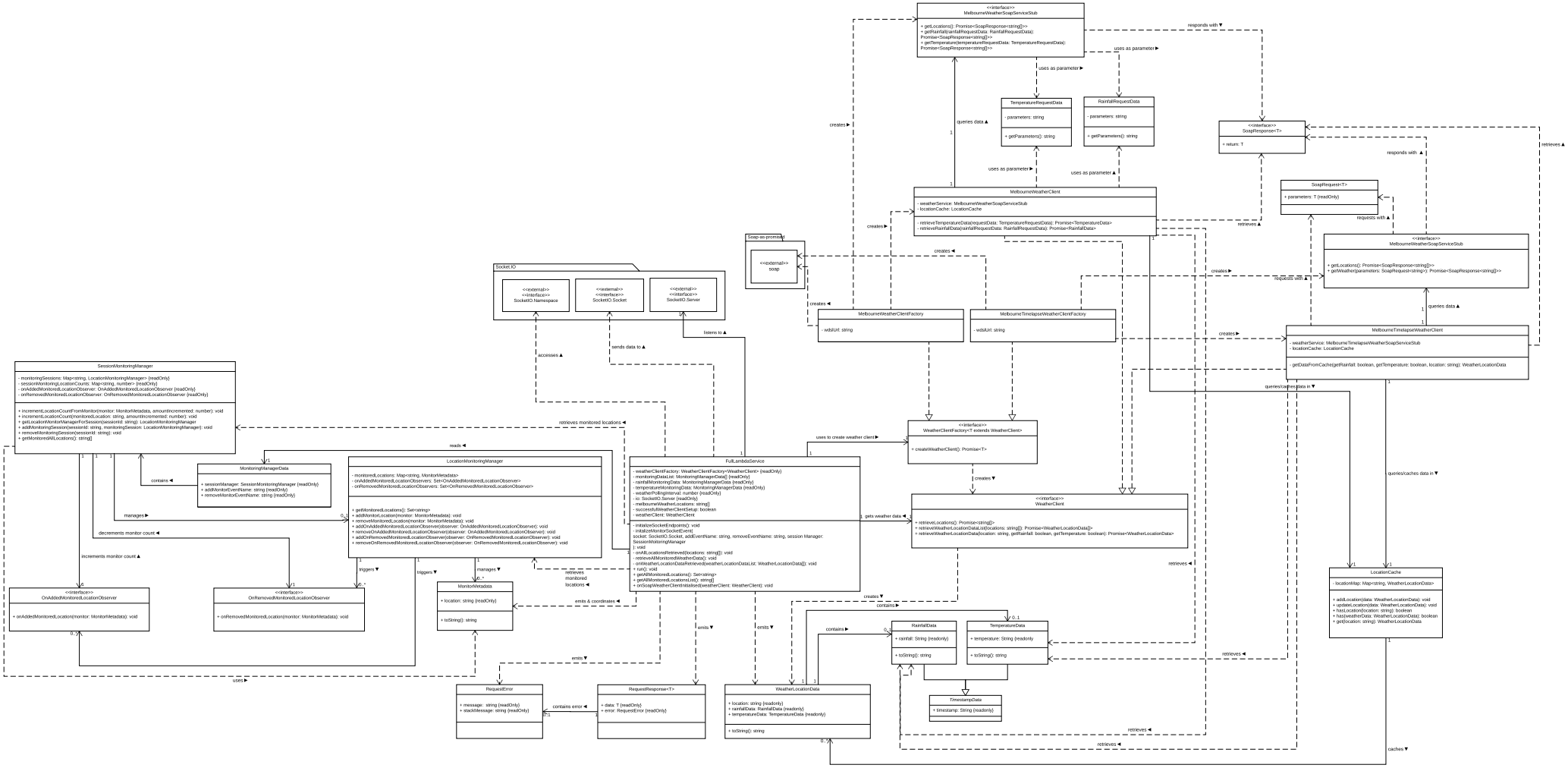
<<interface>>
MelbourneWeatherSoapServiceStub
+ getLocations(): Promise<SoapResponse<string[]>>
+ getRainfall(rainfallRequestData: RainfallRequestData): Promise<SoapResponse<string[]>>
+ getTemperature(temperatureRequestData: TemperatureRequestData): Promise<SoapResponse<string[]>>
responds with
uses as parameter
uses as parameter
TemperatureRequestData
- parameters: string
+ getParameters(): string
RainfallRequestData
- parameters: string
+ getParameters(): string
queries data
creates
<<interface>>
SoapResponse<T>
+ return: T
retrieves
responds with
uses as parameter
uses as parameter
MelbourneWeatherClient
- weatherService: MelbourneWeatherSoapServiceStub
- locationCache: LocationCache
- retrieveTemperatureData(requestData: TemperatureRequestData): Promise<TemperatureData>
- retrieveRainfallData(rainfallRequestData: RainfallRequestData): Promise<RainfallData>
SoapRequest<T>
+ parameters: T (readOnly)
requests with
retrieves
creates
Soap-as-promised
<<external>>
soap
creates
<<interface>>
MelbourneWeatherSoapServiceStub
+ getLocations(): Promise<SoapResponse<string[]>>
+ getWeather(parameters: SoapRequest<string>): Promise<SoapResponse<string[]>>
creates
requests with
Socket.IO
<<external>>
<<interface>>
SocketIO.Namespace
<<external>>
<<interface>>
SocketIO.Socket
<<external>>
<<interface>>
SocketIO.Server
listens to
sends data to
accessses
creates
MelbourneWeatherClientFactory
- wdsUrl: string
MelbourneTimelapseWeatherClientFactory
- wdsUrl: string
creates
queries data
MelbourneTimelapseWeatherClient
- weatherService: MelbourneTimelapseWeatherSoapServiceStub
- locationCache: LocationCache
- getDataFromCache(getRainfall: boolean, getTemperature: boolean, location: string): WeatherLocationData
queries/caches data in
queries/caches data in
SessionMonitoringManager
- monitoringSessions: Map<string, LocationMonitoringManager> (readOnly)
- sessionMonitoringLocationCounts: Map<string, number> (readOnly)
- onAddedMonitoredLocationObserver: OnAddedMonitoredLocationObserver (readOnly)
- onRemovedMonitoredLocationObserver: OnRemovedMonitoredLocationObserver (readOnly)
+ incrementLocationCountFromMonitor(monitor: MonitorMetadata, amountIncremented: number): void
+ incrementLocationCount(monitoredLocation: string, amountIncremented: number): void
+ getLocationMonitorManagerForSession(sessionId: string): LocationMonitoringManager
+ addMonitoringSession(sessionId: string, monitoringSession: LocationMonitoringManager): void
+ removeMonitoringSession(sessionId: string): void
+ getMonitoredAllLocations(): string[]
retrieves monitored locations
reads
contains
MonitoringManagerData
+ sessionManager: SessionMonitoringManager (readOnly)
+ addMonitorEventName: string (readOnly)
+ removeMonitorEventName: string (readOnly)
LocationMonitoringManager
- monitoredLocations: Map<string, MonitorMetadata>
- onAddedMonitoredLocationObservers: Set<OnAddedMonitoredLocationObserver>
- onRemovedMonitoredLocationObservers: Set<OnRemovedMonitoredLocationObserver>
+ getMonitoredLocations(): Set<string>
+ addMonitorLocation(monitor: MonitorMetadata): void
+ removeMonitoredLocation(monitor: MonitorMetadata): void
+ addOnAddedMonitoredLocationObserver(observer: OnAddedMonitoredLocationObserver): void
+ removeOnAddedMonitoredLocationObserver(observer: OnAddedMonitoredLocationObserver): void
+ addOnRemovedMonitoredLocationObserver(observer: OnRemovedMonitoredLocationObserver): void
+ removeOnRemovedMonitoredLocationObserver(observer: OnRemovedMonitoredLocationObserver): void
manages
decrements monitor count
increments monitor count
FullLambdaService
- weatherClientFactory: WeatherClientFactory<WeatherClient> (readOnly)
- monitoringDataList: MonitoringManagerData[] (readOnly)
- rainfallMonitoringData: MonitoringManagerData (readOnly)
- temperatureMonitoringData: MonitoringManagerData (readOnly)
- weatherPollingInterval: number (readOnly)
- io: SocketIO.Server (readOnly)
- melbourneWeatherLocations: string[]
- successfulWeatherClientSetup: boolean
- weatherClient: WeatherClient
- initializeSocketEndpoints(): void
- initializeMonitorSocketEvent(
socket: SocketIO.Socket, addEventName: string, removeEventName: string, session Manager:
SessionMonitoringManager
): void
- onAllLocationsRetrieved(locations: string[]): void
- retrieveAllMonitoredWeatherData(): void
- onWeatherLocationDataRetrieved(weatherLocationDataList: WeatherLocationData[]): void
+ run(): void
+ getAllMonitoredLocations(): Set<string>
+ getAllMonitoredLocationsList(): string[]
+ onSoapWeatherClientInitialized(weatherClient: WeatherClient): void
uses to create weather client
<<interface>>
WeatherClientFactory<T extends WeatherClient>
+ createWeatherClient(): Promise<T>
creates
gets weather data
<<interface>>
WeatherClient
+ retrieveLocations(): Promise<string[]>
+ retrieveWeatherLocationDataList(locations: string[]): Promise<WeatherLocationData[]>
+ retrieveWeatherLocationData(location: string, getRainfall: boolean, getTemperature: boolean): Promise<WeatherLocationData>
retrieves
LocationCache
- locationMap: Map<string, WeatherLocationData>
+ addLocation(data: WeatherLocationData): void
+ updateLocation(data: WeatherLocationData): void
+ hasLocation(location: string): boolean
+ hasLocation(weatherData: WeatherLocationData): boolean
+ get(location: string): WeatherLocationData
<<interface>>
OnAddedMonitoredLocationObserver
+ onAddedMonitoredLocation(monitor: MonitorMetadata): void
<<interface>>
OnRemovedMonitoredLocationObserver
+ onRemovedMonitoredLocation(monitor: MonitorMetadata): void
triggers
triggers
manages
MonitorMetadata
+ location: string (readOnly)
+ toString(): string
retrieves monitored locations
emits & coordinates
creates
contains
contains
RainfallData
+ rainfall: String (readonly)
+ toSiring(): string
TemperatureData
+ temperature: String (readonly
+ toSiring(): string
TimestampData
+ timestamp: String (readonly)
retrieves
retrieves
emits
emits
emits
emits
uses
RequestError
+ message: string (readOnly)
+ stackMessage: string (readOnly)
contains error
RequestResponse<T>
+ data: T (readOnly)
+ error: RequestError (readOnly)
WeatherLocationData
+ location: string (readonly)
+ rainfallData: RainfallData (readonly)
+ temperatureData: TemperatureData (readonly)
+ toString(): string
retrieves
retrieves
caches