

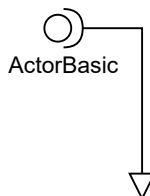
## SuspendableCore

- owner : ActorBasic {val}
- onMessageWhenNotSuspended : suspend (IApplMessage) -> Unit {val}
- suspended : Boolean
- ignoreMessageWhileSuspended : Boolean
- name : String
- mutSuspensionStateFlow : MutableStateFlow<SuspensionState> {val}
- + suspensionStateFlow : StateFlow<SuspensionState> {val}
- msgBuffer : ArrayDeque<IApplMessage>
- restoredMessage : IApplMessage?

SuspendableCore(ActorBasic, suspend (IApplMessage) -> Unit, Boolean, Boolean)

- + handleMessage(IApplMessage) {suspend}
- resume() {suspend}
- tryRestoreMsgFromBuff() {suspend}
- suspend() {suspend}

ActorBasicFsm



## <<Enum>> SuspensionState

**SUSPENDED**  
**RESUMING**  
**NORMAL\_WORK**

## SuspendableActorBasic

+ suspendableCore : SuspendableCore {val}

SuspendableActorBasic(String, CoroutineScope, Boolean, Boolean, Boolean, Boolean, Int)

- + onMessageWhenNotSuspended(IApplMessage) {suspend}
- + actorBody(IApplMessage) {suspend, override}
- + suspend() {suspend}
- + resume() {suspend}

## SuspendableActorBasicFsm

+ suspendableCore : SuspendableCore {val}

SuspendableActorBasicFsm(String, CoroutineScope, Boolean, Boolean, Boolean, Boolean, Int)

- + actorBody(IApplMessage) {suspend, override}
- + suspend() {suspend}
- + resume() {suspend}