

Mobile Application Development

SQLite database applied to MyRent

Waterford Institute of Technology

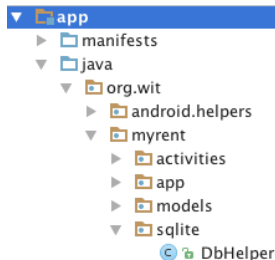
October 23, 2016

John Fitzgerald

SQLite

Relational Database Management System (DBMS)

- SQLite implemented in earlier lab.
- Here we integrate in MyRent
 - Create package **sqlite**.
 - Add copy of *DbHelper.java*
 - No changes necessary.
 - Remove references to *PortfolioSerializer*.
 - Add Residence update v save in *onPause*.
 - Refactor MyRentApp.
 - Refactor Model.



ResidenceFragment

Residence object state persisted in *onPause*

```
@Override
public void onPause() {
    super.onPause();
    portfolio.updateResidence(residence);
}
```

```
@Override
public void onPause() {
    super.onPause();
    portfolio.saveResidences();
}
```

MyRent Application

Refactor MyRentApp.java

- Remove references to PortfolioSerializer.

MyRent Portfolio

Refactor Portfolio.java

- Remove references to PortfolioSerializer.
- Introduce DbHelper.
- Support code to invoke update residence object.
- Recall: primary key unchanged in update.

MyRent Model

Refactor Residence.java

- Remove references to JSON:
 - JSON String constants (example String JSON_DATE).
 - Constructor containing JSONObject argument.
 - toJSON method.
 - JSON import statements.

Portfolio

DbHelper methods invoked

```
// R(ead)
/**
 * Obtain the entire database of residences
 * @return All the residences in the database as an ArrayList
 */
public ArrayList<Residence> selectResidences() {
    return dbHelper.selectResidences();
}
```

Portfolio

DbHelper methods invoked

```
// C(reate)
/**
 * Add incoming residence to both local and database storage
 * @param residence The residence object to be added to local and database
 * storage.
 */
public void addResidence(Residence residence) {
    residences.add(residence);
    dbHelper.addResidence(residence);
}
```


Portfolio

DbHelper methods invoked

```
// U(pdate)
/**
 * Update a residence record – primary key unchanged
 * @param Residence record for update
 */
public void updateResidence(Residence residence) {
    dbHelper.updateResidence(residence);
    updateLocalResidences(residence);
}
```

Portfolio

DbHelper methods invoked

```
/**
 * // D(elete)
 * Delete Residence object from local and remote storage
 * @param residence Residence object for deletion.
 */
public void deleteResidence(Residence residence) {
    dbHelper.deleteResidence(residence);
    residences.remove(residence);
}
```



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology
INSTITIÚD TEICNEOLAÍOCHTA PHOIRT LÁIRGE

