Use a Failsafe or Fallback Selector List When Building Actions

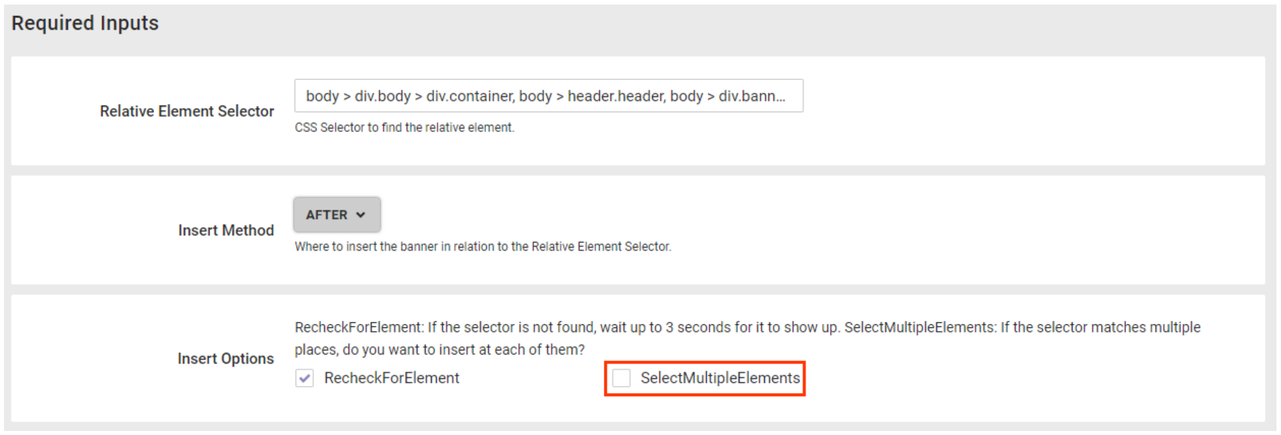
The Element Selector is the cornerstone of most actions. This input is present in the majority of action types that insert new content or edit existing content.

Each site is unique and presents distinct challenges for development. Some sites may have varied layouts for product detail pages. These pages are still categorized as `product` but have different HTML markup and therefore need different selectors for targeting actions.

Presumably, you could solve this by building multiple actions, each with a unique and specific selector, and target each variation to the respective layouts. This increases the time dedicated to development and testing and complicates and adds tedium to your experience setup.

Alternatively, you might need to build actions with deliberately broad selectors, but this carries the inherent risk of inserting an action in the wrong place on a page. Again, you would likely see an increased amount of time spent testing, but would ultimately result in a less tedious experience setup.

Finally, you have an option that takes the best of both worlds. You can deliberately put in multiple selectors that are separated by commas to make an array. If you disable **Select All Elements**, this list of selectors acts as action and experience priority. It attempts to find the first selector in the list. If it fails to do so, it then attempts to find the second in the list and so on.


Required Inputs

Relative Element Selector: body > div.body > div.container, body > header.header, body > div.bann...
CSS Selector to find the relative element.

Insert Method: AFTER
Where to insert the banner in relation to the Relative Element Selector.

Insert Options: RecheckForElement: If the selector is not found, wait up to 3 seconds for it to show up. SelectMultipleElements: If the selector matches multiple places, do you want to insert at each of them?
✓ RecheckForElement
☐ SelectMultipleElements

If your selectors are correctly listed with the more common variations first and less common ones later and each selector in the list is sufficiently unique, the result is a single action that is easy to use and update in an experience as needed to target all product pages regardless of layout.