

# [Import mpage + "and”; out += "new=”](https://assignbuster.com/import-mpage-out-new/)

import com. google. gson. Gson; import com. google. gson. GsonBuilder; import com. google. gson. annotations. SerializedName; import org. jsoup. Jsoup; import org. jsoup. nodes. Element; import org. jsoup. select. Elements; import java. io. IOException; import java. util. ArrayList; import java. util. List; public class CompanyScrap {    class CompanyFilter {        @SerializedName(“ ids”)        private List mIds;        @SerializedName(“ hexdigest”)        private String mDigest;        @SerializedName(“ total”)        private String mTotalCount;        @SerializedName(“ page”)        private int mPage;        @SerializedName(“ sort”)        private String mSort;        @SerializedName(“ new”)        private boolean mNew;        public List getIds() {            return mIds;        }        public String getDigest() {            return mDigest;        }        public String getTotalCount() {            return mTotalCount;        }        public int getpage() {            return mPage;        }        private String buildRequest() {            String out = “ total=” + mTotalCount + “&”;            out += “ sort=” + mSort + “&”;            out += “ page=” + mPage + “&”;            out += “ new=” + mNew + “&”;            for (int i = 0; i < mIds. size(); i++) {                out += " ids=" + mIds. get(i) + "&";            }            out += " hexdigest=" + mDigest + "&";            return out;        }    }    private static class Company {        private String mLink;        private String mName;        private String mDescription;        public Company(String name, String link, String description) {            mLink = link;            mName = name;            mDescription = description;        }        public String getLink() {            return mLink;        }        public String getName() {            return mName;        }        public String getDescription() {            return mDescription;        }    }    private static List getCompanies(final CompanyFilter companyFilter) throws IOException {        List companies = new ArrayList <>();        Element doc = Jsoup. connect(“ https://angel. co/companies/startups?” + companyFilter. buildRequest())                . userAgent(“ Mozilla”)                . ignoreContentType(true)                . get(). body();        Elements data = doc. select(“ divdata-\_tn”);        if (data. size() > 0) {            for (int i = 2; i < data. size(); i++) {                companies. add(new Company(data. get(i). select(" a"). first(). attr(" title"). replace(""", ""),                        data. get(i). select(" a"). first(). attr(" href"). replace(""", ""),                        data. get(i). select(" div. pitch"). first(). text(). replace("", "")));            }        } else {            System. out. println(" no data");        }                      return companies;    }    /\*\*     \* Return company filter object     \*/    private static CompanyFilter getCompanyFilter(final String filter, final int page) throws IOException {        String response = Jsoup. connect(" https://angel. co/company\_filters/search\_data")                . header(" Content-Type", " application/x-www-form-urlencoded; charset= UTF-8")                . header(" X-Requested-With", " XMLHttpRequest")                . data(" filter\_datacompany\_types=", filter)                . data(" sort", " signal")                . data(" page", String. valueOf(page))                . userAgent(" Mozilla")                . ignoreContentType(true)                . post(). body(). text();        GsonBuilder gsonBuilder = new GsonBuilder();        Gson gson = gsonBuilder. create();        return gson. fromJson(response, CompanyFilter. class);    }    public static void main(String args) throws IOException {        int pageCount = 1;        List companies = new ArrayList <>();        for (int i = 0; i < 10; i++) {            //System. out. println(" get page n°" + pageCount);            CompanyFilter companyFilter = getCompanyFilter(" Startup", pageCount);            pageCount++;            //System. out. println(" digest     : " + companyFilter. getDigest());            //System. out. println(" count      : " + companyFilter. getTotalCount());            //System. out. println(" array size : " + companyFilter. getIds(). size());            //System. out. println(" page       : " + companyFilter. getpage());            companies. addAll(getCompanies(companyFilter));                        if (companies. size() == 0) {                break;            }         }        for (int i = 0; i < companies. size(); i++){        System. out. println(" NAME : "+companies. get(i). getName());        System. out. println(" DESCRIPTION : "+companies. get(i). getDescription());        System. out. println(" LINK : "+companies. get(i). getLink());        }        }}