San Jose State

University Fall 2017

KAYAK API DESIGN DOCUMENT

CMPE 273 – Team 7

SUBMITTED BY:

Aditya Parmar

Dishant Kimtani

Meenakshi Parwani

Siddharth Suthar

Sunil Tiwari

SUBMITTED TO:

Prof. Simon Shim

1. **User Management APIs**
   a. **Login API** - This will enable the user to login to the account.

Title	login
URL	/user/login
Method	POST
URL Params	NA

Data Params	{
	   username : String,
	   password : String
	}

Success Response	{
	   status : 200,
	   result: {
	      Message : "Logged In"
	   }
	}

Error Response	{
	   status : 401,
	   error: {
	      Message : "Login Unsuccessful"
	   }
	}

   b. **Sign Up API** - This API will let the user signup and create their own account.

Title	signup
URL	/signup
Method	POST
URL Params	NA

| Data Params | ```
{
   firstname: String,
   lastname: String,
   username: String,
   password: String,
   email: String,
   address: String,
   phone: Integer
}
``` |
| Success Response | ```
{
   status : 200,
   result: {
      Message : "Successfully Registered"
   }
}
``` |
| Error Response | ```
{
   status : 401,
   error: {
      Message : "Error While Registration"
   }
}
``` |

c. **Update User API** - This API will let you update the user's information.

Title	update
URL	/user/update
Method	PUT
URL Params	{userid: String}
| Data Params | ```
{
   firstname: String,
   lastname: String,
   username: String,
   password: String,
   email: String,
   address: String,
``` |

	phone: Integer }

Success Response	{
	  status : 200,
	  result: {
	    Message : "user updated"
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Update Unsuccessful"
	  }
	}

d. User History API - This API will let you update the user's information.

Title	UserHistory
URL	/user/history
Method	GET
URL Params	{type: String}
Data Params	NA

Success Response	{
	  status : 200,
	  history: Array
	}

Error Response	

```
{
   status : 401,
   error: {
      Message : "Update Unsuccessful"
   }
}
```

2. **Flight APIS**
   a. **List flights –** list the flights matching the user entered criteria

Title	List flights
URL	/flight/list
Method	GET
URL Params	NA

Data Params	`{`
	`   type: String (one-way/round-trip/multi-city)`
	`   source: String,`
	`   destination: String,`
	`   departure-date: Date,`
	`   return-date: Date,`
	`   cabin-class: String,`
	`   travelers: Array`
	`}`

Success Response	`{`
	`   status: 200,`
	`   result: {`
	`      flights: Array`
	`   }`
	`}`

Error Response	`{`
	`   status : 300,`
	`   error: {`
	`      Message : "No flights found"`

	} }

b. Book Flight – book the flight/flights selected by the user

Title	Book flight
URL	/flight/book
Method	POST
URL Params	NA

Data Params	{
	   type: String (one-way/round-trip/multi-city)
	   source: String,
	   destination: String,
	   departure-timestamp: TimeStamp,
	   return-timestamp: TimeStamp,
	   cabin-class: String,
	   travelers: Array
	}

Success Response	{
	   status: 200,
	   result: {
	      message: "Flight booked Successfully"
	      flights: Array
	   }
	}

Error Response	{
	   status : 300,
	   error: {
	      Message : "Flight not booked, Please try again"
	   }
	}

c. Get Flights By Vendor – get the flights from each vendor

Title	Get flights By Vendor
URL	/flight/getByVendor
Method	POST
URL Params	NA

Data Params	{
	   type: String (one-way/round-trip/multi-city)
	   source: String,
	   destination: String,
	   departure-date: Date,
	   return-date: Date,
	   cabin-class: String,
	   travelers: Array
	}

Success Response	{
	   status: 200,
	   result: {
	     flights: Array
	   }
	}

Error Response	{
	   status : 300,
	   error: {
	     Message : "No Flights for the specified vendor"
	   }
	}

d. Cancel Flight – cancel the flight selected by user if it satisfies the selected criteria

Title	Cancel the Flight
URL	/flight/cancel

Method	PUT
URL Params	NA

Data Params	{
	   booking-id: Number
	}

Success Response	{
	  status: 200,
	  result: {
	    message: "Flight cancelled Successfully"
	  }
	}

Error Response	{
	  status : 300,
	  error: {
	    Message : "Failed to cancel the flight"
	  }
	}

3. **Car APIs**
   a. **Search Cars API:** Avails search facility to the user for cars based on various parameters such as Type or Price

Title	Car Search
URL	/car/search
Method	POST
URL Params	NA
Data Params	NA

Success Response	{
	   status : 200,
	   result: {
	[carID:123,
	 { Name: Toyota,
	   Model: Fortuner,
	   BookingDate:date,
	   isBooked:false
	}
	],
	[carID:456,
	  { Name: Volkswagen,
	   Model: Golf,
	   BookingDate:date,
	   isBooked:false]
	   }
	]
	   }
	}

Error Response	{
	   status : 401,
	   error: {
	     Message : "Search Unsuccessful"
	   }
	}

b. **Book Cars:** Avails booking facility to the user for cars.

Title	Book Cars
URL	/car/book
Method	POST
URL Params	NA

Data Params	{
	   carID:123,
	   {

	Name: Toyota,
	Model: Fortuner,
	bookingDate:2nd Jan 2018,
	isBooked:false,
	bookingStartTime: Monday 6:00 pm,
	bookingEndTime: Monday 8:00 pm,
	pickup : Location
	paymentInfo : {
	      paymentMethod: Card,
	      amount:$200
	    }
	  }
	}

Success Response	{
	  status : 200,
	  result: {
	    Message : "Flight Booked"
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Booking Unsuccessful"
	  }
	}

c. **Cancel Car Booking:** Let's user cancel an existing booking.

Title	Cancel Book
URL	/car/cancel
Method	DELETE
URL Params	NA
Data Params	{ carID: 123,

	Cancelation Reason : "Sorry, clash in time schedule" }

Success Response	{
	  status : 200,
	  result: {
	    Message : "Flight Canceled",
	    carID : 12345
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Cancelation unsuccessful"
	  }
	}

4. **Hotel APIs**
   a. Get Hotel Data

Title	Get Hotel Data
URL	/hotel
Method	POST
URL Params	null

Data Params	{
	 cityname:"String",
	 startdate:"String",
	 enddate:"String"
	}

Success Response	{
	  status: 200,
	  result: {
	    Message : "get hotel data successfully",
	    Hoteldata : [{
	  hotelId : 1,

```
                        name :"hotelname",
                        address:"address"
                        city:"city",
                        state:"state",
                        zipcode:"zipcode",
                        starts:"starts",
                        rooms:{roomtype:"",prices:""},
                        ratings:"",
                        reviews:""
                  }]
                     }
                  }
```

| Error Response | ```
{
   status : 401,
   error: {
       Message : "Unsuccessful"
   }
}
``` |

b. Hotel Booking

Title	Hotel Booking
URL	/hotel/booking
Method	POST
URL Params	null
| Data Params | ```
{
   userid:"userid"
   Hoteldata : [{
   hotelId : 1,
   name :"hotelname",
   address:"address",
   city:"city",
   state:"state",
   zipcode:"zipcode",
   starts:"starts",
   rooms:{roomtype:"",prices:""},
``` |

	}

Success Response	{
	   status: 200,
	   result: {
	      Message : " hotel booking successfully",
	      BookingID:"5EDWYHSKJK"
	
	   }
	}

Error Response	{
	   status : 401,
	   error: {
	      Message : "Unsuccessful"
	   }
	}

c. Get Hotel Data from vender

Title	Get Hotel Data from vender
URL	/hotel/venderid
Method	POST
URL Params	null

Data Params	{
	   venderid:"int",
	
	}

Success Response	{
	   status: 200,
	   result: {
	      Message : "get hotel data successfully",
	      Hoteldata : [{
	   hotelId : 1,
	   name :"hotelname",
	   address:"address",
	   city:"city",

	state:"state", zipcode:"zipcode", starts:"starts", rooms:{roomtype:"",prices:""}, ratings:"", reviews:"" }] } }

Error Response	{
	  status : 401,
	  error: {
	    Message : "Unsuccessful"
	  }
	}

d. Cancel Hotel Booking

Title	Cancel Hotel Booking
URL	/hotel/cancelbooking
Method	POST
URL Params	null

Data Params	{
	  userid:"int",
	  bookingid:"int"
	}

Success Response	{
	  status: 200,
	  result: {
	    Message : "hotel booking cancel successfully",
	    BookingID:"5EDWYHSKJK"
	
	  }
	}

Error Response	

```
{
    status : 401,
    error: {
        Message : "Unsuccessful"
    }
}
```

5. **Admin APIs**

a. **Add Hotel/Flight/Car List API**
   This API will be used to a new Vendor which will have a list of Hotels/Flights/Cars. It will be used to store End Point URL of the vendor which will be later used for getting a list of Hotels/Flights/Cars from their database.

Title	Add Hotel/Flight/Car List
URL	/admin/addlist
Method	POST
URL Params	N/A

Data Params	{
	    VendorName: String,
	    VendorType: String (Hotel/Flight/Car),
	    Method: String,
	    VendorEndPoint: String
	}

Success Response	{ status : 200,

	result: {
	    Message : "Hotel List successfully added",
	    hotels: Array
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Error adding hotel list"
	  }
	}

b. **Get Hotel/Flight/Car Vendors List API**
   This API will be used to get a list Hotel Vendors.

Title	Get Hotel/Flight/Car Vendor List
URL	/admin/getvendorlist
Method	GET
URL Params	{vendortype: String}
Data Params	N/A

Success Response	{
	  status : 200,
	  result: {
	    vendors: Array
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Error getting hotel/flight/car vendor list"
	  }
	}

c. **Edit Hotel/Flight/Car Vendors List API**
   This API will be used to edit a Hotel/Flight/Car Vendor

Title	Edit Hotel/Flight/Car Vendor
URL	/admin/editvendorlist
Method	PUT
URL Params	{vendorid: String}

Data Params	{
	   VendorName: String,
	   VendorType: String (Hotel/Flight/Car),
	   Method: String,
	   VendorEndPoint: String
	}

Success Response	{
	   status : 200,
	   result: {
	     {
	       VendorName: String,
	       VendorType: String (Hotel/Flight/Car),
	       Method: String,
	       VendorEndPoint: String
	     }
	   }
	}

Error Response	{
	   status : 401,
	   error: {
	     Message : "Error updating hotel/flight/car vendor list"
	   }
	}

d. **Delete Hotel/Flight/Car Vendors List API**
   This API will be used to delete a Hotel/Flight/Car Vendor.

Title	Delete Hotel/Flight/Car Vendor
URL	/admin/deletevendorlist
Method	DELETE
URL Params	{vendorid: String}
Data Params	N/A

Success Response	{
	   status : 200,
	   result: {
	       Message : "Hotel/flight/car vendor deleted successfully"
	
	      }
	}

Error Response	{
	   status : 401,
	   error: {
	    Message : "Error deleting hotel/flight/car vendor list"
	    }
	}

e. **Get user List API**
   This API will be used to get a list of Kayak Users.

Title	Get User List
URL	/admin/getuserlist
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	{ status : 200,

	result: {
	    users: Array
	  }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Error getting user list"
	  }
	}

f. **Delete user account API**
   This API will be used to delete a user account.

Title	Edit Hotel/Flight/Car Vendor
URL	/admin/deleteuser
Method	DELETE
URL Params	{userid: String}
Data Params	N/A

Success Response	{
	  status : 200,
	  result: {
	      Message : "User deleted successfully"
	
	    }
	}

Error Response	{
	  status : 401,
	  error: {
	    Message : "Error deleting user"
	  }
	}

g. **Get bills list API**
   This API will be used to get a list of bills.

Title	Get bills list
URL	/admin/getbilllist
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	``` { status : 200, result: { bills: Array } } ```
Error Response	``` { status : 401, error: { Message : "Error getting bill list" } } ```

h. **Get bill list by date API**
   This API will be used to get a list of bills for a particular date.

Title	Get Bill List by date
URL	/admin/getbilllistbydate
Method	GET
URL Params	{date: DATE}
Data Params	N/A
Success Response	``` { status : 200, result: { bills: Array ```

	} } }

Error Response	{
	   status : 401,
	   error: {
	      Message : "Error getting bill list"
	   }
	}

I. **Get bill list by month API**
   This API will be used to get a list of bills for a particular month.

Title	Get Bill List by month
URL	/admin/getbilllistbymonth
Method	GET
URL Params	{month: String}
Data Params	N/A

Success Response	{
	   status : 200,
	   result: {
	      bills: Array
	   }
	}

Error Response	{
	   status : 401,
	   error: {
	      Message : "Error getting bill list"
	   }
	}