Hibernate - Core - Conf

Hibernate Configuration is handled by the Tapestry Hibernate Core module. This is done in a just-in-time manner, the first time a Hibernate Session is required.

Related Articles

- Hibernate - Core - Conf
- Hibernate - Core
- Hibernate
- Using Tapestry With Hibernate
- Hibernate User Guide
- Hibernate Support FAQ
- Hibernate Statistics

HibernateSessionSource Configuration

One way to configure hibernate is to create a `hibernate.cfg.xml` file and place it in the root of your application (i.e., under src/main/resources). Most Hibernate-specific configuration occurs in this file. Another way is to contribute objects that perform configuration (such as setting event listeners). Example:

```
public static void contributeHibernateSessionSource(OrderedConfiguration<HibernateConfigurer> config)
{
  config.add("Widget", new WidgetHibernateConfigurer());
}
```

Note that the configuration is an OrderedConfiguration. The library contributes two configurers by default:

- **Default** - performs default hibernate configuration
- **PackageName** - loads entities by package name as contributed to the HibernateEntityPackageManager service

HibernateEntityPackageManager Configuration

This configuration is a set of package names, identifying where to search for entity classes.

For each package contributed, the library will:

- Add the package to the configuration, which will load annotations from the package-info class within the named package, if present.
- Every Java class in the package (or any subpackage) will be added as an annotated class. This excludes inner classes, but includes all other classes.

By default, the package *application-root-package.entities* is scanned as described above. If you have additional packages containing entities, you must contribute them to the tapestry.hibernate.HibernateEntityPackageManager service configuration.

Example:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
public static void contributeHibernateEntityPackageManager(Configuration<String> configuration)
{
  configuration.add("org.example.myapp.domain");
}
```

You may add as many packages in this manner as you wish. This option is most often used when the entities themselves are contained in a library included within an application, rather than part of the application directly.

Hibernate Symbols

The Hibernate integration includes a number of symbols used to control certain features:

tapestry.hibernate.provide-entity-value-encoders	If true (the default) then ValueEncoders are automatically provided for all Hibernate entities (ValueEncoders are used to encode the primary keys of entities as strings that can be included in URLs). Set to false if you want direct control over this feature.
tapestry.hibernate.default-configuration	If true (the default), then the application must include a `hibernate.cfg.xml` file. If your application configures itself entirely in code, you should set this symbol to false.
tapestry.hibernate.early-startup	If "true", the Hibernate is initialized when the application starts up. The default is "false", to start Hibernate up lazily, on first use.