EXPEDIA BACKEND DESIGN

Internal API Wrappers

FlightInfo | **FlightRequest**

HotelInfo | **HotelRequest**

PaymentInfo

NOTE there should be an internal API wrapper for each external API

<<interface>> IFlightAPI
- + search(FlightRequest): Collection<FlightInfo>
- + reserve(FlightInfo, FlightRequest): boolean
- + cancelReservation(FlightInfo, FlightRequest): boolean

<<interface>> IHotelAPI
- + search(HotelRequest): Collection<HotelInfo>
- + reserve(HotelInfo, HotelRequest): boolean
- + cancelReservation(HotelInfo, HotelRequest): boolean

<<interface>> IPaymentAPI
- + makePayment(PaymentInfo): boolean

FlightAPIFactory
- + createFlightAPI(FlightInfo): IFlightAPI
- + getAllFlightAPIs(): Collection<IFlightAPI>

HotelAPIFactory
- + createHotelAPI(HotelInfo): IHotelAPI
- + getAllHotelAPIs(): Collection<IHotelAPI>

PaymentAPIFactory
- + createPaymentAPI(PaymentInfo): IPaymentAPI
- + getAllPaymentAPIs(): Collection<IPaymentAPI>

WRAPS

External APIs

- TukishAirlinesOnlineAPI | AirCandaOnlineAPI
- HiltonHotelAPI | MarriottHotelAPI
- PayPalOnlinePaymentAPI | SquarePaymentAPI | StripePaymentAPI

Expedia Reservation Logic

FlightReservationHandler
- - IFlightAPI
- + constructor(FlightInfo, FlightRequest)
- + reserve()
- + cancelReservation()

HotelReservationHandler
- - IHotelAPI
- + constructor(HotelInfo, HotelRequest)
- + reserve()
- + cancelReservation()

NOTE every ReservationHandler uses its corresponding APIFactory to get the specific API object

FlightReservation
- - FlightInfo
- - FlightRequest
- - FlightReservationHandler

HotelReservation
- - HotelInfo
- - HotelRequest
- - HotelReservationHandler

Itinerary
- - Collection<ReservationItem>
- + addReservationItem(ReservationItem): void

<<interface>> ReservationItem
- + totalCost(): double
- + reserve(): boolean
- + cancelReservation(): boolean

Expedia API

ExpediaBookingAPI
- + book(Iternary, PaymentInfo, UserInfo)

ExpediaFlightAPI
- - Collection<IFlightAPI>
- + search(FlightRequest)

ExpediaHotelAPI
- - Collection<IHotelAPI>
- + search(HotelRequest)

UserItinerariesDAO
- + addIternaryToUser(userID, Iternary)

NOTE the frontend should use one of the available API classes to perform its operations

Expedia REST API

ExpediaRestAPI
- + searchFlights(flightRequestJson): string
- + searchHotels(hotelRequestJson): string
- + bookIternary(bookingDataJson): string

The class in this module just converts the json string to objects and then use the Expedia API classes to make the request. NOTE you can also have different API mechanisms but they all should use the main Expedia API to perform the task

User Authentication

UserInfo

UserDAO
- + addUser(UserInfo): boolean
- + getUser(username): UserInfo
- + userExist(username): boolean

AuthenticationStatus
- + authenticatedSuccessfully(): boolean
- + authenticationMsg(): string
- + failed(reason): AuthenticationStatus
- + success(UserInfo, reason): AuthenticationStatus

ExpediaAuthenticationAPI
- + login(username, password): AuthenticationStatus
- + signup(email, username, password, userType): AuthenticationStatus