

ID: b110

Name: Wang Yaochong

Source Code Appeal

* Appeal and describe your source code concisely.

i.e. What did you consider while you were programming?

1. Don't repeat yourself.

Keep only one piece of code for the same logical process. It's easy for you to make logical change on code. Otherwise you need to find all code of that logic and it's easy to make mistake.

2. Keep consistency on code with similar operation logic. (function name/ invoke style)

For similar logic process like such as getting paged data/response value operation/rest API invocation, etc. It can help you a lot while you're coding, because the data structure and invoke style are very consistent. You can code more and more fast on similar code.

3. Good naming on variables and functions.

Good naming on variables and functions help a lot on code comprehension.

4. Clear directory structure.

Clear directory structure helps a lot when you want to locate some files and make some change, it is also useful for you to understand the whole structure and maintain the code.

5. Try to break large function into small ones.

A complicated function with well named small functions is much easier to understand than a whole piece of code.

6. Try to think whether there is a better way to solve the problem.

Thinking in this way can absolutely improve your programming abilities and save your precious coding time.

*Appeal and describe the specific part of your source code that you would like to appeal most.

You may give us the specific packages, class names, functions and lines if you can.

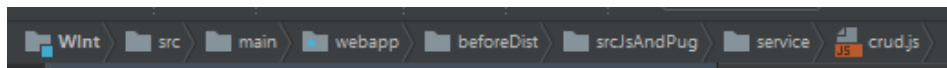
i.e. What was the intention of implementing the specific part?

Code example :

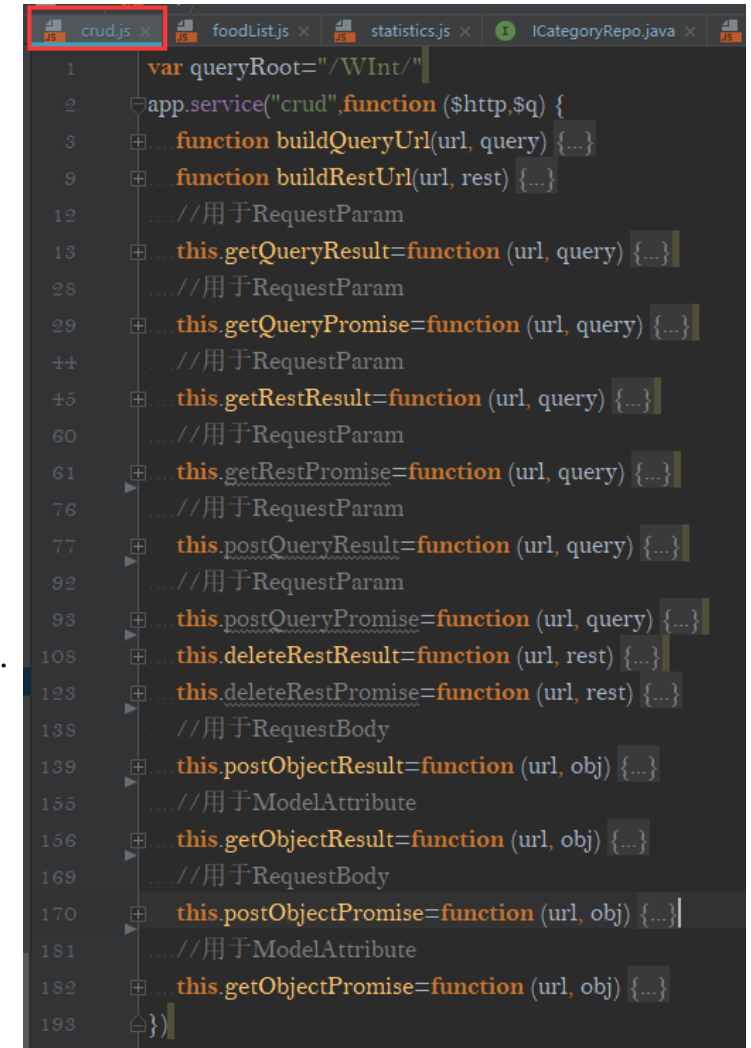
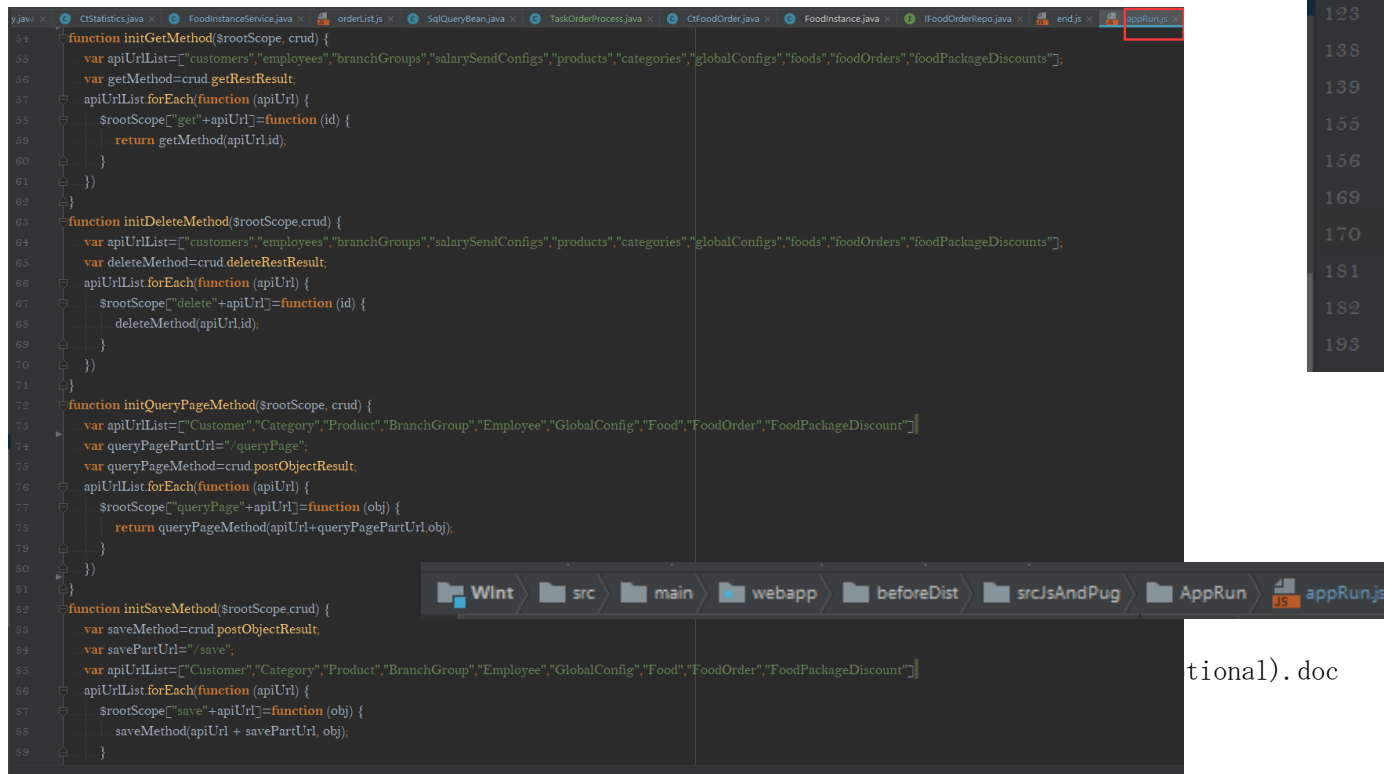
1. Don't repeat yourself.

When I make coding on my software, I need to write a lot of functions to make create/read/update/delete operation both backend and frontend, I don't want to write CRUD functions for every different entity. I made great improvement on frontend. I can use very few lines of code to finish CURD operation.

① Defined common http method inside crud.js



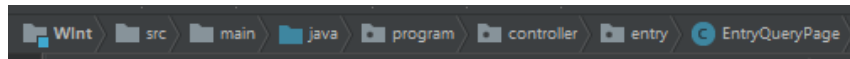
② Build CRUD methods in appRun.js. I can configure CRUD method with string list.



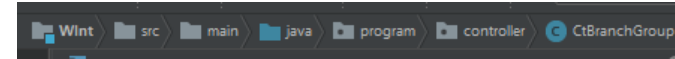
tional). doc

2. Keep consistency on code with similar operation logic. (function name/ invoke style)

When I want to query one page of entities, I used EntryQueryPage<T> to help, and I can get consistent operation for all entities, I take BranchGroup and Food for example. The queryPage method of these two entities are almost the same.



```
@Data
public class EntryQueryPage<T> {
    ....T condition;
    ....Integer pageNum;
    ....Integer pageSize;
    ....String orderBy;
    ....Boolean orderAsc;
}
```



```
@RequestMapping("/queryPage")
public ResponseInfo queryPage(@RequestBody EntryQueryPage<BranchGroup> query){
    Page<BranchGroup> customers = branchGroupService.queryPage(
        ....query.getCondition(),
        ....query.getPageNum(),
        ....query.getPageSize(),
        ....query.getOrderBy(),
        ....query.getOrderAsc()
    );
    ResponseInfo responseInfo = new ResponseInfo();
    responseInfo.setData(customers);
    return responseInfo;
}
```



```
@RequestMapping("/queryPage")
public ResponseInfo queryPage(@RequestBody EntryQueryPage<Food> query){
    Page<Food> customers = foodService.queryPage(
        ....query.getCondition(),
        ....query.getPageNum(),
        ....query.getPageSize(),
        ....query.getOrderBy(),
        ....query.getOrderAsc()
    );
    ResponseInfo responseInfo = new ResponseInfo();
    responseInfo.setData(customers);
    return responseInfo;
}
```

3. Good naming on variables and functions.

I try to make better name for variables and functions for all my code and I used Hump naming rules.

4. Clear directory structure.

My project directory structure is shown left. I tried to make well directory management and I think this directory is good, it helps me a lot.

5. Try to break large function into small ones.

I wrote a complicated function to make initialization over my store data. In this way, I can clearly locate my code. If I write this function with only one piece of code, it would be a mess.

6. Try to think whether there is a better way to solve the problem.

When I want to find order number for each day, I firstly wrote



```
@Query nativeQuery = true,value = "SELECT DATE_FORMAT(orderBeginDateTime,'%Y/%c/%d') as'date' ,count(*)  
List<Object[]> getDateAndOrderCountListByBranchGroup @Param("branchGroupId") String branchGroupId ;
```

When I want to find order number for each day, I firstly wrote but I think If it better to get extracted data,

so I wrote another function.



```
public List<StringAndNumber> getDateAndOrderCountListByBranchGroup(BranchGroup branchGroup){
```

This function can return extracted data directly.

```
@Override  
public void setServletContext(ServletContext servletContext) {  
    if (ObjectUtils.isEmpty(globalConfigRepo.findAll())) { //如果没有初始化数据  
        initGlobalConfig();  
        initBranchGroup();  
        initCustomer();  
        initEmployee();  
        initFood();  
        initProduct();  
        initOrder();  
        initFoodMonthlySellNumber();  
        initPackageDiscount();  
        initProductInstance();  
    }  
}
```

