# Scalars & Enumerations

The scalar and enumeration represent two of the basic types in GraphQL, together referred to as "leaf types". Both types must be bound to C# objects; be that an enum (as is the case with enumeration types) or to a built in value/reference type in the base class library.

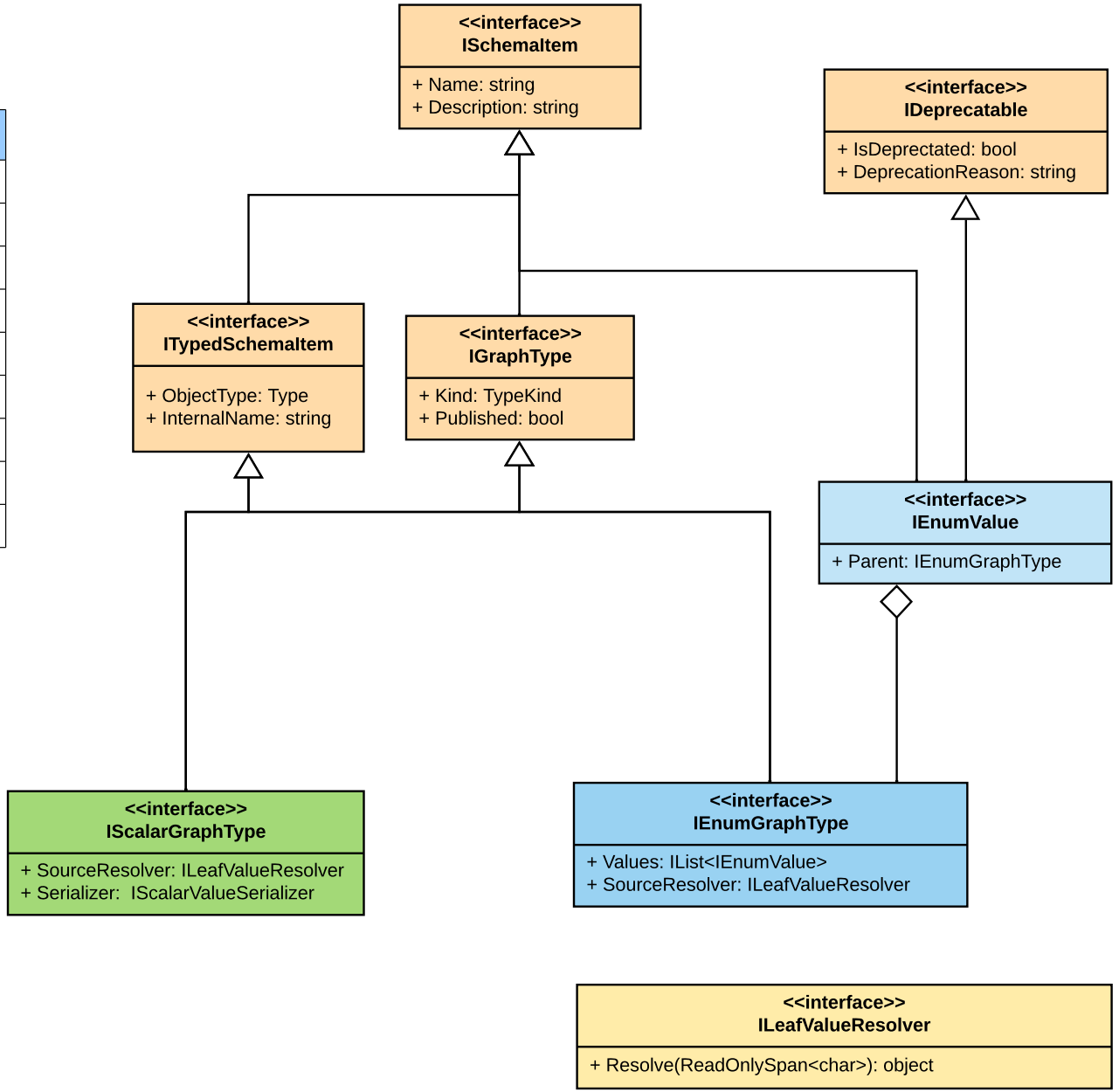
You can add custom scalars to the system by registering your own IScalarGraphType at start up.

| Graph Name | .NET Type | Serialized Type |
|---|---|---|
| STRING | string | string |
| INT | int | number |
| UINT | uint | number |
| LONG | long | number |
| ULONG | ulong | number |
| DECIMAL | decimal | number |
| FLOAT | float | number |
| DOUBLE | double | number |
| DATE* | DateTime | number |

*A complete list of scalar types is available at:*
*https://graphql-aspnet.github.io/docs/types/scalars*

** GraphQL ASP.NET, by default, serializes dates to a number of ticks,* ***in milliseconds,*** *from the unix epoch.*

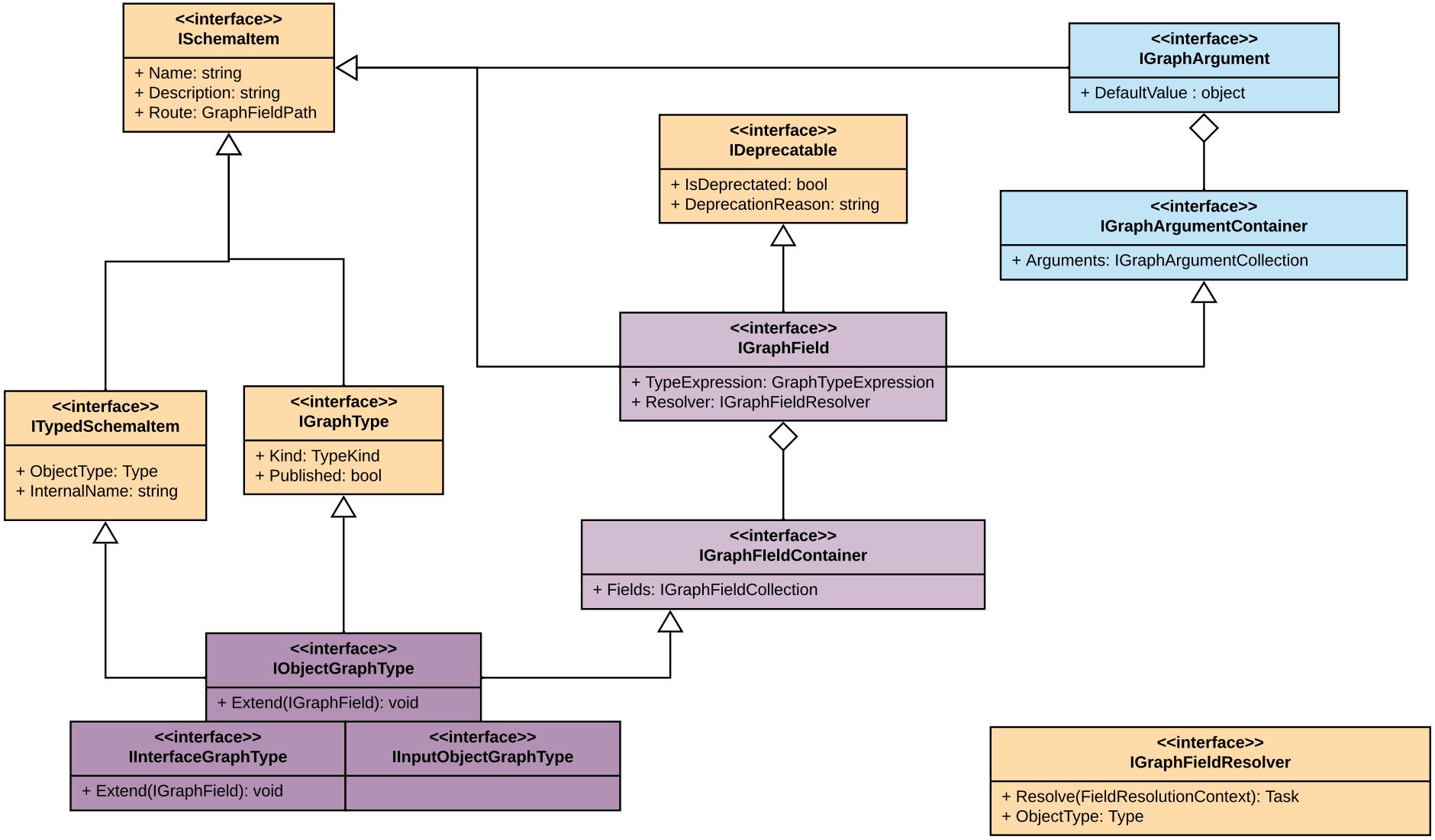
*All value type scalars can be nullable (e.g. int?) . The object graph you construct will be automatically configured for nullable values depending on the property and method return types in your C# code.*




The interfaces in this document do not represent a complete list of properties and methods.
**All interfaces located at**: */src/graphql-aspnet/Interfaces/**

# Input Object, Object, Interface Graph Type

The structure of the INPUT_OBJECT, OBJECT and INTERFACE graph types are nearly identical. They all contain field definitions that potentially have arguments. The contents of and use of these fields will vary significantly at runtime depending on the graph type in question. For instance, the field resolver assigned to an INPUT_OBJECT field is ignored entirely and will be likely set to null in the graph.

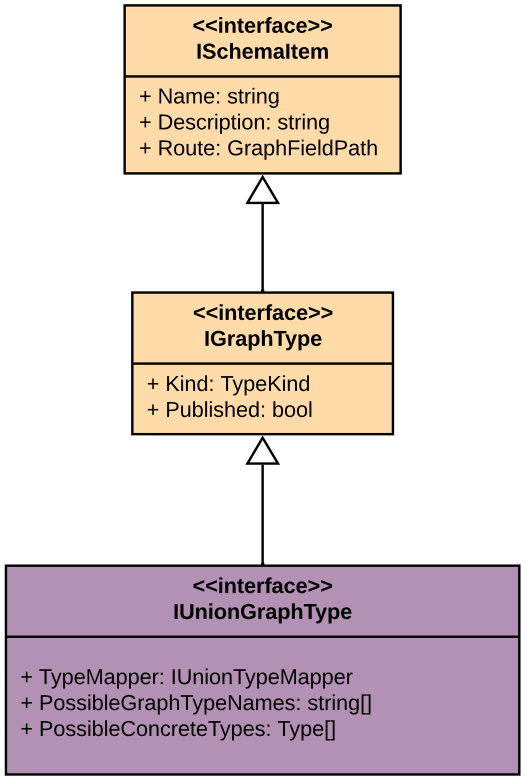



* The interfaces in this document do not represent a complete list of properties and methods.

**All interfaces located at**: */src/graphql-aspnet/Interfaces/**

# Union Graph Type

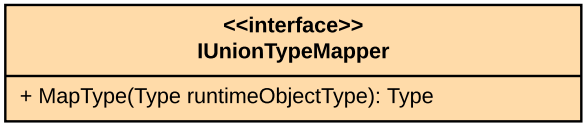
The union type represents an intersection of other graph types. It contains a type mapper to distinguish which type a resolved .NET object should masquerade as when executing a query.




The union graph type represents multiple different possible graph types. It contains the names of the graph types contained in the union.

The **TypeMapper** property points to a class that can map between union types. This is used to resolve some edge cases caused by object inheritance chains when a resolved object could represent more than one graph type in the union.

For instance if a field resolver returned a Teacher object and the union represents both Teachers and Employee objects. Since all teachers are also employees it cannot deteremine which type is being requested without additional criteria.

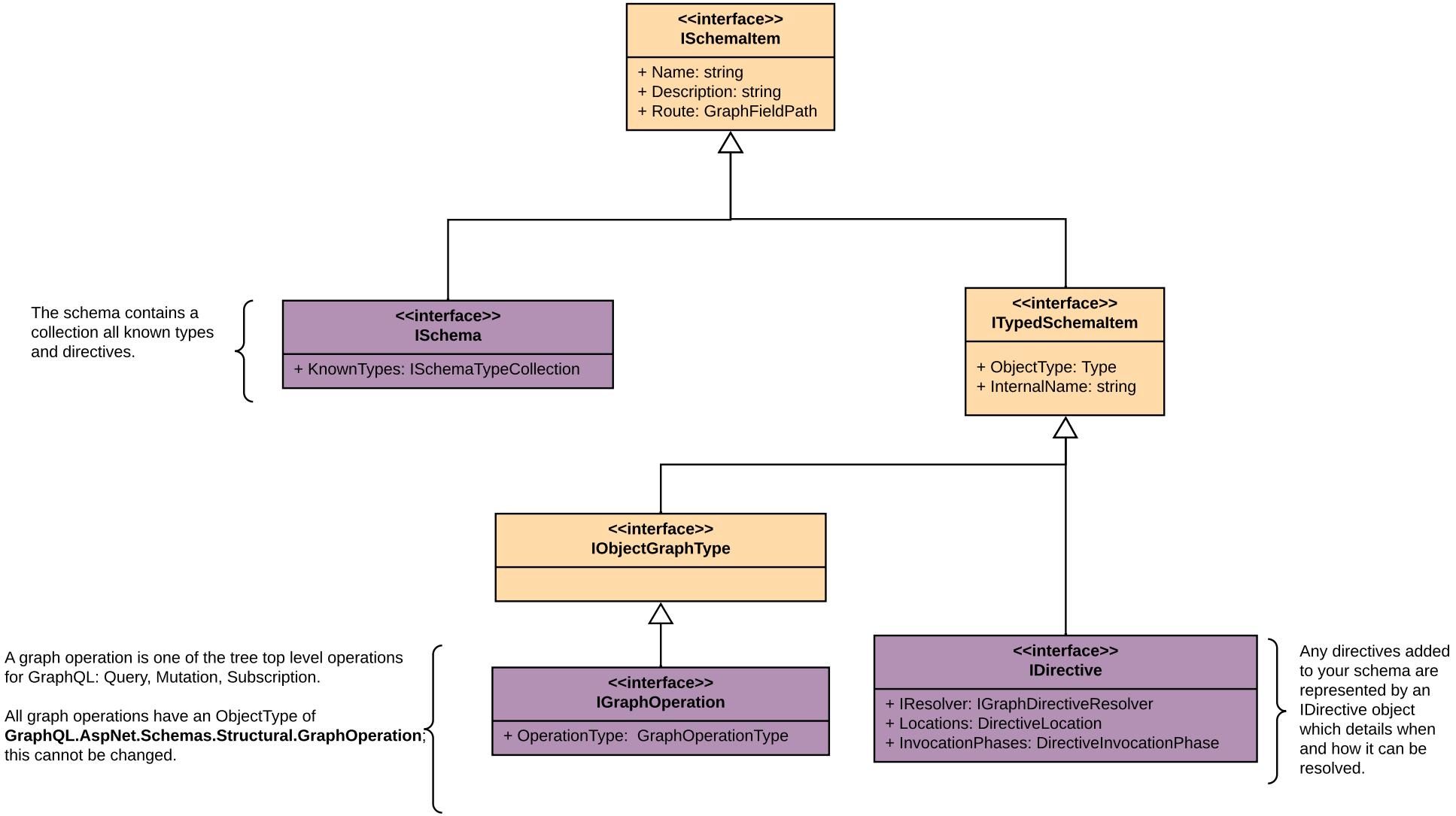



* The interfaces in this document do not represent a complete list of properties and methods.

**All interfaces located at**: */src/graphql-aspnet/Interfaces/**

# Other Schema Items

Some ISchemaItem objects are not related to graph types or fields.




* The interfaces in this document do not represent a complete list of properties and methods.

**All interfaces located at**: */src/graphql-aspnet/Interfaces/**