

# spume::challenge for Padawans

Seguem instruções sobre o uso das ferramentas deste desafio e os resultados esperados.

## Overview

Se trata de uma página simples com três *botões* que disparam requisições via `fetch API` para um servidor `nodejs` pré-configurado na solução, estas requisições retornam uma coleção de artigos diversos encontrados no repositório do desafio.

Encontra-se implementada a requisição que trás todos os artigos (limitados a 50 por motivos de desempenho), e cabe ao desafiado implementar as demais rotinas, sendo a primeira obrigatória para a conclusão deste desafio e a segunda um bônus que deve contar bastante!

## DevOps

Este desafio foi construído com uma *docker nodejs* pré-configurada.

Para iniciar a *docker*, considerando que o ambiente esteja instalado devidamente no seu sistema, basta executar o *script* `./docker/start.sh` . De maneira análoga, para encerrar a *docker* há um script de nome `stop.sh` .

Ao iniciar o *container*, você deverá ser direcionado para o sistema emulado, uma vez dentro do servidor, basta correr o comando `/root/proj/run` e uma instância `nodemon` será executada com o projeto carregado.

### notas:

1. Se o sistema não te direcionar para dentro do contâiner, use o código: `docker exec -it nodejs zsh` para acessá-lo.
2. com o `nodemon` funcionando, cada vez que algum arquivo da aplicação `node` for modificada, basta digitar `rs` e teclar *Enter* para que o servidor seja reiniciado com as novas modificações.
3. Para acessar a aplicação, basta digitar `http://localhost:3000` em qualquer navegador de internet (chrome, edge, opera, etc.).

## Backend

### Pastas de projeto e arquivo de inicio

	Projeto	Servidor
Pasta raiz	<code>./project</code>	<code>/root/proj</code>
Aplicação	<code>./project/app</code>	<code>/root/proj/app</code>
Node start file	<code>./project/app/app.js</code>	<code>/root/proj/app/app.js</code>

Estrutura `mvc` de pastas (do desafio apenas!)

```

project/
\___ docker/           -- docker config files/scripts
\___ app/
  \___ assets/
  |   \___ rss/         -- project`s RSS RAW files
  \___ lib/             -- spume`s framework
  \___ webroot/
    \___ controllers/   -- nodejs specific files
    \___ views/         -- web files, node independent

```

## Arquivos principais do desafio (backend)

```

project/
\___ app/
  \___ assets/
  |   \___ rss/*         -- project`s RSS RAW files
  \___ webroot/
    \___ controllers/
      \___ articles.js   -- where the magic will happen!
      \___ rss.js        -- just in case you need to surprise us (optional)
      \___ themes.js     -- try yourself on customizations! (optional)

```

## Frontend

### Pastas de projeto e arquivo de renderização

	Projeto
Pasta raiz	./project/app

## Arquivos principais do desafio (frontend)

```

project/
\___ app/
  \___ webroot/
    \___ views/
      |   \___ components/
      |   \___ article_tile.htm -- the tile in wich articles are shown (optional customizat
      |   \___ css/
      |   \___ style.css       -- help you write your style (optional)
      |   \___ js/
      |   \___ preset.js       -- boot scheduler (wrong choises messes the system)
      |   \___ home.htm        -- main app container
      |   \___ splash.htm      -- feel free to customize the boot screen (optional)
      \___ index.html

```

## O desafio

Crie código no arquivo `./project/app/webroot/controllers/article.js` , dentro das funções membros: `only_10_days` e `only_techs` (opcional), para customizar o retorno das chamadas que vêm da página `./project/app/webroot/views/home.htm` .

Já há nesta classe/módulo uma `action` implementada com assinatura

`array Articles::all(int limit)` que retorna todos (limitados a 50 por razões de performance) os artigos encontrados pela classe `Rss` (do documento: `./project/app/webroot/controllers/rss.js` )

### **only\_10\_days**

Ao contrário da chamada para `Articles::all` , a função membro `only_10_days` deve retornar **apenas**, única e somente os artigos encontrados com data de publicação dentro dos últimos 10 dias. Atenção para a implementação do *limite* de retorno.

### **only\_techs (opcional, porém bônus)**

Você deve criar um critério mínimo para decidir se o artigo tem conteúdo de tecnologia, aplique seu critério e retorne apenas os artigos que você encontrou. Não se preocupe em perder artigos, o importante é que os que vierem sejam de fato de tecnologia.

## **Avaliação**

Você será avaliado pelo desafio `only_10_days`, este precisa estar concluído.

## **Bônus**

O desafio `only_techs` é totalmente opcional, mas traduz bastante os desafios que você enfrentará na Spume, e por isso te dará pontos extras!

## **Opcionais**

Apenas servirão para que possamos ser surpreendidos!

São eles:

1. Alterações na classe `Rss` : pode te dar pontos se úteis.
2. Lidar com 'possíveis' erros nos arquivos `rss` (inclusive `charset`): pode ser bom!
3. Alterações visuais no `*/article_tile.htm`
4. Alterações na página principal ( `*/home.htm` )
5. Alterações na página de carregamento ( `*/splash.htm` )

# **Que a força esteja com você!**