Take Home Assignment: Senior C++ Developer - IMG.LY SDK

Assessment

⏱️ 3-4 hrs

At IMG.LY we're dealing with various types of resources and need to have a robust infrastructure to manage them at runtime. Estimated completion time is 3–4 hours where emphasis is on the problem-solving approach and code quality over achieving a perfect solution.

Task

1. Setup a minimal C++ project that provides a `ComputePipeline` implementation
2. A `ComputePipeline` is responsible for performing a series of transformations to achieve a final result
3. The initial action is always a `load` of an item from storage
   a. Which `load` action is required, depends on the given initial `uri`, which may be a file path starting with `file://`, a fully qualified URL (`http://` or `https://`) or a path pointing to the application bundle (`bundle://`)
4. The item must then be further processed by different actions, depending on its type:

    a. Images must be decoded
    b. Compressed data must be decompressed
    c. JSON must be turned into a C++ object

5. **You don't need to implement the actual action logic**
6. The result of each action is always an object holding the actions output and additional metadata, which also acts as the input to the next action
7. An item has finished processing, if no more actions are applicable
8. It's up to you how to decide, whether an action is able to process a previous actions output
9. Assume the results of each action to be expensive and make sure to reduce copies to a minimum
10. Think about your solutions performance and memory implications
11. Please take note of how much time this takes you to implement

Deliverables and Technical Requirements

- Please create a GitHub repository for your project and share it with us (either in public or invite:

francisco.martins@img.ly

thierry.seegers@img.ly

and please inform our recruiting Manager Michael about the delivery by email jobs@IMG.LY

- Ideally, use `CMake` to setup your project, but not required
- There is no need for any user or command line interface
- Standard Library, smart pointers and templates are welcome, but not mandatory
- Please don't use any third-party libraries
- **Focus on writing well-structured, understandable code**. Imagine you'd actually want to release your solution for the world to see.
- Last but not least: Add documentation to your code and overall project.