Akka Actor-Based Model - Reference Guide

This reference guide provides an overview of the Akka actor-based model, including commonly used functions, their usage, and small code examples. Use this guide to quickly recall how Akka actors work and to get started with your projects.

1. Creating an Actor System

To create an ActorSystem in Akka, use the following code:

```
ActorSystem<MyMessageType> system = ActorSystem.create(MyActor.create(), "myActorSystem");
```

- ActorSystem: The entry point for creating and managing actors.
- MyMessageType: The type of messages the root actor will handle.
- MyActor: The root actor class.

2. Commonly Used Functions

2.1 createReceive()

Defines how the actor handles incoming messages. Example:

```
public Receive<MyMessageType> createReceive() {
    return newReceiveBuilder()
        .onMessage(MyMessageType.class, this::onMessage)
        .build();
}
```

- onMessage: Specifies the message type and the handler method.

2.2 getContext()

Provides access to the actor's context, allowing you to spawn child actors, log messages, and more.

Example:

```
ActorRef<ChildActor.Message> child = getContext().spawn(ChildActor.create(), "childActor");
```

- spawn: Creates a new child actor.

2.3 tell()

Sends a message to another actor asynchronously. Example:

```
actorRef.tell(new Message(), getContext().getSelf());
```

- actorRef: The ActorRef of the target actor.
- Message: The message to send.
- getSelf: Reference to the current actor.

3. Small Example

Here's a simple example of an actor that prints messages:

```
public class HelloActor extends AbstractBehavior<String> {
    public static Behavior<String> create() {
        return Behaviors.setup(HelloActor::new);
    }

    private HelloActor(ActorContext<String> context) {
        super(context);
    }
```

```
    @Override
    public Receive<String> createReceive() {
        return newReceiveBuilder()
            .onMessage(String.class, this::onMessage)
            .build();
    }

    private Behavior<String> onMessage(String message) {
        System.out.println("Received: " + message);
        return this;
    }
}
```

- This actor prints any string message it receives.