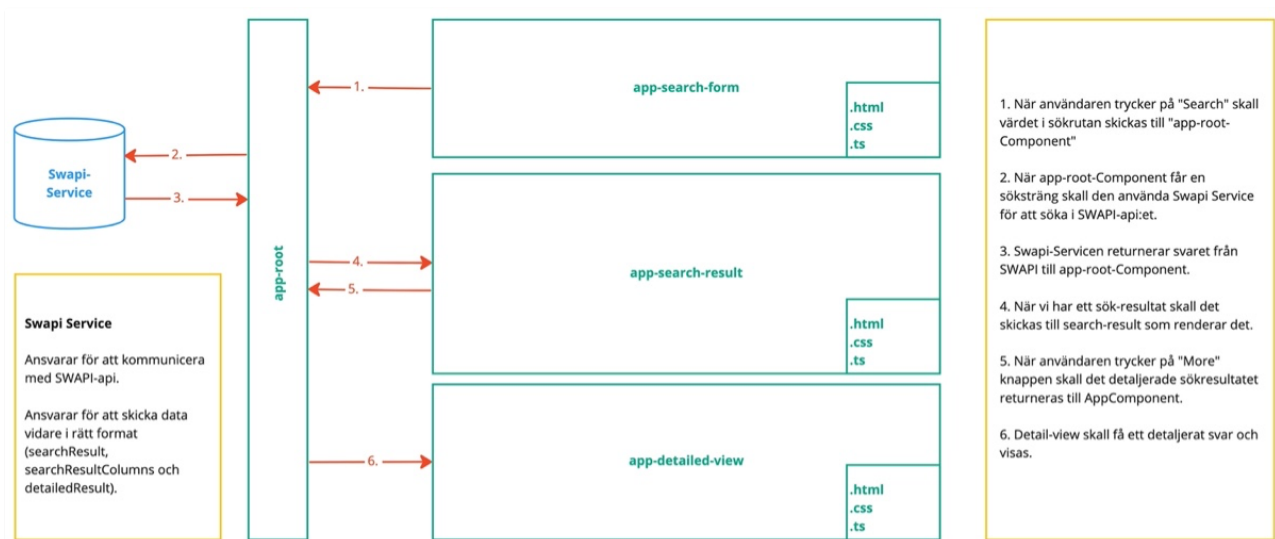


## Övningsuppgift 7

I denna övningsuppgiften skall vi försätta att arbeta med Star Wars Angular-appen.



Bilden visar hur strukturen av vår app skall se ut. I Övningsuppgift 5 skapade vi de gröna elementen som alla är Components. I denna övningen skall vi börja koppla ihop dem olika Components med varandra genom att använda Event- och Property Bindings.

I denna övning har vi ännu inte tittat på "Swapi Service" - den kommer vi till senare. Den kommer att kommunicera med SWAPI med hjälp av Ajax och konvertera data till ett format som våra komponenter vill ha.

### SearchFormComponent

När användaren trycker på "Search"-knappen skall sök-strängen och värdet från resurs-selecten skickas till app-root. Detta så att datan kan skickas vidare till "Swapi Service" för sökning i API:et. Vi saknar en del kunskaper för att göra allt detta. I nuläget nöjer vi oss med att sätta upp länken mellan komponenterna.

1. I search-form komponenten:
  1. Importera Output och EventEmitter
  2. Skapa ett nytt event dvs deklarerar ett nytt @Output()-attribut som du kallar för search - det skall vara en EventEmitter<[string, string]>.
  3. Skapa en ny metod searchBtnClick(). Metoden skall kopplas till "search"-knappen - och ett event skall emit:as. Eftersom vi ännu inte kan läsa från formuläret så skickar vi två hårdkodade strängar i eventet "söktext" och "kategori".
2. I app-component.ts
  1. Skapa en metod "onSearchEventReceived(searchTerms: [string, string])" - metoden skall i nuläget bara console.log:a de två strängarna.
3. I app-component.html

1. I taggen för search-form - koppla ihop (search) eventet med din onSearchEventReceived-metod.

Trycker du nu på din sök-knapp skall du få ut ett meddelande i konsollen.

## SearchResultComponent

I Övning 5 lät vi vår test-data vara hårdkodad i Component Class:en. När vi använder vår app så kommer den datan att komma från Swapi Service via app-root. Nu skall vi ge data till SearchResult component från app-root och få information om vilken "more" knapp som användaren har tryckt på.

### 1. I SearchResultComponent

1. Importera "Input, Output, EventEmitter"
2. Dekorera searchResult och searchResultColumns med @Input() - vi vill inte initialisera dessa med någon test-data — initiera dem med objekt med tomma attribut / tomma arrayer. (Spara din test data - du kommer att behöva den om en lite stund).
3. Skapa ett nytt event dvs deklarerar ett nytt @Output()-attribut som du kallar för selected - det skall vara en EventEmitter<number>.
4. Skapa en ny metod moreBtnClick(index:number).

### 2. I SearchResultComponent - html-mall

1. Uppdatera din ngFor - så att du får rad-nummer (index)  
**Spoiler:** "let result of searchResult; let i=index"
2. I din ngFor skapar du din "More" knapp. Koppla dess (click) till moreBtnClick(i) - i är index:et från ngFor.

### 3. I app-component.ts

1. Definiera searchResult och searchResultColumns - på samma sätt som du gjorde i SearchComponent (dem kan vara tomma).
2. Skapa en metod som heter onSearchResultReceived() - denna metoden skall populera searchResult och searchResultColumns i app-component med din test-data. (Detta kommer senare att komma ifrån Swapi Service).
3. Skapa en metod som heter onMoreResultChosen(index: number). Den kan för tillfället skriva ut index till konsollen. Den kommer senare att skicka den detaljerade datan till detail view Component.

### 4. I app-component.html

1. Uppdatera taggen för search-result - koppla searchResult och searchResultColumn till sina motsvarigheter i search-result-component med hjälp av **property binding**.
  2. Uppdatera taggen för search-result - koppla ihop (selected) eventet med din onMoreResultChosen()-metod.
5. Om du kör din app nu kommer du inte att se något resultat - detta eftersom vi inte fått något sök resultat ännu. Om uppdaterar din "onSearchEventReceived()" i app-component.ts så att den anropar this.onSearchResultReceived() - så kommer du att se sök resultat om du trycker på din sök knapp.

När du kör din app skall du fortfarande se din test-data — men nu finns den inte statistiskt i SearchResult-längre. Du kan också trycka på more-knappen och genom att titta i konsollen kan du se vilket resultat du valde (indexet av resultatet).

I denna övningen har vi implementerat 1, 4 och 5 i skissen över hur appen skall fungera. Detta har vi gjort genom att använda property- och event binding.