ikkuna.export.messages

Message

+tag : string {readOnly}
+global_step : int {readOnly}
+data : Tensor or tuple(Tensor)
 or number = None {readOnly}
+train_step : int {readOnly}
+epoch : int {readOnly}
+>kind : string {readOnly}
+key : object {readOnly}

MessageBundle

+identifier : string {unique} {readOnly}
+expected_kinds : list {unique}
{readOnly}
-_received : dict {unique}
+data : dict {unique} {readOnly}
+seq : int {unique} {readOnly}
+step : int {unique} {readOnly}
+epoch : int {unique} {readOnly}
+complete() : boolean
+add_message(message : Message) : void

Can contain data, but
doesn't have to

NetworkMessage

+key : string {unique} = 'META'
{readOnly}

ModuleMessage

+module : NamedModule {unique} {readOnly}
+key : NamedModule [1] {unique} = self.module
{readOnly}

MessageBus

+name : string {readOnly}
-_subscribers : Subscriber [0..*]
-_meta_kinds : string [0..*] = ikkuna.export.messages._META_KINDS
-_data_kinds : string [0..*] = ikkuna.export.messages._DATA_KINDS
+register_meta_topic() : void
+register_data_topic() : void
+deregister_meta_topic() : void
+deregister_data_topic() : void
+register_subscriber(sub : Subscriber) : void
+publish_network_message(global_step, train_step, epoch, kind : str, data : Tensor =
None)
+publish_module_message(global_step, train_step, epoch, kind : string, named_module :
NamedModule, data : Tensor)