



Backend 2020

Email consulta:

hiring@tiendadolar.com.ar

Entrega: Una semana de recibido

Aviso de confidencialidad. Este documento y la información incluida en él es confidencial y está dirigida únicamente al destinatario.

Tecnología a utilizar

- Node en cualquier versión 10+
- Express
- Cualquier librería que suene necesaria
- Mongo db

Problema

En un miércoles como cualquiera surge la oportunidad de crear una AI llamada Gariboldi que analiza resultados del equipo Leicester City de la Premier League con el fin de predecir el outcome del próximo partido. Como la

Premier es una institución muy antigua y poco tecnológica no hubo caso de conseguir una vía oficial para que Gariboldi se conecte a ellos.

Como nadie te detiene procedemos a activar el plan B.

Se pide

Se debe construir un API Rest que devuelva cierta información básica sobre el equipo de los Foxes. Este API será un microservicio que se conecte a una DB de mongo y se utilize para guardar información en la misma y devolver ciertas cuentas simples para que Gariboldi luego se entrene.

La data debe ser obtenida con un cron una vez por día (o al prenderse el server) de la siguiente web: <https://www.lcfc.com/>. Se puede scrappear el HTML o investigar si se puede utilizar el API que utiliza la página.

Los calls que se piden son los siguientes:

- GET Resultado del último partido
- GET Resultado de un partido en particular
 - Se pueda buscar por fecha o por id
- GET Partidos por intervalo de fecha
- GET para obtener los puntos que tiene Leicester por un rango de fechas
- GET del nombre del equipo que más goles le hizo al Leicester
- Implementar un método POST para agregar un partido a mano, con las mismas características que los partidos provenientes del api. Tener precauciones de realizar todas las validaciones correspondientes

El API propia debe devolver la MONGO DB. De ninguna manera se debe ir a buscar en vivo resultados contra un request.

Considere:

- Buscamos además código ordenado y legible. Levemente comentado mejor.
- Considere como elige las estructuras de datos para que escalable

No es necesario

Implementar nada de Gariboldi.

Adicionales

- Implementar AUTH con jwt token cosa que para poder leer o hacer cualquier request debas antes obtener un authtoken logeandote y permita registrarse con user y password.
- Implementar el servidor enteramente en Typescript.
- Disponibilizar el API en un servicio de hosting como Heroku o similar y la db en un sistema como Mongo Atlas o heroku mismo.



Entrega:

Una vez terminado el ejercicio, se debe sumar a el usuario td-hiring a un repo de github con el código fuente.