

Flow States Hierarchy

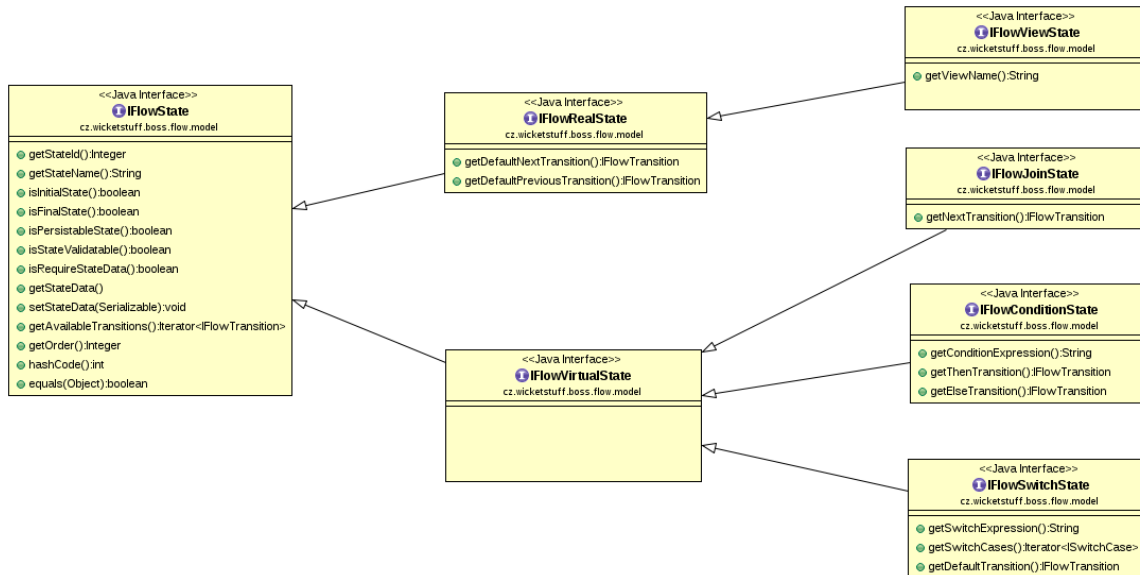
Hierarchy of Flow States

There are few kind of flow states. It is important to understand their hierarchy and meanings to prepare a flow XML file.

Two of most important kinds of states are:

- Real States
- Virtual States

See the UML of Flow States Hierarchy.



Flow States

Flow State is an abstraction of a state or a status. Also it can mean different things such as a use page or a webservice action. The main idea is, that each state can be change just by invoking a transition.

See [interface IFlowState](#).

Initial State

There is none Java interface for an initial state. The initial state is just a behaviour property of any states marked by **isInitialState() == true;**

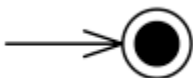


Final State

There is none Java interface for an final state. The final state is just a behaviour property of any states marked by **isFinalState() == true;**

If the flow is in a final state, it is not possible to invoke any other transition. Final state means that flow has been finished forever.

No state can be marked both initial as final state because that is a nonsense one-step flow.



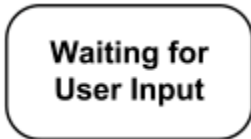
Real States

Real states are state where the flow can stand by. There is no need to stand by in a state infinitely until a user action happens.

Real State

Real State is the ancestor of all real states.

See [interface IRealState](#).



View State

View state is for example a web page or a single step of a form. There is really a small difference between Real State and View State.

View State just contains viewName attribute to simply hold that information. The main idea is to get an opportunity to make difference between an internal state (Real State or its descendant) and a view or a page (View State).

See [interface IFlowViewState](#).



Virtual States

Virtual State

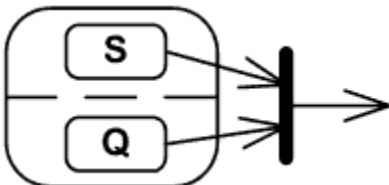
Virtual state is a state that is ever passed through. There is no case when flow can stand by a virtual state. Virtual State must be ever processed and shift flow to another state until one of Real States is reached.

See [interface IFlowVirtualState](#).

Join State

Join State is just a conjunction of transition that can continue itself to a next state, using next transition.

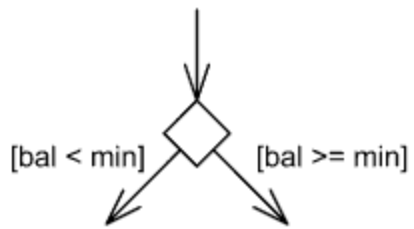
See [interface IFlowJoinState](#).



Condition State

Condition State is similar to Java 'if' statement. It must have ever defined two transitions for 'next' and 'else' cases.

See [interface IFlowConditionState](#).



Switch State

Switch State is similar to Java **'switch'** statement. The default next transition is just mandatory for this state.

See [interface IFlowSwitchState](#).

