در زیر طراحی api ها بر اساس flow application توضیح داده میشود.

در صفحه ی اول user list product ها را میبیند.

GET /products/brief

این API محصولات را با استفاده از تاریخ ایجاد (createdDate) به صورت نزولی یا سعودی مرتب می کند.

برای fetch این اطلاعات productService call میشود.

درخواست نمونه:

GET `http://localhost:8080/insurance-store/api/products/brief?sort=desc`

پاسخ نمونه:

{

"products":[

{

"productId": "1",

"productName": "Product 1",

"productImage": "http://localhost:8080/insurance-store/images/product1.jpg",

"providerName": "Provider 1"

},

{

"productId": "2",

"productName": "Product 2",

"productImage": "http://localhost:8080/insurance-store/images/product2.jpg",

"providerName": "Provider 2"

},

// ...

]

}

اگر user یکی از product ها را انتخاب کند api product with reviews and ratings call میشود.

این api از productService استفاده میکند.

productService.getProductWithRatings(productId);

این service ابتدا product را با id جست و جو کرده, سپس از enquiryService استعلام قیمت میکند, از reviewService لیست 3 کامنت آخر با createdDate و totalRating را میگیرد و همه ی این اطلاعات را با dto مربوط به ui میفرستد.

GET /products/{id}/reviews-and-ratings

درخواست نمونه:

GET http://localhost:8080/insurance-store/products/1/reviews-and-ratings

{

"product": {

"id": "123",

"name": "Insurance Platinum Plan",

"description": "This is the top-tier insurance plan offering maximum coverage.",

"providerName": "BimeBazar Insurance Co.",

"imageUrl": "http://localhost:8000/insuranceStore/products/123.jpg",

"providerId": "456",

"price": 2500000,

"totalRating": 4.3,

"totalComments": 123,

"comments": [

{

"id": "789",

"userName": "Ali V",

"text": "Great insurance plan!",

"createdDate": "2023-06-01T12:00:00",

"lastUpdatedDate": "2023-06-01T12:00:00"

},

{

"id": "790",

"userName": "Sara H",

"text": "Really satisfied with this plan.",

"createdDate": "2023-06-02T15:30:00",

"lastUpdatedDate": "2023-06-02T15:30:00"

},

{

"id": "791",

"userName": "Reza M",

"text": "Could be better in terms of pricing.",

"createdDate": "2023-06-03T10:45:00",

"lastUpdatedDate": "2023-06-03T10:45:00"

}

]

}

}

همچنین api برای حرکت روی کامنت های قبلی نیز وجود دارد. و با توجه به total comments در response ui میتواند محاسبه کند که چه پارامتر های ورودی نیاز است.

GET /api/products/{productId}/comments?page={pageNumber}&size={pageSize}&sort=createdDate,desc

نمونه درخواست:

GET /api/products/123/comments?page=0&size=5&sort=createdDate,desc

{

"meta": {

"totalItems": 100,

"itemsPerPage": 5,

"totalPages": 20,

"currentPage": 1

},

"comments": [

{

"id": "5",

"text": "امکانات بیمه جامع خیلی خوب بود، ممنون.",

"userName": "Reza M",

"productName": "بیمه جامع خودرو",

"createdDate": "2023-06-04T12:00:00Z",

"lastUpdatedDate": "2023-06-04T12:00:00Z",

"userId": "123",

"productId": "456"

},

{

"id": "4",

"text": "سرعت پاسخگویی در خسارت بیمه عمر عالی بود.",

"userName": "Sara N",

"productName": "بیمه عمر",

"createdDate": "2023-06-03T11:00:00Z",

"lastUpdatedDate": "2023-06-03T11:00:00Z",

"userId": "789",

"productId": "321"

},

...

]

}

همچنین ui میتواند برای هر id که توی response آمده است یک link بگذارد و با کیلیک روی آن اطلاعات بیشتری در باره ی آن resource مشاهده شود مثلا با id user میتواند اطلاعات مربوط به user نمایش داده شود.

Request

GET /users/{userId}

مثال:

GET /users/123

{

"user": {

"id": "123",

"name": "Reza M",

"imageUrl": "http://localhost:8000/insuraceStore/users/123/profile.jpg"

}

}

در اینجا، URL برای تصویر پروفایل کاربر در فیلد imageUrl قرار گرفته است.