

Coding Challenge

Hello! In this stage of the interview process, we would like to see what kind of code you write.

Don't worry there won't be any trick questions, it is more grounded in the real-world and serves as a small example of a programming task you might encounter at Expected Behavior. While this isn't a full product by any means, this is an opportunity to show off what you think great software development of a production application looks like.

Problem Description

We're looking for a small [Ruby on Rails](#) application providing an endpoint (`/pdf_metadata`) that takes a list of URLs and returns some information based on using the API of one of our products, [DocRaptor.com](#).

For each URL submitted to this new webapp it should use DocRaptor to generate a PDF and then return some metadata for each PDF based on the [pdf-reader gem](#). The results should be grouped by the number of pages in each PDF, and within each group sorted alphabetically by URL. DocRaptor provides a public key for testing (`YOUR_API_KEY_HERE`) that you can use so you don't need to sign up for a paid account.

The app should be deployed to Heroku (or your preferred host) and accessible publicly.

For example, when this URL is requested from the webapp:

```
/pdf_metadata?urls[]=http://docraptor.com/examples/invoice.html
```

The following JSON would be returned:

```
{
  "1": [
    {
      "url": "http://docraptor.com/examples/invoice.html",
      "pdf_version": 1.5,
      "info": {
        "Producer": "Prince 12.2 (www.princexml.com)",
        "Title": "Your New Project for Our Best Client"
      },
      "metadata": null,
    }
  ]
}
```

```
    "page_count": 1
  }
]
```

Submitting the Application

Push your code to a public repo on GitHub (or your preferred code hosting provider) and send us a link to the repo as well as a link to the publicly accessible webapp.

Conclusion

We value your personal time and tried to frame this exercise to be solvable in around 4 hours or less. In addition, we will pay you \$200 USD for a submission that includes the repo and publicly accessible web application links.

We're looking for tested and documented code. If possible, please send us your solution back within 2 weeks, but if you need more time please reach out to us and keep us in the loop.

And feel free to ask us in case you have questions about the exercise.