**Project introduction**

Mall is an e-commerce shopping website (B2C).

Users can search (by product name, description, and category) and add it to shopping cart without login. If user want to check out, the user need to login. User can login use user name and password, or use third party authentication. The product save in the temporal shopping cart will be merged in the user ‘cart. User can comment the product which user have purchased.

In the backend system

Administrators can manage products (Including product classification, brand, specifications, sale history, and so on),

Administrators can monitor the SQL, scheduled tasks, and manage the interface menus

Administrators can manage user and user authorization



1. For login with pass word and user name:

user

Auth service

User send request to auth

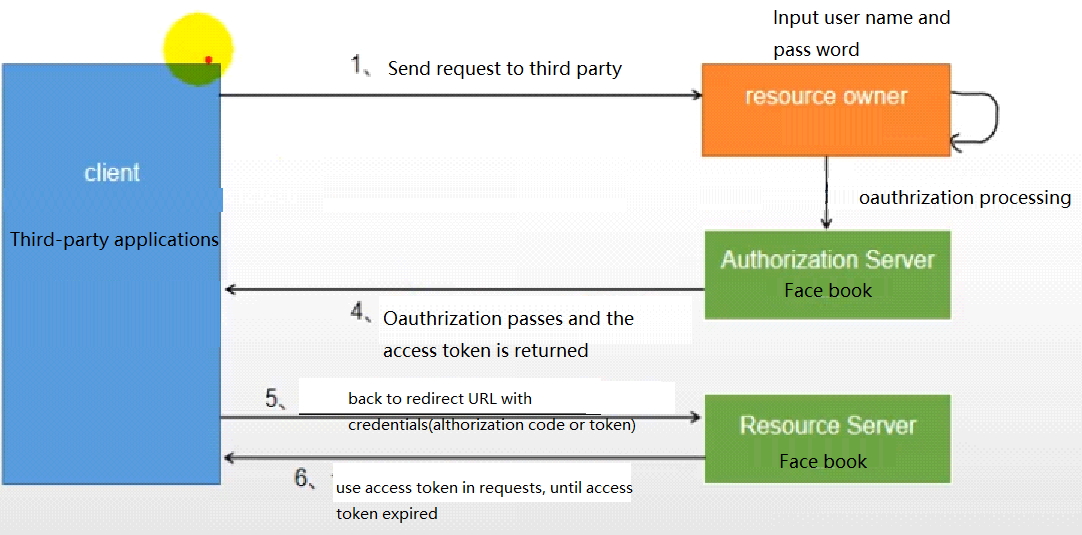
Auth send request to member service to check the authority

Member service

If the user name and password correct, go to the homepage, save a token in the cookie and redis

If not correct, go to the login page, and show the error

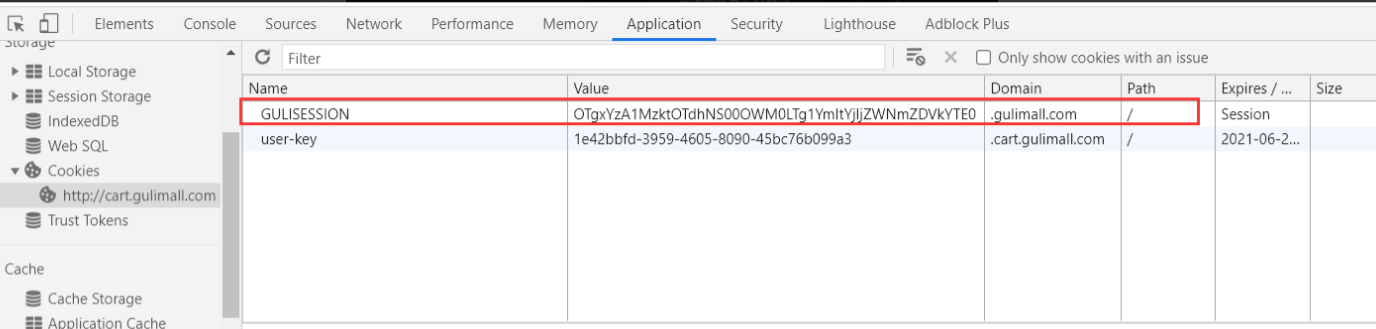
1. For login with oauth2.0:





* If have not used the social account to logged in before, then use the token to call the open API to obtain the authorization information, register and return the result
* If have logged in before, update the token and return the result

After successful login, user information will be saved in Redis and cookies, and the cookies will be added to the browser, the domain name will be the same with home page.(这是为什么登录一次，在其它服务器就不用再认证了) In the future, when call the remote modules（service）, the user can use the information in the Cookie to verify with Redis(that’s the way keep the session ).





Limit request time:

In Redis, use phone-code as the key to store the phone number, verification, and the current time, and set this data valid time in redis for 10 minutes.

if user want to get the code again, we need to check the redis first, if the storage time is less than 60 seconds, the user need to wait.

When after 60 seconds, use want to get the code again, first delete the phone-code in redis

Use JSR303 for verification, use Binding Result to encapsulate the error information and redirect it to the registration page

If the JSR303 verification is passed, check whether the verification code is correct from redis. If the verification code is correct, the user is redirected to the login page;

At the same time, Spring Security uses BCryptPasswordEncoder to encrypt the password.

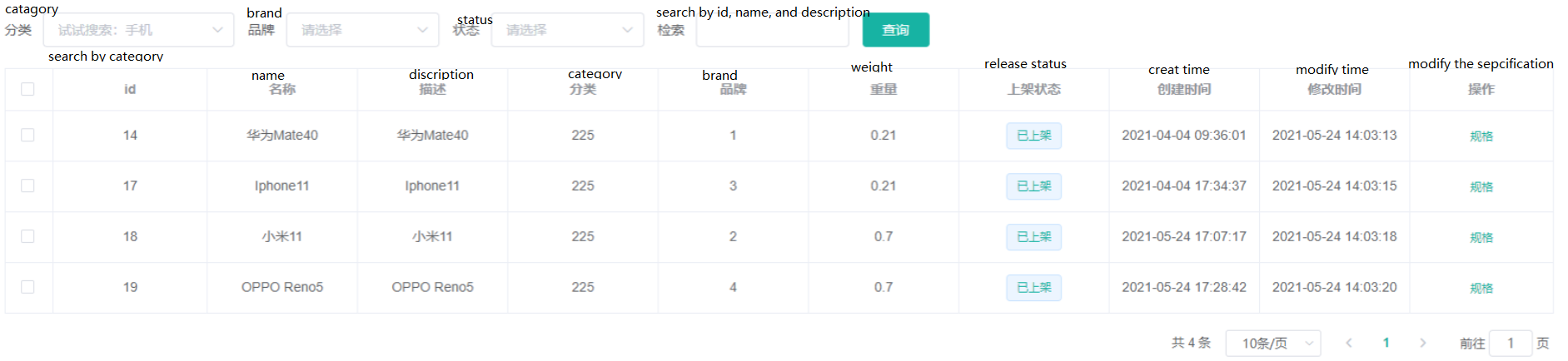
otherwise, the error information returned by the encapsulation from remote service and go to registration page

There is a menu on the website left, it is eager loading and store in the ES, when it is changed, will update data in the ES and database



Product system

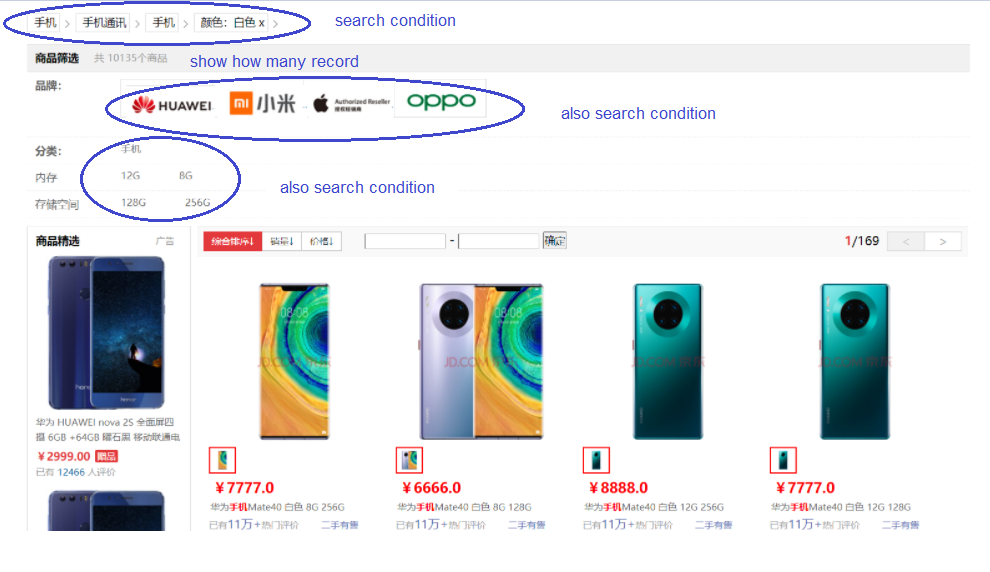
Maintain the data related products. Such as category, brand, name and so on. Use SPU and SKU as structure of product. Provide CURD function for each product.

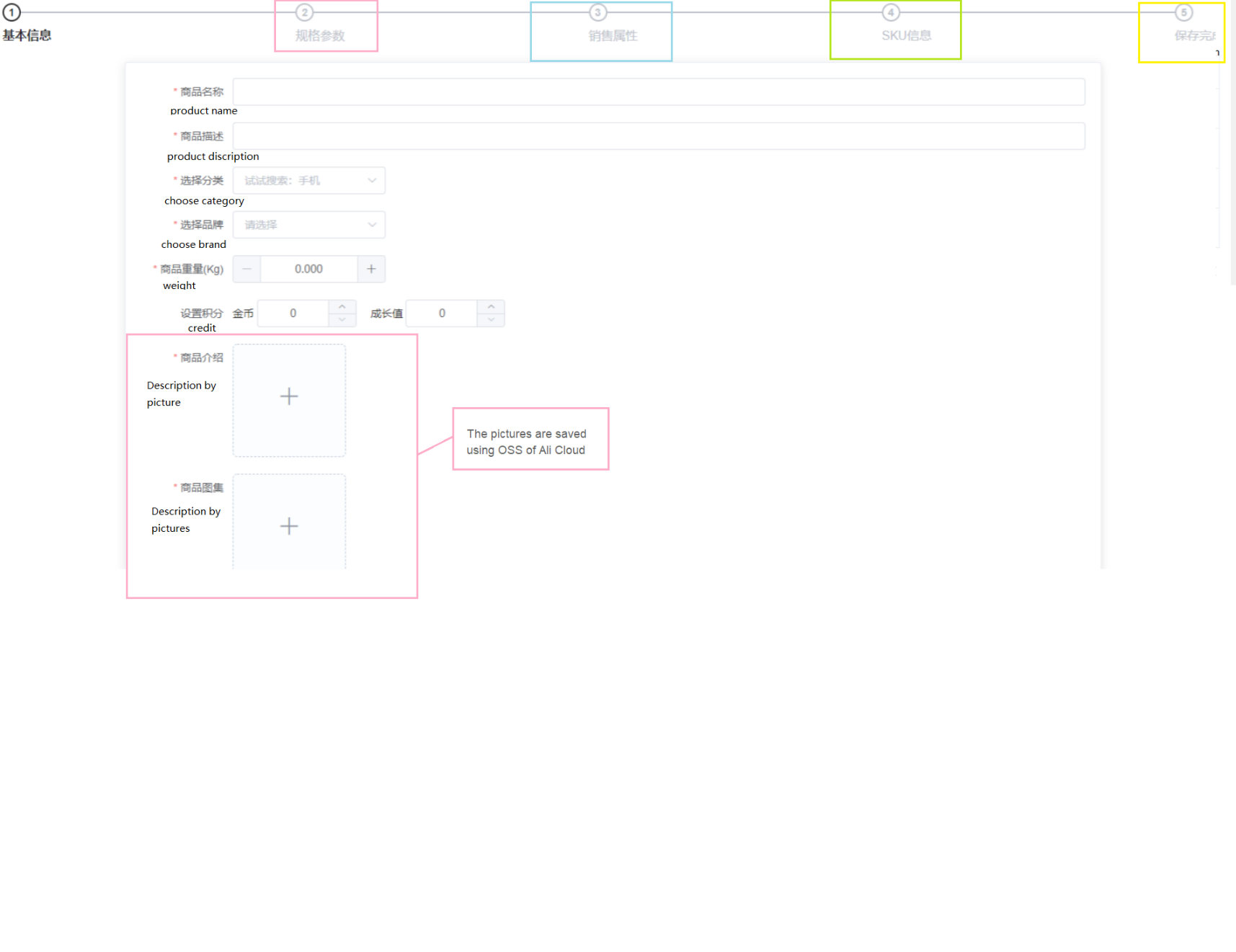


Once save SPU information to the ES, that can be found by using query in the home page.



According to the SKU, Brand, Category, and so to show the related product



Depending on the product information (SPU), generate specific product (SKU) by add

Some report, use ES and paging. （FineReport）

* + Because the data for report are usually so big, when user send request to our service, we response 5 pages and store it in the Redis. 如果报表的数据是从多个服务器来的，我们还要考虑用异步？
  + Some report has pictures, we use lazy loading to show the pictures.
  + Sometimes the data are so big, so we use asynchronization.