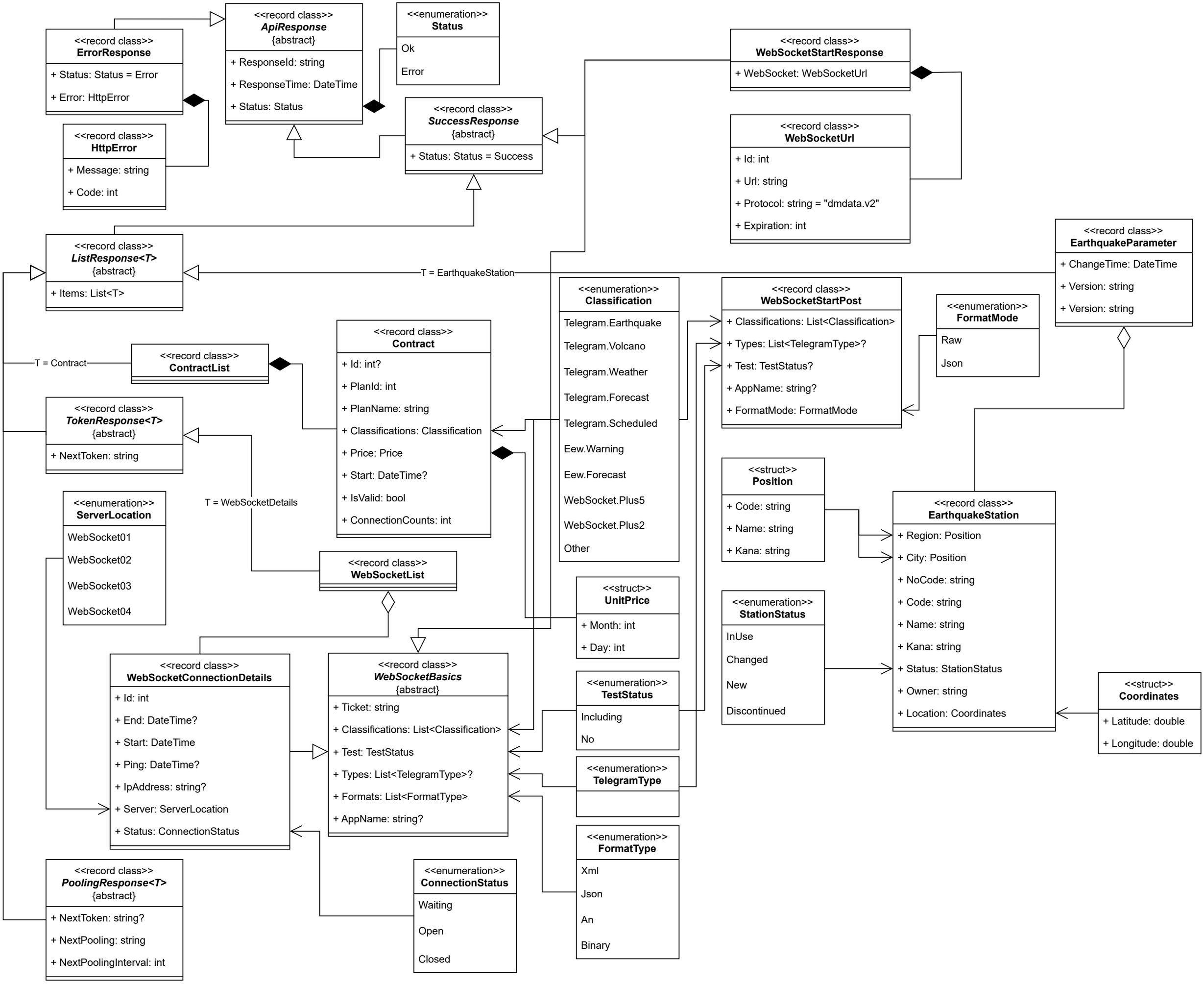
<<record class>>
ErrorResponse
+ Status: Status = Error
+ Error: HttpError
<<record class>>
ApiResponse
{abstract}
+ ResponseId: string
+ ResponseTime: DateTime
+ Status: Status
<<enumeration>>
Status
Ok
Error
<<record class>>
HttpError
+ Message: string
+ Code: int
<<record class>>
SuccessResponse
{abstract}
+ Status: Status = Success
<<record class>>
WebSocketStartResponse
+ WebSocket: WebSocketUrl
<<record class>>
WebSocketUrl
+ Id: int
+ Url: string
+ Protocol: string = "dmdata.v2"
+ Expiration: int
<<record class>>
ListResponse<T>
{abstract}
+ Items: List<T>
T = EarthquakeStation
<<record class>>
EarthquakeParameter
+ ChangeTime: DateTime
+ Version: string
+ Version: string
T = Contract
<<record class>>
ContractList
<<record class>>
Contract
+ Id: int?
+ PlanId: int
+ PlanName: string
+ Classifications: Classification
+ Price: Price
+ Start: DateTime?
+ IsValid: bool
+ ConnectionCounts: int
<<enumeration>>
Classification
Telegram.Earthquake
Telegram.Volcano
Telegram.Weather
Telegram.Forecast
Telegram.Scheduled
Eew.Warning
Eew.Forecast
WebSocket.Plus5
WebSocket.Plus2
Other
<<record class>>
WebSocketStartPost
+ Classifications: List<Classification>
+ Types: List<TelegramType>?
+ Test: TestStatus?
+ AppName: string?
+ FormatMode: FormatMode
<<enumeration>>
FormatMode
Raw
Json
<<record class>>
TokenResponse<T>
{abstract}
+ NextToken: string
T = WebSocketDetails
<<enumeration>>
ServerLocation
WebSocket01
WebSocket02
WebSocket03
WebSocket04
<<record class>>
WebSocketList
<<struct>>
Position
+ Code: string
+ Name: string
+ Kana: string
<<record class>>
EarthquakeStation
+ Region: Position
+ City: Position
+ NoCode: string
+ Code: string
+ Name: string
+ Kana: string
+ Status: StationStatus
+ Owner: string
+ Location: Coordinates
<<struct>>
UnitPrice
+ Month: int
+ Day: int
<<enumeration>>
StationStatus
InUse
Changed
New
Discontinued
<<record class>>
WebSocketConnectionDetails
+ Id: int
+ End: DateTime?
+ Start: DateTime
+ Ping: DateTime?
+ IpAddress: string?
+ Server: ServerLocation
+ Status: ConnectionStatus
<<record class>>
WebSocketBasics
{abstract}
+ Ticket: string
+ Classifications: List<Classification>
+ Test: TestStatus
+ Types: List<TelegramType>?
+ Formats: List<FormatType>
+ AppName: string?
<<enumeration>>
TestStatus
Including
No
<<enumeration>>
TelegramType
<<struct>>
Coordinates
+ Latitude: double
+ Longitude: double
<<enumeration>>
FormatType
Xml
Json
An
Binary
<<record class>>
PoolingResponse<T>
{abstract}
+ NextToken: string?
+ NextPooling: string
+ NextPoolingInterval: int
<<enumeration>>
ConnectionStatus
Waiting
Open
Closed