# URL Shortener

## Preface

Thank you for your interest in Hypefast!
Before proceeding to the technical interview, we'd like you to complete a take-home exercise first. Should you proceed, it would also serves a base work to discuss on the technical interview.

### Scoring

We understand that doing take-home exercise takes time, hence we recommend to spend **3 hours at max** for this exercise.

To give you a sense of what's important, here are the evaluation criterias we use, sorted by the most important first:

1. Correctness (fulfilling the requirements)
2. Code readability
3. System design
4. Code testability

### Contact

Should you need to clarify any requirements, you can send an email to sharon.raissa+rec@hypefast.id. We'll respond within 1 working day.

# Requirements

## Objective

Your task is to create an API-based URL shortener service that provides functionality such as Tiny URL

## Functional Requirements

1. Given a long URL, the service should generate a shorter and **unique** alias of it with **exactly 6 alphanumeric** characters and return it
2. Given a short link, the service should be able to redirect the request to the original link.
3. Given a short link, the service should be able to return these stats:
    - Redirect count
    - Created at

## Non-Functional Requirements

- The solution must be submitted as a git repository.
- Since we're using Golang as our main BE language, it's recommended to use it for this exercise. In case you're not familiar with it, you may use any language you're most comfortable with.
- The service must expose **API endpoints** that satisfy the above requirements. You don't need to implement the view.
- Please provide a simple API documentation in the README (plain text is fine, no need to use fancy formats) for all endpoints you created containing these information:
    - URL
    - Request
    - Response
- You may store the data in-memory instead of database to ease implementation. Extra point if you can make an interface such that it's easy to swap the implementation.

## Nice to Haves

- Unit tests
- Brief documentation explaining your design decisions
- Meaningful git commit history