CS3733-D22 Team X API installation

By: Team X (Theo Coppola)

Your MealRequestAPI and you!

Running the API:

The run method is located at **MealRequestAPI.run()**

For optimal sizing use windowWidth and windowLength 700

The database used to store everything in the API is called “MealRequestAPI_embed_db”

Accessing created MealRequests:

To get a MealServiceRequest that was created through the API you can use...

- **MealServiceRequestDAO.getDAO().getAllRecords()** which returns a list of all MealServiceRequests. (Note: getAllRecords() may return service requests that you did not create. This is because the database is autofilled with example service requests to refer to. This should not interfere with the creation of new service requests)
- **MealServiceRequestDAO.getDAO().getRecord(String requestID)** which returns the MealServiceRequest with the specified requestID. RequestIDs are created automatically when a service request is submitted, and they are unique.

Entity classes:

These all have getters and setters for their attributes

All entities have a DAO and are accessible from the **entity** package

MealServiceRequest

- Has a unique requestID defining the object. Has attributes pertaining to choiceboxes in the UI including destination, assignee, status, mainCourse, side, drink, and patientFor
- For status: the only possible values are “”, “PROC”, and “DONE”.
- It extends ServiceRequest, an abstract class used by Team X for all service request classes

Employee

- Has a unique employeeID defining the object. Has the Strings firstName and lastName.
- Use **EmployeeDAO.getDAO().addRecord(Employee recordObject)** to add a new Employee object to the database. You can use **EmployeeDAO.getDAO().makeID()** to make a new employeeID to use for recordObject

- Use **EmployeeDAO.getDAO().deleteRecord(Employee recordObject)** to remove an Employee object from the database.
- Use **EmployeeDAO.getDAO().updateRecord(Employee recordObject)** to update an existing employee in the database.

Location

- Has a unique nodeID defining the object. Locations are automatically put into the database upon calling run().
- Locations can be retrieved from their nodeID by using **LocationDAO.getDAO.getRecord(String nodeID)** which returns a Location object.

Saving to a CSV:

To create CSVs with the data from the database use the method **DatabaseCreator.saveAllCSV(String dirPath)** where dirPath is the path to the directory where the csv will be saved. If you would rather not choose a path, input “” for dirPath.