

Hi, The TableTop team,

This is Yi Shen, I am the backend person for ShareHome team. The other backend member is Yifei Feng. If you have any questions regarding testing backend, please feel free to send emails to yshen59@wisc.edu, or yfeng59@wisc.edu, we're happy to help!

We are using ZoHo as an external bug tracker, please contact yshen59@wisc.edu, and I'll send you the invitation to the Zoho system for bug tracking.

Our git repo is here: <https://github.com/syncnj/ShareHome>

Here's an overview of the backend system used by the ShareHome team:

1. We use Backendless as our backend platform. Due to some lack of documentation (also the beta stage) of their newest 4.0 Version, we implemented our service based on Version 3.0
2. For security reasons, I apologize for not being able to give you the username and password to our Backendless platform. But you should be able to register a new account on backendless, and deploy all the services (the JAR file generated) to the Custom Business Logic --- API service.
3. Since you will be using a different username and password for testing, please don't forget to change the corresponding APP_ID and Secret_Key in the Android & iOS source code, so that when you run the Android / iOS code for testing, they will connect to the correct (your) Backendless account, and you will be able to see any changes made by those two platforms.
4. I know the documentations for Backendless is sometimes vague, you can always send emails to us so we could explain them.

Here's what the backend team does:

1. We wrote many backend API services, which Android / iOS will call to perform operations like create a new group, add new users to a group, create a new task...

Here're some detailed explanations:

2. Grocery Service: the service which deals with Grocery Lists, it includes add GroceryList item, getGroceryListBy ID, change Status (which will loop the status of the List in sequence),
3. Group Service, it contains a lot of information about the Group, and also many other requests like getAllGrocery, getAllPost...
4. Transaction is most recently written, so there might be some bugs in there, since we're waiting for the two front-end teams to fully integrate it and test it out.
5. Timers: We're using timers to solve the tasks' rotation problems. Here's how it works: In the table (Data) task, we're having the startTime and the duration (it should be called frequency or period, and is measured in hours). Basically, it the timer event will check for all of the tasks, and see if there startTime is the time that's in the past, if it's then we rotate the list, and increment the startTime by the duration, so that it's now a future time.

6. The schema of the tables should be automatically created when you create any of the Posts / Tasks ... If not, the schema file is in the git repo under Server folder, there's a file called TableSchema.zip
7. Always feel free to ask us about any problems! Backendless can be sometimes frustrating, and we've been through this, so if any problems, just ask us.