目录

[FrontEnd 2](#_Toc500362672)

[Add a product to wishlist. 2](#_Toc500362673)

[Show wishlist in accout page. 2](#_Toc500362674)

[计算wishlist item总价 2](#_Toc500362675)

[主要的截图 3](#_Toc500362676)

[Add wishlist button 3](#_Toc500362677)

[Add to wishlist success 3](#_Toc500362678)

[Add to wishlist fail 3](#_Toc500362679)

[Account wishlist 4](#_Toc500362680)

[Calculation wishlist 4](#_Toc500362681)

[主要的REST请求 4](#_Toc500362682)

[Firstly create a Wishlist and product. 4](#_Toc500362683)

[Add a product in existing wishlist 5](#_Toc500362684)

[Get product prices in PriceSvc 5](#_Toc500362685)

[BackEnd 6](#_Toc500362686)

[数据持久化调用YaaS document service 6](#_Toc500362687)

[Bussiness logic层 6](#_Toc500362688)

[REST api层 6](#_Toc500362689)

[主要的REST请求 7](#_Toc500362690)

[add a wishlist 7](#_Toc500362691)

[Get a wishlist 8](#_Toc500362692)

[Miss a token 8](#_Toc500362693)

[主要的截图 9](#_Toc500362694)

**Implementation of Wishlist**

# FrontEnd

概要: 前端实现按照angular MVC模式分层,在相应的controller内注入service.在service内完成rest请求来实现对model的读写.

Controller和service里面大量使用promises和$q的异步调用使得我们的UI更友好.

/ shared/i18n/ lang\_en.json完成对Wishlist的国际化支持.

# Add a product to wishlist.

1)在/products/templates/product-detail.html中加入css中加入add to wishlist事件,把成功或失败信息通过回调显示到页面上.

2)在/products/controller/product-detail-ctrl.js中注入新建的WishlistSvc来完成Wishlist REST请求.

3)在/wishlist/ services/ wishlist-service.js来完成前端主要的逻辑处理,比如查询当前 customer下面是否有wishlist如果有,直接将product放入到已经存在的wishlist,否则创建新的wishlist.

# Show wishlist in accout page.

1)在总路由router.js中/account/下加入WishlistSvc从wishlist里面读取items.

2)在/account/templates/account.html中加入wishlist的section来显示wishlist items

3)在/account/controllers/account-ctrl.js中注入PriceSvc,ProductSvc,WishlistSvc.

4)在ProductSvc通过productId中去get product详细信息.

5)在PriceSvc通过priceId和productId获得originalAmount和effectiveAmount.

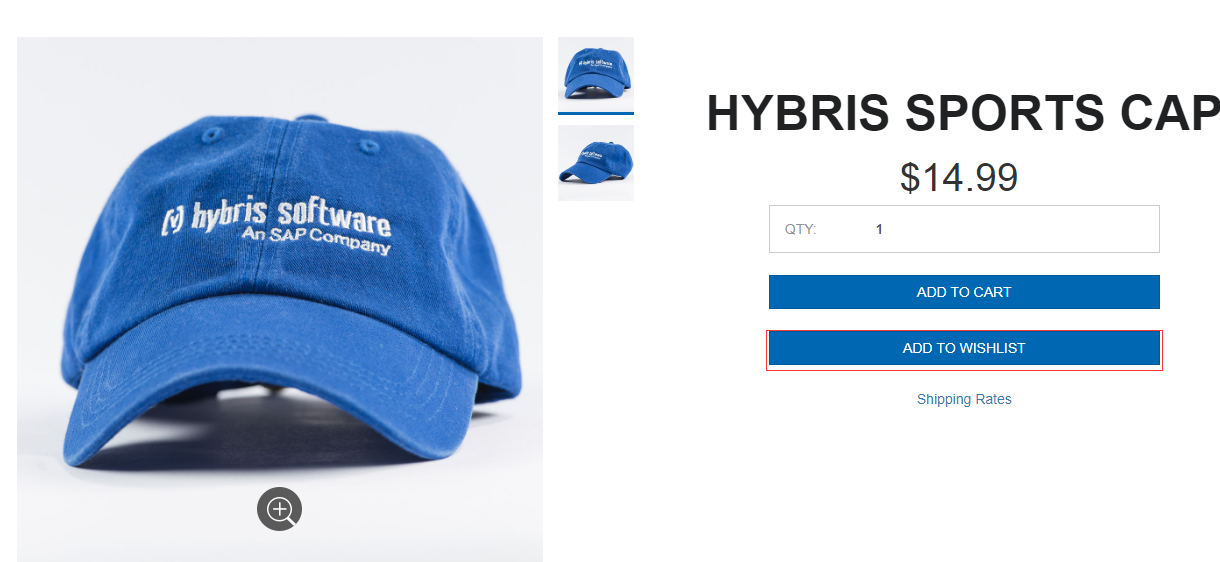
# 计算wishlist item总价

1)在/account/templates/account.html注册calculate事件.

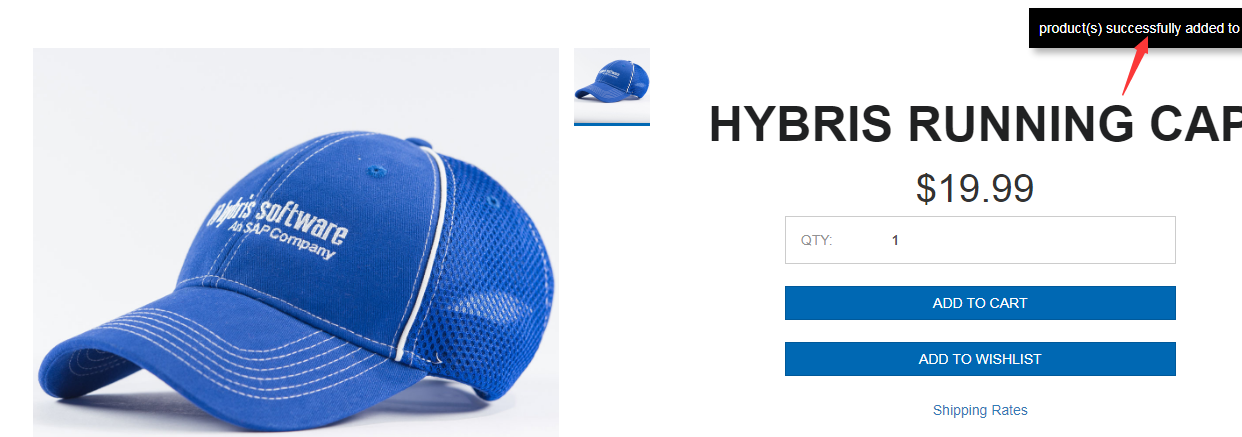
2)在/account/controllers/account-ctrl.js依赖后台的新建的微服务/{wishlistId}/wishlistcalculation来实现对当前用户所有item计算总价.

# 主要的截图

## Add wishlist button



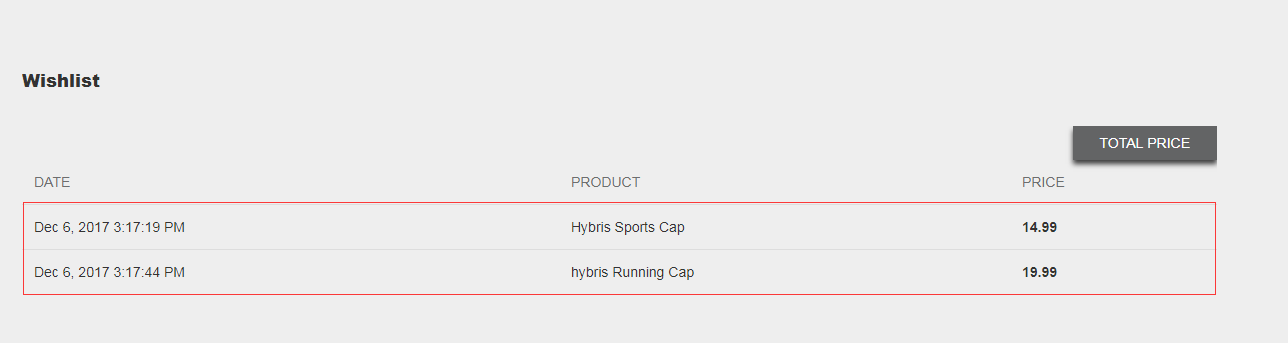
## Add to wishlist success



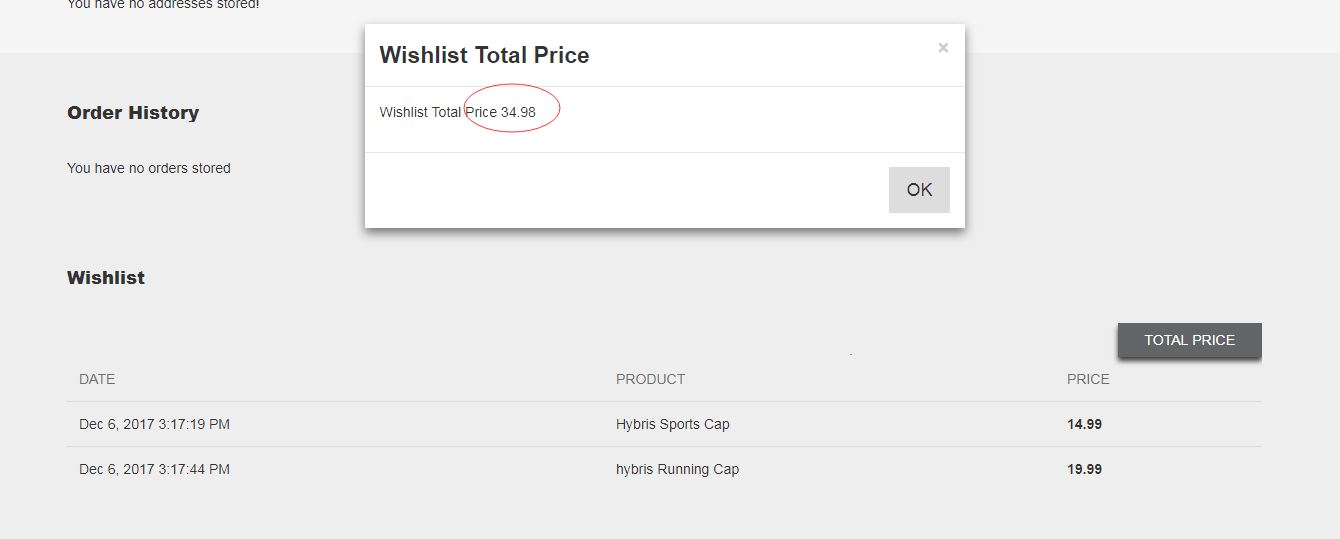
## Add to wishlist fail



## Account wishlist



## Calculation wishlist



# 主要的REST请求

## Firstly create a Wishlist and product.

*curl 'http://10.0.0.13:8080/wishlists' -H 'hybris-site: US' -H 'Origin: http://10.0.0.13:9000' -H 'Accept-Encoding: gzip, deflate' -H 'Accept-Language: zh-CN,zh;q=0.9' -H 'Authorization: Bearer 021-0a36bcdd-bdde-424a-93d6-6eda83cb6591' -H 'hybris-user: Anonymous' -H 'hybris-app: y\_ondemand\_storefront' -H 'hybris-roles: seller' -H 'Connection: keep-alive' -H 'User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36' -H 'Content-Type: application/json;charset=UTF-8' -H 'Accept: application/json, text/plain, \*/\*' -H 'Referer: http://10.0.0.13:9000/' -H 'hybris-tenant: saphybriscaas' --data-binary '{"owner":"C6724366777","id":1512537320899,"title":"test","description":"test","createdAt":"2017-12-06T05:15:27.779Z"}' --compressed*

## Add a product in existing wishlist

*curl 'http://10.0.0.13:8080/wishlists/1512537320899/wishlistItems' -H 'hybris-site: US' -H 'Origin: http://10.0.0.13:9000' -H 'Accept-Encoding: gzip, deflate' -H 'Accept-Language: zh-CN,zh;q=0.9' -H 'Authorization: Bearer 021-0a36bcdd-bdde-424a-93d6-6eda83cb6591' -H 'hybris-user: Anonymous' -H 'hybris-app: y\_ondemand\_storefront' -H 'hybris-roles: seller' -H 'Connection: keep-alive' -H 'User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36' -H 'Content-Type: application/json;charset=UTF-8' -H 'Accept: application/json, text/plain, \*/\*' -H 'Referer: http://10.0.0.13:9000/' -H 'hybris-tenant: saphybriscaas' --data-binary '{"product":"5799cf788f7beb001d0dee20","amount":1,"note":"test","createdAt":"2017-12-06T05:18:25.737Z"}' --compressed*

## Get product prices in PriceSvc

*curl 'https://api.yaas.io/hybris/price/v1/saphybriscaas/prices?currency=USD&effectiveDate=2017-12-06T05:18:50.493Z&productId=5799cf788f7beb001d0dee20,5799cf7e042d5a001de812dd' -H 'Authorization: Bearer 021-0a36bcdd-bdde-424a-93d6-6eda83cb6591' -H 'Origin: http://10.0.0.13:9000' -H 'Accept-Encoding: gzip, deflate, br' -H 'Accept-Language: zh-CN,zh;q=0.9' -H 'User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36' -H 'Accept: application/json, text/plain, \*/\*' -H 'Referer: http://10.0.0.13:9000/' -H 'Connection: keep-alive' --compressed*

*[ {*

*"priceId" : "5799cf7cca4e04001dae5d6f",*

*"metadata" : {*

*"createdAt" : "2016-07-28T09:25:16.731Z",*

*"modifiedAt" : "2016-07-28T09:25:16.731Z",*

*"version" : 1*

*},*

*"productId" : "5799cf788f7beb001d0dee20",*

*"yrn" : "urn:yaas:hybris:price:price:saphybriscaas;5799cf7cca4e04001dae5d6f",*

*"originalAmount" : 14.99,*

*"effectiveAmount" : 14.99,*

*"currency" : "USD"*

*}, {*

*"priceId" : "5799cf80ca4e04001dae5d70",*

*"metadata" : {*

*"createdAt" : "2016-07-28T09:25:20.470Z",*

*"modifiedAt" : "2016-07-28T09:25:20.470Z",*

*"version" : 1*

*},*

*"productId" : "5799cf7e042d5a001de812dd",*

*"yrn" : "urn:yaas:hybris:price:price:saphybriscaas;5799cf80ca4e04001dae5d70",*

*"originalAmount" : 19.99,*

*"effectiveAmount" : 19.99,*

*"currency" : "USD"*

*} ]*

# BackEnd

概要:后端主要采用的是raml http-api定义语言,同时通过api console快速的开发对外的 rest api. 并对项目进行简单分层从数据访问DAO,业务逻辑serivice,工具包utils,constants,plain object 类pojo. 应用spring的元annotation @service @repository进行依赖注入让代码看起更简洁.

# 数据持久化调用YaaS document service

1)采用outh2授权访问.

2)使用Spring restTemplate进行http get and post请求.

3)对两种token用枚举类型进行修饰并用hashmap对view token和manage token分别存取.

4)每次请求先local检查过期时间避免每次对token的获取.

5)对于http 401和403的error handling.如当得到401时应该尝试获取新token再请求.

# Bussiness logic层

1)在WishlistService处理所有关于wishlist业务,比如根据wishlistId获取所有WishlistItems.

2) 在WishlistService依赖注入@repository WishlistDao

3) 在Service层对reponse status进行业务处理根据情况返回数据对象给前端.避免把call stack暴露到前端.

# REST api层

1).项目中的resource是基于JDK的webservice实现

2).创建wishlist totalprice的微服务需要在api.raml定义路径,然后需要在/schemas/创建对象的json文件,在api resource中根据json名创建rest方法,raml对方法名有严格的检查,方法名不对在run的时候失败.

3).尽量不要在api resource层进行复杂的业务逻辑.把复杂的业务逻辑shift to service.

# 主要的REST请求

## add a wishlist

***post*** *https://api.beta.yaas.io/hybris/document/v1/caashiring/caashiring.test/data/wishlist*

***body****:*

*{*

*"url" : null,*

*"owner" : "C6724366777",*

*"title" : null,*

*"createdAt" : 1489899390529,*

*"items" : [ {*

*"productId" : "5799cf45ca4e04001dae5d67",*

*"productName" : "hybris Roller Pen - Blue",*

*"priceId" : "5799cf48fb3ab6001d10ade2",*

*"price" : "1.99",*

*"amount" : 21,*

*"createdAt" : 1489899390529*

*}, {*

*"productId" : "5799cf2efb3ab6001d10ade0",*

*"productName" : "hybris Sunglasses - White",*

*"priceId" : "5799cf34ca4e04001dae5d64",*

*"price" : "12.49",*

*"amount" : 1,*

*"createdAt" : 1489899454101*

*} ],*

*"description" : null,*

*"metadata" : {*

*"createdAt" : "2017-03-15T21:33:21.083+0000",*

*"modifiedAt" : "2017-03-19T04:59:38.577+0000",*

*"version" : 13*

*},*

*"id" : "98hw85Y945e6U35867856776"*

*}*

***response****:*

*{*

*"link" : "https://api.beta.yaas.io/hybris/document/v1/caashiring/caashiring.test/data/wishlist/98hw85Y945e6U35867856776",*

*"id" : "98hw85Y945e6U35867856776"*

*}*

## Get a wishlist

***get*** *https://api.beta.yaas.io/hybris/document/v1/caashiring/caashiring.test/data/wishlist/98hw85Y945e6U3586785*

***Response****:*

*[{*

*"url" : null,*

*"owner" : "C6724366777",*

*"title" : null,*

*"createdAt" : 1489899390529,*

*"items" : [ {*

*"productId" : "5799cf45ca4e04001dae5d67",*

*"productName" : "hybris Roller Pen - Blue",*

*"priceId" : "5799cf48fb3ab6001d10ade2",*

*"price" : "1.99",*

*"amount" : 21,*

*"createdAt" : 1489899390529*

*}, {*

*"productId" : "5799cf2efb3ab6001d10ade0",*

*"productName" : "hybris Sunglasses - White",*

*"priceId" : "5799cf34ca4e04001dae5d64",*

*"price" : "12.49",*

*"amount" : 1,*

*"createdAt" : 1489899454101*

*} ],*

*"description" : null,*

*"metadata" : {*

*"createdAt" : "2017-03-15T21:33:21.083+0000",*

*"modifiedAt" : "2017-03-19T04:59:38.577+0000",*

*"version" : 13*

*},*

*"id" : "98hw85Y945e6U35867856776"*

*}]*

## Miss a token

*https://api.beta.yaas.io/hybris/document/v1/caashiring/caashiring.test/data/wishlists*

*{"status":401,"message":"Unauthorized: Bearer TOKEN is missing","type":"insufficient\_credentials","moreInfo":"https://api.yaas.io/patterns/errortypes.html"}*

# 主要的截图

