Spring Boot Microservices

Beginner to Guru

Introduction to Spring State Machine

What is a State Machine?

- A state machine can be loosely defined as anything with a set of known states.
- As a developer, you've already been writing state machines!
- A 'state machine' can be:
  - if - then - else
  - case statements
  - switch statements
- Each example reflects a 'state'

State Machine History

- The concept of a State Machine dates back to 1943 when McCulloch and Pitts wrote a paper referencing the concept of finite automata.
- Finite Automata consists of:
  - Finite set of states
  - Set of inputs
  - Initial State
  - Final State
  - Transition Function

State Machine Use Cases

- Common use cases for State Machines
  - Message (Event) based applications - ie New Order, Pay Order, Ship Order
  - Events get published based on state changes
  - UI Applications With Actions triggered by Use - Caps Lock On, Caps Lock Off
  - Application behavior changes based on known states

State Machine Terminology

- **States** - The specific state of the state machine. Finite and predetermined values.
  - Frequently defined in an enumeration
- **Events** - Something that happens to the system - may or may not change the state.
- **Actions** - The response of the State Machine to events. Can be changing variables, calling a method or changing to a different state
  - **Transitions** - Type of action which changes state
- **Guards** - Boolean conditions
- **Extended State** - State Machine variables (in addition to state)

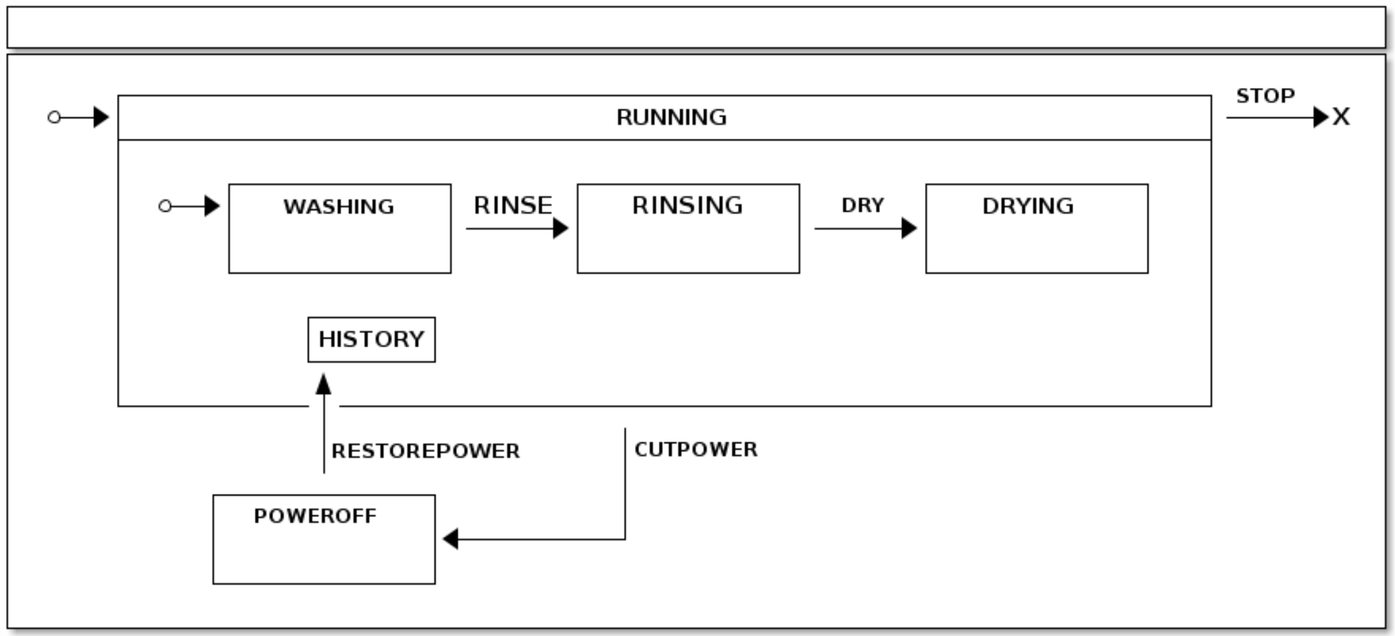
Why Use a State Machine?

- State Machines help define consistent behavior for a finite number of states
- Application logic is defined for specific states or state transitions
- Application logic becomes more modular and more precisely defined
- Long blocks of if, then, else if conditions are difficult to code, debug, and maintain
- Helps avoid spaghetti code for complex conditions

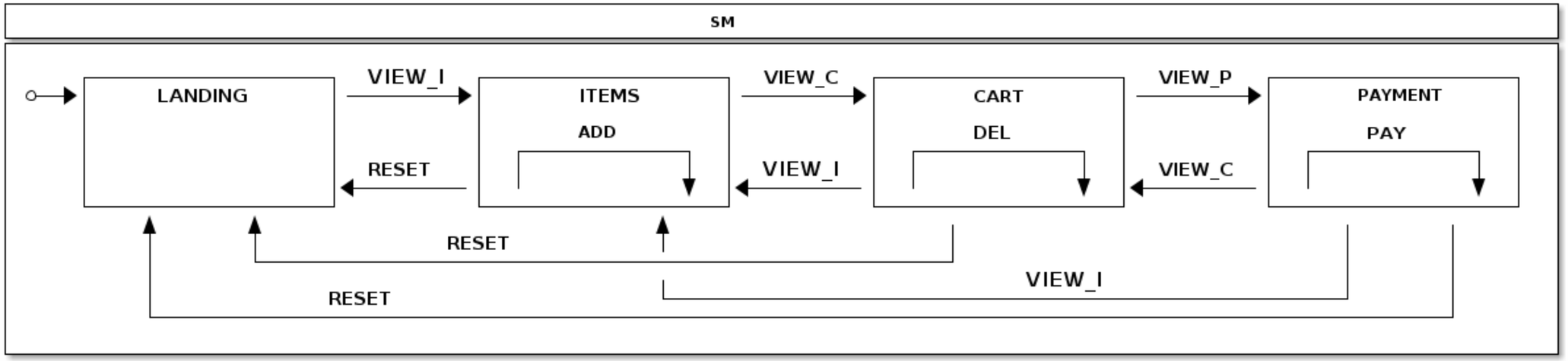
Spring State Machine

- Spring State Machine (SSM) is a mature Spring Framework project
- Initially released in October 2015.
- Current version is 2.1.3
- 3.x is under development, which will introduce non-blocking, reactive types
- SSM has a robust set of features and is integrated with the Spring Framework

Example 1 - Washing Machine



Example 2 - Web Checkout



SPRING FRAMEWORK

GURU