

Retrieving Data Reactively



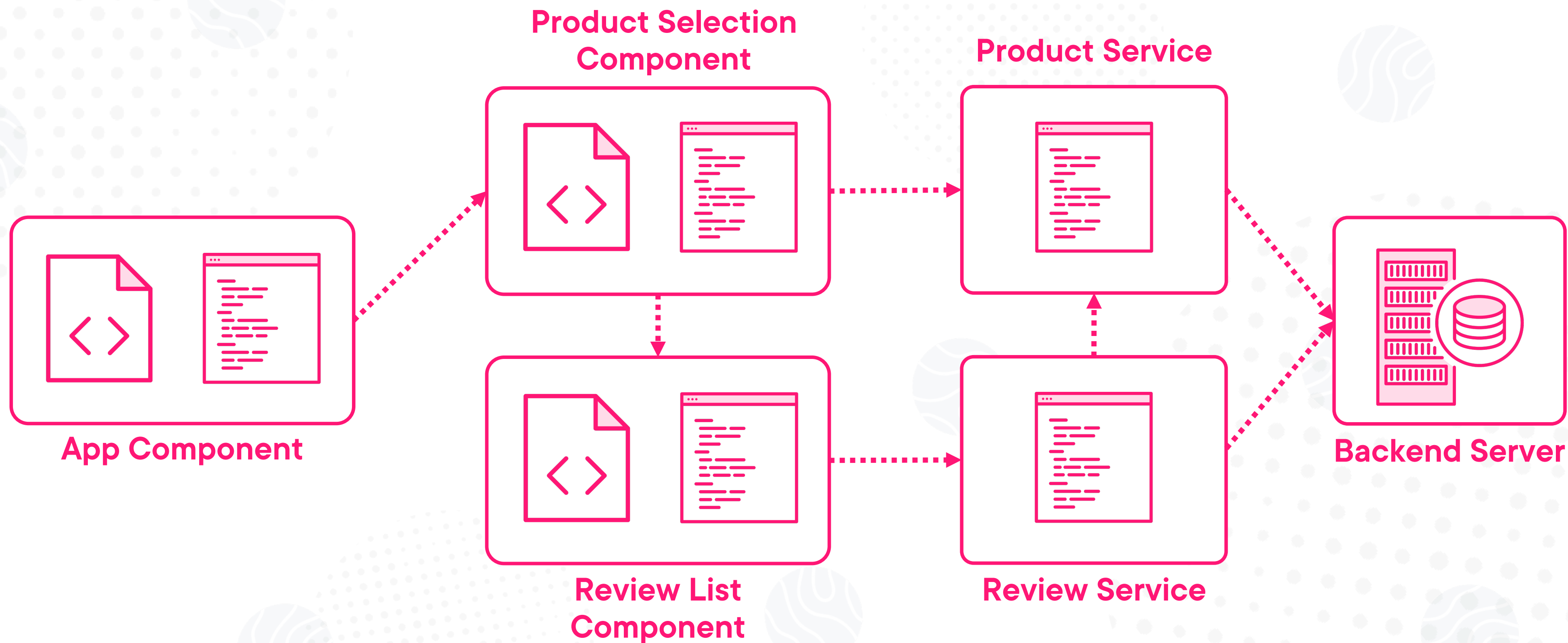
Deborah Kurata

Developer | Content Creator | MVP | GDE

@deborahkurata | https://www.youtube.com/@deborah_kurata



Sample Application Structure



Key `httpResource()` Features

Retrieve data directly into a signal

```
productsResource = httpResource<Product[]>(() => this.productsUrl);  
products = this.productsResource.value;
```

Reactively retrieve data in response to signal changes

```
productResource = httpResource<Product>(() =>  
    `${this.productsUrl}/${this.ps.selectedProd()?.id}`);
```

Get data using signals as query parameters

```
reviewsResource = httpResource<Review[]>(() =>  
    `${this.url}?pid=${this.ps.selectedProd()?.id}`);
```

Avoid an HTTP request by returning undefined

```
reviewsResource = httpResource<Review[]>(() =>  
    this.ps.selectedProd() ?  
        `${this.url}?pid=${this.ps.selectedProd()?.id}` : undefined);
```



Control When the HTTP Request Is Issued

```
getProductsNow = signal(false);  
  
productsResource = httpResource<Product[]>(() =>  
    this.getProductsNow() ? this.productsUrl : undefined,  
    { defaultValue: [] }  
);
```



Key `HttpRequest` Features (continued)

Work with the returned response data as a writable signal

```
(property) ProductSelectionComponent.products: WritableSignal<Product[]>  
products = this.productService.productsResource.value;
```

Access the status of the HTTP request

```
requestStatus = this.ps.productsResource.status();  
isLoading = this.ps.productsResource.isLoading();
```

Display a loading indicator

```
@if (isLoading()) {  
  <div>... loading products</div>
```



Key `httpResource()` Features (continued)

Process error information

```
error = this.ps.productsResource.error();
```

Reload the data when needed

```
this.productsResource.reload();
```

Retrieve other types of data: `arrayBuffer`, `blob`, `text`

```
imageResource = httpResource.blob(() => this.imageUrl);
```



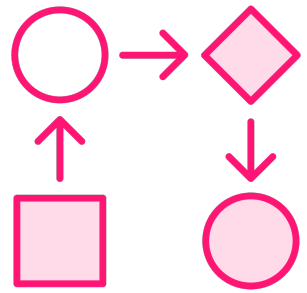
Key `httpResource()` Features (continued)

Use an options object

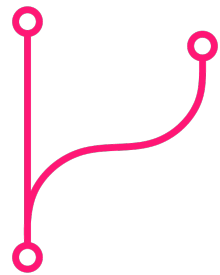
```
reviewsResource = httpResource<Review[]>(() => ({
  url: this.reviewsUrl,
  method: 'GET',
  headers: {
    accept: 'application/json'
  },
  params: {
    productId: `${this.ps.selectedProd()?.id ?? 0}`
  }
})))
```



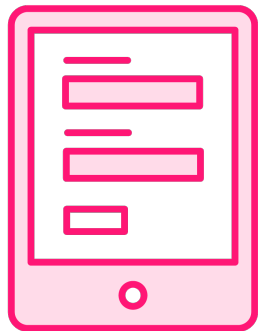
Signals Are Used in Many Other Angular Features



Component communication
Input and model input

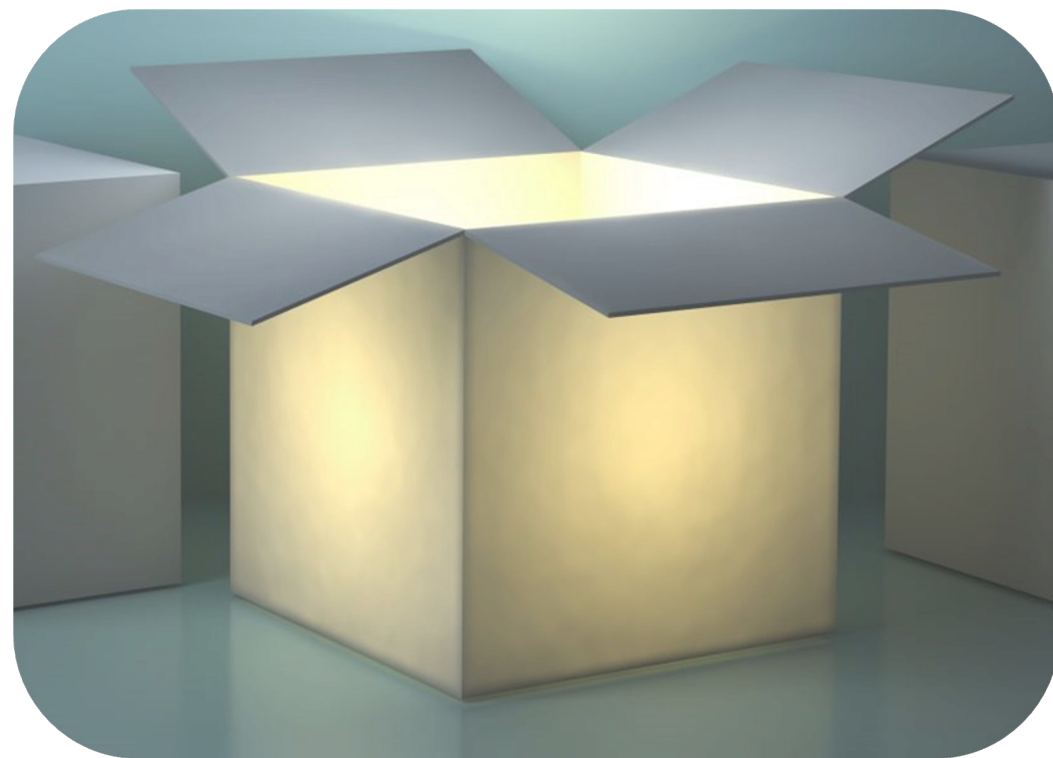


Angular routing



Angular forms





Start to migrate to signals

Use a `signal()` for any state (data) that could change

Use `computed()` for read only values that depend on other signals

Use `linkedSignal()` for writable values that can react to changes in a `source` signal and reset based on a `computation`

Use `httpResource()`

- To retrieve data into a signal
- To reactively re-get data when a signal changes



Thank you!

@deborahkurata | https://www.youtube.com/@deborah_kurata

