Sports News

This task is based on a common business task we have within the backend team. We have a number of microservices that process data from external feed providers. This allows us to transform the data into a consistent and desirable format that the app developers can consume. It also provides stability so that, as is often the case, when the external provider has issues we can still provide data to the apps, albeit stale data.

The project must meet the following requirements:

* Be written in **Golang** - <https://golang.org/doc/> -
* Use **MongoDB** to store news articles - (<https://www.mongodb.com/download-center/community>)
* At regular intervals poll the endpoint for new news articles
* Transform the XML feeds of news articles into appropriate model(s) and save them in the database
* Provide two REST endpoints which return JSON:
  1. Retrieve a list of all articles
  2. Retrieve a single article by ID
* Comments where appropriate
* **Don’t spend more than 1 day on the task - If it's not fully complete (but should compile!) That is fine.**
* **Send over README file listing what works and what doesn't - as well as instructions (if any) on how to run**

### External Feeds

There are two feeds provided by our external provider for Brentford FC club news:

1. Get a list of the latest n news articles  
   <https://www.brentfordfc.com/api/incrowd/getnewlistinformation?count=50>
2. Get addition[Two September games chosen for TV](https://www.brentfordfc.com/api/incrowd/getnewlistinformation?count=50)al details of a news article by ID  
   <https://www.brentfordfc.com/api/incrowd/getnewsarticleinformation?id=173860>

### Example Output

Below are examples of output for the two REST endpoints. Ideally your project should produce output in a similar structure.

1. Retrieve a list of all articles  
   <http://feeds.incrowdsports.com/provider/realise/v1/teams/t94/news>
2. Retrieve a single article by ID  
   <http://feeds.incrowdsports.com/provider/realise/v1/teams/t94/news/e9c26f13-ca6a-5b32-82a5-ecb0d155a644>

### Additional Information

You may find the following links helpful.

Go also has a number of useful guides: <https://golang.org/doc/>

Useful Libraries:

* <https://github.com/gorilla/mux>
* <https://github.com/mongodb/mongo-go-driver>
* <https://github.com/jasonlvhit/gocron> / <https://godoc.org/github.com/robfig/cron>

API Design Docs

* <https://pages.apigee.com/rs/apigee/images/api-design-ebook-2012-03.pdf>
* <http://www.restapitutorial.com/resources.html>
* <https://labs.omniti.com/labs/jsend>

We use **VScode** for Go but you are welcome to use whatever you prefer.

Email **jobs@incrowdsports.com** if you have any questions or need clarification. Asking for help is an important part of being in our team so don’t be afraid to ask if you’re unsure.

**Submit your task to** [**jobs@incrowdsports.com**](mailto:jobs@incrowdsports.com) **as a ZIP using WeTransfer** [**https://wetransfer.com/**](https://wetransfer.com/)