General Notes

Develop a simple **command-line based** Java application with the following basic requirements:

- Programming language: Java (use Java7 or Java 8)
- Database: Up to you. MySQL 5.3+, Mariadb or Postgres are all fine.
- Use properties files to read database access details (like url, user, pass) and anything else you think must be configurable.

Any other technical decision is yours.

Deadline: Please email me by the end of next business day with the deadline you can commit to. Send your Java project to zsofia.unguran@worldofbooks.com or impress us and upload it to a public github repo.

The Task

We simulate a listing reporting system.
Write an application, that:

- synchronize data from a REST API and,
- validate and store it in to the local database and,
- create a report according to the following conditions, and
- upload report as JSON file to FTP

API documentation

GET /listing : array of listings(1000 row)

• id	UUID	Listing identifier
• title	text	Listing title
• description	text	Listing description
• inventory_item_location_id	UUID	Location id
• listing_price	number	Listing price
• currency	text	Currency code
• quantity	number	Item quantity
• listing_status	number	Listing status id
• marketplace	number	Marketplace id
• upload_time	date	Upload time
• owner_email_address	text	Owner email address

API url:
https://my.api.mockaroo.com/listing?key=63304c70

```
curl -H "X-API-Key: 63304c70" https://my.api.mockaroo.com/listing
```

GET /location

• id	UUID	Location identifier
• manager_name	string	Manager name
• phone	string	Phone number
• address_primary	string	Primary address
• address_secondary	string	Secondary address
• country	string	Country
• town	string	Town
• postal_code	string	Postal code

API url:
https://my.api.mockaroo.com/location?key=63304c70

```
curl -H "X-API-Key: 63304c70" https://my.api.mockaroo.com/location
```

GET /listingStatus

• id	number	Listing status id
• status_name	string	Status name

API url:
https://my.api.mockaroo.com/listingStatus?key=63304c70

```
curl -H "X-API-Key: 63304c70" https://my.api.mockaroo.com/listingStatus
```

GET /marketplace

• id	number	Marketplace id
• marketplace_name	string	Marketplace name

API url:
https://my.api.mockaroo.com/marketplace?key=63304c70

```
curl -H "X-API-Key: 63304c70" https://my.api.mockaroo.com/marketplace
```

Data is provided by https://www.mockaroo.com.
If any problem or question came up with the API or reachability of it please let us know, and we will find an alternative way to provide you the API data.
Send a mail to the following address with your problem: andras.boros@worldofbooks.com

Database

Plan the database wisely.

The database tables have to be created by you on your local database, but please include the SQL files in your project.

Validation

Validate the API response:

Listing:

- id — UUID, not null
- title — not null
- description — not null
- location_id — not null, reference to location table
- listing_price — not null, >0, decimals: 2
- currency — not null, length: 3
- quantity — not null, >0
- listing_status — not null, reference to listing status table
- marketplace — not null, reference to marketplace table
- owner_email_address — not null, valid email address format

Collect the invalid lines and write it to **importLog** CSV file with the following content for each row:

ListingId;MarketplaceName;InvalidField

Reporting

Create report that contains the following data:

- Total listing count
- Total eBay listing count
- Total eBay listing price
- Average eBay listing price
- Total Amazon listing count
- Total Amazon listing price
- Average Amazon listing price
- Best lister email address
- Monthly reports:
    - Total eBay listing count per month
    - Total eBay listing price per month
    - Average eBay listing price per month
    - Average Amazon listing price per month
    - Total Amazon listing count per month
    - Total Amazon listing price per month
    - Best lister email address of the month

Monthly reports should contains months without listing.

Use as few queries as you can to collect report data.
Write out the report data to ***report*** JSON file.
Make sure that you create a readable clean structure to the report file.

The finished ***report*** JSON file must be uploaded to an FTP.
Feel free to setup a local FTP on your own, we will manage to test it with our own test FTPs. We use https://filezilla-project.org/ for FTP testing.
Please add sufficient unit tests.