



Front-end Engineer

Exercício de conhecimentos técnicos

Para que se realize o exercício proposto existem várias formas de chegar ao resultado, sem que necessariamente uma linha esteja definindo o certo e o errado. Existem sim, várias maneiras corretas de aplicar seus conhecimentos técnicos e de fazer uso das boas práticas de programação.

Como um(a) bom(a) desenvolvedor(a), você sabe que impressionar, prezar pela qualidade e surpreender são ações fundamentais em nosso meio. Portanto, além de propormos um exercício de lógica de programação, pressupomos que você fará um exercício de criatividade e boas práticas!

Vamos começar?

Introdução

Esse documento descreve o desafio técnico da Meritt para a vaga de Front-end Engineer.

Leia atentamente antes de iniciar seu trabalho e desenvolva a solução solicitada seguindo as melhores práticas de engenharia de software e desenvolvimento que você conhece.

Avaliação

Analisaremos todos os aspectos do código por você produzido para verificarmos se os seus conhecimentos e se a forma que você coloca eles em prática condizem com as nossas expectativas, portanto dê o seu melhor.

Nossa avaliação consiste em atribuição de pontos de acordo com os critérios de qualidade de software e design de código. Nossa intenção é entender melhor o seu nível de conhecimento nos aspectos que julgamos importantes e que são requisitos da vaga que você se candidatou.

Tempo

No recebimento deste briefing, estime o tempo que você levará para resolver o desafio. Você será avaliado pela capacidade de estimar o tempo necessário para realizar a tarefa em relação ao tempo efetivamente utilizado.

As instruções sobre o envio estão no e-mail que você recebeu. Em até 2 horas confirme o recebimento deste arquivo informando qual o seu prazo estimado para entregar a atividade completa.

Boa sorte!

Requisitos

Sua tarefa consiste na implementação da interface de acompanhamento de diversas métricas de performance de servidores. Nos anexos há mais detalhes sobre os dados e o wireframe, com estilo visual aplicado.

Requisitos funcionais

- A interface deve ser responsiva (mobile, tablet, desktop).
- Você deve usar o seguinte pré-processador de CSS: LESS (<http://lesscss.org/>).
- Você deve usar HTML5. Quanto mais válido (<http://validator.w3.org/>) melhor.
- A interface não deve usar nenhuma imagem estática (jpg, gif, png, etc). Você tem liberdade de abrir mão de detalhes no design para não comprometer o prazo e focar nas funcionalidades.
- Você precisa garantir a compatibilidade com o navegador Google Chrome, em sua versão mais atual. Compatibilidade com outros navegadores é opcional.
- Todo o código e documentação gerada deve ser entregue em Inglês.
- Você deve entregar as versões /dist/ e /dev/ do desafio \o/

Observação: ao final do documento estão os requisitos de cada componente da tela.

Informações extras

- Você pode usar bibliotecas 100% JavaScript para ajudar a resolver o desafio: somos práticos e gostamos de código organizado, preferencialmente em uma arquitetura MV*
- Você pode também desenvolver testes, aqui eles são muito bem vindos (ah, temos um carinho especial por JasmineJS)
- Você deve gerar a documentação que achar necessária para que outros desenvolvedores entendam o código

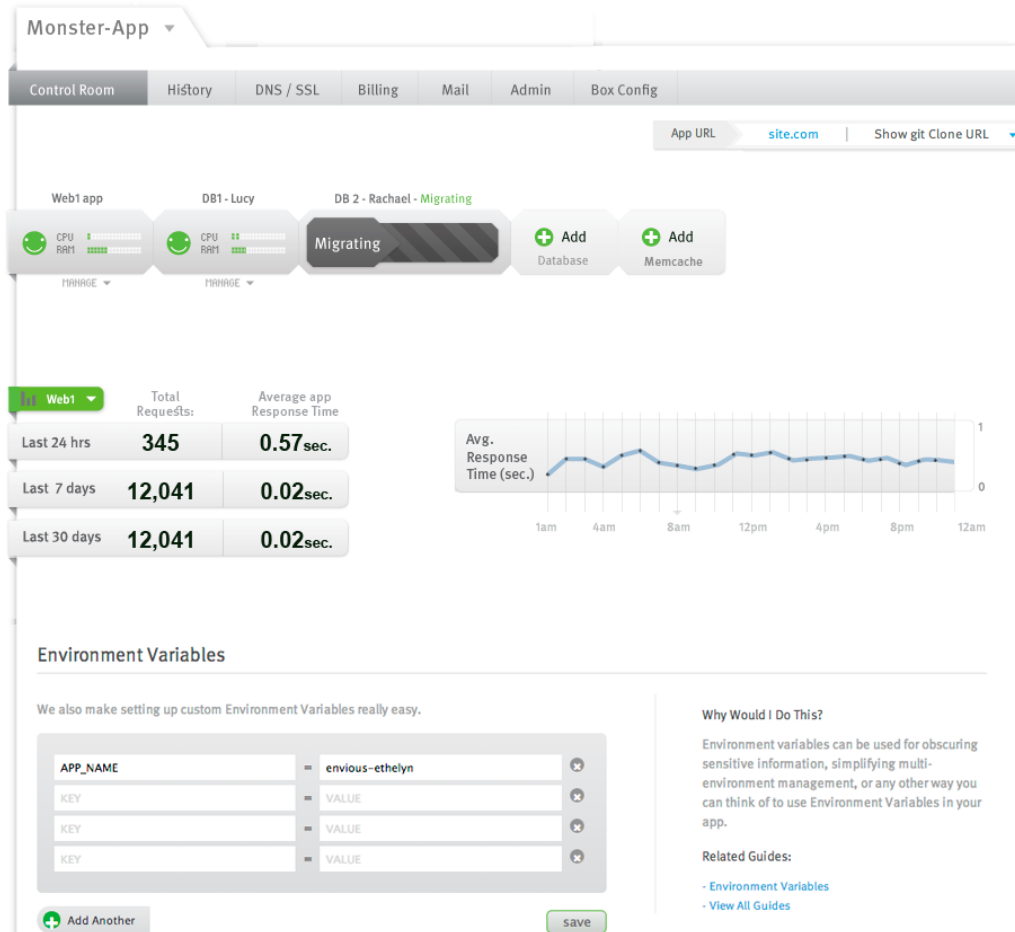
Entregáveis

Você deve nos entregar o resultado final em um repositório git dentro do github. O repositório deve conter:

- O código fonte completo do desafio;
- Histórico de mensagens de commit, em inglês. Olharemos o histórico para ver como você organiza seu trabalho e produção;
- Um arquivo README.md formatado em markdown explicando como devem ser instaladas as dependências e também a forma de execução do seu projeto.

Interface do desafio

(Imagens em tamanho real: <http://goo.gl/zNhNw3>)

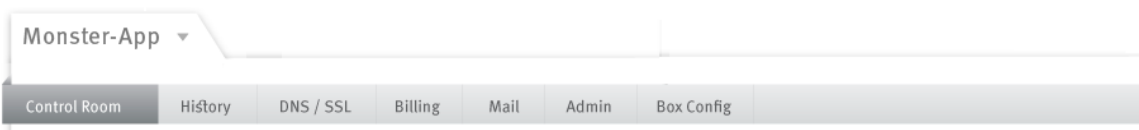


Componentes da interface:

Menu Principal

- Ao clicar em Monster-App, deve abrir um dropdown listando outras aplicações.
- A lista de aplicações pode ser obtida por um Request.
- Você deve mostrar apenas aplicações ativas, filtrando pelo atributo isActive.

<http://demo2177885.mockable.io/apps>



Barra de URL

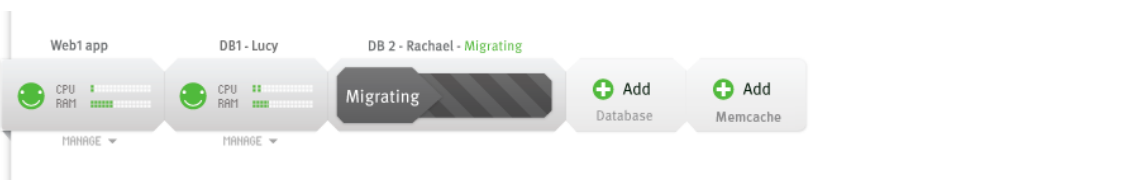
- Ao clicar em Show git Clone URL, deverá ser feito um Request para a URL abaixo, que retornará um erro (HTTP 500).
- O usuário deve ser informado que um erro ocorreu

<http://demo2177885.mockable.io/error>



Barra de serviços

- Os indicadores de CPU e RAM deverão ser atualizados constantemente (a cada 5 segundos).
- Os indicadores deverão ser implementados usando CSS e HTML.
- O barra de Migrating poderá ser animada para dar impressão de "em processamento".
- Os Requests para obter os dados podem ser obtidos nos endereços:
Web1 app: <http://demo2177885.mockable.io/web1app>
DB1 - Lucy: <http://demo2177885.mockable.io/db1lucy>



Barra de status detalhada

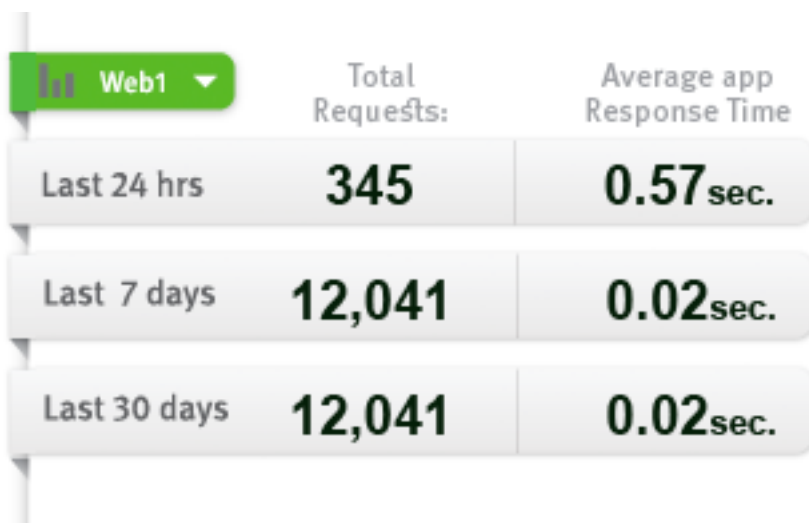
- Os dados devem ser atualizados constantemente (a cada 5 segundos).
- Serão feitos quatro Requests. Quando **os quatro Requests estiverem completos**, você deve agregar os resultados por três espaços de tempo:
 - Últimas 24 horas
 - Últimos 7 dias
 - Últimos 30 dias
- Considere a hora atual como: **2014-07-25T01:00:00+03:00** (defina a data com o dia da realização do desafio).
- O total de requests e o tempo médio de resposta precisam ser calculados, para cada um dos espaços de tempo.
- Os requests devem ser feito pelas seguintes URLs:

<http://demo2177885.mockable.io/homer>

<http://demo2177885.mockable.io/jenkins>

<http://demo2177885.mockable.io/mariadb>

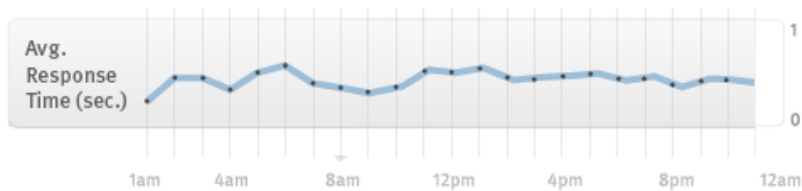
<http://demo2177885.mockable.io/marmitt>



Web1 ▾	Total Requests:	Average app Response Time
Last 24 hrs	345	0.57sec.
Last 7 days	12,041	0.02sec.
Last 30 days	12,041	0.02sec.

Gráfico

- O gráfico deve ser gerado estaticamente – no lado do cliente (navegador).
- Escolha a biblioteca que mais lhe agradar.




Lista de variáveis

- As variáveis devem ser salvas usando localStorage.
- Implemente a interface de forma a tornar o seu uso o mais fácil possível para o usuário. Por exemplo, ao digitar ENTER, o foco pode pular para outro campo.
- Ao deixar um campo em branco, o usuário pode ser avisado com um efeito highlight no campo. Você pode implementar salvamento automático também.

Environment Variables

We also make setting up custom Environment Variables really easy.

APP_NAME	=	envious-ethelyn	✕
KEY	=	VALUE	✕
KEY	=	VALUE	✕
KEY	=	VALUE	✕

 Add Another

save