# Tiny Soldiers

# Small assignment II

In this assignment we want to get familiar (*and even comfortable*) with concepts such as paging, content negotiation and HATEOAS. In order to become familiar with those concepts, we are going to implement routes that make use of those concepts. We are going to create a small web service for a small company called **Tiny Soldiers** which specializes in creating really small paintable models. They want you to create their API because of your expertize on these subjects.

## Template

The template for this assignment can be downloaded in **Canvas** (*template.zip*) and it includes:

- An all-inclusive ticket for two to Seoul, South Korea where you will be staying at a 5-star hotel for two weeks
- Within **Models/** folder
  - **HyperMediaModel.cs**
  - **Envelope.cs**
  - **Model.cs**
- Within **Data/** folder
  - **DataContext.cs** - *It stores all the data in a property called Models, but it won't be accessible until you finish creating your models. The property Models has two extension methods ToDetails() and ToLightWeight() which return either a list of ModelDetailDTO or ModelDTO. It can be accessed statically, e.g. DataContext.Models.ToDetails()*
- Within **Extensions/** folder
  - **ListExtensions.cs** - *It contains extensions methods for the **Models** property within **DataContext***
  - **HyperMediaExtensions.cs** - *It contains an extension method called **AddReference()** which accepts two parameters: **key** and **value**. Which can be used to implement the HATEOAS structure of the model, e.g.* ***DataContext.Models.ToDetails().FirstOrDefault(m => m.Id == id).Links.AddReference("self", "http://link-to-self");***

## Assignment description

Below is a description of the functionality that should be provided in this assignment:

- **(5%)** There should be a single controller called **ModelController**
- **(10%)** In **ModelController**, there should be two routes
  - GetAllModels()
  - GetModelById()
- **(10%)** There should be two models within a folder called **Models/**
  - ModelDTO
  - ModelDetailsDTO
- **(5%)** Each model should derive from a model called **HyperMediaModel**. *This model is already within the **Models/** folder and implemented*
- **(25%)** The route **GetAllModels()** should be implemented like this:
  - It can receive two query parameters called: *pageNumber (default 1)* and *pageSize (default 10)*
  - It should use those query parameters to page the data and return an **Envelope** (*it resides within the **Models/** folder*) of **ModelDTO**
  - The _links property found in **HyperMediaModel** should be properly filled out
- **(25%)** The route **GetModelById()** should be implemented like this:
  - It receives an id as URL parameter and it should be used to find the corresponding model based on that id
  - It should return an **ModelDetailsDTO**
  - The _links property found in **HyperMediaModel** should be properly filled out
- **(20%)** The client can request all data with an **Accept-Language** header, which supports both **en-US** and **de-DE**. The data should be served on the language requested. It should default to **en-US** if the **Accept-Language** header is not present. When accessing the

extension methods **ToDetails()** and **ToLightWeight()** they accept an optional parameter called language which can be passed in to change to either English (*en-US*) or German (*de-DE*). *This does not need to be perfect, but must at least handle the basics.*

# Models

The models are described here below:

**MODELDTO**

Properties:

- Id : int
- Name : string
- Race : string
- Price : double

**MODELDETAILSDTO**

Properties:

- Id : int
- Name : string
- Race : string
- Price : double
- Description : string
- Rarity : int
- DifficultyLevel : string
- YearOfRelease : int
- ImageUrl : string

# Submission

A single compressed file (**.zip, .rar**) should be submitted to **Canvas**. *Don't forget to comment your group members in the comment section below (excluding the one who submitted).*