



For one of the apps that I was developing recently, I needed some information on the elements of the periodic table.

Now Wikipedia gives neat intros/summaries in the first paragraph of it's articles.

So I decided I would just grab that. Now, I needed to find a way to do that as efficiently and as easily as possible.

After exploring a few options I decided it would be best if I parse the HTML rather than the XML markup.

Fortunately, there is this great library available for Java, which lets you parse HTML. It's called [Jsoup](#). The best part is it is open source.

The library is really powerful and parsing just the first paragraph is a piece of cake using this library as it can do much more advanced things than that.

So, let's see how to use it.

Tutorial:

[NOTE: don't forget to add Internet permissions

```
<uses-permission android:name="android.permission.INTERNET" />
```

Step 1: Add the Gradle dependency in your Build.gradle file.

```
compile 'org.jsoup:jsoup:1.10.2'
```

Step 2: Now, in this step we would parse the first para of the article. But there is an issue. We cannot do it simply as the documentation of the library describes.

Rather we would need to do that as an Async Task.

So let's say you have a TextView in your activity and you want to display the first paragraph of the article on Hydrogen.

Then just add the following code to your activity(Note: it should be outside your onCreate method but inside the Activity class)

```
private class JsoupAsyncTask extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected Void doInBackground(Void... params) {
        try {
            Document doc =
```

```

        Parsing the first paragraph of Wikipedia articles using Jsoup- Android Development | 2
Jsoup.connect("https://en.wikipedia.org/wiki/Hydrogen").get();
    org.jsoup.select.Elements paragraphs=doc.select(".mw-content-ltr p");
    Element firstParagraph = paragraphs.first();
    para=firstParagraph.text();

    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(Void result) {
    textView.setText(desc);
}
}

```

where `para` and `textView` are `String` and `TextView` objects defined in the enclosing class. Once the first paragraph is obtained the `setText` command in the `onPostExecute()` method will set the text of the `textView` with the first paragraph.

Step 3:

Now you just need to call/execute this task.

```

JsoupAsyncTask jsoupAsyncTask = new JsoupAsyncTask();
jsoupAsyncTask.execute();

```

and Voila! the parse is done.

Now let me also explain some of the logic of the code.

```
Document doc = Jsoup.connect("https://en.wikipedia.org/wiki/Hydrogen").get();
```

This line fetches the given url and parses it to a DOM. This documents consists of `Elements`.

The next line is the crucial part.

In the next line we create an object `paragraphs` of the type `Elements` which selects the elements with the `p`(paragraph tag) belonging to the class `mw-content-ltr`.

```
org.jsoup.select.Elements paragraphs=doc.select(".mw-content-ltr p");
```

If you look at a Wikipedia article you can notice that the article contents are in the class `mw-content-ltr`.

Now in the next line I store the first paragraph in a new object `paragraph` of the type `Element`.

```
Element firstParagraph = paragraphs.first();
```

Then use the `.text()` to get the text of in the element.

```
para=firstParagraph.text();
```

Now, all that was easy peasy but there are a lot of unhandled errors there like `NP exception` etc.

So below I am adding a more elegant code that takes care of all the errors and returns the first paragraph of a Wikipedia article.

Better Code:

[**Note:** This one has a progress dialog spinner also that shows while the paragraph/intro/summary is being fetched.]

In your parent/enclosing class declare the following:

```
TextView article;
```

```

ProgressDialog progress;
String articleUrl="https://en.wikipedia.org/wiki/Carbon";
Boolean success;
String error;
String para;
Context context=MainActivity.this;

```

Then find(initialize) the textView in the onCreate Method of your activity(Make sure you have added a textView with the id:article to your layout of activity.):

```
article=(TextView)findViewById(R.id.text);
```

Then add the following class to it:

```

private class wikiArticleParser extends AsyncTask<Void, Void, Void> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progress=new ProgressDialog(context);
        progress.show();
    }

    @Override
    protected Void doInBackground(Void... params) {
        try {
            Connection.Response response = Jsoup.connect(articleUrl)
                .userAgent("Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/535.21
(KHTML, like Gecko) Chrome/19.0.1042.0 Safari/535.21")
                .timeout(4000)
                .ignoreHttpErrors(true)
                .execute();

            int statusCode = response.statusCode();
            if (statusCode == 200) {
                success = true;
                Document doc = Jsoup.connect(articleUrl).get();
                org.jsoup.select.Elements paragraphs = doc.select(".mw-content-ltr
p");

                if (!paragraphs.isEmpty()){
                    Element firstPara=paragraphs.first();
                    para=firstPara.text();
                }
                else{
                    success=false;
                    error="No para found";
                }
            }
            else {
                success = false;
                error=Integer.toString(statusCode);
            }
        }
    }
}

```

```

    }catch (IOException e) {
        e.printStackTrace();
        success=false;
        error=e.getLocalizedMessage();
    }
    return null;
}

@Override
protected void onPostExecute(Void result) {
    progress.dismiss();
    if(success){
        article.setText(para);
    }
    else{
        article.setText("Sorry, there was an error: "+error);
    }
}
}

```

Then finally execute this task as:

```

wikiArticleParser jsoupAsyncTask = new wikiArticleParser();
jsoupAsyncTask.execute();

```

The above is a more refined version of the previous code, handling all the errors, and displaying the issues if encountered.

An even better and desirable way could be to have the AsyncTask class, as a separate class rather than an enclosing class.

So that all the paragraph fetching is done separately.

That can be done as follows:

Create a new class `firstPara` that extends `AsyncTask` and pass the `context`, `article url` and the `TextView` to by implementing a constructor:

```

public class firstPara extends AsyncTask<Void, Void, Void> {

    ProgressDialog progress;
    String articleUrl;
    Boolean success;
    String error;
    String para;
    Context context;
    TextView textView;

    //constructor
    protected firstPara(Context context, String url, TextView textView){
        articleUrl=url;
        this.context=context;
        this.textView=textView;
        progress=new ProgressDialog(context);
    }
}

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    progress.show();
}

@Override
protected Void doInBackground(Void... params) {
    try {
        Connection.Response response = Jsoup.connect(articleUrl)
            .userAgent("Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/535.21
(KHTML, like Gecko) Chrome/19.0.1042.0 Safari/535.21")
            .timeout(4000)
            .ignoreHttpErrors(true)
            .execute();

        int statusCode = response.statusCode();
        if (statusCode == 200) {
            success = true;
            Document doc = Jsoup.connect(articleUrl).get();
            org.jsoup.select.Elements paragraphs = doc.select(".mw-content-ltr
p");

            if (!paragraphs.isEmpty()){
                Element firstPara=paragraphs.first();
                para=firstPara.text();
            }
            else{
                success=false;
                error="No para found";
            }
        }
        else {
            success = false;
            error=Integer.toString(statusCode);
        }

    }catch (IOException e) {
        e.printStackTrace();
        success=false;
        error=e.getLocalizedMessage();
    }
    return null;
}

@Override
protected void onPostExecute(Void result) {
    progress.dismiss();
    if(success){
        textView.setText(para);
    }
    else{
        textView.setText("Sorry, there was an error: "+error);
    }
}
}

```

}

Then execute this task in your MainActivity or any other activity similarly as before, except this time pass the arguments for context, url and textView:

```
new  
firstPara(MainActivity.this, "https://en.wikipedia.org/wiki/Carbon", article).execute();
```



[Manas Sharma](#)

I'm a physicist specializing in theoretical, computational and experimental condensed matter physics. I like to develop Physics related apps and softwares from time to time. Can code in most of the popular languages. Like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

Share this:

- [Click to share on Facebook \(Opens in new window\)](#)
- [Click to share on Twitter \(Opens in new window\)](#)
- [Click to share on Google+ \(Opens in new window\)](#)
- [Click to share on WhatsApp \(Opens in new window\)](#)
- [Click to share on Pinterest \(Opens in new window\)](#)
- [Click to share on Reddit \(Opens in new window\)](#)
- [Click to share on LinkedIn \(Opens in new window\)](#)
- [Click to share on Skype \(Opens in new window\)](#)
- [Click to email this to a friend \(Opens in new window\)](#)
- [Click to print \(Opens in new window\)](#)
- [Click to share on Tumblr \(Opens in new window\)](#)
- [Click to share on Pocket \(Opens in new window\)](#)
- [Click to share on Telegram \(Opens in new window\)](#)

Like this:

Like Loading...

Consider donating if you found the information useful

Appreciate your blog: \$3 ▼

