Backend Coding Challenge

Specification

- For this project, you will develop a simple web application in Go which accepts math problems via the URL and returns the response in JSON
- The application should be simple to test using `curl` or `wget`
- Example:
  http://localhost/add?x=2&y=5
  Output:
  `{“action”: “add”, “x”: 2, “y”: 5, “answer”, 7, “cached”: false}`
- Implement **add**, **subtract**, **multiply**, and **divide** operations
- Only two arguments will ever be passed: **x** and **y** (no need to handle a variable number of arguments)
- Cache results so that repeated calls with the same problem will return the answer from the cache
    - Show in the output JSON whether the cache was used
    - Expire anything in the cache that has not been hit for one minute

Additional notes and requirements

- The code should be delivered via Github (if you choose to create a private repo, add proper access for the reviewer)
- Use your judgment in handling anything not explicitly defined in the requirements
    - The result should be simple but robust
    - We will be looking for some things which are not specifically mentioned in order to gauge your experience and knowledge of best practices
- Mistakes due to an obvious lack of familiarity with Go will not be judged harshly
- Describe how would you deploy your application (in the cloud) based on your experience
- Readability is key. When in doubt, choose clarity over cleverness
- Assume that we will try to break your app

Final notes

- Projects without a README.md file will be ignored
- After writing and testing your code, take a break and code-review it hours later, or the next day. Better to proofread before the editor gets his hands on it.