Challenge

Backend Engineering

At Dalia, we look beyond CVs and try to understand early in the process how you think, work and tackle concrete problems that are relevant for this role. Please answer the questions below to the best of your ability but keep in mind that there are no perfect answers.

For you, the challenge is a great opportunity to demonstrate your capabilities, explain your thought process and share your toolset, while at the same time getting a first glimpse of what your day-to-day work could look like at Dalia.

The multi website agenda

Imagine that Dalia is about to start a spin-off project that collates all the cultural activities in Berlin into one site and you are the designated engineer for this new product.

The specifications**:**

You will need to collect the information for your website from other web sources**.** There are plenty of resources you can draw, for example:

- https://www.co-berlin.org/en/calender
- https://www.deutscheoperberlin.de/en_EN/calendar
- https://gorki.de/en/programme/2018/08/all
- http://berghain.de/events/

The challenge here is that the information you need from these resources is in different formats and therefore difficult to analyse. We need to standardize it in a unified format that can be easily digested by our engine.

Once all the information has been standardized we want to make it easily accessible to website visitors via a centralized calendar.

The requirements for this Application are:

- Events filter: User can filter the events based on different criteria:
  - Web Source
  - Dates
  - Simple partial text search on title (no need to implement complex text, search SQL "`like`" or similar is enough)

- The events are updated frequently so visitors can always access fresh information.

The implementation

Based on these requirements we need to have a working prototype.

Your deliverables for the challenge are:

1. Implement the code needed to parse 2 of the "web sources" (of your choice) into the standardized format.
2. Implement the code needed to collect the standardized format and render it in a website. Ad some simple filtering mechanism (based on backend filtering, not frontend JS filtering)
3. Prepare your code for easy deployment