

1. Pull

Eat Smart's architecture is based off of the pull architecture where the client checks the 3rd-party API server for updates. When the client wishes to make a search request to update the recipe list, the code fetches data from the URL endpoint pre-defined in the `pantry-route.js` routes. While the code doesn't explicitly have pull in it, making a fetch request is synonymous as such.

Below is a snippet of code found in `EatIn.js` where pull architecture is being used. When the client clicks on the Save & Search button in the Eat In page, it triggers the `handleSearch()` function which makes a request to the Spoonacular API to retrieve data and consequently updates the state using the results.

```
handleSearch() {  
  fetch('/pantry/recipes/'+this.state.id)  
    .then(res => {  
      console.log(res);  
      return res.json()  
    })  
    .then(recipes => {  
      this.setState({ recipes: recipes })  
    });  
}
```