

Nuri - Coding Challenge

Instructions:

Thanks for applying at Nuri. The purpose of this coding challenge is for us to see your personality as a coder and give you the opportunity to show off your skills!

Here are some quick, easy-to-follow instructions for this coding challenge:

- **We recommend that you read through all of the instructions in this document before starting :) You should have been given write access to a private Github repository. If you haven't - please let us know and we'll give you access.**
- This coding challenge should take around 2 - 3 hours to complete and we would like you to complete it within two days (let us know if you need more time).
- Include a readme file in your solution explaining how to run your solution and any comments on your design. You can also include any suggestions for how you might improve or expand on your program - and even how you might host it.
- As part of this coding challenge we ideally want you to create a back-end and front-end component. However, we know that many people are stronger in one area than the other and so if you want to focus on doing an awesome job on one component - then that's ok. If you were unable to complete, or finish, a component then that's also ok. Either way, please include an explanation of this in your readme file.
- You can use whatever languages, technologies and frameworks you like - ideally we'd like you to keep these as close to the tech stack that we're using here at Nuri (take a look at our page here: <https://stackshare.io/nuri-gmbh/bitwala-gmbh>).
- Develop your solution in a way that represents a "real world" solution. Think about how you might test your solution, carry out logging, host your application, document your application, scale your application, improve your data access layer, improve the design of the UI etc. - you don't necessarily have to implement all of this stuff, but you could talk about it in your readme file.
- The structure of the application is up to you. For example, you can put all front-end and API code into the same application or you can separate the API and front-end as different applications in the same repo - we'll let you decide on the best approach.
- Once you've finished - don't forget to commit your code and send us an email letting us know that you've finished the coding challenge.
- **You don't need to know about cryptocurrency or blockchain to complete this coding challenge :)**

The Task:

Create a web-based client for <http://blockchain.info> that allows users to list the latest blocks and details of each block. A block is just a data structure which groups transactions. For example:

LATEST BLOCKS

[SEE MORE →](#)

Height	Age	Transactions	Total Sent	Relayed By	Size (kB)	Weight (kWU)
534978	13 minutes	748	1,350.84 BTC	ViaBTC	1,234.19	3,992.96
534977	14 minutes	2162	9,034.03 BTC	Unknown	1,205.36	3,992.61
534976	30 minutes	2361	30,406.68 BTC	SlushPool	1,093.97	3,993.08
534975	31 minutes	2576	8,322.15 BTC	BTC.com	1,384.62	3,992.84

The UI (frontend) component:

Implement a frontend (e.g. React) that contains the following functionality:

- The first view (i.e. the landing screen) should display a list of blocks and the following details for each block:
 - ◆ Block hash
 - ◆ Block time
 - ◆ Block height
 - ◆ If you get some time you might want to implement pagination on the results.
- A user should be able to click on each block and view the following details for the block:
 - ◆ Size
 - ◆ Block index
 - ◆ Previous hash
 - ◆ If you get some time it would be great if you could also display the list of transactions for each block - you can decide on what details should be displayed.

If you're not an artist and UI/UX is not your strength - that's fine. We don't mind if your UI uses unstyled HTML. But if you're going to focus more on the back-end component, we'll expect something magical in your API work.

The API component:

To make your UI work you'll need to provide an API to fetch the latest blocks and block details from blockchain.info. You can use the following boilerplate repo to quickly and easily create this API component:

