

Examen del Curso (1, 2, 3) - EXAMEN ATC - Agregar contacto - [preguntas]

? Descripción

Obtiene todas las preguntas de un examen (Quiz) registrado en el ERP, junto con sus opciones, respuestas correctas y metadatos.

El servicio se encarga de:

- Consultar el examen (*Quiz*) en la base ERP.
- Traer todas las preguntas asociadas.
- Traer todas las opciones de cada pregunta.
- Reorganizar los resultados del SQL (que vienen aplanados) en una estructura jerárquica:
 - **Pregunta**
 - **Opciones**
- Convertir el contenido HTML de la pregunta a texto plano.
- Detectar si la pregunta incluye una **imagen** embebida.
- Preparar la respuesta final en un formato limpio y listo para usar en la app.

? Endpoint

POST `/preguntas`

? Request Body

```
{  
  "examen": "QUIZ-0001"  
}
```

Parámetros

Campo	Tipo	Obligatorio	Descripción
examen	string	✓	ID del cuestionario (Quiz.name)

? Seguridad

Este servicio requiere acceso a la API interna del ERP a través de:

- `dbErp()` → Consulta SQL interna
No usa token del usuario final. La autenticación es interna del servidor.

? Flujo del Servicio (Explicación Real)

1. Construye consulta SQL

El servicio arma un query que une:

- `tabQuiz`
- `tabQuiz Question`
- `tabQuestion`
- `tabOptions`

De esta manera obtiene:

- Datos de la **pregunta**
- Datos de **cada opción**
- Datos del examen

2. Envía la consulta

Se ejecuta: `POST method/send-query-database`

3. Reorganiza la data

El ERP devuelve una fila por cada opción.

El servicio agrupa todo por: `question.name`

Generando esta estructura:

```
[
  {
    question: "...",
    question_link: "...",
    options: [
      { option: "...", is_correct: 0/1 }
    ]
  }
]
```

4. Convierte HTML ? texto

Las preguntas vienen como HTML.

El servicio:

- Analiza el HTML con `html_to_obj()`
- Extrae texto plano (`strip_tags`)
- Busca imágenes incrustadas y genera la URL completa.

5. Construye la respuesta final

Para cada pregunta:

- `name`
- `question` (texto plano)
- `imagen` (si existe)
- `options` (lista de alternativas)

? Response 200 – Ejemplo

```
{
  "valor": true,
  "data": [
    {
      "name": "PREG-001",
      "owner": "admin@example",
      "creation": "2025-01-01",
      "modified": "2025-01-01",
      "modified_by": "admin@example",
      "idx": 1,
      "docstatus": 0,
      "question": "¿Cuál es la capital de Francia?",
      "question_type": "Opciones",
      "question_link": "PREG-001",
      "options": [
        {
          "name": "OPT-001",
          "option": "Madrid",
          "is_correct": 0,
          "doctype": "Options"
        },
        {
          "name": "OPT-002",
          "option": "París",
          "is_correct": 1,
          "doctype": "Options"
        }
      ]
    },
    {
      "imagen": ""
    }
  ]
}
```

? Posibles Errores

1. Examen no encontrado

```
{
  "valor": false,
  "msn": "No existe información del examen",
  "data": []
}
```

2. Error en consulta SQL

```
{
  "valor": false,
  "msn": "Error al obtener preguntas",
  "data": "<mensaje_de_error>"
}
```

3. Examen sin preguntas

```
{
  "valor": true,
  "data": []
}
```

? Schemas Usados

? Quiz

Campo	Tipo
name	string

? Question

Campo	Tipo
name	string
question	HTML
question_type	string

? Options

Campo	Tipo
name	string

Campo	Tipo
option	string
is_correct	int (0 o 1)
parent	question.name

? Lógica en pseudo-código

```
examen = request.examen

rows = SQL:
  SELECT preguntas + opciones
  FROM Quiz JOIN Quiz Question JOIN Question JOIN Options
  WHERE qz.name = examen

grouped = reorganizarPorPregunta(rows)

foreach pregunta in grouped:
  procesar HTML
  extraer texto e imagen

return JSON(valor=true, data=preguntasProcesadas)
```

Revisión #3

Creado 2025-11-25 17:51:12 -05 por Elian

Actualizado 2025-11-26 11:06:58 -05 por Elian