# ING Coding Challenge

This is the ING Coding Challenge for the assignment NASA.

The Coding Challenge is part of the ING interview process.

Time limit

For regular developers takes approximate 2 hours.  
For fullstack developers takes approximate 3 hours.  
Setting up your environment and reading documentation must not be taken into account.

Setup

First of all, you'll need an API-key from NASA. You'll need to request by using this page.  
Get familiar with the NASA API by trying out the resources here using i.e. Postman.  
More information on [api.nasa.gov](http://api.nasa.gov/)

Assignment

The assignment consists of two parts, a web user interface and a back-end application supporting this web user interface.

**Web user interface**

Please create a simple web user interface which could search and show Mars Rover photos. You'll need to read the following documentation by selecting Browse APIs on [api.nasa.gov](http://api.nasa.gov/) and choosing Mars Rover Photos.

**Backend API**

A backend API must support the web user interface. Based on the search criteria's, the backend will make the calls to NASA. The results must be parsed from JSON.  
Each API call must be logged into a database. An in-memory database should be sufficient.  
Make sure that this will not affect the application to stop working. Criteria for storing the API call request:

* Generated ID
* Requested method name
* Response time in milliseconds
* Date created

Configuration

The application must be configurable, this means that there should be using some kind of configuration file.

Testing

It is not possible to cover the application entirely with tests, but at least cover the most important part of the application like services or controllers.

Bonus points

* Make the application multilingual (EN, NL or any other language)
* Add more?

Building the API

In order to build the API a simple gradle build will do.  
Running the API locally

gradle bootRun

If port 8080 is already bound you can change te port number. Go to the file application.yml and update the following setting:

server:  
    port: 8090

The application is available with the URL: [http://localhost:8080](http://localhost:8080/)

Objectives for ING

Check on the following aspects:

* Dependency injection
* Clean coding
* Usage of enumerations
* REST solution
* Catching exceptions
* Usage of collections
* Externalize property configuration