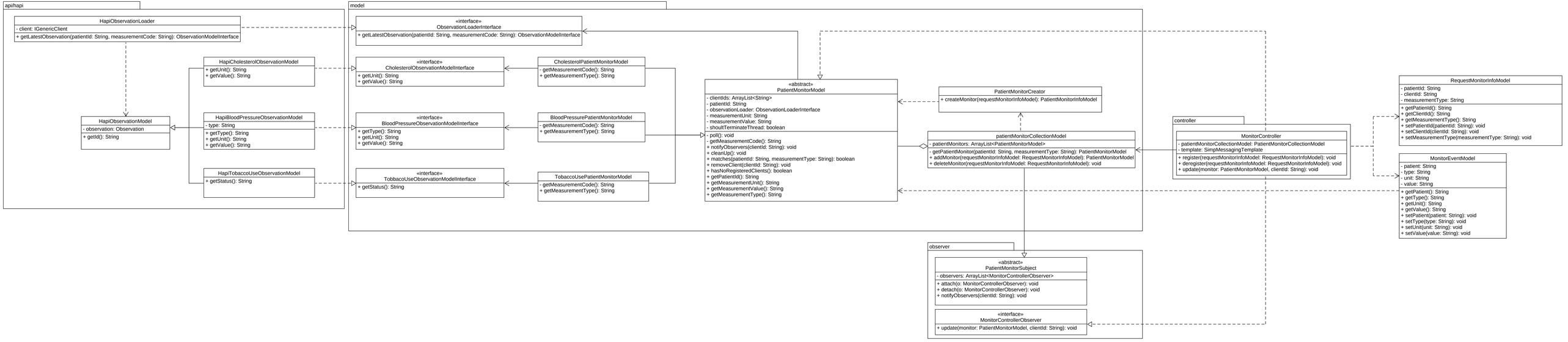
api/hapi
HapiObservationLoader
- client: IGenericClient
+ getLatestObservation(patientId: String, measurementCode: String): ObservationModelInterface
HapiCholesterolObservationModel
+ getUnit(): String
+ getValue(): String
HapiObservationModel
- observation: Observation
+ getId(): String
HapiBloodPressureObservationModel
- type: String
+ getType(): String
+ getUnit(): String
+ getValue(): String
HapiTobaccoUseObservationModel
+ getStatus(): String
model
«interface»
ObservationLoaderInterface
+ getLatestObservation(patientId: String, measurementCode: String): ObservationModelInterface
«interface»
CholesterolObservationModelInterface
+ getUnit(): String
+ getValue(): String
CholesterolPatientMonitorModel
- getMeasurementCode(): String
+ getMeasurementType(): String
«interface»
BloodPressureObservationModelInterface
+ getType(): String
+ getUnit(): String
+ getValue(): String
BloodPressurePatientMonitorModel
- getMeasurementCode(): String
+ getMeasurementType(): String
«interface»
TobbacoUseObservationModelInterface
+ getStatus(): String
TobaccoUsePatientMonitorModel
- getMeasurementCode(): String
+ getMeasurementType(): String
«abstract»
PatientMonitorModel
- clientIds: ArrayList<String>
- patientId: String
- observationLoader: ObservationLoaderInterface
- measurementUnit: String
- measurementValue: String
- shoultTerminateThread: boolean
- poll(): void
- getMeasurementCode(): String
+ notifyObservers(clientId: String): void
+ cleanUp(): void
+ matches(patientId: String, measurementType: String): boolean
+ removeClient(clientId: String): void
+ hasNoRegisteredClients(): boolean
+ getPatientId(): String
+ getMeasurementUnit(): String
+ getMeasurementValue(): String
+ getMeasurementType(): String
PatientMonitorCreator
+ createMonitor(requestMonitorInfoModel): PatientMonitorInfoModel
patientMonitorCollectionModel
- patientMonitors: ArrayList<PatientMonitorModel>
- getPatientMonitor(patientId: String, measurementType: String): PatientMonitorModel
+ addMonitor(requestMonitorInfoModel: RequestMonitorInfoModel): PatientMonitorModel
+ deleteMonitor(requestMonitorInfoModel: RequestMonitorInfoModel): void
controller
MonitorController
- patientMonitorCollectionModel: PatientMonitorCollectionModel
- template: SimpMessagingTemplate
+ register(requestMonitorInfoModel: RequestMonitorInfoModel): void
+ deregister(requestMonitorInfoModel: RequestMonitorInfoModel): void
+ update(monitor: PatientMonitorModel, clientId: String): void
RequestMonitorInfoModel
- patientId: String
- clientId: String
- measurementType: String
+ getPatientId(): String
+ getClientId(): String
+ getMeasurementType(): String
+ setPatientId(patientId: String): void
+ setClientId(clientId: String): void
+ setMeasurementType(measurementType: String): void
MonitorEventModel
- patient: String
- type: String
- unit: String
- value: String
+ getPatient(): String
+ getType(): String
+ getUnit(): String
+ getValue(): String
+ setPatient(patient: String): void
+ setType(type: String): void
+ setUnit(unit: String): void
+ setValue(value: String): void
observer
«abstract»
PatientMonitorSubject
- observers: ArrayList<MonitorControllerObserver>
+ attach(o: MonitorControllerObserver): void
+ detach(o: MonitorControllerObserver): void
+ notifyObservers(clientId: String): void
«interface»
MonitorControllerObserver
+ update(monitor: PatientMonitorModel, clientId: String): void