API Operations

Tours & Activities

Contents

[B. HOME PAGE 2](#_Toc162358339)

[Description 2](#_Toc162358340)

[Functionalities 2](#_Toc162358341)

[C. Landing pages (Destination Page/ Attraction pages) 4](#_Toc162358342)

[Description 4](#_Toc162358343)

[Functionalities 5](#_Toc162358344)

[Example 5](#_Toc162358345)

[Data to fetch from 5](#_Toc162358346)

[D. PRODUCT PAGE 6](#_Toc162358347)

[Description 6](#_Toc162358348)

[Functionalities 6](#_Toc162358349)

[Example 6](#_Toc162358350)

[Data to fetch from 7](#_Toc162358351)

[E. Cart management 8](#_Toc162358352)

[Description 8](#_Toc162358353)

[Functionalities 8](#_Toc162358354)

[Data to fetch from 9](#_Toc162358355)

[Example 9](#_Toc162358356)

[F. Other code implementations at site end 10](#_Toc162358357)

# HOME PAGE

### Description

Site home page is displayed when user open the site.

### Functionalities

#### Login management (For logged in users/ On login user )

* + It is checked if user login is saved in cookies or not
  + If yes, then look for saved cart in db
  + Load cart and check of availability for each product that was added in cart to make sure the option is still available to book or not
  + If any option in cart has past travel date, remove such options from cart
  + Update cart and save in db if user is logged in.
  + Get option count in from cart and show on cart icon

Graphical user interface, website

Description automatically generated

#### Language Change

1. User can change language if cart is empty
2. If any tour option is added in cart then language drop down becomes disable

#### Multiple Sites Management

Same code is used to load multiple sites using its affiliate id

<https://www.hop-on-hop-off-bus.com/>?partner =hoho

This allows us to run all sites from the same code base.

#### Example

<https://www.hop-on-hop-off-bus.com/>

#### Data to fetch from

.net core front end API

Home page : Landing Module : Home Controller

* + 1. Search Module : Products Controller
    2. Helper Module : SearchPageProducts

SEO URL Mappings From DB and cache into Site to re-use

All data upto product page should come from cache after 1st time load from db and API’s.

# Landing pages (Destination Page/ Attraction pages)

### Description

Sites non-home pages that can be grouped as destination and attraction pages which list out the products based on destination and categories

A screenshot of a computer

Description automatically generated with medium confidence

### Functionalities

* These are pages which show the list of products based on destinations like
* The SEO URL mapping module provides URL mapping that is used to get SEO friendly URL that we see in browser’s address bar but its actual URL is like [https://www.hop-on-hop-off-bus.com/{paris-regionId}](https://www.hop-on-hop-off-bus.com/%7bparis-regionId%7d)
* Site Should have this mapping information so that it can translate requested URL’s <https://www.hop-on-hop-off-bus.com/paris-hop-on-hop-off-bus-tours> to “[https://www.hop-on-hop-off-bus.com/{paris-regionId}](https://www.hop-on-hop-off-bus.com/%7bparis-regionId%7d)/{categoryid}” to get data from API based on region-Ids and category-Ids
* To build Search Data like below api will need these ids to fetch data for that destination
* Example Product list based on region id and category id for ***Paris*** (Region) page showing ***hop on hop off tours***(category)
* Prices in product list is associated with Affiliate ID it means that prices varies as per site being loaded.
* Redirection rules can be there for landing page which needs to be handled by site’s code.

Graphical user interface

Description automatically generated

### Example

<https://www.hop-on-hop-off-bus.com/paris-hop-on-hop-off-bus-tours> (Destination Page)

### Data to fetch from

#### .Net Core API Landing Pages :

Search Module : Products Controller

Helper Module : SearchPageProducts

SEO URL Mappings From DB and cache into Site to re-use

All data upto product page should come from cache after 1st time load from db and API’s.

# PRODUCT PAGE

### Description

This page shows product details and from this page booking flow starts

User can check for product availabilities from booking window

Graphical user interface, website

Description automatically generated

### Functionalities

* Prices are all cached, after doing check availabilities prices are updated with real time prices
* User can change currency of site and view price based on selected currencies
* Activity Data contains binding for guest and calendar too, Calendar date is marked as closed/greyed out if activity is not bookable for that date
* Cancellation Policy is updated based on response from check availability API result.
* Redirection rules can be there for an activity which needs to be handled by site code for product page

### Example

<https://www.hop-on-hop-off-bus.com/paris/paris-open-tour-hop-on-hop-off_3733>

This is SEO URL which is translated to [https://www.hop-on-hop-off-bus.com/activity /detail/3733](https://www.hop-on-hop-off-bus.com/activity%20/detail/3733) to get data required to bind product page for tour activity id 3733.

### Data to fetch from

#### .Net Core API Product Pages :

* Activity Module : Detail Controller (For Product Data)
* Activity Module : CheckAvailabilityFrombookingWindow (For Product Availability)

SEO URL Mappings From DB and cache into Site to re-use

All data upto product page should come from cache after 1st time load from db and API’s.

# Cart management

### Description

After doing check availability user can add product to cart. Cart management at API end is not needed

as for booking multiple options from cart booking request needs Availability reference ids and relevant data associated with option to build cart at api end

use bumble bee API’s call after cart page to create booking

Diagram

Description automatically generated

### Functionalities

* Add product option to cart
* Add bundle option to cart
* Apply discount
* Remove discount
* Remove product Options from cart
* Remove bundle Options from cart
* On change of currency update prices of cart as per selected currencies
* Multi-save Discount
  + It is configured at affiliate level
  + If its yes, then user will get additional discount on adding 2nd product option in cart
  + Discount can’t go beyond max capping
  + Max Discount Capping can be as per currencies
* Check for Discount Process in bumble API for after any change in cart to have updated price of the cart

### Data to fetch from

* Use Bumblebee API to syc prices
* Use Bumblebee API to process discount
* Use Bumblebee API from cart page onwards to booking confirmation.

Use Bumblebee API for further operation from cart page onwards :

* Discount coupon
* Payment extra detail
* CreateBooking from cart
* CreateBookingCallback
* BookingDetails
* Voucher

### Example



# Other code implementations at site end

#### Session management

#### Login Management

* Facebook
* Google
* Site’s login
* Logged in user’s cart is saved into database, it is used for sending campaign
* If user does not book and leave site and when he retunes and login the cart is restored from db on Login event.
* Options in cart if are of past travel dated then such options are removed from cart.
* For option with future date are again recheck for availability and added as per previously added criteria

#### Caching Strategy

* All data for the site is cached in site’s memory upto product page
* Cached data varies as per affiliated and language code
* Product data varies as per affiliate id, currency code and language code
* Data being cached
  1. Hope page
  2. Landing Page
  3. Product page (Except check availability result)
  4. Affiliate data
  5. SEO URL mappings
  6. Product List as per regions and destination
  7. Currencies
  8. Other master data if needed.
  9. Cached data expires after 2 hour and this duration is configurable from web-config file
  10. CurrencyExchangeRate

#### SEO URLS mappings

SEO mapping are crucial part of site code and it take care of dynamic URL mapping. In site all URLS upto Product page and corporate pages are SEO friendly URLS.

These are translated into ids by code to fetch respective resource data from db and API’s

#### Redirection rules

Site’s code and IIS take cares of all the redirections for destination pages and product pages.