

Omedym Scenario-Based Skills Assessments

Hello! Thank you for your interest in working with us here at Omedym. As part of our screening process we often require engineering candidates to complete one (or sometimes more) scenario-based skills assessments.

What Is A Scenario-Based Skills Assessment?

The assessment is an evaluation exercise based on a real life scenario we've previously encountered in the course of building the Omedym Platform. It enables you, the candidate, to truly showcase your skills through solving a true-to-life problem.

The entire exercise and communication/interaction once you start is as much about surfacing *how* you approach and solve a problem (and uncertainty!) as much as producing the working solution to the problem.

Because we use the same one(s) with all candidates applying to a particular opportunity, we're able to compare and contrast a pool of candidates and how they work in a uniform fashion. It also helps us level set expectations and make better decisions as we weigh tradeoffs between hard and soft skills, and frankly, rationalize value.

How Much Time Will It Take?

An assessment is typically designed to take several hours in total for the level of expertise it targets. We realize this constitutes an investment on your part, but believe it is fair given we make long term commitments with our team members.

What's The Process?

You'll be given a "take home" package. In the package, along with a copy of these instructions, you'll find other artifacts:

- A 3-4 page scenario document that describes the background, a problem, and the solution you need to build. It will also contain technical and other guidance, even futures, that must influence your approach.
- Sample datasets, containing reference data or pseudo data structures related to the scenario. These are provided in various formats such as: tabular CSV/TSV; JavaScript objects; and/or line feed separated raw JSON objects.

You'll first need to closely review the scenario and sample dataset(s) that may be included, and ask as you see fit any questions you need in order for you to proceed.

Then, within the stated parameters of the scenario's solution approach and other guidance as stated, you'll build an implementation that fulfills the solution through working code that is: encapsulated, validatable, maintainable.

What Should You Watch For?

When it comes to a scenario always be mindful of the following:

- A scenario is often purposefully information dense -- How you consume, parse, and deconstruct it is extremely important.
- As with the real world, sometimes a scenario may provide conflicting explanations or parts may be ambiguous -- So don't hesitate to ask questions.
- But your attention to detail also matters, a lot. Avoid asking questions for which clear answers exist. For example, a typical red flag? Asking "What stack should I use?" when the scenario has a specific section, *Technical Guidelines & Concerns* that stipulates the *Languages & Frameworks* you can use.

- Sample datasets might only provide an approximation of the data, e.g. pseudo data, and may not cover every use case outlined in the scenario's problem to solve.
- Sample datasets might also include artifacts that provide a deeper or more nuanced understanding of the overall data and models involved. But these may, in fact, not be needed to solve the problem.

STOP! BEFORE YOU BEGIN...

Read the following instructions *carefully*!

How To Take The Assessment?

1. Review the scenario, sample data, and guidance as outlined.
2. Ask any questions you have along the way, but especially within the first hour.
3. When you're ready to begin, create a new, *local* Git repository on your machine and make a first commit indicating you have started. *Do not create this repository on GitHub.*
4. As you work, structure and layer your commits. [^atomic-commits]
5. When finished, submit your assessment project repository *exactly* per the instructions below.

What Should The Results Include?

The submitted project itself will need to contain:

1. The working code that expresses the solution approach to the scenario's problem and within the confines of its guidelines.
2. Executable tests that exercise the code and validate that it solves the problem correctly.
3. The project's Git repository subfolder (.git) with its history.
4. Optionally: A Markdown based "readme" with any special instructions, commentary, feedback, etc. you believe should be shared with us.

How To Submit Your Results?

Upon completion of the assessment:

1. Create a compressed archive (RAR or ZIP) of the assessment project repository that contains your work.

You *must* make sure what you send us includes the `.git` subfolder so we can review your commit history.

2. Name the archive file using your last name and first name as `last-name-YYYYMMDD.{rar|zip}`. For example, if your name is John Doe, `doe-john-20210715.zip`.
3. Send us the archive. How and where you send will depend on how we've already been introduced. If you're unsure, ask!

[^atomic-commits]: *Why I Create Atomic Commits*: <https://curiousprogrammer.io/blog/why-i-create-atomic-commits-in-git>