**Homework**

**Common**

This homework based on results from **Introduction** module.

**Task 1. DI/IoC and unit testing for controllers**

In this task you should:

* Switch all available controllers to inject dependencies trough constructor
* Register in Startup.ConfigureServices() all necessary dependencies classes
* Add separate projects with tests
* Create at least one test for every public action
  + Every external dependency should be mocked or stubbed!

**Materials:**

* [Dependency injection in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection)
* [Dependency injection into controllers in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/dependency-injection)
* [Testing controller logic in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/testing)

**Task 2. Action Results and File Uploads**

Add new Action into Category Controller, which return category image:

* Image should be sending as binary stream with correct Content-Type (for referencing from HTML pages)

**Note**. Please note that test data for Northwind Categories contain broken images (it’s BMP pictures, but first 78 bytes – garbage)

Add links to images in category list.

Add edit form for change image in Category (upload new image)

**Materials:**

* [Format response data in ASP.NET Core MVC](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/formatting)
* [File uploads in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/file-uploads)

**Task 3. Routing configuration**

Configure routing for enable getting images not only by standard path **{controller}/{action}/…**, but also by **images/{image\_id}**

**Materials:**

* [Routing in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/routing)
* [Routing to controller actions in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/routing)

**Task 4. Custom Middleware**

Create own middleware component for image caching. Middleware should:

* Check Content-Type every response and if returned any valid image format:
  + Keep image on the disk (as file)
  + If next request accessing to the same image, get it from cache directory
* Support follow options:
  + Path for cache directory
  + Max count of cached images
  + Cache expiration time (if no requests during this time, cache cleaned)

Include the middleware into request pipeline of your application

**Materials:**

* [ASP.NET Core Middleware](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware)

**Task 5. Filters**

Add MVC filter for logging Action calls. Filter should:

* Logs Action start/end
* Use standard ASP.Net Core logging infrastructure
* Provide option, which on/off logging parameters of Action method (by default – it’s off)

**Materials:**

* [Filters in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/filters)