Available Base Types

The Designer supports seven base types:

- Array
- Blob
- Boolean
- DateTime
- Float
- Integer
- String

All base types are derived from the abstract **Any** type. The default base type for properties is **String**. All built-in base types of the Designer are located in a dedicated package **Base Types** in the **Service** panel (see also Base Types).

The following table shows a short description of its types and example values:

Base Type	Type Definition	Example Values
Blob	A blob represents base64-encoded arbitrary binary data.	e.g. "`YWJjZGVmZ2g=`" for the encoded value "abcdefgh"
Boolean	Boolean values represent binary-valued logic (true, false).	`true, false, 0, 1`
DateTime	`CCYY-MM-DDThh:mm:ss[.sss] [Z|[+|-]hh:mm]` Whereas "Z" stands for the time zone: Coordinated Universal Time (UTC). The W3C value space of **DateTime** is closely related to the dates and times described in ISO 8601.	`2021-12-01T00:00:00.0.Z` This value represents the 1st of December of the year 2021 at midnight.
Float	A float corresponds to the IEEE single-precision 32-bit floating-point type. Lexical representation: `[white spaces] [+|-][nnn][.nnn][e|E[+|-]nnn]` (where `white spaces` are any tab or space character; `nnn` may be any number of digits)	`1.234, 1e-5`
Integer	The W3C defines integers as "decimal", which represents arbitrary precision decimal numbers.	`5`
String	A string is a set of finite-length sequences of a character set (the Bridge uses UTF-8 internally).	`Hello World!`

Most of these base types are only able to hold one single piece of information, like text in a string, true or false in a boolean, or binary data in a blob. Only the base type **Array** can store multiple pieces of information like multiple array elements. If you wish to associate several bits of information, you have to define a complex type that combines a number of independent base types and possibly other complex types. Arrays are characterized by multiplicity and type.

The base type **Array** is an exception from the rule that base types do not have any sub-structure. Arrays allow multiple elements to be stored, but the array elements must be all of the same type. For example, an array may not contain string and integer elements, but only elements, which are of one of these types. Arrays may also contain complex types.

> ⓘ Arrays of arrays are not supported by the Designer.