

# PassiveReader

Singleton

```
-PassiveReader()
-init(inventory_listener: AbstractInventoryListener, reader_listener: AbstractReaderListener, response_listener: AbstractResponseListener)
+getInstance(inventory_listener: AbstractInventoryListener, reader_listener: AbstractReaderListener, response_listener: AbstractResponseListener)
+connect(reader_address: string)
+disconnect()
+close()
+isAvailable(reader_address: string): boolean
+getSecurityLevel()
+setSecurityLevel(level: enumeration)
+isHF(): boolean
+isUHF(): boolean
+testAvailability()
+sound(frequency: integer, step: integer, duration: integer, interval: integer, repetition: integer)
+light(led_status: boolean, led_blink: integer)
+getBatteryStatus()
+getFirmwareVersion()
+setShutdownTime(time: integer)
+getShutdownTime()
+setInventoryMode(mode: enumeration)
+setInventoryType(standard: enumeration)
+setInventoryParameters(feedback: enumeration, timeout: integer, interval: integer)
+setRfPower(level: integer, mode: enumeration)
+getRfPower()
+doInventory()
+getBatteryLevel()
+setRfForISO15693tunnel(delay: integer, timeout: integer)
+getRfForISO15693tunnel()
+setISO15693optionBits(option_bits: enumeration)
+getISO15693optionBits()
+setISO15693extensionFlag(flag: boolean, permanent: boolean)
+getISO15693extensionFlag()
+setISO15693bitrate(bitrate: enumeration, permanent: boolean)
+getISO15693bitrate()
+setEPCfrequency(frequency: enumeration)
+getEPCfrequency()
+ISO15693tunnel(command: byte[])
+ISO15693encryptedTunnel(flag: byte, command: byte[])
```

-response\_listener

-reader\_listener

-inventory\_listener

## AbstractReaderListener

```
+connectionFailedEvent(error: enumeration)
+connectionSuccessEvent()
+disconnectionSuccessEvent()
+securityLevelEvent(level: enumeration)
+availabilityEvent(available: boolean)
+resultEvent(command: enumeration, error: enumeration)
+batteryStatusEvent(status: enumeration)
+firmwareVersionEvent(major_number: integer, minor_number: integer)
+shutdownTimeEvent(time: integer)
+RfPowerEvent(level: enumeration, mode: enumeration)
+batteryLevelEvent(level: real)
+RfForISO15693tunnelEvent(delay: integer, timeout: integer)
+ISO15693optionBitsEvent(option_bits: enumeration)
+ISO15693extensionFlagEvent(flag: boolean, permanent: boolean)
+ISO15693bitrateEvent(bitrate: enumeration, permanent: boolean)
+tunnelEvent(data: byte[])
+EPCfrequencyEvent(frequency: enumeration)
```

## ReaderListener

```
+ReaderListener()
+connectionFailureEvent(error: enumeration)
+connectionSuccessEvent()
+disconnectionSuccessEvent()
+securityLevelEvent(level: enumeration)
+availabilityEvent(availability: boolean)
+resultEvent(command: enumeration, error: enumeration)
+batteryStatusEvent(status: enumeration)
+firmwareVersionEvent(major_number: integer, minor_number: integer)
+shutdownTimeEvent(time: integer)
+RfPowerEvent(level: integer, mode: enumeration)
+batteryLevelEvent(level: real)
+RfForISO15693tunnelEvent(delay: integer, timeout: integer)
+ISO15693optionBitsEvent(option_bits: enumeration)
+ISO15693extensionFlagEvent(flag: boolean, permanent: boolean)
+ISO15693bitrateEvent(bitrate: enumeration, permanent: boolean)
+EPCfrequencyEvent(frequency: enumeration)
+tunnelEvent(data: byte[])
```

## AbstractInventoryListener

```
+inventoryEvent(tag: Tag)
```

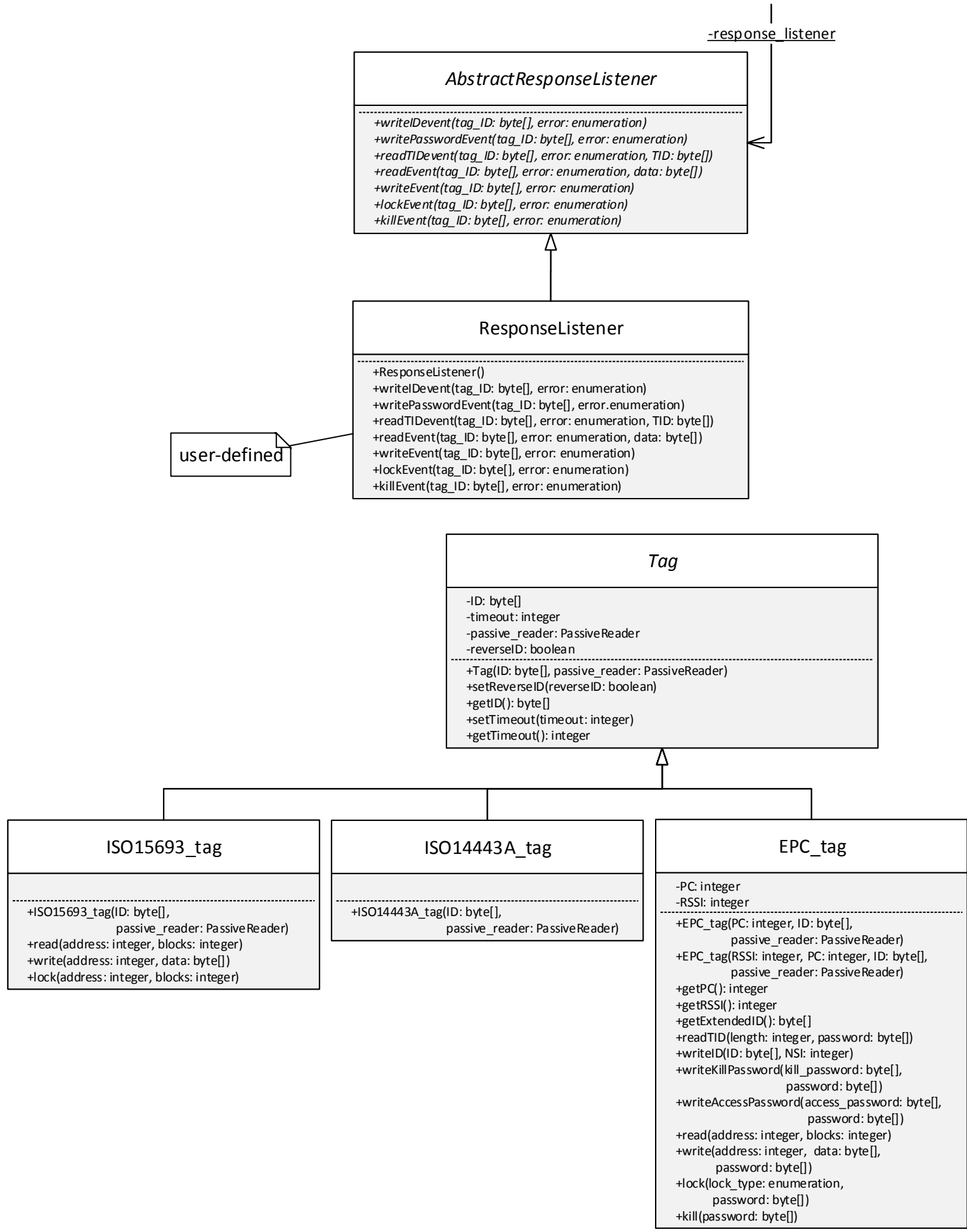
## InventoryListener

```
-tags: Tag[]
+InventoryListener()
+inventoryEvent(tag: Tag)
+reset()
+getTagsNumber(): integer
+getTags(): Tag[]
```

user-defined

user-defined

TERTIUM  
Technology  
NFC-scanner  
API  
2020-10-09



user-defined

