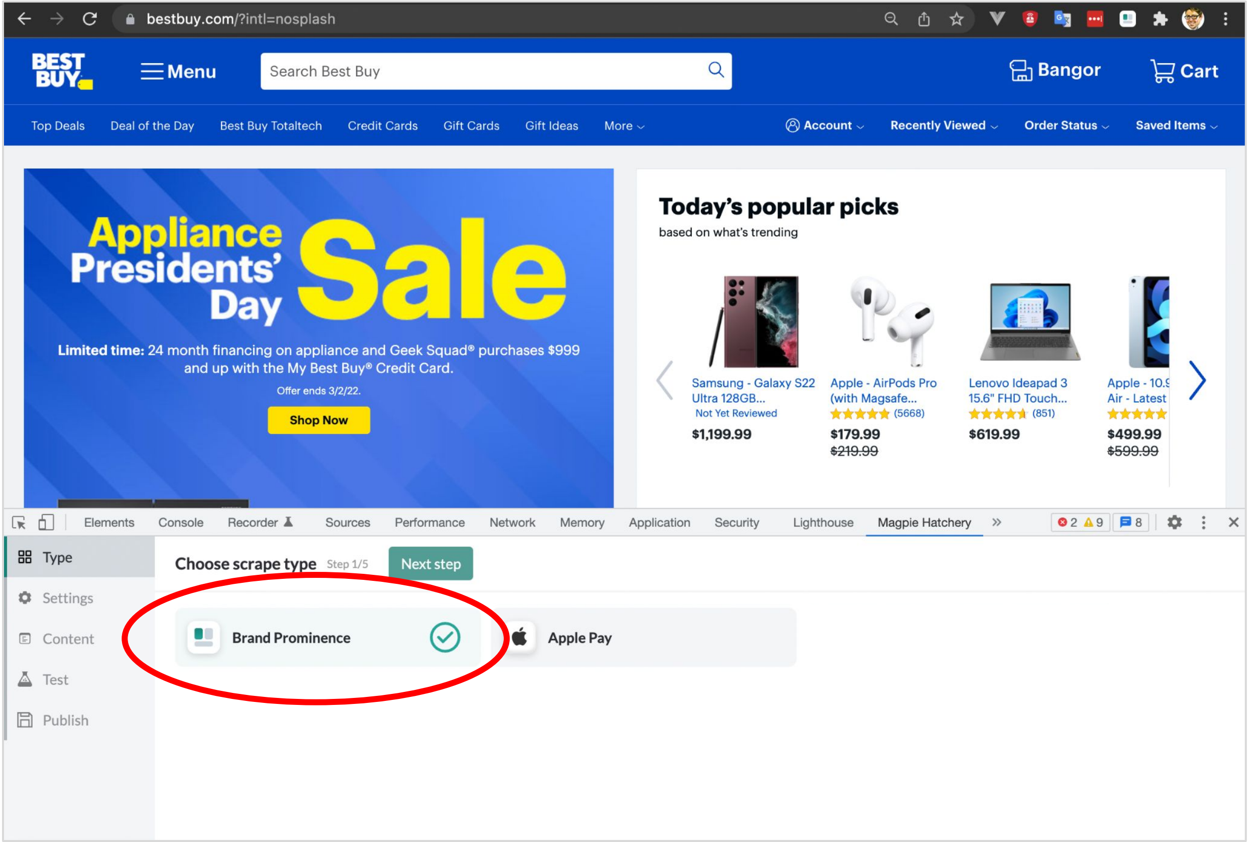
Overview

The goal of this project is to create a **Chrome Developer Tools Extension** that enables the point and click capture of elements on a web page and the saving of the position of those elements into a JSON file. Once saved it should then download screenshots of all the slots/elements identified when pointing and clicking.

This is to enable to capture of page content by a user for the purposes of page analysis.

Step one: Type



Choose scrape type Step 1/5

Brand Prominence ✓

The first step creates page record and starts the process of creating the JSON file.

This will need to show the user 2 options:

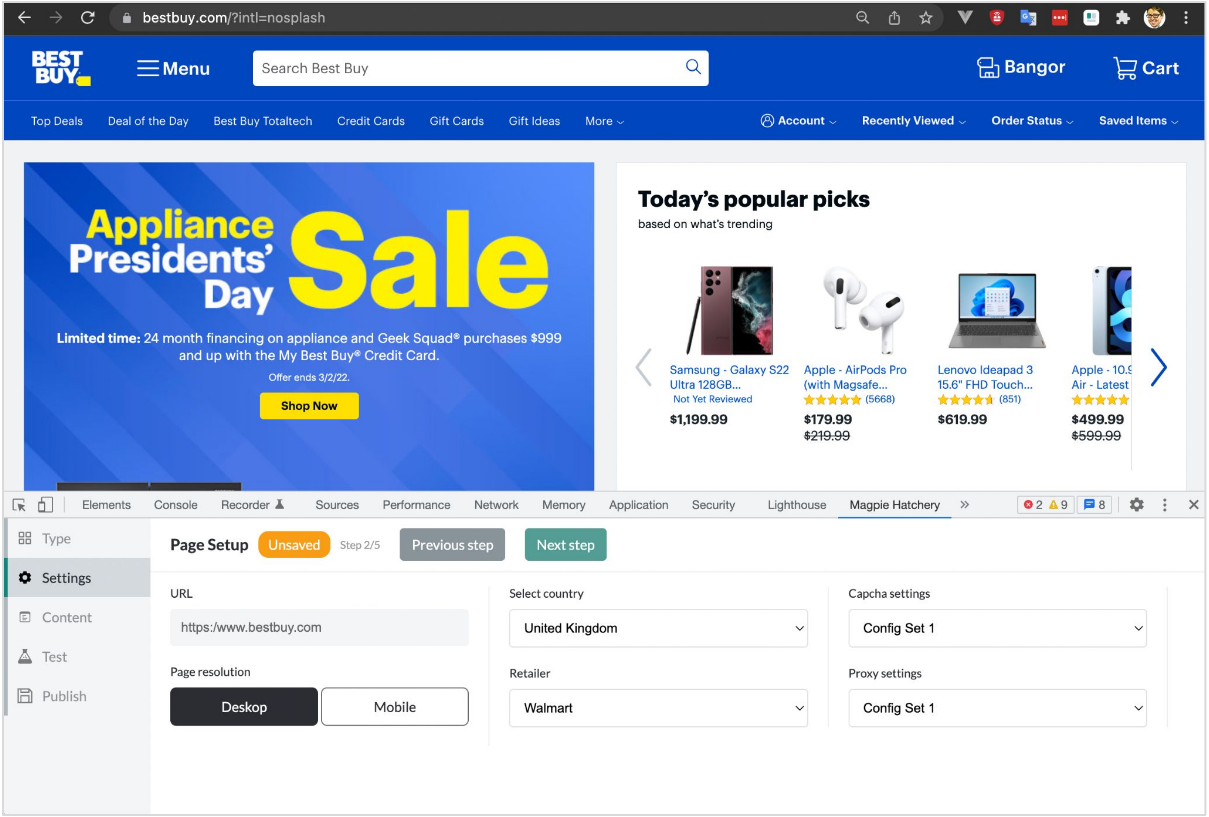
- Brand Prominence
- Apple Pay

It will need to send a post request to:

And when the response returns an ID start creating the JSON file with:

```
{
“id”: [response-id],
“scrapeType”:[chosen-option]
}
```

Step two: Settings



Page Setup Unsaved Step 2/5

URL: https:/www.bestbuy.com

Page resolution: Deskop

Select country: United Kingdom

Retailer: Walmart

Capcha settings: Config Set 1

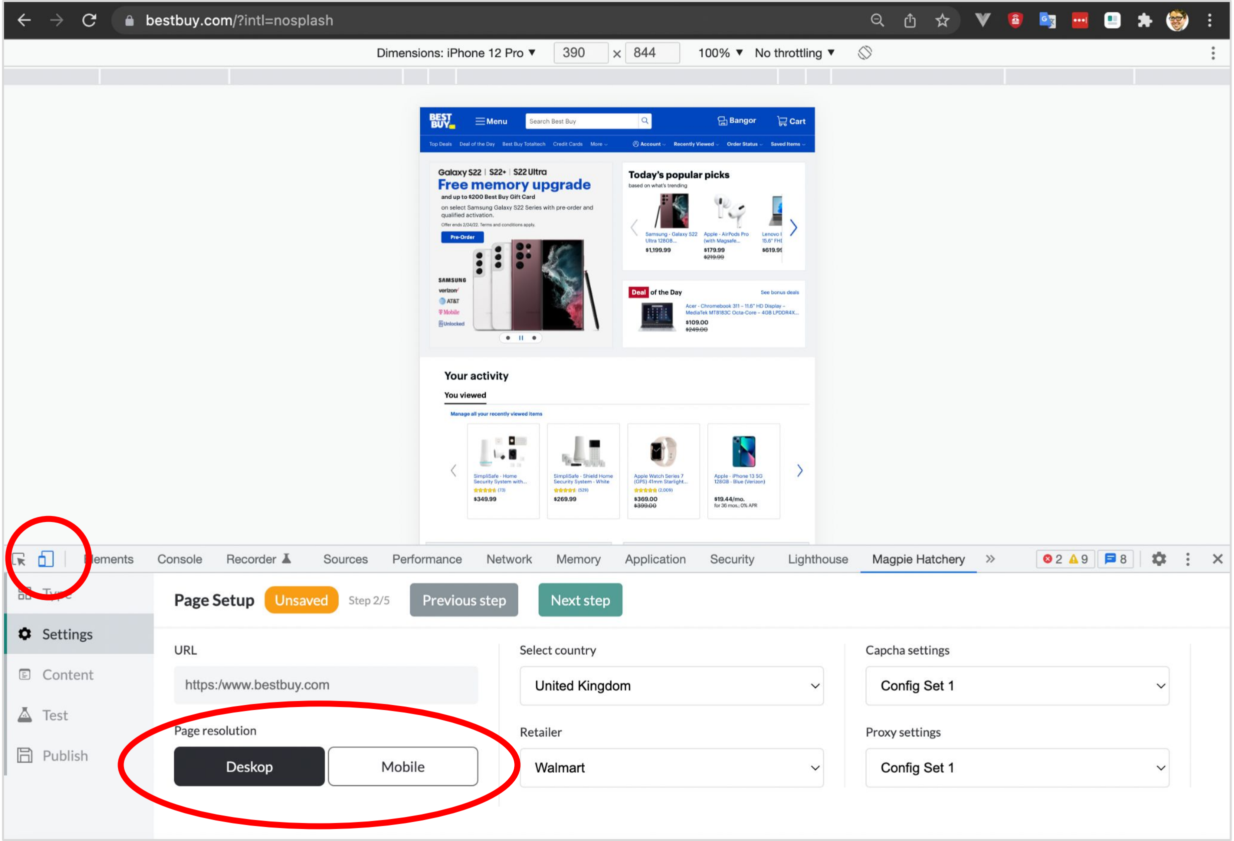
Proxy settings: Config Set 1

The settings step sets a number of options in the JSON file.

```
{
“id”: [response-id],
“scrapeType”:[chosen-option]
"Partner_name":[chosen-retailer]
"url": [current-url],
"Url_name":[page-title],
"Partner_homepage_url":
[root-url],
"partner_country":
[chosen-country-name],
"partner_country_id":
[chosen-country-value],
}
```

When ‘Next Step’ is clicked it needs to post these values to a URL to save online.

Step two: Settings - page resolution



Page Setup Unsaved Step 2/5 Previous step Next step

URL: https:/www.bestbuy.com

Select country: United Kingdom

Capcha settings: Config Set 1

Page resolution: Deskop | Mobile

Retailer: Walmart

Proxy settings: Config Set 1

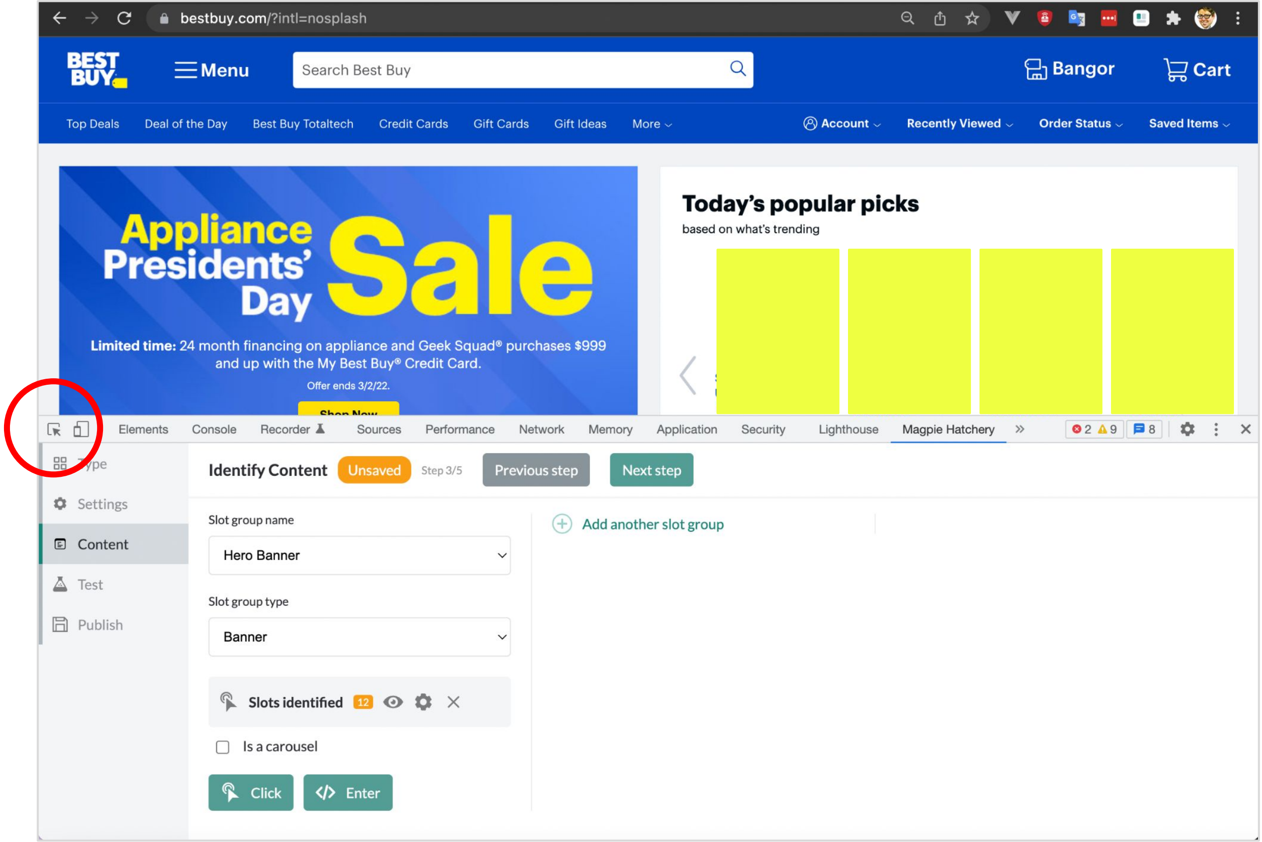
The user can chose the page resolution. This needs to use the Chrome tools ‘Toggle device toolbar’ to set the page dimensions in the browser. For example, if ‘Mobile’ was chosen then the browser would show mobile dimensions as shown on the left. This needs adding to the JSON file too:

```
“resolution”:
[mobile-or-desktop],
```

Options are:
DESKTOP', '1280x1024'
MOBILE', '414x896';

Clicking ‘Next Step’ will need to post these values to a supplied URL to save online again.

Step three: Content (click)



Identify Content Unsaved Step 3/5

Slot group name: Hero Banner

Slot group type: Banner

Slots identified 12

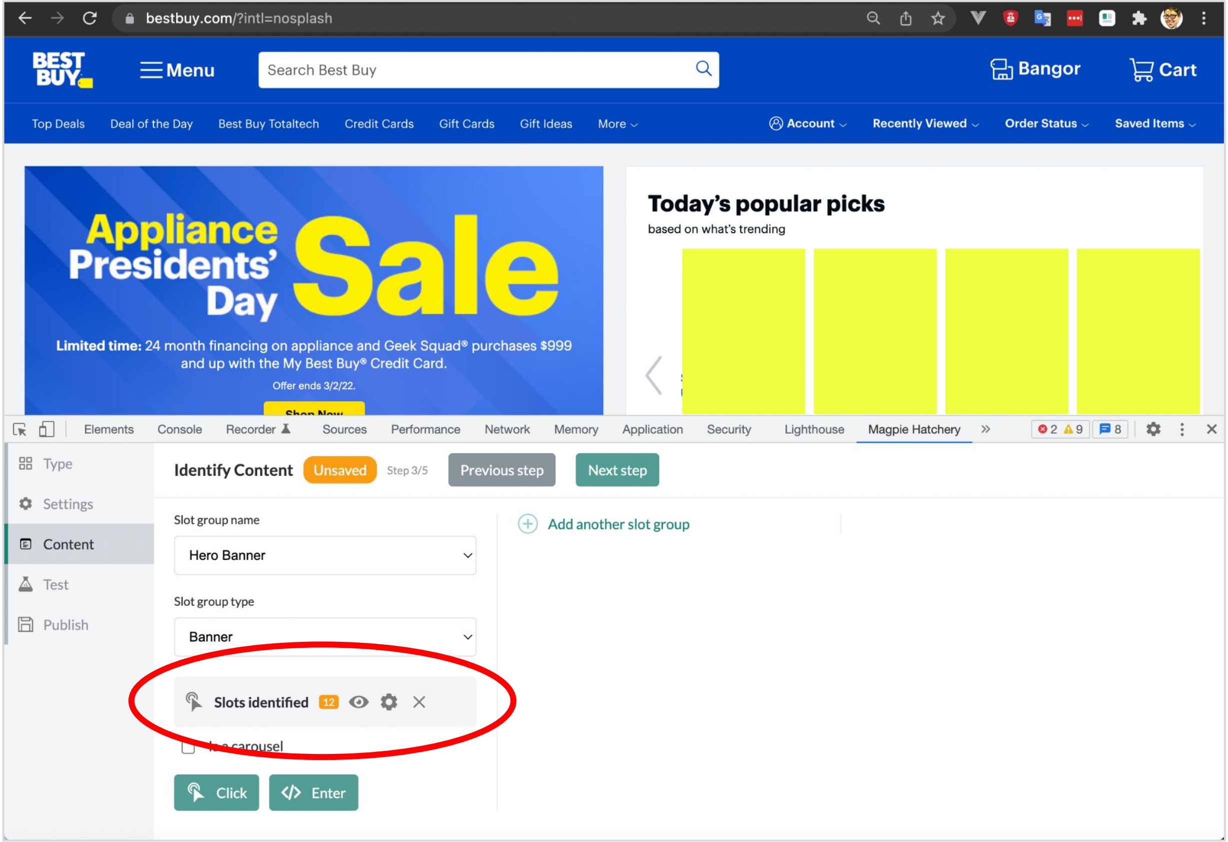
□ Is a carousel

The content stage is the core part of the tool. Here we need to first create a slot group. The user will give it a name and type from two drop down fields.

Then they will either click, or enter div class/ids, to identify slots on the page.

Firstly, by clicking on a html element using the Chrome Dev tools element inspector (or similar) they can select an element on the page. The extension will then identify any other elements with the same name on the page in order to identify repeating slots- such as a list of products in a product grid, or items in a carousel. The system will say how many slots have been found.

Step three: Content (click)



Slot group name: Hero Banner

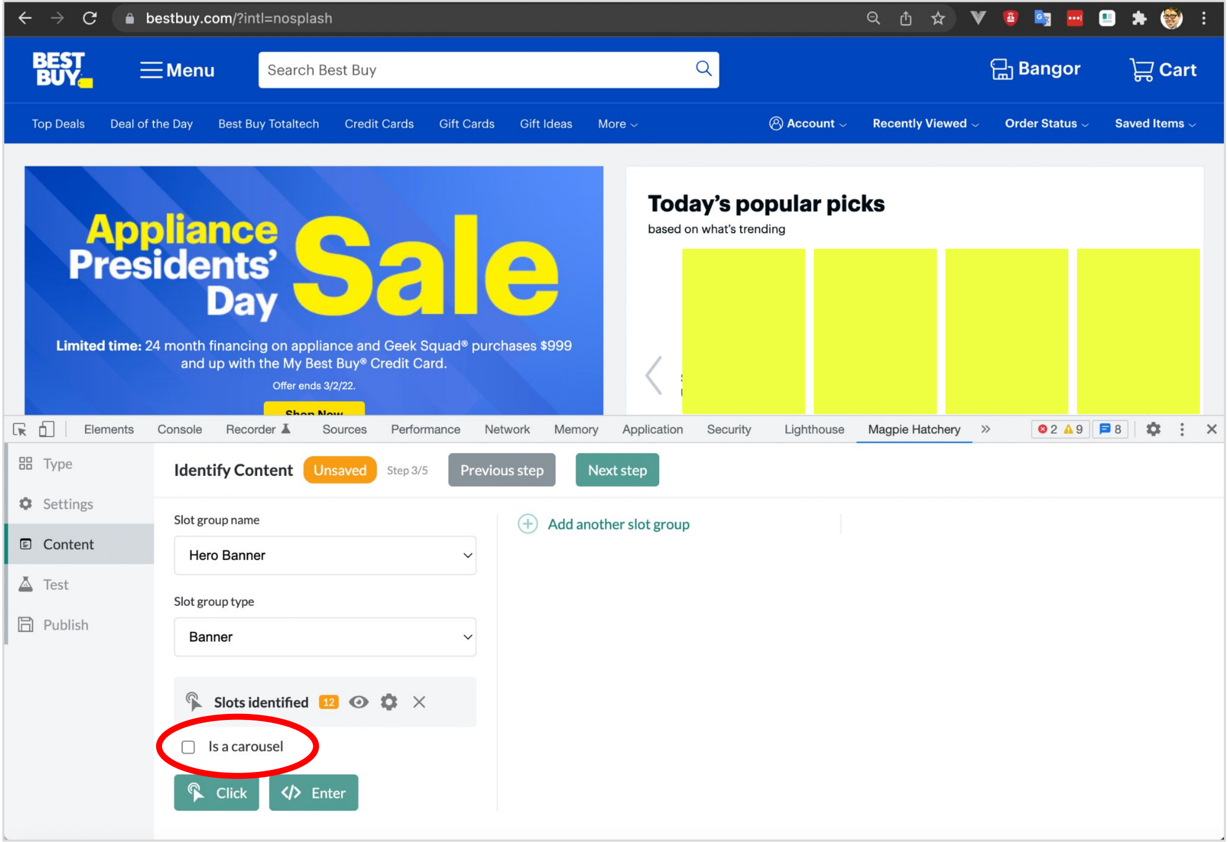
Slot group type: Banner

It will then add each element to the JSON file within a 'slots' object, like so:

```
"slots":[
       {
       "name": "Hero Banner",
       "section_type": "Banner",
       "position": 1,
       "x_position":0,
       "y_position":262,
       "width":1280,
       "Height":445,
       "Screenshot": unique-id.png
       },
       {
       "name": "Hero Banner",
       "section_type": "Banner",
       "position": 2,
       "x_position":300,
       "y_position":262,
       "width":1280,
       "Height":445
       "Screenshot": unique-id.png
       }
```

Please note, each slot needs to be given a unique ID so it can be matched to a screenshot PNG of that area of the page, later on.

Step three: Content (Is a carousel)



Slot group name: Hero Banner

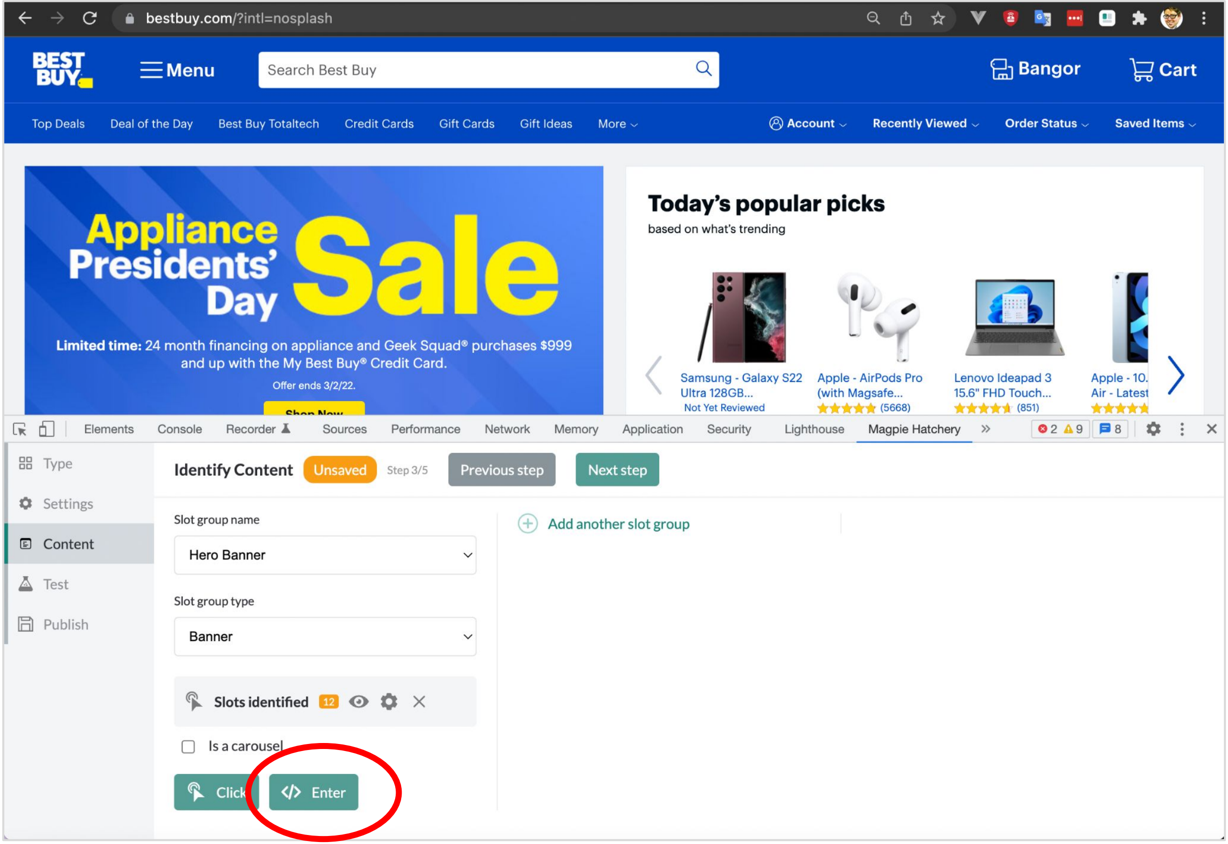
Slot group type: Banner

□ Is a carousel

If the user ticks the ‘Is a carousel’ tick box for this slot group, the JSON is slightly different:

```
"slots":[
       {
       "name": "Hero Banner",
       "section_type": "Banner",
       "position": 1,
       "x_position":null
       "y_position":null,
       "width":null,
       "Height":null,
      "is_in_carousel": true,
       "carousel_total_frames":3,
       "carousel_frame_number":1,
       "carousel_x_position":0,
       "carousel_y_position":262,
       "carousel_width":1280,
       "Carousel_height":445,
      “Screenshot”: id.png
      },
```

Step three: Content (enter class/id)



Identify Content Unsaved Step 3/5

Slot group name: Hero Banner

Slot group type: Banner

Slots identified 12

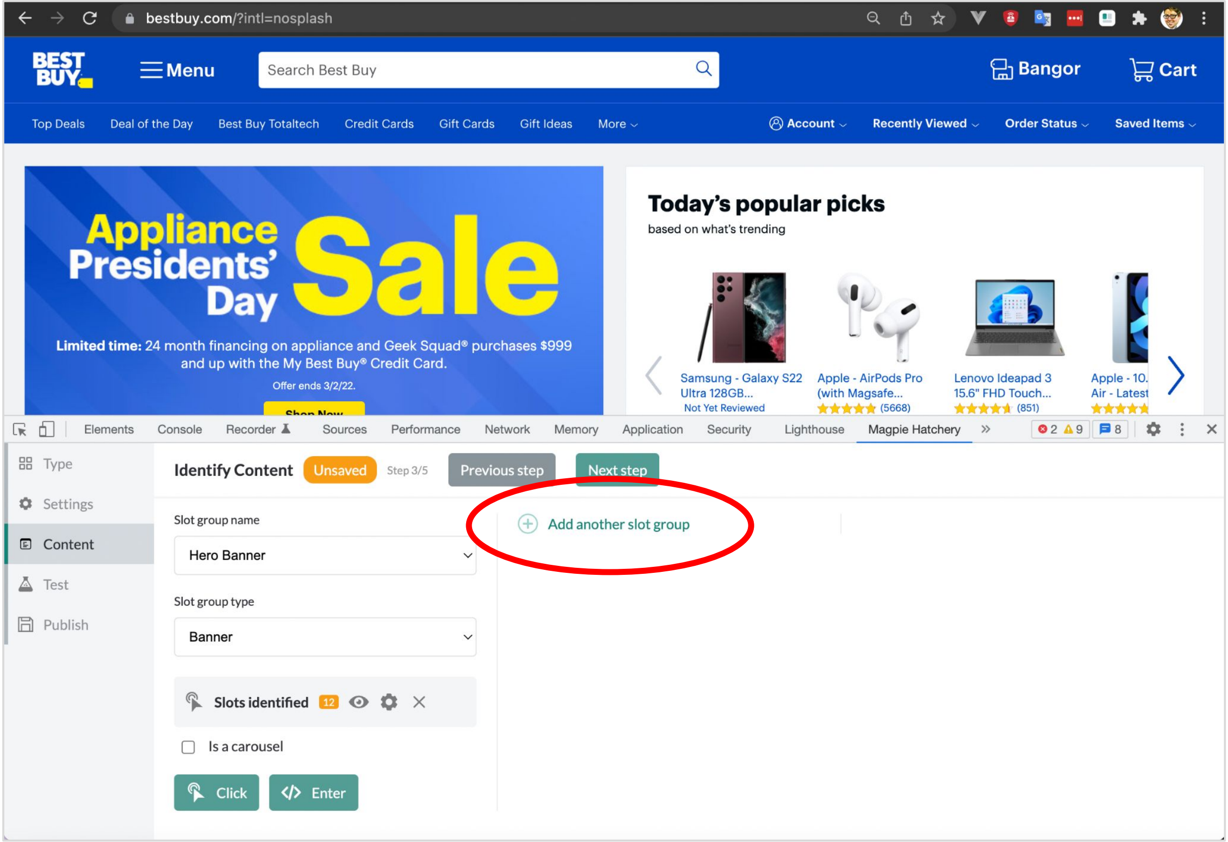
☐ Is a carousel

A second option for finding slots on the page is for the user to manually enter a html class or ID.

If they click the Enter button, they will be given a text field to enter a class or ID .class #id

The extension then scans the page to identify all elements with that class or ID and save the details to the slot JSON file.

Step three: Add another slot group



Slot group name: Hero Banner

Slot group type: Banner

Slots identified 12

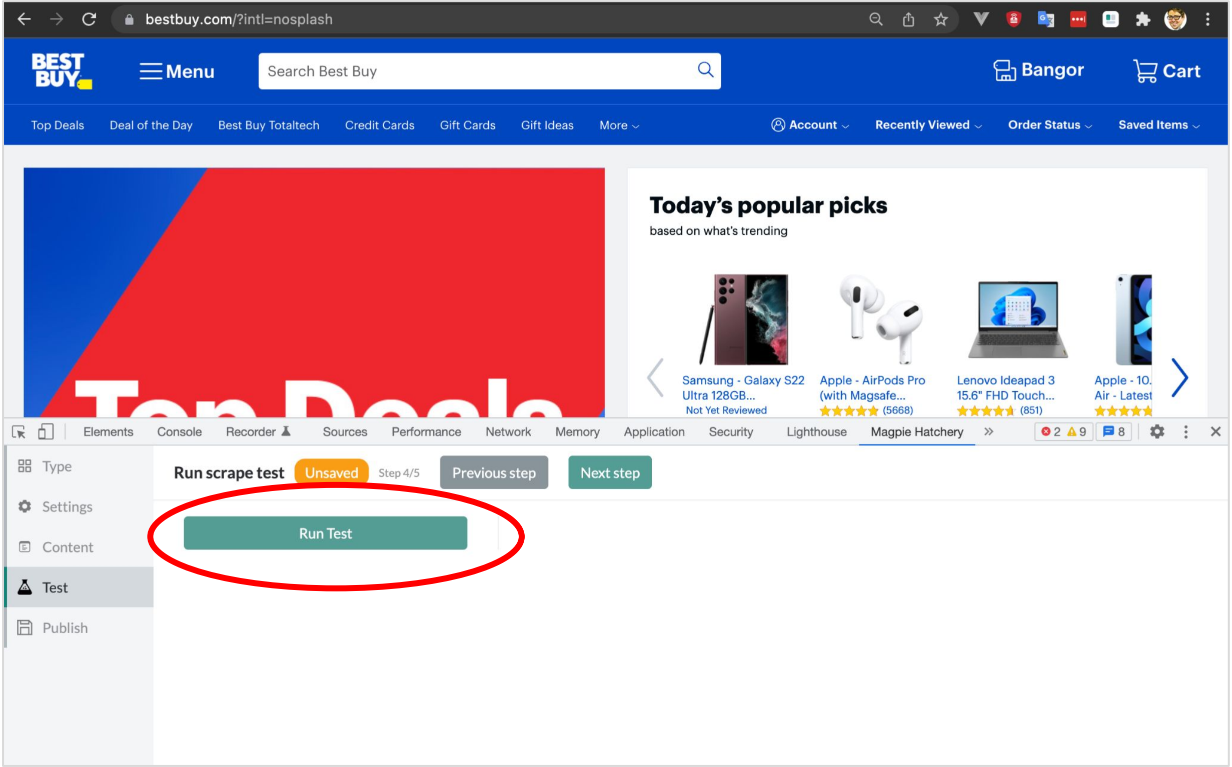
☐ Is a carousel

Once the user has finished identifying the elements on the page for that slot group they can either go to the next step, or add another slot group. If they add another one, the same process as before is done. But they can obviously give this group another name/type, plus the count of the positions of the elements in the slot group restarts at 1.

```
{
    "name": "Slot group 2",
    "section_type": "Banner",
    "position": 1,

etc…
```

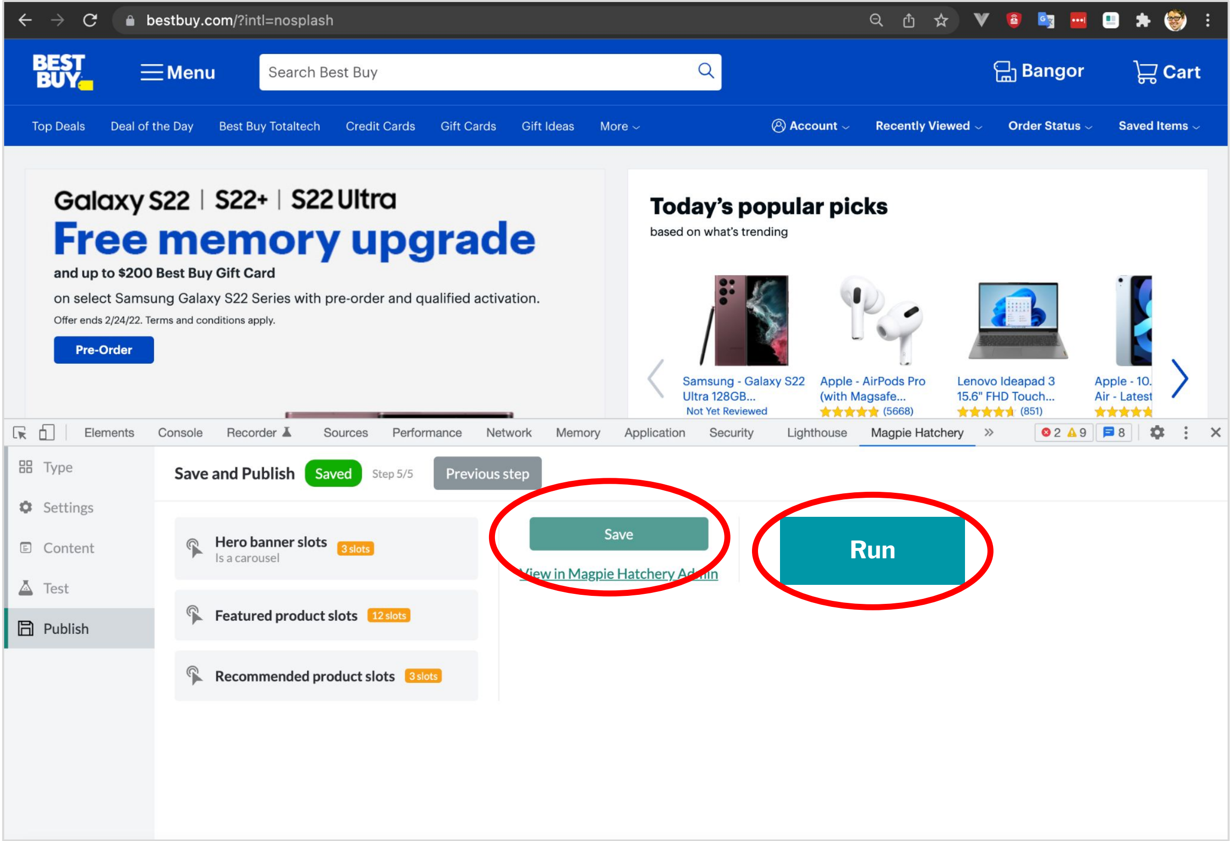
Step four: Test



Once the user has identified all the elements on the page they wish to capture, they can run a test. This will generate the full JSON file for them to preview.

See included JSON file for example. This will ideally be previewed in a popup modal on the page.

Step five: Save and Run



Once the user is happy with the JSON preview, they click Next and can save the full settings.

This will show them a summary of the slots they have identified. The Save button will post the full JSON result to a supplied URL, so it can be saved in a database for later use.

Finally, the user can RUN this. When they click run it will need to capture node screenshots of all the elements identified in the slots in the JSON file. It will save each node screenshot based on the unique-id given to the slot.