Technical Information - SweetCherrySync

1. *Which CherryTree file structures are supported?*

   Only the files with the CTB extension are supported. Actually they are SQLite database files.

2. *Which CherryTree versions are supported?*

   Version 1.1.3 is supported (without any responsibility).

   You need to be careful, because a new version can modify your database structure.

   Then the sync process will have no guarantee, errors or missing data on the destination datasource can happen.

   For example, the shared nodes feature made a small change in the database structure. But it resulted a major update over my codes.

   Therefore, make keep backup of your CTB files and CherryTree application versions.

3. *Is syncing safe?*

   If you set the "from" and "destination" datasource settings correctly, then your source file is safe.

   Because it is one directional sync process. From source database to destination database.

   Anyway, use it at your own risk. You can first try it on a copy of the database. If you have a smaller test database(CTB file), the first sync duration will be significantly less.

4. *Can I use the CherryTree application with SweetCherrySync at the same time?*

   Yes. Because SweetCherrySync is using the source database only for reading the data.

5. *How to change some default settings?*

   Create an empty file named application.properties in the same folder of the application file.

6. *How to add variables and their values into application.properties file?*

   For each beginning of the lines, use this format below;

   ```
   variable.name=value
   ```

7. *How to change port?*

   Add server.port variable and its value into application.properties file.

8. *How to change sync duration?*

   Add my.scheduler.delayInSeconds and its value into application.properties file.

9. *How to configure the source and destination datasource connection settings?*

   This is the most important question.

   The answer has its own documentation page. Read it, it will show you how to apply correctly.

10. *How to run?*

    First you need to have your application.yml file with your desired settings. Then you can run the app by following these options below:

    Run by typing;

    ```
    java -jar SweetCherrySync.java
    ```

    command. + If you want to compile it from source, read my general installation page.

11. *How to stop?*

    Press **CTRL** + **C** keys to break and stop the application from the command line if you started it within a terminal window before.

12. *How the application starts?*

    `SweetCherrySyncApplication.java` is the main starting point. But actually the `SchedulerConfig.java` is doing the job.

13. *Where is sync mechanism?*

    Examine SchedulerConfig.java file, syncDatabases() method.

14. *Is there a web GUI to monitor sync results?*

    http://localhost:<port number>/

    default is:

    http://localhost:8081/

    Its controller file is SettingsController.java.

    Its html file is index.html.

    You need to refresh the page manually.

    For each beginning of the sync interval, MyService.java resetStatistics() method resets all comparison counters.

    Some counters need to be tested or have logical misinformation. But looking to the table row counters and seeing their equality can make you satisfied.

15. *Where are those counters and the other keys value pairs are stored?*

    Examine the application.yml file inside the project source.

    The variables starting with my.datasource key are related with the stored keys and values.

    By default it is a H2 database file.

    You can change the datasource by overwriting them using an application.properties file.

    The source code classes related with it are located under the "my" folder.

This database is for storing various keys and values during application run.

```
./
```

represents the application folder.

Example:

```
my.datasource.url=jdbc:h2:file:./SweetCherrySync
```

If you set the database name to SweetCherrySync, on your filesystem you will see it as

```
SweetCherrySync.mv.db
```

because it adds ".mv.db" automatically.

16. *Can I sync manually with a command?*

    Yes.

    NodeController.java is responsible for that.

    add /sync to your server URL and type enter.

    Example:

    ```
    http://localhost:8081/sync
    ```

    It will call the synching method.

    Because there is no setting to disable the scheduler, as a workaround, set its value to very high as seconds. (ugly solution).

    But this way the application runs a sync during startup too.

    Or wait until I develop a better solution or develop for yourself.

    I dont know what will happen if you call it manually during scheduled sync.

    Always do regular backups using the backup settings feature of the CherryTree application or do it manually.

17. *How the SweetCherrySync software can differentiate source and destination databases technically?*

    In a normal situation, Java Spring Boot application is configured for a single database by default.

    In our case, to be able to use two datasources, we are using these config files below:

```
FromDataSourceConfig.java
DestinationDataSourceConfig.java
```

And we have separate repository classes for both datasources.

You can differentiate them by examining the repositories folder.

Other classes(entities and services) are shared by both datasources.

Some classes are specific to one of those databases, but you can differentiate them easily by looking their names, for example;

```
DestinationRecordCountService.java
FromRecordCountService.java
```

18. *Any questions?*

    Ask to me.

Author: Türker Öztürk