

ClientStateStorageImpl

```
package scm.hivemind.statefulservice;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import javax.servlet.http.HttpSession;
import javax.servlet.http.HttpSessionBindingEvent;
import javax.servlet.http.HttpSessionBindingListener;

import org.apache.commons.logging.Log;

/**
 *
 * @author Marcus Schulte
 */
public class ClientStateStorageImpl implements ClientStateStorage {
    protected Log myLog;

    private HttpSession session = null;
    private boolean invalidSession = false;
    private Map clientStores = new HashMap();
    private Map map = new HashMap();

    private class StoredProxy implements HttpSessionBindingListener {
        private StateStorageClearanceListener clearanceListener;

        public StoredProxy ( StateStorageClearanceListener object ){
            clearanceListener = object;
        }

        public void valueBound( HttpSessionBindingEvent arg0 ) {
            // TODO Auto-generated method stub
        }

        public void valueUnbound( HttpSessionBindingEvent e ) {
            invalidSession = true;
            clearanceListener.clientStateCleared();
            myLog.debug("Session expired or invalidated.");
        }

        public StateStorageClearanceListener getObject() {
            return clearanceListener;
        }
    }

    /** (non-Javadoc)
     * @see com.bmw.sample.ClientStateStorage#provideSession(javax.servlet.http.HttpSession)
     */
    public void provideSession(HttpSession s ) {
        this.session = s;
        this.invalidSession = false;
    }

    public void setLog( Log log ) {
        myLog = log;
    }

    /** (non-Javadoc)
     * @see com.bmw.sample.ClientStateStorage#store(java.lang.String, java.lang.Object)
     */
    public void store(String key, StateStorageClearanceListener obj) {
```

```

        if (session==null) {
            myLog.debug( "storing key: "+key );
            map.put(key,obj);
        } else {
            if ( ! invalidSession ) {
                StoredProxy p = (StoredProxy) session.getAttribute( key );

                // avoid unbind-events due to replacing an object with itself...
                if ( p!=null && p.getObject()==obj)
                    return;

                session.setAttribute( key, new StoredProxy(obj) );
            } else
                myLog.debug("skipping store due to invalid Session");
        }
    }

    /* (non-Javadoc)
     * @see com.bmw.sample.ClientStateStorage#retrieve(java.lang.String)
     */
    public StateStorageClearanceListener retrieve(String key) {
        if (session == null) {
            myLog.debug( "retrieving key: "+key );
            return (StateStorageClearanceListener) map.get(key);
        } else {
            StoredProxy p = (StoredProxy) session.getAttribute(key);
            if ( p != null )
                return p.getObject();
            else
                return null;
        }
    }

    /* (non-Javadoc)
     * @see com.bmw.sample.ClientStateStorage#setClientId(java.lang.String)
     */
    public void setClientId(String cid) {
        Map m = (Map) clientStores.get( cid );
        if ( m == null) {
            m = new HashMap();
            clientStores.put( cid, m );
        }
        map = m;
    }

    /* (non-Javadoc)
     * @see com.bmw.sample.ClientStateStorage#clear()
     */
    public void clear( String clientId ) {
        Map store = ( (Map)clientStores.get(clientId) );
        for ( Iterator it = store.values().iterator(); it.hasNext(); ) {
            StateStorageClearanceListener obj
                = (StateStorageClearanceListener) it.next();
            obj.clientStateCleared();
        }
        store.clear();
    }
}

```