;; Auto-generated. Do not edit!

(when (boundp 'gazebo\_msgs\_new::GetJointProperties)

(if (not (find-package "GAZEBO\_MSGS\_NEW"))

(make-package "GAZEBO\_MSGS\_NEW"))

(shadow 'GetJointProperties (find-package "GAZEBO\_MSGS\_NEW")))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIES")

(make-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIES"))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESREQUEST")

(make-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESREQUEST"))

(unless (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE")

(make-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(in-package "ROS")

(defclass gazebo\_msgs\_new::GetJointPropertiesRequest

:super ros::object

:slots (\_joint\_name ))

(defmethod gazebo\_msgs\_new::GetJointPropertiesRequest

(:init

(&key

((:joint\_name \_\_joint\_name) "")

)

(send-super :init)

(setq \_joint\_name (string \_\_joint\_name))

self)

(:joint\_name

(&optional \_\_joint\_name)

(if \_\_joint\_name (setq \_joint\_name \_\_joint\_name)) \_joint\_name)

(:serialization-length

()

(+

;; string \_joint\_name

4 (length \_joint\_name)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; string \_joint\_name

(write-long (length \_joint\_name) s) (princ \_joint\_name s)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; string \_joint\_name

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_joint\_name (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(intern "\*REVOLUTE\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*REVOLUTE\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*REVOLUTE\* 0)

(intern "\*CONTINUOUS\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*CONTINUOUS\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*CONTINUOUS\* 1)

(intern "\*PRISMATIC\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*PRISMATIC\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*PRISMATIC\* 2)

(intern "\*FIXED\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*FIXED\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*FIXED\* 3)

(intern "\*BALL\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*BALL\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*BALL\* 4)

(intern "\*UNIVERSAL\*" (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(shadow '\*UNIVERSAL\* (find-package "GAZEBO\_MSGS\_NEW::GETJOINTPROPERTIESRESPONSE"))

(defconstant gazebo\_msgs\_new::GetJointPropertiesResponse::\*UNIVERSAL\* 5)

(defclass gazebo\_msgs\_new::GetJointPropertiesResponse

:super ros::object

:slots (\_type \_damping \_position \_rate \_success \_status\_message ))

(defmethod gazebo\_msgs\_new::GetJointPropertiesResponse

(:init

(&key

((:type \_\_type) 0)

((:damping \_\_damping) (make-array 0 :initial-element 0.0 :element-type :float))

((:position \_\_position) (make-array 0 :initial-element 0.0 :element-type :float))

((:rate \_\_rate) (make-array 0 :initial-element 0.0 :element-type :float))

((:success \_\_success) nil)

((:status\_message \_\_status\_message) "")

)

(send-super :init)

(setq \_type (round \_\_type))

(setq \_damping \_\_damping)

(setq \_position \_\_position)

(setq \_rate \_\_rate)

(setq \_success \_\_success)

(setq \_status\_message (string \_\_status\_message))

self)

(:type

(&optional \_\_type)

(if \_\_type (setq \_type \_\_type)) \_type)

(:damping

(&optional \_\_damping)

(if \_\_damping (setq \_damping \_\_damping)) \_damping)

(:position

(&optional \_\_position)

(if \_\_position (setq \_position \_\_position)) \_position)

(:rate

(&optional \_\_rate)

(if \_\_rate (setq \_rate \_\_rate)) \_rate)

(:success

(&optional \_\_success)

(if \_\_success (setq \_success \_\_success)) \_success)

(:status\_message

(&optional \_\_status\_message)

(if \_\_status\_message (setq \_status\_message \_\_status\_message)) \_status\_message)

(:serialization-length

()

(+

;; uint8 \_type

1

;; float64[] \_damping

(\* 8 (length \_damping)) 4

;; float64[] \_position

(\* 8 (length \_position)) 4

;; float64[] \_rate

(\* 8 (length \_rate)) 4

;; bool \_success

1

;; string \_status\_message

4 (length \_status\_message)

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; uint8 \_type

(write-byte \_type s)

;; float64[] \_damping

(write-long (length \_damping) s)

(dotimes (i (length \_damping))

(sys::poke (elt \_damping i) (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

)

;; float64[] \_position

(write-long (length \_position) s)

(dotimes (i (length \_position))

(sys::poke (elt \_position i) (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

)

;; float64[] \_rate

(write-long (length \_rate) s)

(dotimes (i (length \_rate))

(sys::poke (elt \_rate i) (send s :buffer) (send s :count) :double) (incf (stream-count s) 8)

)

;; bool \_success

(if \_success (write-byte -1 s) (write-byte 0 s))

;; string \_status\_message

(write-long (length \_status\_message) s) (princ \_status\_message s)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; uint8 \_type

(setq \_type (sys::peek buf ptr- :char)) (incf ptr- 1)

;; float64[] \_damping

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_damping (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_damping i) (sys::peek buf ptr- :double)) (incf ptr- 8)

))

;; float64[] \_position

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_position (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_position i) (sys::peek buf ptr- :double)) (incf ptr- 8)

))

;; float64[] \_rate

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_rate (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_rate i) (sys::peek buf ptr- :double)) (incf ptr- 8)

))

;; bool \_success

(setq \_success (not (= 0 (sys::peek buf ptr- :char)))) (incf ptr- 1)

;; string \_status\_message

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setq \_status\_message (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

;;

self)

)

(defclass gazebo\_msgs\_new::GetJointProperties

:super ros::object

:slots ())

(setf (get gazebo\_msgs\_new::GetJointProperties :md5sum-) "7b30be900f50aa21efec4a0ec92d91c9")

(setf (get gazebo\_msgs\_new::GetJointProperties :datatype-) "gazebo\_msgs\_new/GetJointProperties")

(setf (get gazebo\_msgs\_new::GetJointProperties :request) gazebo\_msgs\_new::GetJointPropertiesRequest)

(setf (get gazebo\_msgs\_new::GetJointProperties :response) gazebo\_msgs\_new::GetJointPropertiesResponse)

(defmethod gazebo\_msgs\_new::GetJointPropertiesRequest

(:response () (instance gazebo\_msgs\_new::GetJointPropertiesResponse :init)))

(setf (get gazebo\_msgs\_new::GetJointPropertiesRequest :md5sum-) "7b30be900f50aa21efec4a0ec92d91c9")

(setf (get gazebo\_msgs\_new::GetJointPropertiesRequest :datatype-) "gazebo\_msgs\_new/GetJointPropertiesRequest")

(setf (get gazebo\_msgs\_new::GetJointPropertiesRequest :definition-)

"string joint\_name # name of joint

---

# joint type

uint8 type

uint8 REVOLUTE = 0 # single DOF

uint8 CONTINUOUS = 1 # single DOF (revolute w/o joints)

uint8 PRISMATIC = 2 # single DOF

uint8 FIXED = 3 # 0 DOF

uint8 BALL = 4 # 3 DOF

uint8 UNIVERSAL = 5 # 2 DOF

# dynamics properties

float64[] damping

# joint state

float64[] position

float64[] rate

# service return status

bool success # return true if get successful

string status\_message # comments if available

")

(setf (get gazebo\_msgs\_new::GetJointPropertiesResponse :md5sum-) "7b30be900f50aa21efec4a0ec92d91c9")

(setf (get gazebo\_msgs\_new::GetJointPropertiesResponse :datatype-) "gazebo\_msgs\_new/GetJointPropertiesResponse")

(setf (get gazebo\_msgs\_new::GetJointPropertiesResponse :definition-)

"string joint\_name # name of joint

---

# joint type

uint8 type

uint8 REVOLUTE = 0 # single DOF

uint8 CONTINUOUS = 1 # single DOF (revolute w/o joints)

uint8 PRISMATIC = 2 # single DOF

uint8 FIXED = 3 # 0 DOF

uint8 BALL = 4 # 3 DOF

uint8 UNIVERSAL = 5 # 2 DOF

# dynamics properties

float64[] damping

# joint state

float64[] position

float64[] rate

# service return status

bool success # return true if get successful

string status\_message # comments if available

")

(provide :gazebo\_msgs\_new/GetJointProperties "7b30be900f50aa21efec4a0ec92d91c9")