8

Creating a Session Persistence Module

GlassFish Server enables you to create a session persistence module in the web container for high availability-related functionality by extending the StrategyBuilder interface . Using the PersistenceStrategyBuilder class in an HK2 service makes the session manager extensible because you can implement a new persistence type without having to modify the web container code.

For information about other high-availability, session persistence solutions, see Chapter 9, "Configuring High Availability Session Persistence and Failover," in *GlassFish Server Open Source Edition 3.1 High Availability Administration Guide*.

The following topics are addressed here:

Implementing the **PersistenceStrategyBuilder** Class

You can implement the PersistenceStrategyBuilder class by creating a new web session manager type.

```
package com.sun.enterprise.web;

import com.sun.enterprise.deployment.runtime.web.SessionManager;
import org.apache.catalina.Context;
import org.jvnet.hk2.annotations.Contract;

@Contract
public interface PersistenceStrategyBuilder {

    public void initializePersistenceStrategy(
            Context ctx,
            SessionManager smBean,
            ServerConfigLookup serverConfigLookup);

    public void setPersistenceFrequency(String persistenceFrequency);
```

```
        public void setPersistenceScope(String persistenceScope);

        public void setPassedInPersistenceType(String persistenceType);
}
```

Here is an example of how to implement the `PersistenceStrategyBuilder` class by creating a new web session manager and setting a store for it:

```
@Service(name="xyz")
public class XYZStrategyBuilder implements PersistenceStrategyBuilder {

    private String persistenceFrequency = null;
    private String persistenceScope = null;
    private String persistenceType = null;

    public void init(StandardContext ctx, SessionManager sessionManager,
      ServerConfigLookup serverConfigLookup) {
        // add listeners, valves, etc. to the ctx
        // Set the manager and store
    }

    public void setPersistenceFrequency(String persistenceFrequency) {
        this.persistenceFrequency = persistenceFrequency;
    }

    public void setPersistenceScope(String persistenceScope) {
        this.persistenceScope = persistenceScope;
    }

    public void setPassedInPersistenceType(String persistenceType) {
        this.passedInPersistenceType = persistenceType;
    }
}
```

If a `Manager` is provided, then it will be used in GlassFish Server.

Note – If a backing store is required, it is the responsibility of the `Manager` to make sure that the `findSession` method correctly uses the `Store` that the `Manager` provides.

EXAMPLE 8–1   Implementing `PersistenceStrategyBuilder` With a Custom Web Session Manager

This example defines a session manager type that is named `MyHASolution`.

```
@Service(name="MyHASolution")
public class MyHASolutionStrategyBuilder implements PersistenceStrategyBuilder {

    private String persistenceFrequency = null;
    private String persistenceScope = null;
    private String persistenceType = null;

    public void init(StandardContext ctx, SessionManager sessionManager,
      ServerConfigLookup serverConfigLookup) {
        // add listeners, valves, etc. to the ctx
        // Set the manager and store
```

EXAMPLE 8–1   Implementing `PersistenceStrategyBuilder` With a Custom Web Session Manager *(Continued)*

```
        MyManager myManager = new MyManager(persistenceType, persistenceFrequency);
        // (You could also make this a service and look it up in the habitat.
        // For simplicity we are just doing a new implementation of the class here.)
        MyStore store = new MyStore();
        myManager.setStore(store);
        ctx.setManager(myManager);
    }

    public void setPersistenceFrequency(String persistenceFrequency) {
        this.persistenceFrequency = persistenceFrequency;
    }

    public void setPersistenceScope(String persistenceScope) {
        this.persistenceScope = persistenceScope;
    }

    public void setPassedInPersistenceType(String persistenceType) {
        this.passedInPersistenceType = persistenceType;

    }

}
```

EXAMPLE 8–2   Session Manager Configuration in the `glassfish-web.xml` File

This example sets the `persistence-type` attribute of the `session-manager` element of `glassfish-web.xml` to `myHASolution`

Based on the `domain.xml` and `glassfish-web.xml` settings, the web container looks up the appropriate `PersistenceStrategyBuilder` in the Habitat and uses it.

```
        <glassfish-web-app>
          <session-config>
            <session-manager persistence-type="myHASolution"/>
          <session-config>
        <glassfish-web-app>
```