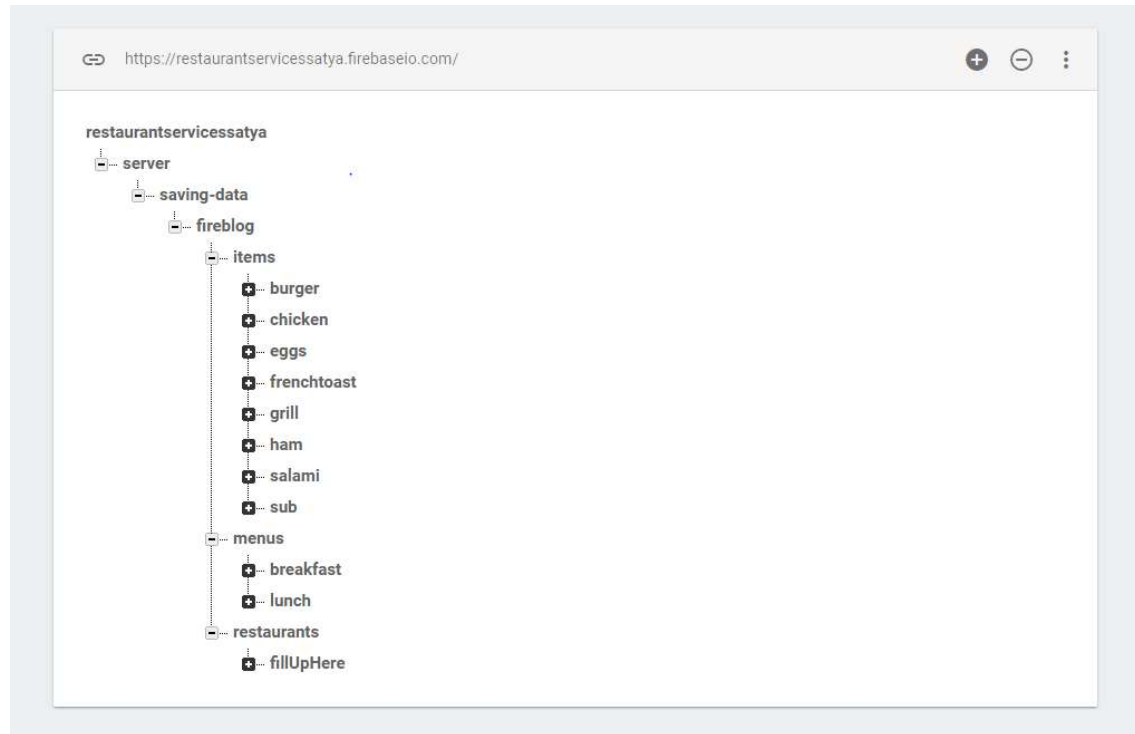


### Online Food Ordering Related Rest Services:

- 1) Created a set of Restful services which allow the user to perform CRUD operations on food items, Menus, Restaurants.
- 2) Each item, Menu and Restaurant is Identified by its name which is unique.
- 3) Used Firebase DB to store data. This allows me to avoid creating local databases every time I move the code.



**Firebase data structuring**

- 4) Created the Rest service layer with Spring and Java. Used Maven for dependency management.
- 5) Test cases were written using SOAPUI as it is suitable for testing Rest services. Test Suite xml file is attached.
- 6) Once hosted, the base url would be "host:port/restaurantesservices-0.0.1-SNAPSHOT".
- 7) All the urls and corresponding services are listed below.

<b>Method</b>	<b>URL</b>	<b>Function</b>	<b>Input/Output</b>
GET	{baseUrl}/getItem	Fetches all available items	Outputs Json with all available items
GET	{baseUrl}/getItem?itemName={name}	Fetches item with itemName = "name"	Outputs Json : item with itemName : "name"
POST	{baseUrl}/addItem	Adds one or more items; Updates the item If an item with the itemName already available in DB	Takes Json Input of items being added/updated
DELETE	{baseUrl}/deleteItem?itemName={name}	Deletes item by itemName	Takes itemName as input
GET	{baseUrl}/getMenu	Fetches all available Menus	Outputs Json with all available Menus
GET	{baseUrl}/getMenu?name={name}	Fetches menu with name = "name"	Outputs Json : menu with name : "name"
POST	{baseUrl}/addMenus	Adds one or more menus; Updates the menu If a menu with the given name already available in DB	Takes Json Input of menus being added/updated
DELETE	{baseUrl}/deleteMenu?name={name}	Deletes menu by name	Takes name as input
GET	{baseUrl}/getRestaurant	Fetches all available restaurants	Outputs Json with all available restaurants
GET	{baseUrl}/getRestaurant?name={name}	Fetches restaurants with name = "name"	Outputs Json : restaurant with name : "name"
POST	{baseUrl}/addRestaurants	Adds one or more restaurants; Updates the restaurant If name already available in DB	Takes Json Input of restaurants being added/updated
DELETE	{baseUrl}/deleteRestaurant?name={name}	Deletes restaurant by name	Takes name as input