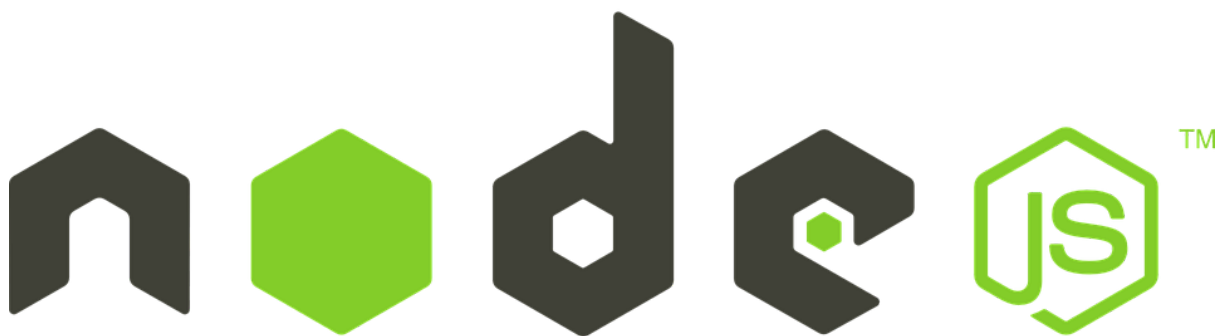




CodigoDelSur

Evaluación técnica Nodejs

Proceso de reclutamiento de CodigoDelSur



Introducción

El objetivo es evaluar a un candidato para ser parte del equipo de Android de CodigoDelSur.

El candidato tendrá que desarrollar un producto Android basado en la especificación provista por CodigoDelSur. El candidato tendrá que analizar los requerimientos y plantear las dudas que surjan a través de las siguientes vías:

- Mail a vmengui@codigodelsur.com

Proceso

1. Presentación de los requerimientos por parte de CodigoDelSur
2. Implementación y envío del producto por parte del candidato.
3. Defensa del producto.

Envío de código

Es válido entregar la implementación a través de un repo publico en github o un zip conteniendo todos los archivos (sin incluir la carpeta node_modules)

Defensa del Producto

La defensa del producto consiste en una instancia presencial o una videollamada entre el candidato y los técnicos de CodigoDelSur.

Algunos puntos claves que se evaluarán:

- Comunicación con el cliente
- Análisis de los requerimientos
- Documentación del código
- Decisiones de diseño y arquitectura
- Calidad del producto
- Proactividad

Deadline

El plazo máximo para la entrega del producto es de 2 semanas a partir de la presentación de la evaluación.

Requerimientos

Desarrollar una API que debe ser capaz de dar soporte a las siguientes acciones que serán consumidas por aplicaciones clientes :

- registrar usuario
- autenticar usuario
- obtener películas (con opción de búsqueda por una keyword)
- agregar película a favoritos
- obtener películas favoritas

En la sección “Detalles de Endpoints” se detallara las necesidades de cada acción en particular.

Se valorará aparte de el llevado a cabo de la implementación en sí, la prolijidad y simplicidad con la cual esta es llevada a cabo.

Notas :

- No se solicita el desarrollo de una aplicación cliente, los endpoints implementados en la API pueden ser consumidos con la aplicación “PostMan” , “Curl” o “Insomnia”
- Se sugiere usar versión de Nodejs 10 o 12
- Se sugiere usar Express como framework para la implementación del server que atenderá los endpoints implementados
- Se requiere la integración de la API implementada con una API externa para obtener un listado de películas
<https://www.themoviedb.org/documentation/api?language=es>
- Se necesitará registrarse en dicho sitio para obtener una api key, es **GRATIS**. Si hay problemas con este paso contactarse con CódigoDelSur
- Ante eventuales problemas integrando nuestra API con la API de themoviedb.org se puede simular la obtención de películas desde otra fuente local (un archivo llamado películas.txt almacenado en el proyecto por ejemplo, que contenga una lista de películas)
- No se requiere conexión ni setup de una base de datos real, pero si se requiere el almacenamiento de la información que la API recibe (usuarios registrados y favoritos).
- Se sugiere simular la base de datos simplemente teniendo archivos almacenados en el root folder del proyecto que puedan ser leídos / escritos y que contendrán la información que la API maneja en formato JSON. (**users.txt** y **favoritos.txt**)
- No es necesario que la autenticación soporte múltiples usuarios, requiriendo que cada usuario tenga sus propios datos, esto significa que el usuario que esté “autenticado” va a tener acceso a todos los datos.

Detalles de Endpoints

Registrar usuario

Se debe poder realizar un request con el método “POST” a una url convenientemente decidida por el implementador

En el body del post se debe poder mandar un JSON con los siguientes campos

email
firstName
lastName
password

Se espera que la API :

- verifique de que no exista un usuario con ese email ya registrado en el sistema (recurriendo a **users.txt**)
- En caso de no existir dicho usuario será almacenado en el archivo users.txt (sin borrar usuarios previamente creados.)
- En caso de existir la API deberá responder con un error

Nota : Dicho usuario podrá ser utilizado posteriormente para autenticarse

OPCIONAL : no se solicita q la contraseña (password) sea encriptada antes de ser almacenada , pero si el implementador así lo desea puede hacerlo

Autenticar usuario

Se debe poder realizar un request con el metodo “POST” a una url convenientemente decidida por el implementador

En el body del post se debe poder mandar un JSON con los siguientes campos:

email
password

Se espera que la API :

- verifique que dicho usuario esté registrado en el sistema (recurriendo a **users.txt**)
 - indicar error oportuno en caso contrario
- verifique si el password es el correcto
 - indicar error oportuno en caso contrario
- en caso de que no haya error se deberá retornar al cliente un “token” que posteriormente será utilizado para autenticar al usuario en futuros requests

Nota: La implementación del mecanismo de autenticación queda librada a la imaginación del implementador. Lo unico q se requiere es la existencia de dicho token y que los endpoints que requieran autenticación efectivamente no puedan ser consumidos si la aplicación cliente no envía dicho token.

Obtener películas

Se debe poder realizar un request con el metodo "GET" a una url convenientemente decidida por el implementador.

Se debe proveer en el request un parámetro llamado keyword (NO REQUERIDO) que la API usara para filtrar resultados en la búsqueda.

Se espera que la API :

- Verifique que el usuario invocando el request esté autenticado
 - indicar error oportuno en caso contrario
- Realizar un request a la API de themoviedb.org para traerse peliculas (incluyendo el keyword, si aplica, para filtrado)
- Para cada película retornada la API debe agregar un campo nuevo que se llame "suggestionScore" que debe ser un número aleatorio entre 0 y 99
- La API debe retornar a la aplicación cliente el listado de peliculas ordenado por el campo "suggestionScore"

Agregar películas a favoritos

Se debe poder realizar un request con el método "POST" a una url convenientemente decidida por el implementador.

En el body del post se debe poder mandar un JSON con los cambios obtenidos en la respuesta de una invocación a "**Obtener películas**"

Se espera que la API :

- Verifique que el usuario invocando el request esté autenticado
 - indicar error oportuno en caso contrario
- Inserte en el archivo favoritos.txt el JSON de la película con un campo extra que incluirá la fecha de adición del favorito en el archivo (el campo debe llamarse "addedAt")
- Responder a la aplicacion cliente si hubo error o éxito en la operacion

Obtener películas favoritas

Se debe poder realizar un request con el metodo “GET” a una url convenientemente decidida por el implementador.

Se espera que la API :

- Verifique que el usuario invocando el request esté autenticado
 - indicar error oportuno en caso contrario
- Obtenga del archivo favoritos.txt las películas favoritas de el usuario
- Para cada película agregue un campo extra que se llame “suggestionForTodayScore” que debe ser un número aleatorio entre 0 y 99
- La API debe retornar a la aplicación cliente el listado de peliculas ordenado por el campo “suggestionForTodayScore”

OPCIONALES

Algunos opcionales que el implementador puede llevar a cabo en caso que lo desee.

- Encriptamiento del password
- Manejo de múltiples usuarios
 - Esto significa que un usuario autenticado solo tendrá acceso a agregar favoritos a SU lista de favoritos
 - Tambien debera poder ver sólo1 SU lista de favoritos
 - La implementación de esta solución queda librada a la imaginación del implementador
- Capacidad de hacer Logout
 - crear un endpoint que permita que un “token” de autenticación pueda ser inválido

¡MUCHA SUERTE!