



WEB PORTAL - API



1 INTRODUCTION

1.1 Purpose of the document

The purpose of this document is to detail the Booking API for Courier-S solution:

1.2 Scope

The scope of this document is limited to the following

- API description
- Available API controllers
- Security
- Description of the models, the data type and format.
- Error Handling
- Data Mapping

2 API DESCRIPTION

The Courier-S solution handles Pickup and Delivery bookings created on Booking Portal.

Business requires the new source of bookings sent directly from the 3rd party solution via WebAPI

The Booking Web API provides the following functionality:

- Allow to get the estimated price
- Allow to create a Booking
- Allow to create a Quote

All the controllers are sharing the same set of Models.

The API is using the HereMaps service for the geo-coding the provided addresses.

For the Quote and Booking the street addresses for both Pickup Stop and Delivery Stop have to be provided in full with street number, street name, street type, suburb, postcode and state. Those have to be “clean” and “resolvable” addresses. With no extra info or comments.

3 API CONTROLLERS

3.1 GetPrice

The GetPrice controller calculates the estimated price based on provided booking details.

To get the most accurate estimate the full street address should be provided. But in case if the street number, name and type are unknown user can provide the suburb, state and postcode only.

The GetPrice interface does not create the actual booking or Quote records therefore it does not return the Booking number.

3.2 CreateQuote

The CreateQuote controller allows user to create a quote in Courier-S database and get the estimated price for that quote. At the later stage the quote can be converted into actual Booking by one of the Users that Customer using the Booking Portal.

For the Quote the street addresses for both Pickup Stop and Delivery Stop have to be provided in full.

Courier-S solution does not create any actionable events or data for the quotes. The dispatchers and AutoAllocation service will not see quotes and will not assign them to drivers for the execution.

3.2 CreateBooking

The CreateBooking controller allows user to create a new booking in Courier-S database and get the estimated price for that booking.

For the Bookings the street addresses for both Pickup Stop and Delivery Stop have to be provided in full.

As soon as the new booking is created it is processed by the Courier-S Rules Engine which generates all the required actionable events and data.

The dispatchers and Auto Allocation service will be able see the booking related records and assign them to drivers for the execution.

The following are the interfaces and systems involved in the flow

SECURITY

The security of the Booking Web API is implemented on multiple levels:

- Courier-S web server allows the HTTPS communication only which automatically encrypts the data.
- There is a the API Client record saved in Courier-S for each customer using the API. where each API Client record has a unique token and set of allowed Account Numbers. Each API request must have the token included.
- The API Client can create booking for the allowed Account numbers only
- It has to be a Booking Portal user created for the Account Number. The login of that user has to be provided as part of the request.

In case if the API client is a broker working with multiple customers there will be a Customer record created for each one of them. Each customer will have a unique Account Number and a list of the Booking portal users. Each portal user will have a LOGIN unique for that customer

In order for the request to be successful the API client should provide the valid Account Number and the user LOGIN.

5 DESCRIPTION OF THE MODELS, DATA TYPES

Each request has to use the following models:

- Request Booking Model
- Request Booking ItemModel
- Request Booking Stop Model
- Booking Result Model

5.1 Request Booking Model

```
RequestBookingModel {
    externalId          String, maxLength: 32, optional

    accountNumber       String ,maxLength: 16, mandatory for all controllers

    requestedBy         String, maxLength: 64, mandatory for all controllers
                        (Account LOGIN)

    bookingDate         string($date-time), dd/MM/yyyy, mandatory for all
                        controllers

    bookingReference1   String, maxLength: 40, optional

    bookingReference2   String, maxLength: 40, optional

    product             String, maxLength: 64, mandatory for all controllers

    priority            String, maxLength: 64, mandatory for all controllers

    service             String, maxLength: 64, mandatory for all controllers

    subService          String, maxLength: 64, mandatory for all controllers

    items              [RequestBookingItemModel{...}]Collection of
                        RequestBookingItemMode , mandatory for all controllers

    pickupDetails       RequestBookingStopModel{...} , mandatory for all
                        controllers

    deliveryDetails     RequestBookingStopModel{...} , mandatory for all
                        controllers
}
```

5.2 Request Booking Item Model

```
RequestBookingItemModel {
```

```

itemType          String, maxLength: 64, mandatory for controllers

quantity          integer($int32), maximum: 10000, minimum: 1, mandatory for
                  all controllers

totalWeight       number($double), maximum: 10000.00 minimum: 0.1, mandatory
                  all for controllers

length           integer($int32), maximum: 10000, optional

width            integer($int32), maximum: 10000, optional

height           integer($int32), maximum: 10000, optional

}

```

5.3 Request Booking Stop Model

```

RequestBookingStopModel {
    companyName     String, maxLength: 100, mandatory for CreateBooking and
                  CreateQuote

    address1        String, maxLength: 255, mandatory for CreateBooking and
                  CreateQuote

    address2        String, maxLength: 255, optional

    suburb          String, maxLength: 64, mandatory for all controllers

    state           String, maxLength: 64, mandatory for all controllers

    postcode        String, maxLength: 5, mandatory for all controllers

    contactPhone    String, maxLength: 16, optional

    contactName     String, maxLength: 64, optional

    readyTime       string($date-time), dd/MM/yyyy HH:mm (24 hr format),
                  optional

    closeTime       string($date-time), dd/MM/yyyy HH:mm (24 hr format),
                  optional

    specialInstructions String, maxLength: 255, optional

    notificationEmail String, maxLength: 64, optional

}

```

5.3 Request Booking Stop Model

```
RequestBookingStopModel {
    companyName      String, maxLength: 100, mandatory for CreateBooking and
                      CreateQuote
    address1          String, maxLength: 255, mandatory for CreateBooking and
                      CreateQuote
    address2          String, maxLength: 255, optional
    suburb            String, maxLength: 64, mandatory for all controllers
    state             String, maxLength: 64, mandatory for all controllers
    postcode          String, maxLength: 5, mandatory for all controllers
    contactPhone      String, maxLength: 16, optional
    contactName       String, maxLength: 64, optional
    readyTime         string($date-time), dd/MM/yyyy HH:mm (24 hr format),
                      optional
    closeTime         string($date-time), dd/MM/yyyy HH:mm (24 hr format),
                      optional
    specialInstructions String, maxLength: 255, optional
    notificationEmail String, maxLength: 64, optional
}
```

5.4 Request Result Model

```
BookingResult {
    documentNumber    String, (nor available for GetPrice controller)
    distanceKM        number($double)
    price             number($double)
}
```

6 ERROR HANDLING

The Booking API validates all the provided values and does all the required matching to avoid the processing of invalid data or data in an incorrect format.

In case if any data related issues the API returns the Error code 400 and provides problem description in Response Body.

Example:

Error 400:

Response body

```
{
  "ExternalId": [
    "The field ExternalId must be a string with a maximum length of 32."
  ],
  "BookingDate": [
    "Booking Date cannot be empty and must be in format 'dd/MM/yyyy'"
  ]
}
```

7 DATA MAPPING

The current version of the Booking API allows to make the Pickup and Delivery bookings and Quotes only. In terms of Felix business it's a "Courier" product.

There is a provision for the "Taxi Truck" or "Hourly Hire" bookings as well but this option is not available yet.

There are the following properties of the **RequestBookingModel** to have the predefined (mapped) values in order for Couier-S to process the request successfully :

Property	Code	Name
Product	C	Courier
Service	C	Courier
Service	W	Wagon
Service	HT	Half Tonne
Service	1T	1 Tonne
Service	RR	1 Tonne - Roof Rack
Subservice	V	Van
Subservice	TT	Tray
Subservice	E	Either
Priority	S	Standard
Priority	E	Express
Priority	V	VIP

