \PFC\ModelBundle\Entity\Answer**

Set content

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| $content | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Answer | |

**setQuestion(\PFC\ModelBundle\Entity\Question $question = null) : \PFC\ModelBundle\Entity\Answer**

Set question

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| $question | \PFC\ModelBundle\Entity\Question | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Answer | |

**setTolerance( $value) : \PFC\ModelBundle\Entity\Answer**

Set tolerance

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$value** | | |

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Answer | |

---

**setValue**(float $value) : \PFC\ModelBundle\Entity\Answer

Set value

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$value** | float | |

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Answer | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/Entity/Category.php

Show: **inherited**

## \PFC\ModelBundle\Entity\Category

**Package:** global
» Entidad Category
**ORM\Entity**

**ORM\Table**
   (name="Categories")
**UniqueEntity**
   ("slug")

### Properties
### Methods

---

#### getDescription() : string

Get description

**Returns**

| Type | Description |
|------|-------------|
| string | |

---

#### getId() : integer

Get id

**Returns**

| Type | Description |
|------|-------------|
| integer | |

---

#### getName() : string

Get name

**Returns**

| Type | Description |
|------|-------------|
| string | |

---

#### getPage() : \PFC\ModelBundle\Entity\Page

Get page

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Page | |

**getSlug() : string**

Get slug

**Returns**

| Type | Description |
|---|---|
| string | |

**getSubject() : \PFC\ModelBundle\Entity\Subject**

Get subject

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Subject | |

**getTitle() : string**

Get title

**Returns**

| Type | Description |
|---|---|
| string | |

**setDescription(string $description) : \PFC\ModelBundle\Entity\Category**

Set description

**Parameters**

| Name | Type | Description |
|---|---|---|
| $description | string | |

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Category | |

**setName(string $name) : \PFC\ModelBundle\Entity\Category**

Set name

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$name** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

**setPage**(\PFC\ModelBundle\Entity\Page $page = null) : \PFC\ModelBundle\Entity\Category

Set page

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$page** | \PFC\ModelBundle\Entity\Page | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

**setSlug**(string $slug) : \PFC\ModelBundle\Entity\Category

Set slug

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$slug** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

**setSubject**(\PFC\ModelBundle\Entity\Subject $subject = null) : \PFC\ModelBundle\Entity\Category

Set subject

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$subject** | \PFC\ModelBundle\Entity\Subject | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

### setTitle(string $title) : \PFC\ModelBundle\Entity\Category

Set title

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$title** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/Entity/Page.php

Show:

## \PFC\ModelBundle\Entity\Page

**Package:** global

» Entidad Page

**ORM\Entity**

   (repositoryClass="PFC\ModelBundle\Repositories\PageRepository")

**ORM\Table**

   (name="Pages", indexes = { @ORM\Index(name="IDX_TYPE", columns={"type"}),
   @ORM\Index(name="IDX_ACTIVE", columns={"active"}), @ORM\Index(name="IDX_PRIORITY",
   columns={"priority"}) }, uniqueConstraints={@ORM\UniqueConstraint(name="UNIQ_SLUG",
   columns={"slug"})})

## Properties
## Methods

### **getActive**() : boolean

Get active

#### Returns

| Type | Description |
| --- | --- |
| boolean | |

### **getId**() : integer

Get id

#### Returns

| Type | Description |
| --- | --- |
| integer | |

### **getPriority**() : integer

Get priority

#### Returns

| Type | Description |
| --- | --- |
| integer | |

### **getSlug**() : string

Get slug

**Returns**

| Type | Description |
| --- | --- |
| string | |

### getTitle() : string

Get title

**Returns**

| Type | Description |
| --- | --- |
| string | |

### getType() : integer

Get type

**Returns**

| Type | Description |
| --- | --- |
| integer | |

### setActive(boolean $active) : \PFC\ModelBundle\Entity\Page

Set active

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$active** | boolean | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

### setPriority(integer $priority) : \PFC\ModelBundle\Entity\Page

Set priority

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$priority** | integer | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

### setSlug(string $slug) : \PFC\ModelBundle\Entity\Page

Set slug

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$slug** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

### setTitle(string $title) : \PFC\ModelBundle\Entity\Page

Set title

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$title** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

### setTranslatableLocale(string $locale) : void

Set TranslatableLocale

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$locale** | string | |

### setType(integer $type) : \PFC\ModelBundle\Entity\Page

Set type

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$type** | integer | |

## Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/Entity/PageUrl.php

## \PFC\ModelBundle\Entity\PageUrl

**Package:** global
» Entidad PageUrl
**ORM\Entity**

**ORM\Table**
　(name="PagesUrl", indexes = {@ORM\Index(name="IDX_LANG", columns={"lang"})})

### Properties
### Methods

---

**getId**() : integer

Get id

**Returns**

| Type | Description |
| --- | --- |
| integer | |

---

**getLang**() : string

Get lang

**Returns**

| Type | Description |
| --- | --- |
| string | |

---

**getPage**() : \PFC\ModelBundle\Entity\Page

Get page

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

---

**getUrl**() : string

Get url

**Returns**

| Type | Description |
|------|-------------|
| string | |

### setLang(string $lang) : \PFC\ModelBundle\Entity\PageUrl

Set lang

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$lang** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\PageUrl | |

### setPage(\PFC\ModelBundle\Entity\Page $page = null) : \PFC\ModelBundle\Entity\PageUrl

Set page

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$page** | \PFC\ModelBundle\Entity\Page | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\PageUrl | |

### setUrl(string $url) : \PFC\ModelBundle\Entity\PageUrl

Set url

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$url** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\PageUrl | |

# ModelBundle/Entity/Question.php

## \PFC\ModelBundle\Entity\Question

**Package:** global
» Entidad Question
**ORM\Entity**

**ORM\Entity**
   (repositoryClass="PFC\ModelBundle\Repositories\QuestionRepository")
**ORM\Table**
   (name="Questions")
**UniqueEntity**
   ("id")

### Properties
### Methods

---

**getCategory() : \PFC\ModelBundle\Entity\Category**

Get category

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

---

**getDescription() : string**

Get description

**Returns**

| Type | Description |
|------|-------------|
| string | |

---

**getId() : integer**

Get id

**Returns**

| Type | Description |
|------|-------------|
| integer | |

---

**getLevel() : float**

Get level

## Returns

| Type | Description |
| --- | --- |
| float | |

**getNumAnswers**() : integer

Get numAnswers

## Returns

| Type | Description |
| --- | --- |
| integer | |

**getNumCorrectAnswers**() : integer

Get numCorrectAnswers

## Returns

| Type | Description |
| --- | --- |
| integer | |

**getPenalty**() : float

Get penalty

## Returns

| Type | Description |
| --- | --- |
| float | |

**getSingle**() : boolean

Get single

## Returns

| Type | Description |
| --- | --- |
| boolean | |

**getSubcategory**() : \PFC\ModelBundle\Entity\Subcategory

Get subcategory

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Subcategory | |

**getSubject() : \PFC\ModelBundle\Entity\Subject**

Get subject

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Subject | |

**getTitle() : string**

Get title

**Returns**

| Type | Description |
| --- | --- |
| string | |

**getType() : integer**

Get type

**Returns**

| Type | Description |
| --- | --- |
| integer | |

**setCategory(\PFC\ModelBundle\Entity\Category $category = null) : \PFC\ModelBundle\Entity\Question**

Set category

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$category** | \PFC\ModelBundle\Entity\Category | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

**setDescription**(string $description) : \PFC\ModelBundle\Entity\Question

Set description

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$description** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setLevel**(float $level) : \PFC\ModelBundle\Entity\Question

Set level

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$level** | float | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setNumAnswers**(integer $numAnswers) : \PFC\ModelBundle\Entity\Question

Set numAnswers

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$numAnswers** | integer | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setNumCorrectAnswers**(integer $numCorrectAnswers) : \PFC\ModelBundle\Entity\Question

Set numCorrectAnswers

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$numCorrectAnswers** | integer | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setPenalty(float $penalty) : \PFC\ModelBundle\Entity\Question**

Set penalty

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$penalty** | float | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setSingle(boolean $single) : \PFC\ModelBundle\Entity\Question**

Set single

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$single** | boolean | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setSubcategory(\PFC\ModelBundle\Entity\Subcategory $subcategory = null) : \PFC\ModelBundle\Entity\Question**

Set subcategory

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subcategory** | \PFC\ModelBundle\Entity\Subcategory | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setSubject**(\PFC\ModelBundle\Entity\Subject $subject = null) : \PFC\ModelBundle\Entity\Question

Set subject

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$subject** | \PFC\ModelBundle\Entity\Subject | |

### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setTitle**(string $title) : \PFC\ModelBundle\Entity\Question

Set title

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$title** | string | |

### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

---

**setType**(integer $type) : \PFC\ModelBundle\Entity\Question

Set type

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| **$type** | integer | |

### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Question | |

# ModelBundle/Entity/Subcategory.php

» Entidad Subcategory

1.0 @date 2013-10-30

## \PFC\ModelBundle\Entity\Subcategory

**Package:** global
Subcategory
**ORM\Entity**

**ORM\Table**
   (name="Subcategories")
**UniqueEntity**
   ("slug")

### Properties
### Methods

---

**getCategory() : \PFC\ModelBundle\Entity\Category**

Get category

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

---

**getDescription() : string**

Get description

**Returns**

| Type | Description |
| --- | --- |
| string | |

---

**getId() : integer**

Get id

**Returns**

| Type | Description |
| --- | --- |
| integer | |

## getName() : string

Get name

### Returns

| Type | Description |
| --- | --- |
| string | |

## getPage() : \PFC\ModelBundle\Entity\Page

Get page

### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

## getSlug() : string

Get slug

### Returns

| Type | Description |
| --- | --- |
| string | |

## getSubject() : \PFC\ModelBundle\Entity\Subject

Get subject

### Returns

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Subject | |

## getTitle() : string

Get title

### Returns

| Type | Description |
| --- | --- |
| string | |

## setCategory(\PFC\ModelBundle\Entity\Category $category = null) : \PFC\ModelBundle\Entity\Subcategory

Set category

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$category** | \PFC\ModelBundle\Entity\Category | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Subcategory | |

**setDescription**(string $description) : \PFC\ModelBundle\Entity\Category

Set description

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$description** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

**setName**(string $name) : \PFC\ModelBundle\Entity\Category

Set name

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$name** | string | |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\ModelBundle\Entity\Category | |

**setPage**(\PFC\ModelBundle\Entity\Page $page = null) : \PFC\ModelBundle\Entity\Category

Set page

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$page** | \PFC\ModelBundle\Entity\Page | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

### setSlug(string $slug) : \PFC\ModelBundle\Entity\Category

Set slug

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$slug** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

### setSubject(\PFC\ModelBundle\Entity\Subject $subject = null) : \PFC\ModelBundle\Entity\Category

Set subject

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$subject** | \PFC\ModelBundle\Entity\Subject | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

### setTitle(string $title) : \PFC\ModelBundle\Entity\Category

Set title

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$title** | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Category | |

# ModelBundle/Entity/Subject.php

Show: **inherited**

» Entidad Subject

1.0 @date 2013-10-30

## \PFC\ModelBundle\Entity\Subject

**Package:** global
Subject
**ORM\Entity**

**ORM\Table**
   (name="Subjects")
**UniqueEntity**
   ("slug")

## Properties
## Methods

### getDescription() : string

Get description

**Returns**

| Type | Description |
| --- | --- |
| string | |

### getId() : integer

Get id

**Returns**

| Type | Description |
| --- | --- |
| integer | |

### getName() : string

Get name

**Returns**

| Type | Description |
| --- | --- |
| string | |

### getPage() : \PFC\ModelBundle\Entity\Page

Get page

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Page | |

---

### getSlug() : string

Get slug

**Returns**

| Type | Description |
| --- | --- |
| string | |

---

### setDescription(string $description) : \PFC\ModelBundle\Entity\Subject

Set description

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| $description | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Subject | |

---

### setName(string $name) : \PFC\ModelBundle\Entity\Subject

Set name

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| $name | string | |

**Returns**

| Type | Description |
| --- | --- |
| \PFC\ModelBundle\Entity\Subject | |

---

### setPage(\PFC\ModelBundle\Entity\Page $page = null) : \PFC\ModelBundle\Entity\Subject

Set page

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$page** | \PFC\ModelBundle\Entity\Page | |

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Subject | |

**setSlug**(string $slug) : \PFC\ModelBundle\Entity\Subject

Set slug

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$slug** | string | |

**Returns**

| Type | Description |
|---|---|
| \PFC\ModelBundle\Entity\Subject | |

Documentation was generated by <u>phpDocumentor 2.5.0</u> .

# ModelBundle/ModelBundle.php

Show:

## \PFC\ModelBundle\ModelBundle

**Package:** global

**Parent(s)**

\Symfony\Component\HttpKernel\Bundle\Bundle

Documentation was generated by phpDocumentor 2.5.0 .

# ModelBundle/Repositories/PageRepository.php

» Repositorio PageRepository

@date 2013-08-20

## \PFC\ModelBundle\Repositories\PageRepository

**Package:** global

**Parent(s)**

   \Doctrine\ORM\EntityRepository

### Methods

**getBreadCrumb**(\PFC\ModelBundle\Repositories\Page $page, $lang) : void

» Obtiene la estructura de páginas sobre la página actual (breadcrumbs)

@param string $lang Idioma

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$page** | \PFC\ModelBundle\Repositories\Page | *Página @return array $breadCrumb Array de páginas que forman el breadcrumb* |
| **$lang** | | |

Documentation was generated by [phpDocumentor 2.5.0](#) .

# ModelBundle/Repositories/QuestionRepository.php

» Repositorio QuestionRepository

## \PFC\ModelBundle\Repositories\QuestionRepository

**Package:** global
**Parent(s)**
    \Doctrine\ORM\EntityRepository

## Methods

---

**findByTypeIdLevelCount**( $selector,  $id,  $level = 0,  $numQuestions = '') : array

» Método para que devuleve un array de preguntas, con los filtros indicados

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| $selector | | String Filtro de consulta [subject, category o subcategory] |
| $id | | Integer Id del tipo indicado en selector |
| $level | | Integer Filtro del nivel de preguntas a consultar. Level = 0 indica no discriminar por nivel |
| $numQuestions | | String Número de preguntas a exportar |

### Returns

| Type | Description |
| --- | --- |
| array | Array Array de preguntas |

**Details**
**Package**
    PFC\ModelBundle\Repositories;
**Version**
    1.0

---

**getLevelQuestions**(string $category) : array

» Obtiene los de niveles disponibles de una categoría de cuestiones

### Parameters

| Name | Type | Description |
| --- | --- | --- |
| $category | string | Categoria |

### Returns

| Type | Description |
| --- | --- |
| array | $levels Array con el nº de niveles del que existen cuestiones [1,4,8] |

Documentation was generated by phpDocumentor 2.5.0 .

# ModelBundle/Tests/Controller/DefaultControllerTest.php

## \PFC\ModelBundle\Tests\Controller\De...

**Package:** global

**Parent(s)**

\Symfony\Bundle\FrameworkBundle\Test\WebTestCase

### Methods

**testIndex**() : void

Documentation was generated by phpDocumentor 2.5.0 .

# WebBundle/Controller/BaseController.php

## \PFC\WebBundle\Controller\Base

**Package:** global

» BaseController es la clase de la que deben heredar todos los controladores.
Define comportamientos y métodos comunes para todos ellos

**Parent(s)**

   \Symfony\Bundle\FrameworkBundle\Controller\Controller

**Children**

   \PFC\WebBundle\Controller\PagesController

## Properties
## Methods

---

**setContainer**( $container = null) : void

» Método para dar valor a las propiedades de la clase. Nos evita repetir este código en cada controlador que lo necesite

@param ContainerInterface $container

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$container** | | |

---

Documentation was generated by [phpDocumentor 2.5.0](#) .

# WebBundle/Controller/PagesController.php

## \PFC\WebBundle\Controller\Page

Classes
\PFC\WebBundle\Controller\PagesController

**Package:** global

» Controlador para la renderización de páginas y la ejecución de los cuestionarios para el usuario. Define comportamientos y métodos comunes para todos ellos

**Parent(s)**

\PFC\WebBundle\Controller\BaseController <
\Symfony\Bundle\FrameworkBundle\Controller\Controller

## Properties
## Methods

**TestPostAction**(string $subjectSlug, integer $level, \Symfony\Component\HttpFoundation\Request $request) : \PFC\WebBundle\Controller\Render

» Envío de los resultados de un cuestionario, y almacenamiento de las evaluaciones de las preguntas.

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$subjectSlug** | string | *slug de la asignatura* |
| **$level** | integer | *Nivel del cuestionario. 0 indica aleatorio de cualquier nivel* |
| **$request** | \Symfony\Component\HttpFoundation\Request | *Objeto que contiene el formulario enviado* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\WebBundle\Controller\Render | Devuelve la vista WebBundle:Pages:test-post.html.twig con el resultado del cuestionario. |

**Details**
**Package**
  PFC\WebBundle\Controller;
**Version**
  1.0

**indexAction**() : void

**levelSelectionAction**(string $subjectSlug, string $categorySlug = '', string $subCategorySlug = '') : \PFC\WebBundle\Controller\Render

» Renderización de la página con la selección de nivel del cuestionario.

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$subjectSlug** | string | *slug de la asignatura* |
| **$categorySlug** | string | *opcional, slug de la categoría* |
| **$subCategorySlug** | string | *opcional, slug de la subcategoría* |

**Returns**

| Type | Description |
|---|---|
| \PFC\WebBundle\Controller\Render | Devuelve la vista WebBundle:Pages:level-selection.html.twig |

**Details**
**Package**
   PFC\WebBundle\Controller;
**Version**
   1.0

---

**setContainer**( $container = null) : void

» Método para dar valor a las propiedades de la clase. Nos evita repetir este código en cada controlador que lo necesite

@param ContainerInterface $container
**Parameters**

| Name | Type | Description |
|---|---|---|
| **$container** | | |

---

**subjectAction**(string $subjectSlug) : \PFC\WebBundle\Controller\Render

» Método que realiza las tareas del controlador después de seleccionar una aisgnatura.

**Parameters**

| Name | Type | Description |
|---|---|---|
| **$subjectSlug** | string | *slug de la subcategoría* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\WebBundle\Controller\Render | Devuelve la vista WebBundle:Pages:subject.html.twig que muestra la asignatura, sus categorías y sus subcategorías, con los cuestionarios a realizar. |

**Details**
**Package**
  PFC\WebBundle\Controller;
**Version**
  1.0

**testAction**( $request, string $subjectSlug, string $categorySlug = '', string $subCategorySlug = '') : \PFC\WebBundle\Controller\Render

» Renderización de la página con la lista de preguntas del nivel, asignatura, categoría o subcategoría seleccionadas

**Parameters**

| Name | Type | Description |
|------|------|-------------|
| **$request** | | |
| **$subjectSlug** | string | *slug de la asignatura* |
| **$categorySlug** | string | *opcional, slug de la categoría* |
| **$subCategorySlug** | string | *opcional, slug de la subcategoría* |

**Returns**

| Type | Description |
|------|-------------|
| \PFC\WebBundle\Controller\Render | Devuelve la vista WebBundle:Pages:test.html.twig |

**Details**
**Package**
  PFC\WebBundle\Controller;
**Version**
  1.0

# WebBundle/DependencyInjection/Configuration.php

## \PFC\WebBundle\DependencyInjec

**Package:** global

This is the class that validates and merges configuration from your app/config files
To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html#cookbook-bundles-extension-config-class}

**Implements**

## Methods

**getConfigTreeBuilder**() : void

{@inheritDoc}

Documentation was generated by phpDocumentor 2.5.0 .

## \PFC\WebBundle\DependencyInject...

Classes
\PFC\WebBundle\DependencyInjection\WebExtension

**Package:** global

This is the class that loads and manages your bundle configuration

To learn more see {@link http://symfony.com/doc/current/cookbook/bundles/extension.html}

**Parent(s)**

\Symfony\Component\HttpKernel\DependencyInjection\Extension

## Methods

**load**( $configs, $container) : void

{@inheritDoc}

**Parameters**

| Name | Type | Description |
| --- | --- | --- |
| **$configs** | | |
| **$container** | | |

Documentation was generated by phpDocumentor 2.5.0 .

# WebBundle/Tests/Controller/DefaultControllerTest.php

Show: **inherited**

## \PFC\WebBundle\Tests\Controller\De

**Package:** global

**Parent(s)**

   \Symfony\Bundle\FrameworkBundle\Test\WebTestCase

## Methods

**testIndex**() : void

Documentation was generated by phpDocumentor 2.5.0 .

# WebBundle/WebBundle.php

## \PFC\WebBundle\WebBundle

**Package:** global

**Parent(s)**

\Symfony\Component\HttpKernel\Bundle\Bundle

Documentation was generated by phpDocumentor 2.5.0 .