

# Stateful Controls Proposal

## Table of content

- [Background](#)
- [Proposal](#)

## Background

With the [deprecation of stateful](#) pages we need a way to preserve control state across requests.

## Proposal

Add a new interface that stateful controls can implement:

### Stateful.java

```
interface Stateful {

    public Object getState() {
        // Assemble state and return as object. This allows developers to save state in cookie instead.
        // It also allows parent containers to assemble its children state recursively
        // If this control is a container it should assemble all its child control state.
    }

    public void setState(Object state) {
        // Cast object back to its type and set state to this control and its children
    }
}
```

Some useful utilities include:

### ClickUtils.java

```
public static void saveState(Stateful stateful, Context context) {
    // public static void saveState(Stateful stateful, Context context, Map options)??
    // By default save state in session in a map that is stored against the request page path
    // This method can call stateful.getState to retrieve this components state
    // Optimization includes GZIP compressing the state
}

public static void restoreState(Stateful stateful, Context context) {
    //public static void restoreState(Context context, Map options)??
    // Grab state from session and invoke stateful.setState
}
}
```

Each control will be responsible for storing its own state as well as its children. It's up to the control or container how to best store the state, but should keep the following in mind:

- state stored in the session is replicated in a cluster and should be as compact as possible eg: object array instead of a List
- while arrays are compact and has order, child controls can be reordered meaning the wrong state could be applied to a control. In this instance a Map would be a more flexible solution by storing the control state against its name, and structural order doesn't matter, as long as the controls are moved around inside that container only. If a control is moved to another container it won't be possible to restore its state automatically, however it would still be possible for the developer to find the original state and apply it manually on the changed structure.