

Rival Frontend Web Developer Coding Exercise

Thank you for taking the time to do this exercise! We're excited to get a better idea of where you might fit at Rival, and hopefully you'll have some fun showing off your skills in the process. Our goal is to get a sense of your experience, style, and personality in code, rather than a completely perfect solution.

You have 3 hours for the exercise, but most people do not finish in that time, so please don't feel stressed if you're not getting through it all. We are far more interested in the approach you take and the quality of your code than we are in your beating the clock.

If after 3 hours you wish you had more time, we'd appreciate it if you'd spend an extra 15 minutes writing an overview of what your next steps would have been and include that in the email pointing us to your challenge (see delivery instructions below).

Instructions

Build a web UI for the fictitious Academy of Mars Research (AMR), showcasing the photos taken over the years by the various Mars rovers.

Accompanying this README, you will find folders for **mocks**, **specs**, and **images**. AMR has signed off on them, and they are expecting your submission to match them.

Sadly, AMR cannot provide the data and images to bring these designs to life as a fully functioning single-page app. Instead, you will need to interact with the [NASA API for the Mars Rover Photos](#).

Be sure you understand the data model, as outlined at the API link above. As a quick example, the following URL snippet shows the relevant pieces needed to construct the URL that will return Curiosity's images from June 3, 2015.

`.../curiosity/photos?earth_date=2015-06-03&api_key=pvU4y4gG0TZtzvjuLLTVXg86sxJazkQtg1NkFXzt`

You may use `pvU4y4gG0TZtzvjuLLTVXg86sxJazkQtg1NkFXzt` as the API key, or you may [sign up for your own key](#).

Implementation Details

- The user should be able to select from among the three rovers.
- The user should be able to filter images by date, defaulting to today's (earth) date.
- As the user updates the rover and/or date, the gallery should be updated with the feed of photos based on the selections.
- Include the timestamp of the photo in a human-readable format, as well as other metadata if available.
- Given the time constraint of 3 hours, implementation should work in the current version of Chrome and Firefox. Getting it working in current versions of Edge and/or Safari is considered a bonus.
- Implementation should accommodate users on mobile browsers.

Preferred Architecture, Frameworks, etc.

- Please develop using a modern API-based single-page web app architecture with modular components.
- Directly manipulating the DOM to display interactive elements is discouraged.
- Given the time allotted, the project will be small and likely incomplete; but it should be developed using UI patterns and data models designed to scale. For example, we would rather see your responsive design patterns rather than a pixel perfect implementation at only one screen size.
- JS frameworks like React, Angular, Vue, and the like are good starting places.
- If you prefer not to use third-party frames, a framework-like architecture using WebComponents is encouraged.

- For reference, Rival is currently built using React Native (and runs in a browser via React Native Web). This will not bias our assessments of your tooling choices.
- Frameworks like Bootstrap, Foundation, Ant, Semantic UI, Material UI, and similar are useful to quickly create a functional UI without spending time on basic styling boilerplate.

Time Frame and Delivery

Please confine your work to the allotted 3 hours (+15 minutes if needed). While seeing what can be developed during a limited time frame is useful, we do not want to take up too much of your time.

Once you've finished, please zip the project, put it on a file-hosting service (like [DropBox](#), [Box](#), [Google Drive](#), etc.), and email us a link that gives us read access to your zipped submission. If after 3 hours you wish you had more time, we'd appreciate it if you'd spend an extra 15 minutes writing an overview of what your next steps would have been and include that in the email pointing us to your challenge.

Please do not share or publish your solution beyond submitting it to us.

Thank you, have fun, and good luck!



Figure 1: The UX you'll build.