

Directory /src/github.com/abitoftelp/motominderapi

Name	Synopsis
..	
clean	
adapter	
gateway	
api	Package api contains the restful web service.
dto	Package dto contains data transfer objects sent from client applications.
repository	Package repository contains implementations of data repositories.
security	Package security contains implementations of interfaces dealing security, authentication, and authorization.
presenter	Package presenter performs the translation of a response message into a view model.
viewmodel	Package viewmodel translates a response message into a view model.
configuration	
web	
main	Package main is the entry point for the API web service.
domain	
constant	Package constant contains values for the domain.
contract	Package contract contains contracts for entities and other objects.
entity	Package entity contains the domain entities.
enumeration	Package enumeration defines enumerations for the application.
usecase	
interactor	Package interactor contains use cases, which contain the application specific business rules.
request	Package request contains the request messages for the use cases.
response	Package response contains the response messages for the use cases.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package api

```
import "github.com/abitoofhelp/motominderapi/clean/adapter/gateway/api"
```

[Overview](#)[Index](#)[Subdirectories](#)

Overview

Package api contains the restful web service.

Index

type [Api](#)

```
func NewApi(roles map[enumeration.AuthorizationRole]bool, authService *security.AuthService,  
motorcycleRepository *repository.MotorcycleRepository, router *httprouter.Router) (*Api, error)
```

```
func (api *Api) Start() error
```

```
func (api *Api) Stop() error
```

```
func (api Api) Validate() error
```

Package files

[api.go](#)

type Api

Api is a web service.

```
type Api struct {  
    Roles                map[enumeration.AuthorizationRole]bool  
    AuthService          *security.AuthService  
    MotorcycleRepository *repository.MotorcycleRepository  
    Router               *httprouter.Router  
}
```

func NewApi

```
func NewApi(roles map[enumeration.AuthorizationRole]bool, authService
```

```
*security.AuthService, motorcycleRepository *repository.MotorcycleRepository, router
*httprouter.Router) (*Api, error)
```

NewApi creates a new instance of an Api. Returns (an instance of APi, nil), otherwise (nil, error)

func (*Api) Start

```
func (api *Api) Start() error
```

Start launches the web service.

func (*Api) Stop

```
func (api *Api) Stop() error
```

Stop terminates the web service.

func (Api) Validate

```
func (api Api) Validate() error
```

Validate verifies that a api's fields contain valid data. Returns nil on success, otherwise error.

Subdirectories

Name	Synopsis
------	----------

..	
----	--

dto	Package dto contains data transfer objects sent from client applications.
-----	---

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package dto

```
import "github.com/abitoofhelp/motominderapi/clean/adapter/gateway/api/dto"
```

[Overview](#)[Index](#)

Overview

Package dto contains data transfer objects sent from client applications.

Index

[type InsertMotorcycleDto](#)

Package files

[insertmotorcycledto.go](#)

type InsertMotorcycleDto

InsertMotorcycleDto contains the data for creating a new motorcycle in the repository.

```
type InsertMotorcycleDto struct {  
    Make   string `json:"make"`  
    Model  string `json:"model"`  
    Year   int    `json:"year"`  
    Vin    string `json:"vin"`  
}
```

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package repository

```
import "github.com/abitoofhelp/motominderapi/clean/adapter/gateway/repository"
```

[Overview](#)[Index](#)

Overview

Package repository contains implementations of data repositories.

Index

```
type MotorcycleRepository
func NewMotorcycleRepository() (*MotorcycleRepository, error)
func (repo *MotorcycleRepository) Delete(motorcycle *entity.Motorcycle) error
func (repo *MotorcycleRepository) FindByID(id int) (*entity.Motorcycle, error)
func (repo *MotorcycleRepository) FindByVin(vin string) (*entity.Motorcycle, error)
func (repo *MotorcycleRepository) Insert(motorcycle *entity.Motorcycle) (*entity.Motorcycle, error)
func (repo *MotorcycleRepository) List() ([]entity.Motorcycle, error)
func (repo *MotorcycleRepository) Save() error
func (repo *MotorcycleRepository) Update(motorcycle *entity.Motorcycle) (*entity.Motorcycle, error)
func (repo MotorcycleRepository) Validate() error
```

Package files

motorcyclerepository.go

type MotorcycleRepository

MotorcycleRepository provides CRUD operations against a collection of motorcycles.

```
type MotorcycleRepository struct {
    // These items are unordered.
    Motorcycles []entity.Motorcycle `json:"motorcycles"`
}
```

func NewMotorcycleRepository

```
func NewMotorcycleRepository() (*MotorcycleRepository, error)
```

NewMotorcycleRepository creates a new instance of a MotorcycleRepository. Returns (nil, error) when there is an error, otherwise a (MotorcycleRepository, nil).

func (*MotorcycleRepository) Delete

```
func (repo *MotorcycleRepository) Delete(motorcycle *entity.Motorcycle) error
```

Delete an existing motorcycle from the repository. If the motorcycle does not exist, an error is returned. Returns nil on success, otherwise an error.

func (*MotorcycleRepository) FindByID

```
func (repo *MotorcycleRepository) FindByID(id int) (*entity.Motorcycle, error)
```

FindByID a motorcycle in the repository using its primary key, ID. Returns (motorcycle, nil) on success, otherwise (nil, error).

func (*MotorcycleRepository) FindByVin

```
func (repo *MotorcycleRepository) FindByVin(vin string) (*entity.Motorcycle, error)
```

FindByVin a motorcycle in the repository using its VIN. Returns (motorcycle, nil) on success, otherwise (nil, error).

func (*MotorcycleRepository) Insert

```
func (repo *MotorcycleRepository) Insert(motorcycle *entity.Motorcycle)
(*entity.Motorcycle, error)
```

Insert adds a motorcycle to the repository. Do not permit duplicate ID values. Returns the (new motorcycle, nil) on success, otherwise (nil, error).

func (*MotorcycleRepository) List

```
func (repo *MotorcycleRepository) List() ([]entity.Motorcycle, error)
```

List gets the unordered list of motorcycles in the repository. Returns the (list of motorcycles, nil) or an (nil, error).

func (*MotorcycleRepository) Save

```
func (repo *MotorcycleRepository) Save() error
```

Save all of the changes to the repository (assuming some kind of unit of work/dbContext). Returns nil on success, otherwise an error.

func (*MotorcycleRepository) Update

```
func (repo *MotorcycleRepository) Update(motorcycle *entity.Motorcycle)
(*entity.Motorcycle, error)
```

Update replaces a motorcycle an existing motorcycle in the repository. If the motorcycle does not exist, an error is returned. Returns (updated motorcycle, nil) on success, otherwise an (nil, error).

func (MotorcycleRepository) Validate

```
func (repo MotorcycleRepository) Validate() error
```

Validate test that a motorcycle repository is valid. Returns nil on success, otherwise an error.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package security

```
import "github.com/abitoofhelp/motominderapi/clean/adapters/gateway/security"
```

[Overview](#)[Index](#)

Overview

Package security contains implementations of interfaces dealing security, authentication, and authorization.

Index

```
type AuthService
```

```
func NewAuthService(authenticated bool, roles map[enumeration.AuthorizationRole]bool) (*AuthService, error)
```

```
func (authService *AuthService) IsAuthenticated() bool
```

```
func (authService *AuthService) IsAuthorized(role enumeration.AuthorizationRole) bool
```

```
func (authService AuthService) Validate() error
```

Package files

[authservice.go](#)

type AuthService

AuthService is a contract that provides authentication and authorization services.

```
type AuthService struct {
    Authenticated bool
    Roles         map[enumeration.AuthorizationRole]bool
}
```

func NewAuthService

```
func NewAuthService(authenticated bool, roles map[enumeration.AuthorizationRole]bool)
(*AuthService, error)
```


NewAuthService creates a new instance of an AuthService. Returns (nil, error) when there is an error, otherwise (authService, nil).

func (*AuthService) IsAuthenticated

```
func (authService *AuthService) IsAuthenticated() bool
```

IsAuthenticated determines whether the User has been authenticated by the system. Returns true if the User has passed authentication, otherwise false.

func (*AuthService) IsAuthorized

```
func (authService *AuthService) IsAuthorized(role enumeration.AuthorizationRole) bool
```

IsAuthorized determines whether the User possesses the required authorization role(s). Returns true if "Admin" is in the roles, otherwise false.

func (AuthService) Validate

```
func (authService AuthService) Validate() error
```

Validate verifies that an AuthService's fields contain valid data. Returns nil if the AuthService contains valid data, otherwise an error.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package presenter

```
import "github.com/abitoofhelp/motominderapi/clean/adapter/presenter"
```

[Overview](#)[Index](#)

Overview

Package presenter performs the translation of a response message into a view model.

Index

```
type InsertMotorcyclePresenter
func NewInsertMotorcyclePresenter() (*InsertMotorcyclePresenter, error)
func (presenter *InsertMotorcyclePresenter) Handle(responseMessage
*response.InsertMotorcycleResponse) (*viewmodel.InsertMotorcycleViewModel, error)
func (presenter InsertMotorcyclePresenter) Validate() error
```

Package files

[insertmotorcyclepresenter.go](#)

type InsertMotorcyclePresenter

InsertMotorcyclePresenter translates the response message from the InsertMotorcycleInteractor to a view model.

```
type InsertMotorcyclePresenter struct {
}
```

func NewInsertMotorcyclePresenter

```
func NewInsertMotorcyclePresenter() (*InsertMotorcyclePresenter, error)
```

NewInsertMotorcyclePresenter creates a new instance of a InsertMotorcyclePresenter. Returns (instance of InsertMotorcyclePresenter, nil) on success, otherwise (nil, error).

func (*InsertMotorcyclePresenter) Handle

```
func (presenter *InsertMotorcyclePresenter) Handle(responseMessage
*response.InsertMotorcycleResponse) (*viewModel.InsertMotorcycleViewModel, error)
```

Handle performs the translation of the response message into a view model. Returns (instance of InsertMotorcycleViewModel, nil) on success, otherwise (nil, error)

func (InsertMotorcyclePresenter) Validate

```
func (presenter InsertMotorcyclePresenter) Validate() error
```

Validate verifies that a InsertMotorcyclePresenter's fields contain valid data. Returns (an instance of InsertMotorcyclePresenter, nil) on success, otherwise (nil, error)

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package viewmodel

```
import "github.com/abitoofhelp/motominderapi/clean/adapters/viewmodel"
```

[Overview](#)[Index](#)

Overview

Package viewmodel translates a response message into a view model.

Index

```
type InsertMotorcycleViewModel
func NewInsertMotorcycleViewModel(id int, message string, err error) (*InsertMotorcycleViewModel, error)
func (viewmodel *InsertMotorcycleViewModel) Handle(responseMessage
*response.InsertMotorcycleResponse) (*InsertMotorcycleViewModel, error)
func (viewmodel InsertMotorcycleViewModel) Validate() error
```

Package files

[insertmotorcycleviewmodel.go](#)

type InsertMotorcycleViewModel

InsertMotorcycleViewModel translates a InsertMotorcycleResponse to a InsertMotorcycleViewModel. by the Configuration ring.

```
type InsertMotorcycleViewModel struct {
    ID      int    `json:"id"`
    Message string `json:"message"`
    Error   error  `json:"error"`
}
```

func NewInsertMotorcycleViewModel

```
func NewInsertMotorcycleViewModel(id int, message string, err error)
(*InsertMotorcycleViewModel, error)
```

NewInsertMotorcycleViewModel creates a new instance of a InsertMotorcycleViewModel. Returns an (instance of InsertMotorcycleViewModel, nil) on success, otherwise (nil, error)

func (*InsertMotorcycleViewModel) Handle

```
func (viewModel *InsertMotorcycleViewModel) Handle(responseMessage
*response.InsertMotorcycleResponse) (*InsertMotorcycleViewModel, error)
```

Handle performs the translation of the response message into a view model. Returns (instance of InsertMotorcycleViewModel, nil) on success, otherwise (nil, error)

func (InsertMotorcycleViewModel) Validate

```
func (viewModel InsertMotorcycleViewModel) Validate() error
```

Validate verifies that a InsertMotorcycleViewModel's fields contain valid data. Returns (an instance of InsertMotorcycleViewModel, nil) on success, otherwise (nil, error).

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Command main

Package main is the entry point for the API web service.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package constant

```
import "github.com/abitoofhelp/motominderapi/clean/domain/constant"
```

[Overview](#)[Index](#)

Overview

Package constant contains values for the domain.

Index

[Constants](#)

Package files

[constant.go](#)

Constants

InvalidEntityID is used when there is an invalid ID for an entity.

```
const InvalidEntityID = -1
```

MaxMakeLength is the maximum length string for a make.

```
const MaxMakeLength = 20
```

MaxModelLength is the maximum length string for a model.

```
const MaxModelLength = 20
```

MaxYear is the maximum year.

```
const MaxYear = 2020
```

MinEntityID is the minimum ID value.

```
const MinEntityID = 1
```

MinMakeLength is the minimum length string for a make.

```
const MinMakeLength = 1
```

MinModelLength is the minimum length string for a model.

```
const MinModelLength = 1
```

MinYear is the minimum year.

```
const MinYear = 1999
```

VinLength is the length of a VIN.

```
const VinLength = 17
```

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package contract

```
import "github.com/abitofofhelp/motominderapi/clean/domain/contract"
```

[Overview](#)[Index](#)

Overview

Package contract contains contracts for entities and other objects.

Package contract contains contracts for entities and other objects.

Package contract contains contracts for entities and other objects.

Package contract contains contracts for entities and other objects.

Package contract contains contracts for entities and other objects.

Index

[type AuthService](#)
[type MotorcycleRepository](#)
[type RequestHandler](#)
[type RequestMessage](#)
[type ResponseMessage](#)

Package files

[authservice.go](#) [motorcyclerepository.go](#) [requesthandler.go](#) [requestmessage.go](#) [responsemessage.go](#)

type AuthService

AuthService is a contract that provides authentication and authorization services.

```
type AuthService interface {  
    IsAuthenticated() bool  
    IsAuthorized(role enumeration.AuthorizationRole) bool  
    Validate() error  
}
```

type MotorcycleRepository

MotorcycleRepository defines the contract for its actions.

```
type MotorcycleRepository interface {
    List() ([]entity.Motorcycle, error)
    Insert(motorcycle *entity.Motorcycle) (*entity.Motorcycle, error)
    FindByVin(vin string) (*entity.Motorcycle, error)
    Update(motorcycle *entity.Motorcycle) (*entity.Motorcycle, error)
    Delete(motorcycle *entity.Motorcycle) error
    FindByID(id int) (*entity.Motorcycle, error)
    Save() error
    Validate() error
}
```

type RequestHandler

RequestHandler is the default handler for a requests.

```
type RequestHandler interface {
    Handle(requestMessage RequestMessage) (ResponseMessage, error)
}
```

type RequestMessage

RequestMessage defines the base contract for all request messages.

```
type RequestMessage interface {
    // Validate verifies that a RequestMessage's fields contain valid data.
    // Returns nil if the RequestMessage contains valid data, otherwise an error.
    Validate() error
}
```

type ResponseMessage

ResponseMessage defines the base contract for all response messages.

```
type ResponseMessage interface {
    // Validate verifies that a ResponseMessage's fields contain valid data.
    // Returns nil if the ResponseMessage contains valid data, otherwise an error.
    Validate() error
}
```

```
}
```

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package entity

```
import "github.com/abitoofhelp/motominderapi/clean/domain/entity"
```

[Overview](#)[Index](#)

Overview

Package entity contains the domain entities.

Index

```
func Is17Characters(value interface{}) error
func IsInvalidManufacturer(value interface{}) error
type Motorcycle
    func NewMotorcycle(make string, model string, year int, vin string) (*Motorcycle, error)
    func (m Motorcycle) Validate() error
```

Package files

[motorcycle.go](#)

func Is17Characters

```
func Is17Characters(value interface{}) error
```

Is17Characters verifies that a string has 17 characters. Returns nil if the string does not contain 17 characters, otherwise an error.

func IsInvalidManufacturer

```
func IsInvalidManufacturer(value interface{}) error
```

IsInvalidManufacturer verifies that a motorcycle's make is not an invalid manufacturer. Returns nil if the manufacturer is valid, otherwise an error.

type Motorcycle

Motorcycle is an entity

```
type Motorcycle struct {
    ID          int          `json:"id"`
    Make        string       `json:"make"`
    Model       string       `json:"model"`
    Year        int          `json:"year"`
    Vin         string       `json:"vin"`
    CreatedUtc  time.Time   `json:"createdUtc"`
    ModifiedUtc time.Time   `json:"modifiedUtc"`
}
```

func NewMotorcycle

```
func NewMotorcycle(make string, model string, year int, vin string) (*Motorcycle, error)
```

NewMotorcycle creates a new instance of a Motorcycle. Returns (nil, error) when there is an error, otherwise (motorcycle, nil).

func (Motorcycle) Validate

```
func (m Motorcycle) Validate() error
```

Validate verifies that a motorcycle's fields contain valid data that satisfies enterprise's common business rules. Returns nil if the motorcycle contains valid data, otherwise an error.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package enumeration

```
import "github.com/abitoofhelp/motominderapi/clean/domain/enumeration"
```

[Overview](#)[Index](#)

Overview

Package enumeration defines enumerations for the application.

Index

[Constants](#)[type AuthorizationRole](#)[func \(AuthorizationRole\) Accounting\(\) string](#)[func \(AuthorizationRole\) Admin\(\) string](#)[func \(AuthorizationRole\) General\(\) string](#)

Package files

[authorizationrole.go](#)

Constants

```
const (  
    // NoAuthorizationRole is when the user does not require any authorization roles to  
    access resources.  
    NoAuthorizationRole = iota  
    // AdminAuthorizationRole is a user with authorization to access administrative  
    resources.  
    AdminAuthorizationRole  
    // AccountingAuthorizationRole is a user with authorization to access accounting  
    resources.  
    AccountingAuthorizationRole  
    // GeneralAuthorizationRole is a user with administrative authorization to access  
    general resources.  
    GeneralAuthorizationRole  
)
```

type AuthorizationRole

AuthorizationRole is an authorization given to an authenticated user to access a resource.

```
type AuthorizationRole int
```

func (AuthorizationRole) Accounting

```
func (AuthorizationRole) Accounting() string
```

Accounting provides a useful string for the value. Returns a string for the Accounting constant.

func (AuthorizationRole) Admin

```
func (AuthorizationRole) Admin() string
```

Admin provides a useful string for the value. Returns a string for the Admin constant.

func (AuthorizationRole) General

```
func (AuthorizationRole) General() string
```

General provides a useful string for the value. Returns a string for the General constant.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package interactor

```
import "github.com/abitoofhelp/motominderapi/clean/usecase/interactor"
```

[Overview](#)[Index](#)

Overview

Package interactor contains use cases, which contain the application specific business rules. Interactors encapsulate and implement all of the use cases of the system. They orchestrate the flow of data to and from the entity, and can rely on their business rules to achieve the goals of the use case. They do not have any dependencies, and are totally isolated from things like a database, UI or special frameworks, which exist in the outer rings. They Will almost certainly require refactoring if details of the use case requirements change.

Index

```
type InsertMotorcycleInteractor
func NewInsertMotorcycleInteractor(motorcycleRepository contract.MotorcycleRepository, authService
contract.AuthService) (*InsertMotorcycleInteractor, error)
func (interactor *InsertMotorcycleInteractor) Handle(requestMessage *request.InsertMotorcycleRequest)
(*response.InsertMotorcycleResponse, error)
func (interactor InsertMotorcycleInteractor) Validate() error
```

Package files

[insertmotorcycleinteractor.go](#)

type InsertMotorcycleInteractor

InsertMotorcycleInteractor is a use case for adding a motorcycle to the motorcycle repository.

```
type InsertMotorcycleInteractor struct {
    MotorcycleRepository contract.MotorcycleRepository
    AuthService          contract.AuthService
}
```

func NewInsertMotorcycleInteractor

```
func NewInsertMotorcycleInteractor(motorcycleRepository contract.MotorcycleRepository,
authService contract.AuthService) (*InsertMotorcycleInteractor, error)
```

NewInsertMotorcycleInteractor creates a new instance of a InsertMotorcycleInteractor. Returns (nil, error) when there is an error, otherwise (InsertMotorcycleInteractor, nil).

func (*InsertMotorcycleInteractor) Handle

```
func (interactor *InsertMotorcycleInteractor) Handle(requestMessage
*request.InsertMotorcycleRequest) (*response.InsertMotorcycleResponse, error)
```

Handle processes the request message and generates the response message. It is performing the use case. The request message is a dto containing the required data for completing the use case. On success, the method returns the (response message, nil), otherwise (nil, error).

func (InsertMotorcycleInteractor) Validate

```
func (interactor InsertMotorcycleInteractor) Validate() error
```

Validate verifies that a InsertMotorcycleInteractor's fields contain valid data. Returns nil if the InsertMotorcycleInteractor contains valid data, otherwise an error.

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package request

```
import "github.com/abitoofhelp/motominderapi/clean/usecase/request"
```

[Overview](#)[Index](#)

Overview

Package request contains the request messages for the use cases.

Index

```
type InsertMotorcycleRequest
func NewInsertMotorcycleRequestMessage(make string, model string, year int, vin string)
(*InsertMotorcycleRequest, error)
func (request InsertMotorcycleRequest) Validate() error
```

Package files

[insertmotorcyclerequest.go](#)

type InsertMotorcycleRequest

InsertMotorcycleRequest is a simple dto containing the required data for the InsertMotorcycleInteractor.

```
type InsertMotorcycleRequest struct {
    Make   string `json:"make"`
    Model  string `json:"model"`
    Year   int    `json:"year"`
    Vin    string `json:"vin"`
}
```

func NewInsertMotorcycleRequestMessage

```
func NewInsertMotorcycleRequestMessage(make string, model string, year int, vin string)
(*InsertMotorcycleRequest, error)
```

NewInsertMotorcycleRequestMessage creates a new instance of a InsertMotorcycleRequest. Returns (nil,

error) when there is an error, otherwise (InsertMotorcycleRequest, nil).

func (InsertMotorcycleRequest) Validate

```
func (request InsertMotorcycleRequest) Validate() error
```

Validate verifies that a InsertMotorcycleRequest's fields contain valid data. Returns (an instance of InsertMotorcycleRequest, nil) on success, otherwise (nil, error)

Build version go1.9.2.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Package response

```
import "github.com/abitoofhelp/motominderapi/clean/usecase/response"
```

[Overview](#)[Index](#)

Overview

Package response contains the response messages for the use cases.

Index

```
type InsertMotorcycleResponse
func NewInsertMotorcycleResponse(id int, err error) (*InsertMotorcycleResponse, error)
func (response InsertMotorcycleResponse) Validate() error
```

Package files

[insertmotorcycleresponse.go](#)

type InsertMotorcycleResponse

InsertMotorcycleResponse is a simple dto containing the response data from the InsertMotorcycleInteractor.

```
type InsertMotorcycleResponse struct {
    ID      int    `json:"id"`
    Error   error  `json:"error"`
}
```

func NewInsertMotorcycleResponse

```
func NewInsertMotorcycleResponse(id int, err error) (*InsertMotorcycleResponse, error)
```

NewInsertMotorcycleResponse creates a new instance of a MotorcycleResponseMessage. Returns (nil, error) when there is an error, otherwise (InsertMotorcycleResponse, nil).

func (InsertMotorcycleResponse) Validate

```
func (response InsertMotorcycleResponse) Validate() error
```

Validate verifies that a `InsertMotorcycleResponse`'s fields contain valid data. Returns nil if the `InsertMotorcycleResponse` contains valid data, otherwise an error.

Build version `go1.9.2`.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)