If you are using Turbine only for a MVC framework, and are not trying to use the built in security features, then you need to do a couple things to make Turbine happy. Turbine requires an Anonymous user to be available, however, if you never try and save that Anonymous user, then the fact that there is no persistent data store (a user database) is fine. To configure things, set these values in your TurbineResources.properties:

Use the default security service

```properties
services.SecurityService.classname=org.apache.turbine.services.security.db.DBSecurityService
```

Make sure you aren't trying to persist any users

```properties
services.SecurityService.user.manager = org.apache.turbine.services.security.passive.PassiveUserManager
```

You will still be able to do things like:

```java
User user = data.getUser()
user.setTemp("myval","something to store");
```

but you will always be working with an anonymous user!

One more note. If you use the PassiveUserManager, you cannot use any of the secure Screen or Action classes that ship with Turbine because you have no way of logging a user in. For an essentially insecure app, this is fine however. And if you wish to use your own security implementation, you can easily create your own secure Screen and Action class hierarchy.

Also, this does not remove your dependency on the Torque and Village jar files. You still need those in your lib dir. Until the turbine User object is fully abstracted from torque, this will continue to be the case.