

Prueba técnica - Backend JavaScript

Versión 1.0

El objetivo de esta prueba es conocer tus habilidades como desarrollador backend JavaScript con un reto que refleja el tipo de trabajo que tú podrías hacer en La Manicurista.

Mecanografiar o autocompletar es una característica común que la gente encuentra en los sitios web. Por ejemplo, cuando esté buscando en Google, notará que una palabra se completa antes de que termine de escribir.

Para esta tarea, implementará el backend de un sistema de escritura anticipada simple. Se le proporcionará una lista de nombres y su popularidad.

Se le pedirá que redacte un sistema de backend con anticipación utilizando prefix-tree ordenado por popularidad.

Instrucciones

Debes escribir un sistema de backend con anticipación utilizando **prefix-tree** ordenado por popularidad.

Por ejemplo, si los datos proporcionados son: `{"John": 21, "James": 43, "Joanna": 13, "Ja": 3}` y un usuario consulta "j", entonces devuelve: `[{"name": "James ", "times": 43}, {"name ": " John ", " times ": 21}, {"name ": " Joanna ", " times": 13}, {"name ":" Ja ", "times": 3}]`. Tenga en cuenta que la búsqueda no distingue entre mayúsculas y minúsculas.

Además, una coincidencia exacta tendría prioridad a pesar de la popularidad. Por ejemplo, una palabra clave de búsqueda "ja" devolvería `[{"name": "Ja", "times": 3}, {"name": "James", "times": 43}]` ya que "ja" tiene una coincidencia exacta.

Requisitos específicos

Cuando ejecute su aplicación, tendrá dos variables de entorno disponibles:

- `PORT`: su aplicación debe recibir solicitudes usando este puerto
- `SUGGESTION_NUMBER`: esta será la cantidad máxima de resultados para el endpoint `GET`

Todos los endpoints de esta **API REST** devolverán y recibirán JSON (cuando corresponda).

No es un requisito tener persistencia en los reinicios de la aplicación, pero mientras se ejecuta, la aplicación debe comportarse como si tuviera persistencia, cuando se le solicita modificar datos (es decir, si envía una solicitud para modificar un nombre y luego consulta ese nombre, la aplicación debe devolver el nombre modificado).

Los datos iniciales están en el archivo `names.json` junto a esta prueba, en el formato {
`[nombre1]: [popularidadOfNombre1], [nombre2]: [popularidadOfNombre2],`
`...}`.

Es extremadamente importante que no cambie este archivo, ya que las pruebas automatizadas asumirán este archivo exacto como los datos iniciales para su aplicación.

Deberá escribir dos endpoints de esta API REST para aprobar esta tarea:

GET /typeahead/{prefix}

Recibe un **prefijo** en la ruta y devuelve un array de objetos, cada uno de los cuales tiene los atributos `name` y `times`. El resultado contiene todos los nombres que tienen el `prefijo` dado, hasta un máximo de nombres correspondiente al límite `SUGGESTION_NUMBER`, ordenados por mayor popularidad (`times`), y el `name` en orden ascendente si tienen la misma popularidad, siempre dejando la coincidencia exacta en el principio, si lo hay (ver ejemplos en la descripción general).

Si la parte `prefix` de la solicitud no se proporciona o está vacía, devuelve los nombres, correspondiente al límite `SUGGESTION_NUMBER` con mayor popularidad (y ordenados por nombre ascendente en caso de igual popularidad).

Este punto final debe distinguir entre mayúsculas y minúsculas, según la descripción inicial.

Ejemplo

```
$ curl -X GET http://host:port/typeahead/ja
[{"name": "Janetta", "times": 973}, {"name": "Janel", "times": 955}, {"name": "Jazmin", "times": 951}, {"name": "Janette", "times": 947}, {"name": "Janet", "times": 936}, {"name": "Janeva", "times": 929}, {"name": "Janella", "times": 916}, {"name": "Janeczka", "times": 915}, {"name": "Jaquelin", "times": 889}, {"name": "Janaya", "times": 878}]
```

POST /typeahead

Recibe un objeto JSON con un nombre como cuerpo de la solicitud (ejemplo: {`"name": "Joanna"`}), aumenta la popularidad de ese nombre en 1 y devuelve un objeto con las propiedades `name` y `times` considerando el nuevo estado.

Si el nombre de pila no existe en los datos (`names.json`), entonces este extremo debería devolver un error HTTP 400 (por lo que no se agregarán nuevos nombres, solo aumentará la popularidad de los nombres existentes).

Este endpoint debe distinguir entre mayúsculas y minúsculas, según la descripción inicial.

```
$ curl -X POST -H "Content-Type: application/json" -d '{"name":  
"Joanna"}' http://localhost:8080/typeahead  
{ "name": "Joanna", "times": 441 }
```

Envío de tu tarea

Por favor no subas tu código en un repositorio público, en su lugar usa un repositorio privado, comparte un archivo zip en Google Drive o envía el zip adjunto por correo.

Evaluación

Por la explicación anterior, sabe que hay pruebas automatizadas y una revisión humana una vez que pasan esas pruebas automatizadas. El criterio de evaluación para las pruebas automatizadas es simple: sea verde. Las pruebas cubren una gran cantidad de casos, por lo que ser ecológico garantiza la corrección de su solución.

En cuanto a la revisión humana, el resultado real no solo dependerá de lo que envió, sino también del nivel de antigüedad que esperamos de usted (tomado de su solicitud para el trabajo). Pero en general, la revisión humana se centra en:

- **buen código:** ¿qué significa esto exactamente? queremos saber lo que significa para usted mirando el código que escribe
- **código de rendimiento:** no esperamos la mejor solución de rendimiento posible, pero esperamos que tenga en cuenta el rendimiento. Puedes decirlo por el código que escribes.
- **buen solución:** aunque se trata de una tarea ficticia y bastante corta, esperamos que la implemente como lo haría en un trabajo real; no se concentre en mostrar habilidades o conocimientos específicos, concéntrese en mostrar lo gran ingeniero que puede ser para una empresa real.

Consideraciones

1. Tú tienes máximo **4 días** para completar esta prueba, iniciando desde la fecha en la que recibiste este documento en tu correo electrónico.
2. Incluye todas las instrucciones claras y detalladas de cualquier paso requerido para ejecutar la aplicación.
3. Las buenas prácticas y estilo de código serán evaluado.
4. Por favor no subas tu código en un repositorio público, en su lugar usa un repositorio privado, comparte un archivo zip en Google Drive o envía el zip adjunto por correo.
5. Cualquier cosa no especificada arriba será de tu consideración.

♥ Disfrutamos lo que hacemos, esperamos que tú también, ¡Que te diviertas!