

## Plugin Framework

Maximum amount of time allowed: 2 hours

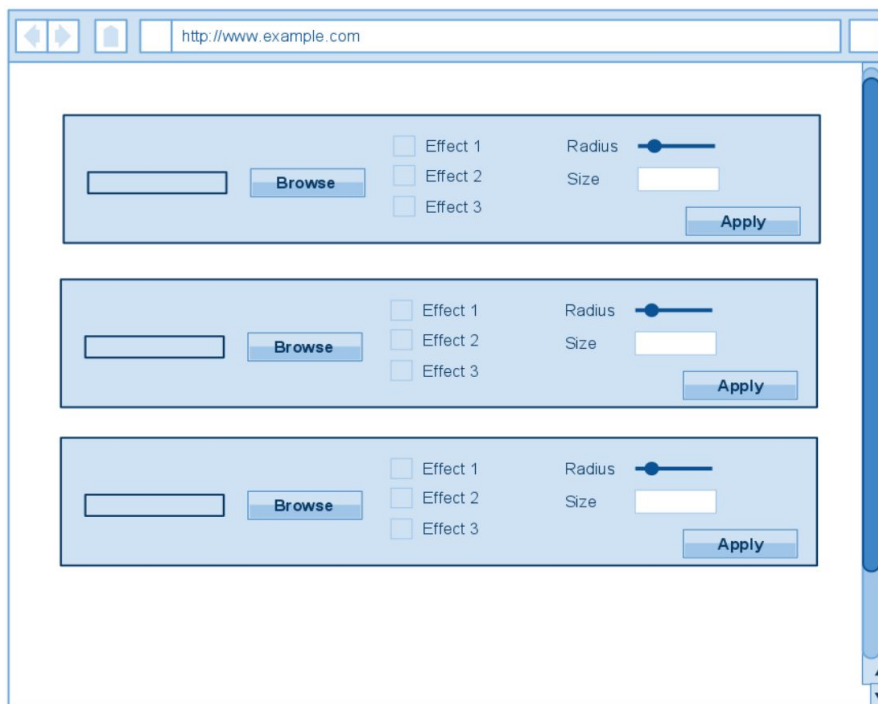
### Task

Imagine you work at a company that provides software on the web to manipulate images.

You are given the task of developing a .net framework to:

- Allow handling of multiple images at once, each one with its desired effects
- Allow an easy system of adding/removing plugins (by plugin we assume a specific image operation, like Adobe Photoshop effects and filters) without touching the code of the rest of the application. You can either use a config file containing only associative arrays, or rely on some other method of managing the plugins configuration.
- Allow multiple effects for each of the photos
- You are not supposed to implement any UI, another engineer will be doing that. Now, the APIs you provide will be used to build a UI like the one below, which is provided as a guidance.

Wireframe:



For example, we might want the following:

- Image#1: resize to 100 pixels, add blur 2 pixels size
- Image#2: resize to 100 pixels
- Image#3: resize to 150 pixels, add blur 5 pixels size, convert to grayscale

The API is to be an internal .net API so it will only communicate with other parts of a .net application, not server to server, nor server to client.

## Requirements

- Use object oriented
- Do not use any 3 party framework. Start all code from scratch.
- Coding actual plugins is not necessary. Focus on the plugin framework and a fake/ dummy one to give a sample of how they would be coded, but without doing actually anything related to image manipulation.
  - Ignore the presentation layer, the wireframe presented here are for better understanding of the problem. The only aspect important for your task is that an effect might have one parameter which can be of different kinds (a value slider/ selector, numeric input box,...), but it's optional.
- Supported image types, resolutions, etcetera are not important for this test.
- Task is meant to be an assessment of your personal skills, please do not ask for help.
- Simulate image data in order to test/show how the API works

What we look at this task is designed to give us an idea of how you think when faced with a very limited amount of time to solve a task of significant complexity. We are also interested in how you structure your code so that it's easily extendable, complies with best OO practices, and is easy to modify / understand by others. We are also interested in seeing how well you architecture a specific application.