

Artículo

[Kurro Lopez](#) · Mar 16, 2020 Lectura de 5 min

## RESTForms - REST API para tus clases. Parte 2: Consultas

En el [primer artículo](#) empecé a analizar RESTForms - REST API para tus clases persistentes y hablamos sobre las características básicas. Ahora, me gustaría hablar sobre las características avanzadas, principalmente las capacidades de consultas:

- Consultas básicas
- Consultar argumentos
- Consultas personalizadas

### Consultas

Las consultas pueden obtener segmentos de datos, criterios en criterios arbitrarios. Hay dos tipos de consulta en RESTForms:

- Las consultas básicas funcionan para todas las clases RESTForms una vez definidas y solo difieren en la lista de campos
- Las consultas personalizadas solo funcionan para las clases en las que están especificadas y disponibles, pero el desarrollador tiene acceso completo al texto de la consulta

### Consultas básicas

Definido una vez e inmediatamente disponible para todas o algunas clases. Algunas consultas básicas están definidas por el sistema, los desarrolladores pueden agregar más, y todas estas consultas solo definen la lista de campos SELECT. Todo lo demás (filtrado, paginación, etc. se realiza mediante RESTForms)

Invocar `form/objects/:class/:query`, para ejecutar una consulta simple. El segundo parámetro `:query` determina el nombre de la consulta: el contenido de la consulta entre SELECT y FROM. Estos son los tipos de consulta predeterminados:

Query	Descripción
all	Toda la información
info	displayName e id
infoclass	displayName, id, class
count	número de filas

Por ejemplo, para obtener información básica sobre los objetos `Form.Test.Person`, podemos ejecutar consultas de infoclasses:

```
form/objects/Form.Test.Person/infoclass
```

```
{"children": [  
  {"_id": "1", "displayName": "Alice", "_class": "Form.Test.Person"},  
  {"_id": "2", "displayName": "Charlie", "_class": "Form.Test.Person"},  
  {"_id": "3", "displayName": "William", "_class": "Form.Test.Person"}  
]
```

```
1}
```

RESTForms busca una consulta denominada `myq` en los siguientes lugares (hasta el primer éxito):

1. Método de clase `queryMYQ` en su clase de formulario
2. Parameter `MYQ` in your queries class
3. Método de clase `queryMYQ` en su clase de consultas
4. Parámetro `MYQ` en la clase `Form.REST.Objects`
5. Método de clase `queryMYQ` en la clase `Form.REST.Objects`

Puede definir su propia clase de consultas (para las entradas 2, 3 en la lista anterior): clase especial que contiene definiciones de consulta disponibles para todas las clases. Para definir su propia consulta llamada `myq` allí:

1. (Una vez) Definir una clase `YourClassName`
2. Definir allí un parámetro `MYQ` o método de clase `queryMYQ`. El parámetro tiene prioridad sobre el método.
3. El método o el parámetro deben devolver la parte de la consulta SQL entre `SELECT` y `FROM`
4. (Una vez) Ejecuta en un terminal: `Do ##class(For.Settings).setSetting("queryclass", YourClassName)`

La firma del Method es:

```
ClassMethod queryMYQ(class As %String) As %String
```

También puede definir una consulta específica de clase. Para definir su propia consulta de clase denominada `myq`:

1. Define un método de clase `queryMYQ` en tu clase de formulario
2. La firma del método es: `ClassMethod queryMYQ() As %String`
3. El método debe devolver la parte de la consulta SQL entre `SELECT` y `FROM`

## Argumentos URL

Puede suministrar filtros y otros parámetros en URL. Todos los argumentos son opcionales..

Argumento	Valor de Ejemplo	Descripción
<code>size</code>	2	tamaño de la página
<code>page</code>	1	número de página
<code>filter</code>	Value+contains+W	WHERE cláusula
<code>orderby</code>	Value+desc	ORDER BY cláusula
<code>collation</code>	UPPER	COLLATION cláusula
<code>nocount</code>	1	Eliminar el recuento de filas (ace consulta)

Aquí hay información sobre estos argumentos.

## ORDER BY cláusula

Cambia el orden de los resultados. El valor puede ser: `Column` o `Column+desc`. `Column` es una column de la tabla sql o un número de columna.

## WHERE cláusula

Condición de filtro en un formato: Column+condition+Value.

Son posibles varias condiciones: Column+condition+Value+Column2+condition2+Value2.

La sintaxis de flecha y los objetos en serie también son compatibles: Column\_ColumnField+condition+Value

Si Value contiene espacios en blanco, reemplácelos con tabulaciones antes de enviarlos al servidor.

URL	SQL
neq	!=
eq	=
gte	>=
gt	>
lte	<=
lt	<
startswith	%STARTSWITH
contains	[
doesnotcontain	['
in	IN
like	LIKE

Ejemplo de peticiones:

```
form/objects/Form.Test.Simple/info?size=2&page=1&orderby=text  
form/objects/Form.Test.Simple/all?orderby=text+desc  
form/objects/Form.Test.Simple/all?filter=text+eq+Hello  
form/objects/Form.Test.Person/infoclass?filter=company_name+contains+a  
form/objects/Form.Test.Simple/all?filter=text+in+A9044~B5920
```

Tenga en cuenta que para el acceso SQL, el usuario debe tener privilegios relevantes de SQL (SELECT en la tabla de formulario).

## COLLATION cláusula

En un formato: collation=UPPER o collation=EXACT. Fuerza la recopilación específica en la cláusula WHERE. Si se omite, se utiliza la clasificación predeterminada.

## Paginación

La paginación está disponible con 25 filas. por página por defecto. Para cambiar el tamaño de página y la página actual, proporcione argumentos de size y page (basados en 1).

## Consultas personalizadas

Invocar form/objects/:class/custom/:query, para ejecutar una consulta personalizada. La consulta

personalizada permite al desarrollador determinar el contenido completo de la consulta. Los parámetros de URL además de `size` y `pages` no están disponibles. Su método debe analizar todos los demás parámetros de URL (o llamar a analizadores predeterminados desde `Form.JSON.SQL`).

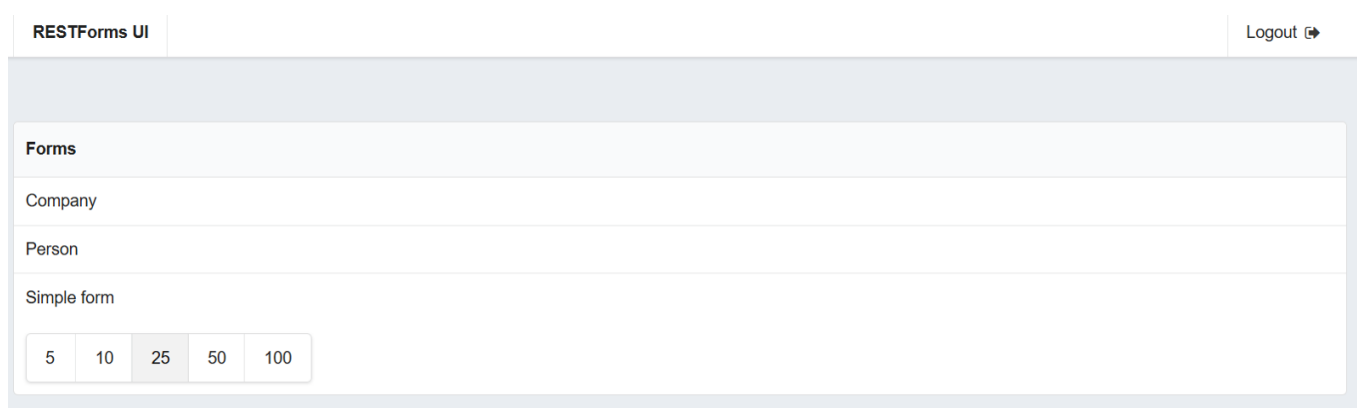
Para definir su propia consulta personalizada llamada `myq`:

1. Define un método de clase `customqueryMYQ` en tu clase de formulario
2. La firma del método es: `ClassMethod customqueryMYQ() As %String`
3. El método debe devolver una consulta SQL válida

## Demo

Puedes probar RESTForms en línea [aquí](#) (user: Demo, pass: Demo) .

Además, hay una aplicación RESTFormsUI: editor de datos RESTForms, échale un vistazo [aquí](#) (user: Demo, pass: Demo). Captura de pantalla de la lista de clases.:



## Conclusión

RESTForms proporciona capacidades de consulta extensas y personalizables.

## Lo siguiente

En el siguiente artículo, me gustaría contarle algunas características avanzadas:

- Traducción de metadatos
- Seguridad y permisos
- Nombre del objeto

## Links

- [RESTForms GitHub repository](#)
- [RESTForms UI GitHub repository](#)

[#Frontend](#) [#SOAP](#) [#Caché](#)