FireApp – Domain Objects

This file contains a short description of the objects that are needed to interact with the FireApp-API

Table of contents

FireEvent

```csharp
/// <summary>
/// Is an abstract version of a message that was sent by a fire alarm system.
/// </summary>
public class FireEvent
{
    // A composite primary key consisting of sourceId and eventId.
    public FireEventId Id { get; set; }

    // Time when the FireEvent accored.
    public DateTime TimeStamp { get; set; }

    // Name of the Fire detector (e.g. MG 13/5).
    public string TargetId { get; set; }

    // Description/location of the fire detector (e.g. Melder Büro).
    public string TargetDescription { get; set; }

    // Type of the event that ocurred.
    public EventTypes EventType { get; set; }
}
```

FireEventId

```
/// <summary>
/// This class is needed because liteDB can not create a composite key itself.
/// </summary>
public class FireEventId
{
    public FireEventId() { }
    public FireEventId(int sourceId, int eventId)
    {
        this.SourceId = sourceId;
        this.EventId = eventId;
    }

    // Id of the FireAlarmSystem.
    public int SourceId { get; set; }

    // This id distinguishes this FireEvent from FireEvents
    // of the same FireAlarmSystem.
    public int EventId { get; set; }
}
```

EventTypes

```
/// <summary>
/// Makes it easier to determine the type of a FireEvent.
/// </summary>
public enum EventTypes
{
    alarm = 1,
    disfunction = 2,
    test = 3,
    reset = 4,
    info = 5,
    deactivated = 6,
    prealarm = 7,
    activation = 8
};
```

FireAlarmSystem

```csharp
/// <summary>
/// This class represents a fire alarm system.
/// </summary>
public class FireAlarmSystem
{
    // Identifier of the FireAlarmSystem.
    public int Id { get; set; }

    // Name of the company that owns the FireAlarmSystem.
    public string Company { get; set; }

    // Short description of the FireAlarmSystem.
    public string Description { get; set; }

    // Country where the FireAlarmSystem is installed.
    public string Country { get; set; }

    // City where the FireAlarmSystem is installed.
    public string City { get; set; }

    // Postal code of the city where the FireAlarmSystem is installed.
    // Not an integer becaus it may contain characters (e.g. A-4020).
    public string PostalCode { get; set; }

    // Address of the building where the FireAlarmSystem is installed.
    public string Address { get; set; }

    // List of the identifiers of FireBrigades
    // that should have access to certain information.
    public HashSet<int> FireBrigades { get; set; }

    // List of the identifiers of ServiceGroups
    // that should have access to certain information.
    public HashSet<int> ServiceGroups { get; set; }
}
```

UserGroup

```csharp
public class UserGroup
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

FireBrigade

```csharp
/// <summary>
/// This class is used as a link between Users and FireAlarmSystems.
/// </summary>
public class FireBrigade : UserGroup
{
}
```

ServiceGroup

```csharp
/// <summary>
/// This class is used as a link between Users and FireAlarmSystems.
/// </summary>
public class ServiceGroup : UserGroup
{
}
```

User

```csharp
/// <summary>
/// This class represents a user of this application.
/// </summary>
public class User
{
    private string token;
    public string Id { get; set; }
    public string Password { get; set; }

    public UserTypes UserType { get; set; }

    // This property is a set, because a User can be part of
    // several institutions of the same type (fire brigade, service group, ...).
    public HashSet<int> AuthorizedObjectIds { get; set; }

    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }

    // Is used to identify the user when sending an request to the API.
    public string Token
    {
        get { return token; }
        set { this.token = value; TokenCreationDate = DateTime.Now; }
    }

    public DateTime TokenCreationDate { get; set; }
}
```

UserTypes

```csharp
/// <summary>
/// Helps to distinguish the different types of users.
/// </summary>
public enum UserTypes
{
    unauthorized = -1,
    admin = 0,
    fireSafetyEngineer = 1,
    fireFighter = 2,
    servicemember = 3
}
```

UserLogin

```csharp
/// <summary>
/// This class is used to transfer the login credentials between two systems.
/// </summary>
public class UserLogin
{
    public string Username { get; set; }
    public string Password { get; set; }
}
```