```
package org.ekkescorner.dsl.qtws {

    import org.lunifera.dsl.ext.cpp.qt.datatypes.*;
    import org.lunifera.dsl.ext.cpp.qt.lib.types.annotation.*;

    @TargetOS("Qt")
    @FriendsClass("DataUtil,DataServer")
    dto SettingsData {
        domainKey int id;
        var int version;
        var QString apiVersion;
        var Timestamp lastUpdate;
        var bool isProductionEnvironment;
        var int primaryColor;
        var int accentColor;
        var bool darkTheme;
        var bool useMarkerColors;
        var bool defaultMarkerColors;
        var QString markerColors;
        var bool hasPublicCache;
        var bool useCompactJsonFormat;
        var int lastUsedNumber;
        var QString publicRoot4Dev;
        var bool autoUpdate;
        var int autoUpdateEveryHours;
        var Timestamp lastUpdateStamp;
        // -1,0: oneHandComfort, 1: bottom, 2: classic
        var int navigationStyle;
        var bool oneMenuButton;
        var bool classicStackNavigation;
    }

    @CachePolicy("-R-")
    @TargetOS("Qt")
    dto Conference {
        domainKey int id;
        // per ex 201801 (BOSTON) 201802 (BERLIN)
        var QString conferenceName;
        var QString conferenceCity;
        var QString address;
        // TODO not used yet
        var QString timeZoneName;
        var int timeZoneOffsetSeconds;
        @DateFormatString("yyyy-MM-dd")
        var Date conferenceFrom;
```

```
    @DateFormatString("yyyy-MM-dd")
    var Date conferenceTo;
    var QString hashTag;
    var QString homePage;
    // TODO Map Integration GeoCoordinate
    var QString coordinate;
    var int lastRoomId;
    var int lastSessionTrackId;
    // same as id + nnn per ex 201801001
    var int lastGenericSessionId;
    // implicitly ref lazy
    // sorted: day,start
    var Day [ 0 .. * ] days;
    var SessionTrack [ 0 .. * ] tracks;
    var Room [ 0 .. * ] rooms;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Room {
    domainKey int roomId;
    var int conference;
    var QString roomName;
    var bool inAssets;

    // implicitly ref lazy
    // sorted day,start
    var Session [ 0 .. * ] sessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Session {
    domainKey int sessionId;
    var int conference;
    var bool isDeprecated;
    // sort day start 010900
    var QString sortKey;
    var bool isTraining;
    var bool isLightning;
    var bool isKeynote;
    var bool isSession;
    var bool isCommunity;
    var bool isUnconference;
    var bool isMeeting;
```

```
    // Generic Session
    var bool isGenericScheduleSession;
    var bool isBreak;
    var bool isLunch;
    var bool isEvent;
    var bool isRegistration;
    //
    var QString title;
    var QString description;
    var QString sessionType;
    @DateFormatString("HH:mm")
    var Time startTime;
    @DateFormatString("HH:mm")
    var Time endTime;
    var int minutes;
    var QString abstractText;

    // don't want to cache Sessions if marked as Favorite
    // isFavorite will be updated at Startup
    transient bool isFavorite;

    // implicitly ref lazy
    var Speaker [0 .. *] presenter;
    var SessionTrack [0 .. *] sessionTracks;
    ref lazy Day [ 1 ] sessionDay;
    ref lazy Room [ 0 .. 1 ] room;
}

@TargetOS("Qt")
// Scheduled Sessions
dto Favorite {
    domainKey int sessionId;
    var int conference;
    ref lazy Session [1] session;
}

@TargetOS("Qt")
@CachePolicy("-R-T-")
// session lists created on demand
dto SessionLists {
    domainKey QString uuid;
    var int conference;
    // implicitly ref lazy
    // scheduled sessions: favorites
    var Session [ 0 .. * ] scheduledSessions;
```

```
    var Session [ 0 .. * ] sameTimeSessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Speaker {
    domainKey int speakerId;
    var bool isDeprecated;
    // 'SCHUL'
    var QString sortKey;
    // 'S'
    var QString sortGroup;
    var QString name;
    var QString publicName;
    var QString bio;

    ref lazy SpeakerImage [0..1] speakerImage;
    // implicitly ref lazy
    // sort day, start
    var Session [0 .. *] sessions;
    var Conference [0 .. *] conferences;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto SpeakerImage {
    domainKey int speakerId;
    var QString originImageUrl;
    var bool downloadSuccess;
    var bool downloadFailed;
    var bool inAssets;
    var bool inData;
    var QString suffix;
    // 0 .. 4
    var int maxScaleFactor;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto SessionTrack {
    domainKey int trackId;
    var int conference;
    var QString name;
    var QString color;
    var bool inAssets;
```

```
    // implicitly ref lazy
    // sort day start
    var Session [0 .. *] sessions;
}

@CachePolicy("-R-")
@TargetOS("Qt")
dto Day {
    domainKey int id;
    var int conference;
    // 0..6
    var int weekDay;
    @DateFormatString("yyyy-MM-dd")
    var Date conferenceDay;
    // implicitly ref lazy
    // sort day start
    var Session [0 .. *] sessions;
}

// A P I   data delivered as payload from server via REST API
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto SessionAPI {
    @ForeignPropertyName("id")
    domainKey int sessionId;
    var QString title;
    var QString description;
    @DateFormatString("HH:mm")
    @ForeignPropertyName("start")
    var Time startTime;
    var QString duration;
    var QString room;
    @ForeignPropertyName("tracks")
    var SessionTrackAPI [0 .. *] sessionTracks;
    @ForeignPropertyName("persons")
    var PersonsAPI [0 .. *] presenter;

}
@CachePolicy("-R-T-")
@TargetOS("Qt")
dto PersonsAPI {
    @ForeignPropertyName("id")
    domainKey int speakerId;
}
@CachePolicy("-R-T-")
```

```
    @TargetOS("Qt")
    dto SessionTrackAPI {
        domainKey QString uuid;
        var QString name;
        var QString color;
    }
    @CachePolicy("-R-T-")
    @TargetOS("Qt")
    dto SpeakerAPI {
        domainKey int id;
        @ForeignPropertyName("first_name")
        var QString firstName;
        @ForeignPropertyName("last_name")
        var QString lastName;
        @ForeignPropertyName("abstract")
        var QString bio;
        var QString avatar;
    }

}
```