**G2A**

If you put in <https://www.g2a.com/lucene/search/filter?=> it gives you numFound. This is the max value of number of pages of results to search through. Then search through 0... numFound-1 with link <https://www.g2a.com/lucene/search/filter?=&start=numFoundCounter>. This is every app on the market.

Things we want to filter out: anything with the type other than ‘e-goods’. Anything like ‘random keys’, ‘xbox’, ‘ps3’, ‘ps4’ etc etc. Go through a few pages and try find shit to exclude. We’ll go from there and build it up.

If an app passes filtering add to db. We would only want to do this if the app is not already in the list, however this may be complicated as G2A deals with global keys, european keys, russia keys etc. Then on different platforms too. Have to think about this and how we will store these.

If it’s passed filtering we add it to the DB; but that link doesn’t give us the market prices. To do this we need to put the ID through [https://www.g2a.com/marketplace/product/auctions/?id=](https://www.g2a.com/marketplace/product/auctions/?id=aID)ID and inspect and store the values stored in the JSON.

If we want to update the list of G2A games again, we can then go from the last entry forwards as the catalog is populated sequentially. So the first time will take a long time, but then afterward it shouldn’t.

**GOG**

<https://www.gog.com/games?sort=popularity&page=1> gives the number of pages their API has

Then we can query [https://embed.gog.com/games/ajax/filtered?page=36&sort=popularity&countryCode=USD](https://embed.gog.com/games/ajax/filtered?page=36&sort=date&countryCode=USD)

With page 1...number of pages, document all the data and pricings.

Not best solution for multi threading. Need to handle one at a time. If needs be we can do sequentially as the parsing shouldn’t take long and there’s only 36 pages, should be comparatively quick in comparison to others. Plus there’s only 1700 listings so that’s useful.