ValidatorOptions

August 18, 2026

ValidatorOptions

Kind: Interface

Source: `packages/common/interfaces/external/validator-options.interface.ts`

Part of: Common

Options passed to validator during validation.

`ValidatorOptions` configures how validation is performed for incoming objects. It controls property handling, validation groups, debug and default messages, and whether unknown properties are allowed or rejected.

Properties

Property	Type
`enableDebugMessages`	`boolean`
`skipUndefinedProperties`	`boolean`
`skipNullProperties`	`boolean`
`skipMissingProperties`	`boolean`
`whitelist`	`boolean`
`forbidNonWhitelisted`	`boolean`
`groups`	`string[]`
`always`	`boolean`
`strictGroups`	`boolean`
`dismissDefaultMessages`	`boolean`
`validationError`	`{ target?: boolean; value?: boolean; }`
`forbidUnknownValues`	`boolean`

Property	Type
`stopAtFirstError`	`boolean`

Diagram

```
graph LR
  A[Validation request] --> B[ValidatorOptions]
  B --> C[Property handling]
  B --> D[Validation groups]
  B --> E[Validation messages]
  B --> F[Whitelist behavior]

  C --> C1[skipUndefinedProperties]
  C --> C2[skipNullProperties]
  C --> C3[skipMissingProperties]

  D --> D1[groups]
  D --> D2[always]
  D --> D3[strictGroups]

  E --> E1[enableDebugMessages]
  E --> E2[dismissDefaultMessages]

  F --> F1[whitelist]
  F --> F2[forbidNonWhitelisted]
```

Usage

```
import type { ValidatorOptions } from './interfaces/external/validator-options.interface';

const validatorOptions: ValidatorOptions = {
  whitelist: true,
  forbidNonWhitelisted: true,
  skipMissingProperties: false,
  skipUndefinedProperties: true,
  skipNullProperties: false,
  groups: ['create'],
  always: false,
  strictGroups: true,
  enableDebugMessages: process.env.NODE_ENV !== 'production',
  dismissDefaultMessages: false,
};

// Pass the options to the validation layer or validator integration.
const errors = await validate(createUserDto, validatorOptions);
```

AI Coding Instructions

- Enable `whitelist` when validating DTOs or request payloads to prevent unrecognized input properties from being retained.
- Use `forbidNonWhitelisted` together with `whitelist` when unknown properties should produce validation errors instead of being silently removed.
- Choose only the skip option that matches the API contract: `skipUndefinedProperties`, `skipNullProperties`, and `skipMissingProperties` have different behaviors.
- Use `groups` with `strictGroups` when validation decorators are explicitly scoped to operations such as `create` or `update`.
- Keep `enableDebugMessages` disabled in production environments to avoid exposing implementation details in validation responses.