

Overview: Built-In functionality

Short overview about faketastic's built-in functionality. Table of contents:

- [BuilderFns](#)
- [AttachedFns](#)
 - [Architects](#)
 - [Processors](#)
- [ModelFns](#)
- [ValueFns](#)
- [FactoryFns](#)

Builder Functions

Builder functions are the "nouns" of your models. They generate the data that attached functions then can manipulate. They are also hosts of attached functions.

- [Overview: BuilderFns](#)
 - [combine builder](#)
 - [itself builder](#)
 - [oneOf builder](#)
 - [range builder](#)
 - [ref builder](#)
 - [someOf builder](#)

Attached Functions

Attached functions can be considered as the "adjectives" of your models. They define the shape and behavior of your data.

There are three subtypes of attached functions:

- [TreeReaderFns](#) (skipped, as not really important for usual modelling)
- [ArchitectFns](#)
- [ProcessorFns](#)

Architects

Architect functions are only affecting the [build-tree](#) in which their associated property is located. They do not touch the result-value of the property itself, but rather moves/copies/removes it structurally within the [build-tree](#). For example, the [quantity](#) architect multiplies the [build-tree](#) node it is attached to and turns it into an [array](#)-type node. This way, it changes a single-value result to be a multiple-value result being wrapped within an array. These kinds of work are done by architect functions.

- [Overview: ArchitectFns](#)
 - [quantity architect](#)

Processors

Processor functions manipulates the value they are attached to. They can be attached to any builder function and affect the result value by either altering or replacing it completely.

- [Overview: ProcessorFns](#)
 - [canBe processor](#)
 - [map processor](#)

Model Functions

Modelling functions allow you to define, extend and reuse existing models. They are the core of faketastic.

- [Overview: ModelFns](#)
 - [build model-function](#)
 - [extend model-function](#)
 - [model model-function](#)
 - [use model-function](#)

Value Functions

Value functions continously create randomized data of a specific type (such as [Date](#) or [string](#)), that respects given input-restrictions (e.g. [min/max](#) parameters). They are used directly on properties, similar to [builder functions](#) as they return [Buildables](#), that yields a new random value each time it gets called.

- [Overview: ValueFns](#)
 - [range value function](#)
 - [time value function](#)

Factory Functions

Factory functions are similiar to [value functions](#) as they take user-input and turns it into randomized data of a specific type, that respects given input-restrictions. But in contrast to value functions, factory functions are used within processor functions as they directly return the created, randomized value.

- [Overview: FactoryFns](#)
 - [duration factory](#)

Related Topics

- [Getting Started](#)
- [ModelFns](#)
- [BuilderFns](#)
- [AttachedFns](#)