



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

P1 - Online/Web Store

Interoperability

Sistemas de Informação
MIEIC - 2017/2018

Group AB:

Andreia Rodrigues - up201404691 - up201404691@fe.up.pt

Eduardo Leite - gei12068 - gei12068@fe.up.pt

Francisco Queirós - up201404326 - up201404326@fe.up.pt

José Carlos Coutinho - up201404293 - jczelik@gmail.com

Index

Core Views	2
1. Product view	2
2. Customer profile	4
3. Catalogue View	6
4. Shopping cart	7
5. Admin	10
Interoperability with Primavera	12

Core Views

1. Product view



1 - Search bar

This bar allows the customer to search a book by its title, or part of it. When something is searched here, there will be made a call to the Primavera API "GET api/Artigos/{expression}/{page}" with the searched expression.

This element is present through the other views (that are available to the customer) in the navigation bar as seen here.

2 - Search by category

This dropdown box allows the customer to easily search for products from a certain category. By clicking in one of the categories, the website will make a call to the Primavera API "GET api/Artigos/category/{category}" with the desired category.

This element is present through the other views (that are available to the customer) in the navigation bar as seen here.

3 - Get product information

The information about the product is acquired from the Primavera API "GET api/Artigos/{id}" considering the product's isbn value as the "id" parameter. We acquire information about the product to show to the user, including stock information, necessary when adding the product to the cart.

4 - Get related products (from the same author)

The related products information is acquired from the Primavera API "GET api/Artigos/suggested/{autor}" using the same author as the product presented in the top part of the page.

5 - Add product to cart

This button allows the user to add the desired product to the cart, with the quantity and warehouse chosen. When clicked, there will be a call to the website's database cart API with a POST method, adding the product to the cart with the previously mentioned additional information (quantity and warehouse).

6 - Add product to wishlist

This button allows the user to add a product to his wishlist. When clicked, there will be a call to the website's database wishlist API with a POST method, adding the product to the user's wishlist.

2. Customer profile

Website logo

Search Here

Categories

Wishlist

Account

Cart

Login

Personal Information:

Name: Eduardo
Birthdate: 1993-01-20
Email: eduardo@gmail.com
PhoneNumber: 913456789
NIF: 123456789

Shipping Address: Rua do Eduardo, Porta 2
4400-445
Billing Address: Rua do Edu, Porta 3
4400-444

Edit Information

Order Information and Tracking

Filter Order State

Order number: 095025BF-CB25-11E7-B0DB-080027747A58
Total price: 170.01
Payment received on day: 2017-11-16
Estimated arrival time: 3 days
Order tracking state: Registered

Shipping method: Express
Shipping address: Rua do Edu Porta 3
4400-444
Billing address: Rua do Edu Porta 3
4400-444

Show Details
Problem with the Order?

Order number: 0F6FA7E8-CB94-11E7-B186-080027747A58
Total price: 100.01
Payment received on day: 2017-11-17
Estimated arrival time: 3 days
Order tracking state: Registered

Shipping method: Express
Shipping address: Rua do Edu Porta 3
4400-444
Billing address: Rua do Edu Porta 3
4400-444

Show Details
Problem with the Order?

Order number: 258F3EE3-CB87-11E7-B186-080027747A58
Total price: 15.01
Payment received on day: 2017-11-17
Estimated arrival time: 3 days
Order tracking state: Registered

Shipping method: Express
Shipping address: Rua do Edu Porta 3
4400-444
Billing address: Rua do Edu Porta 3
4400-444

Show Details
Problem with the Order?

Order number: 329CB574-CB25-11E7-B0DB-080027747A58
Total price: 52.82
Payment received on day: 2017-11-16
Estimated arrival time: 3 days
Order tracking state: Registered

Shipping method: Express
Shipping address: Rua do Edu Porta 3
4400-444
Billing address: Rua do Edu Porta 3
4400-444

Show Details
Problem with the Order?

12345

Contact Us

Administration

SINF 2017/2018

1 - Profile information

The user's information is kept in the website's database and is acquired by calling the user API with a GET method. This information will be kept to make the future order process faster.

2 - Order information

Here is going to be displayed the list of orders made by the user.

This list is acquired through a call to the website's order API which will list all the orders made by the currently logged in user.

Each item of the list will display all the necessary information about the order, acquired through a call made to the Primavera API "GET api/Clientes/Venda/{id}", using each order's id.

3 - Filter Order state

This dropdown allows the user to filter the list of orders by their state.

It calls the website's database orders API, obtaining the user's orders id with the chosen state. Each of the orders' information is acquired through a call to the Primavera API, equal to the call mentioned above.

4 - Order product list information

This button will allow the user to see the remaining information about the order. It will list all the products and amount of each ordered. It will also be possible to rate the ordered products here, once their state is changed to "Arrived".

This information is previously loaded through the call to the Primavera API mentioned in the point 2 (Order Information).

3. Catalogue View



1 - Suggestions

Here are going to be shown some suggestions to the user, based on the last products ordered by the website's users.

The information about the suggested products is acquired from the Primavera API "GET api/Artigos/suggested".

2 - Best Sellers







Here are going to be shown the best selling products, the products that were sold the most. To acquire this information, a call to the Primavera API "GET api/Artigos/best_sellers" is made.

3 - Add product to wishlist

This button allows the user to add a product to his wishlist. When clicked, there will be a call to the website's database wishlist API with a POST method, adding the product to the user's wishlist.

4. Shopping cart

The screenshot shows a shopping cart interface. At the top, there is a navigation bar with 'Website logo', a search bar, and links for 'Categories', 'Wishlist', 'Account', 'Cart', and 'Login'. The main cart area is a table with columns: 'Price per Unit', 'Quantity', and 'Total Price'. It lists two items: 'Joao no Espaco' and 'ERP Para Totos'. Each item row includes a product image, title, description, price, discount, IVA, stock status, warehouse selection, and quantity controls. A 'Summary' section at the bottom shows the total number of products (3) and total cost (83.20€). A 'Check Out' button is located to the right of the summary. Numbered annotations (1-6) point to specific elements: 1 points to the cart table header, 2 points to the warehouse selection dropdown, 3 points to the quantity input field, 4 points to the remove button (X), 5 points to the summary table, and 6 points to the 'Check Out' button.

	Price per Unit	Quantity	Total Price
 Joao no Espaco O Joao vai ao espaco In Stock: Yes Warehouse Chosen: Armazém 4(10) 	15€ 12% Off 23% IVA	<div>3</div> <div>+ 1 -</div>	13.20€ 
 ERP Para Totos Manual de ERP Para Totos In Stock: Yes Warehouse Chosen: Armazém 4(15) 	70€ 50% Off 23% IVA	<div>3</div> <div>+ 2 -</div>	70.00€ 

Summary

Number of products:	3
Total Cost:	83.20€

Check Out

1 - Cart

In this table allows the user to have an overview of all the products he added to his cart, as well as the total price, price per unit, discount, IVA percentage, quantity, warehouse chosen and some additional details.

The list of products in the user's cart is acquired through a call to the website's cart API with the GET method.

Then, several calls will be made to the Primavera API "GET /api/Artigos/{id}" in order to retrieve the information about each of the products in the acquired shopping cart list, using their isbn value as the "id" parameter.

2 - Change Warehouse

This button will allow the user to change the warehouse from which each of the products will come from. A call to the website's cart API with a PUT method will be done to update the cart with the new selected warehouse.

The list of warehouses where the product is available and respective stock have been requested previously for each product when the call to the Primavera API mentioned above is done.

3- Change Quantity

This interface will allow the user to change the quantity of each product in the shopping cart. A call to the website's cart API with a PUT method is going to be necessary if there is a change in the current quantity value. This call will update the quantity value of the product in the database. The "Total Price" presented will also be updated accordingly.

4- Remove from Cart

This button will allow the user to remove a certain product from the shopping cart. When clicked, it will make a call to the website's cart API with the DELETE method and with the product isbn as the passed parameter. This will delete the product from the user's shopping cart in the database.

5 - Cart summary

This table contains a summary of the user's shopping cart state: the total number of products and total cost of the products in the cart. These values will be updated everytime a change is made to the cart (quantity of a product is changed or a product is removed).

6 - Check Out

This button allows the user to confirm the purchase and make the order. When clicked, it will make a call to the website's orders API with the POST method. In this method, a call to the Primavera API "POST /api/Clientes/Venda" is done, with the order data, and the order id is returned. This will add the order to the Primavera DB.

The order's API will then proceed to add the order to the user's orders database, with the id returned by the Primavera API mentioned above.

A final call is made to the website's cart API with the DELETE method, in order to delete the products that have been checked out from the cart.

5. Admin

Website logo

From: 11 / 01 / 2017

To: 11 / 30 / 2017

Login

Best Sellers

300 x 540

☆☆☆☆☆

1. Livrinho do Nuno
Um Livrinho escrito pelo Nuno
50€ 50% Off

Nº Sales: 97
Total Cash: 587.00€

300 x 540

☆☆☆☆☆

2. Joao no Espaco
O Joao vai ao espaco
15€ 12% Off

Nº Sales: 66
Total Cash: 478.00€

300 x 540

☆☆☆☆☆

3. ERP Para Totos
Manual de ERP Para Totos
70€ 50% Off

Nº Sales: 14
Total Cash: 262.00€

300 x 540

☆☆☆☆☆

4. Queiros vai a escola
O Queiros vai a escola
10€ 0% Off

Nº Sales: 32
Total Cash: 192.00€

Top Wishlist Products

300 x 540

☆☆☆☆☆

1. Queiros vai a escola
O Queiros vai a escola
10€ 0% Off

Nº Customers: 1
Total Cash: 10.00€

300 x 540

☆☆☆☆☆

2. ERP Para Totos
Manual de ERP Para Totos
70€ 50% Off

Nº Customers: 1
Total Cash: 35.00€

300 x 540

☆☆☆☆☆

3. Joao no Espaco
O Joao vai ao espaco
15€ 12% Off

Nº Customers: 1
Total Cash: 13.20€

Top 4 Categories

300 x 350

1. Mystery

Nº Sales: 97
Total Cash: 587.00€

300 x 350

2. Science Fiction

Nº Sales: 73
Total Cash: 478.00€

300 x 350

3. Guide

Nº Sales: 22
Total Cash: 330.00€

300 x 350

4. Children's

Nº Sales: 32
Total Cash: 192.00€

Orders Waiting for Shipping

Nº Sales: 234
Total Cash: 1,718.00€

Orders Being Shipped

Nº Sales: 0
Total Cash: 0.00€

Orders Concluded

Nº Sales: 0
Total Cash: 0.00€

Contact Us · Administration

SINF 2017/2018

1 - Time interval to be analysed

These inputs will allow the administrator of the website to check the information presented based on a certain period of time. Whenever he changes one of the dates, having both set, the page is refreshed with the new information, making all the calls to the API's mentioned below with the new date values.

2 - Best Sellers

Here will be shown the best sellers information, according to the time period inserted in the navigation bar inputs by the website's administrator. This information is acquired through a call to the Primavera API "GET api/Admin/best_sellers/{inicio}/{fim}", with the parameters "inicio" as the beginning of the time period and "fim" as the end of it.

3 - Most Wishlisted Products

Here will be shown the most wishlisted products by the website's users. To get this information it is necessary to make a call to the website's wishlist API to get the most wishlisted products list. The information of each of these will be acquired through a call to the Primavera API "GET /api/Artigos/{id}" where "id" is the product's isbn value.

4 - Top Categories

Here will be shown the top selling categories information, according to the time period inserted in the navigation bar inputs by the website's administrator. This information is acquired through a call to the Primavera API "GET api/Admin/best_categories/{inicio}/{fim}", with the parameters "inicio" as the beginning of the time period and "fim" as the end of it.

5 - Orders Information

Here will be shown the orders information, according to the time period inserted in the navigation bar inputs by the website's administrator.

They are separated into three categories: waiting for shipping, being shipped and concluded. "Waiting for shipping" means that the orders are in the state of "Registered" or "Admitted", "Being shipped" means that the orders are in the "Shipping" state and the "Concluded" means that the orders are in "Arrived" state.

To get the information required, each of these categories has to make a call to the Primavera API.

The "Waiting for shipping" information is acquired through the call "GET api/Admin/awaiting_shipment/{inicio}/{fim}", the "Being shipped" information is acquired through the call "GET api/Admin/being_shipped/{inicio}/{fim}" and the "Concluded" orders information is acquired through the call "GET api/Admin/orders_concluded/{inicio}/{fim}".

The parameters "inicio" represent the beginning of the time period selected and "fim" as the end of it.

Interoperability with Primavera

Webservice ID	Webservice Description	Related Core View(s)	Route + Verbs	Input Example	Expected Output
get_product	Return the product with the specified ID	PRODUCT_VIEW and SHOPPING_CART	GET api/Artigos?id={id}	ProductID	{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }
get_product_suggested	Return the products related to the product with the specified ID	PRODUCT_VIEW	GET api/Artigos/suggested/{author}	Author	[{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", }

					<pre> "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 </pre>
--	--	--	--	--	--

					<pre> }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }] </pre>
get_order	Get order	CUSTOMER_P	GET	OrderID	{

	information identified by the ID	ROFILE	api/Clientes/Venda/{id}		"Pago": 1.1, "Data": "2017-11-18T22:13:43.2910208+00:00", "Morada": "sample string 3", "Morada2": "sample string 4", "CodPostal": "sample string 5", "MoradaEntrega": "sample string 6", "MoradaEntrega2": "sample string 7", "CodPostalEntrega": "sample string 8", "TrackingState": "sample string 9", "ListaEncomendas": [{ "ISBN": "sample string 1", "Titulo": "sample string 2", "Subtitulo": "sample string 3", "Quantidade": 4.1 }, { "ISBN": "sample string 1", "Titulo": "sample string 2", "Subtitulo": "sample string 3", "Quantidade": 4.1 }, { "ISBN": "sample string 1", "Titulo": "sample string 2", "Subtitulo": "sample string 3", "Quantidade": 4.1 }] }
get_suggested	Return the suggested products	CATALOGUE_VIEW	GET api/Artigos/suggested	N/A	[{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1,

					<pre> "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }], { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 </pre>
--	--	--	--	--	---

					<pre> }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }], { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }] </pre>
get_best_sellers	Return the best seller products	CATALOGUE_VIEW	GET api/Artigos/best_sellers	N/A	<pre> [{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string </pre>

				<pre> 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", </pre>
--	--	--	--	---

					<pre> "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }] </pre>
--	--	--	--	--	---

get_category	Return the products of a set category	CATALOGUE_VIEW	GET api/Artigos/category/{category}/{page}	Category and Page	{ "Count": 3, "Artigos": [{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1,
--------------	---------------------------------------	----------------	---	-------------------	--

					<pre> "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }], { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", </pre>
--	--	--	--	--	--

					<pre> "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }] } </pre>
get_search	Get the products of a set search	CATALOGUE_VIEW	GET api/Artigos/{expression}/{page}	Expressio n and Page	<pre> { "Count": 3, "Artigos": [{ "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string </pre>

				<pre> 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] }, { "ISBN": "sample string 1", "Title": "sample string 2", "Subtitle": "sample string 3", "Author": "sample string 4", "Edition": "sample string 5", "Type": "sample string 6", "Language": "sample string 7", "Discount": 8.1, "Price": 9.1, "Pages": 10, "Synopsis": "sample string 11", "Editor": "sample string </pre>
--	--	--	--	---

					<pre> 12", "Storage": [{ "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }, { "Code": "sample string 1", "Name": "sample string 2", "Stk": 3.1 }] } </pre>
add_sale	Checkout the order made by the client	SHOPPING_CART	POST api/Clientes/Venda	<pre> { "Cliente": { "CodCliente": "sample string 1", "NomeCliente": "sample string 2", "NumContribuinte": "sample string 3", "Morada": "sample string 4", "Morada2": "sample string 5", "CodPostal": "sample string 6", "MoradaEnvio": "sample </pre>	N/A

				string 7", "MoradaEnvio2": "sample string 8", "CodPostalEnvio": "sample string 9", "Telefone": "sample string 10", "Email": "sample string 11" }, "Vendas": [{ "Artigo": "sample string 1", "Quantidade": 2, "Armazem": "sample string 3" }, { "Artigo": "sample string 1", "Quantidade": 2, "Armazem": "sample string 3" }, { "Artigo": "sample string 1", "Quantidade": 2, "Armazem": "sample	
--	--	--	--	---	--

				string 3" }] }	
get_admin_best_sellers	Get the best selling products	ADMIN_DASH	GET /api/Admin/best_sellers/{inicio}/{fim}	StartDate and EndDate	[{ "ISBN": "sample string 1", "Sales": 2, "Value": 3.1 }, { "ISBN": "sample string 1", "Sales": 2, "Value": 3.1 }, { "ISBN": "sample string 1", "Sales": 2, "Value": 3.1 }]
get_admin_best_categories	Get the top sold categories	ADMIN_DASH	GET api/Admin/best_categories/{inicio}/{fim}	StartDate and EndDate	[{ "Categoria": "sample string 1", "Sales": 2, "Value": 3.1 }, { "Categoria": "sample string 1", "Sales": 2, "Value": 3.1 }, { "Categoria": "sample string 1", "Sales": 2, "Value": 3.1 }]
get_admin_orders_being_shipped	Get stats on the orders being shipped	ADMIN_DASH	GET api/Admin/being_shipped/{inicio}/{fim}	StartDate and EndDate	{ "TotalSales": 1, "TotalValue": 2.1 }
get_admin_orders_awaiting_shipment	Get stats on the orders awaiting shipment	ADMIN_DASH	GET api/Admin/awaiting_shipment/{inicio}/{fim}	StartDate and EndDate	{ "TotalSales": 1, "TotalValue": 2.1 }
get_admin_orders_concluded	Get stats on the concluded orders	ADMIN_DASH	GET api/Admin/orders_concluded/{inicio}/{f	StartDate and EndDate	{ "TotalSales": 1, "TotalValue": 2.1 }

			im}		
--	--	--	-----	--	--