

**Acceso a API MATFER para consulta al repositorio de preguntas MATFERLINE:**

La obtención de preguntas se basa en el estándar API REST, si bien en este caso no haremos uso de todos los métodos comunes de esta tecnología, solo de aquellos que nos son útiles.

El proceso consiste en una autenticación previa, mediante una llamada a un método API, parametrizado con las credenciales suministradas por Ediciones MATFER, que nos devuelve un token, el cual será enviado al método API que devuelve las preguntas en formato JSON.

Métodos disponibles:

**Auth:**

URL: [https://matferline.com/api\\_matfer/auth/](https://matferline.com/api_matfer/auth/)

Method: POST

Headers: Content-Type: application/x-www-form-urlencoded

Parámetros:

- username
- password

Este método devuleve los posibles valores (siempre en formato JSON):

*Error: Acceso al método sin incluir alguno de los parámetros:*

```
{  
  status: "error",  
  code: 400,  
  message: "Acceso Denegado"  
}
```

*Error: Login Incorrecto (usuario o clave incorrectos):*

```
{  
  status: "error",  
  code: 400,  
  message: "Login Incorrecto"  
}
```

*Error: No se ha podido generar el token (errores internos tipificados):*

```
{  
  status: "error",  
  code: 400,  
  message: "Error al generar el Token Autorizado (n)"  
}
```

*Ok: Usuario autenticado y token generado:*

```
{  
  status: "ok",  
  code: 200,  
  message: "Usuario autorizado correctamente",  
  token: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"  
}
```

Una vez devuelto el token tras la autenticación correcta, podremos invocar al siguiente método.

**Preguntas:**

URL: [https://matferline.com/api\\_matfer/preguntas/](https://matferline.com/api_matfer/preguntas/)

Method: GET

Headers: Authorization: Bearer `xxxxxxxxxxxxxx` (el token devuelto en el método anterior)

Parámetros: No necesita parámetros

Este método devuleve los posibles valores (siempre en formato JSON):

*Error: No se ha encontrado una cabecera que incluya el token:*

```
{  
    status: "error",  
    code: 400,  
    message: "No se ha encontrado el Token autorizado"  
}
```

*Error: Se ha encontrado el token, pero no en el formato adecuado (Bearer token):*

```
{  
    status: "error",  
    code: 400,  
    message: "Formato de Token autorizado incorrecto"  
}
```

*Ok: Listado de preguntas generado correctamente:*

```
{  
    status: "ok",  
    code: 200,  
    message: "Listado de Preguntas MATFERLINE. Ediciones MATFER",  
    listado: JSON con todas las preguntas(*)  
}
```

En la siguiente página se muestra un ejemplo del formato en que se presentan las preguntas devueltas por `JSON con todas las preguntas(*)`, y que corresponde con este esquema:

```
{ /* contenedor principal */  
  
    preguntas: [ /* Array que contiene todas las preguntas devueltas */  
        { /* contenedor de cada una de las preguntas */  
            Pregunta: texto del enunciado de la pregunta,  
            imagen: URL de la imagen, si existe,  
            respuestas: [ /* Array con todas las respuestas de una pregunta */  
                texto de la respuesta  
            ]  
            correcta: Indica la respuesta correcta del array anterior,  
            permisos: [ /* Array con todos los permisos de una pregunta */  
                id: Identificador del permiso (número secuencial)  
                nombre: Nombre del permiso  
            ]  
        }  
    ]  
}
```

```
        ]  
    }  
}
```

Esta sería un ejemplo real para *JSON con todas las preguntas (\*)*, con una sola pregunta:

```
{  
  
    preguntas:  
    [  
  
        {  
  
            pregunta: "Si un grupo de peatones dirigidos por una persona  
circulan por una v&iacute;a de fuera de poblado, &iquest;deben  
se&ntilde;alizar su presencia?",  
            imagen: "URL completa a la imagen, si existe. Si no, vacío",  
            respuestas:  
  
                [  
  
                    "S&iacute;, con una se&ntilde;al de STOP incorporada a una  
paleta.",  
                    "S&iacute;, con las luces necesarias para precisar su  
situaci&oacute;n y dimensiones, las cuales ser&aacute;n de  
color blanco o amarillo hacia delante y rojo hacia  
atr&aacute;s.",  
                    "No, no es necesario."  
  
                ],  
  
            correcta: 2,  
            permisos:  
                [  
  
                    {  
  
                        id: "1",  
                        nombre: "PERMISO B (Te&oacute;rica Com&uacute;n)"  
  
                    }  
  
                ]  
  
        }  
  
    ]  
}
```

Como se podrá observar, los textos del enunciado y las respuestas incluyen caracteres HTML.

Si fuese necesario, habría que “parsear” dichos textos, antes de incluirlos en base de datos, para convertir las entidades HTML a sus correspondientes caracteres naturales (acentos, eñes, comillas, etc.). Por ejemplo, en PHP bastaría con usar *html\_entity\_decode*, o cualquier comando similar en otros lenguajes de programación.

Por último, se incluye código de ejemplo en PHP sobre como invocar los dós métodos necesarios:

```
<?PHP

$urlAuth="https://matferline.com/api_matfer/auth/";

$login = "*****";
$password="*****";

$ch1 = curl_init();

curl_setopt($ch1, CURLOPT_URL, $urlAuth);

curl_setopt($ch1, CURLOPT_RETURNTRANSFER, 1);

curl_setopt($ch1, CURLOPT_CUSTOMREQUEST, 'POST');

curl_setopt($ch1, CURLOPT_POSTFIELDS, "username=$login&password=$password");

$headers = array();

$headers[] = 'Content-Type: application/x-www-form-urlencoded';

curl_setopt($ch1, CURLOPT_HTTPHEADER, $headers);

$result = curl_exec($ch1);

if (curl_errno($ch1)) {

    echo 'Error: ' . curl_error($ch1);

} else {

    $accessData = json_decode($result,true);

    if (isset($accessData["status"])){

        if ($accessData["status"] == "ok") {

            $accessToken = $accessData["token"];

            $ch2 = curl_init();

            $urlApi="https://matferline.com/api_matfer/preguntas/";

            curl_setopt($ch2, CURLOPT_URL, $urlApi);

            curl_setopt($ch2, CURLOPT_RETURNTRANSFER, 1);

            curl_setopt($ch2, CURLOPT_CUSTOMREQUEST, 'GET');

            $headers = array();

            $headers[] = 'Authorization: Bearer ' . $accessToken;

            curl_setopt($ch2, CURLOPT_HTTPHEADER, $headers);

            $result = curl_exec($ch2);

            if (curl_errno($ch2)) {

                echo 'Error: ' . curl_error($ch2);

            } else {

                echo utf8_decode($result);

            }

            curl_close($ch2);

        } else {

            echo $result;

        }

    }

}

echo $result;
```

```
    }
} else {
    $resultData = array (
        "status" => "error",
        "code" => 400,
        "message" => "Error en el proceso de Login"
    );
echo json_encode($resultData);
}
}

curl_close($ch1);

?>
```

**NOTAS:**

- El token devuelto es único para cada llamada al método auth, si bien, dicho token tiene una caducidad por motivos de seguridad en entornos de conexiones a Internet problemáticas.
- Se ha incluido un único método que devuelve todas las preguntas de los permisos B, A1 y A2 y AM tipificadas por permiso, por lo que el tiempo de procesamiento de la consulta puede variar según la calidad de la conexión a Internet, siendo de unos 5 segundos aprox., en condiciones normales.