## Projelerde Kullanılacak Bazı Kütüphaneler

Bu bölümde projelerde yaygın olarak kullanılan bazı kütüphaneler ve kullanım amaçlarına dair özet bilgiler yer almaktadır.

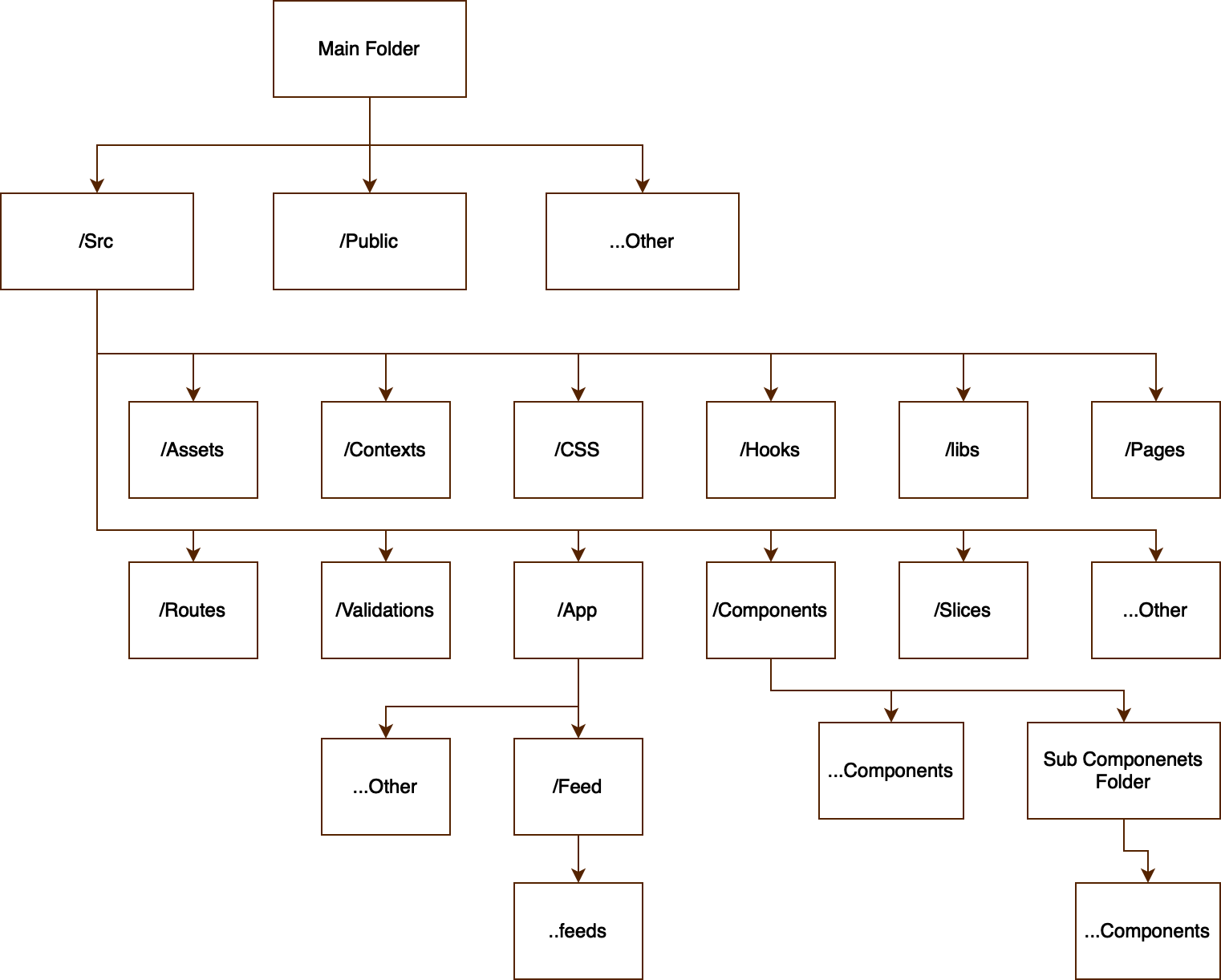
|  |  |
| --- | --- |
| **Kütüphane/Framework Adı** | **Kullanım Amacı** |
| react | Şirketteki tüm frontend projelerinde kullanılan UI kütüphanesi |
| react-router-dom | Projelerdeki routing işlemleri için |
| react-redux | Global state management için kullanılan kütüphane |
| @reduxjs/toolkit | Global state management kullanılan yardımcı kütüphane |
| @tanstack/react-query | HTTP isteklerini yönetmek için |
| react-data-table— component | Veri tabloları için |
| formik | Güncel projelerde form işlemleri için |
| axios | HTTP isteklerini yönetmek için |
| @mui/material | Hazır React componentleri sunmaktadır. |
| @radix-ui | Hazır React componentleri sunmaktadır. |
| tailwindcss | CSS kütüphanesi |
| react-icons | Birçok ikon kütüphanesini bir araya getirmiş olan react kütüphanesi |

## Projenin mimarisi

Bu bölümde projelerde yaygın olarak kullanılan mimari yapıya ait diagrama yer verilmiştir.

## Projelerdeki dosya yapısı ve amaçları ile ilgili özet bilgi

Bu bölümde projelerde genel olarak kullanılan dosya yapısı ve klasörler hakkında bilgi verilmiştir.



Klasör Görevleri

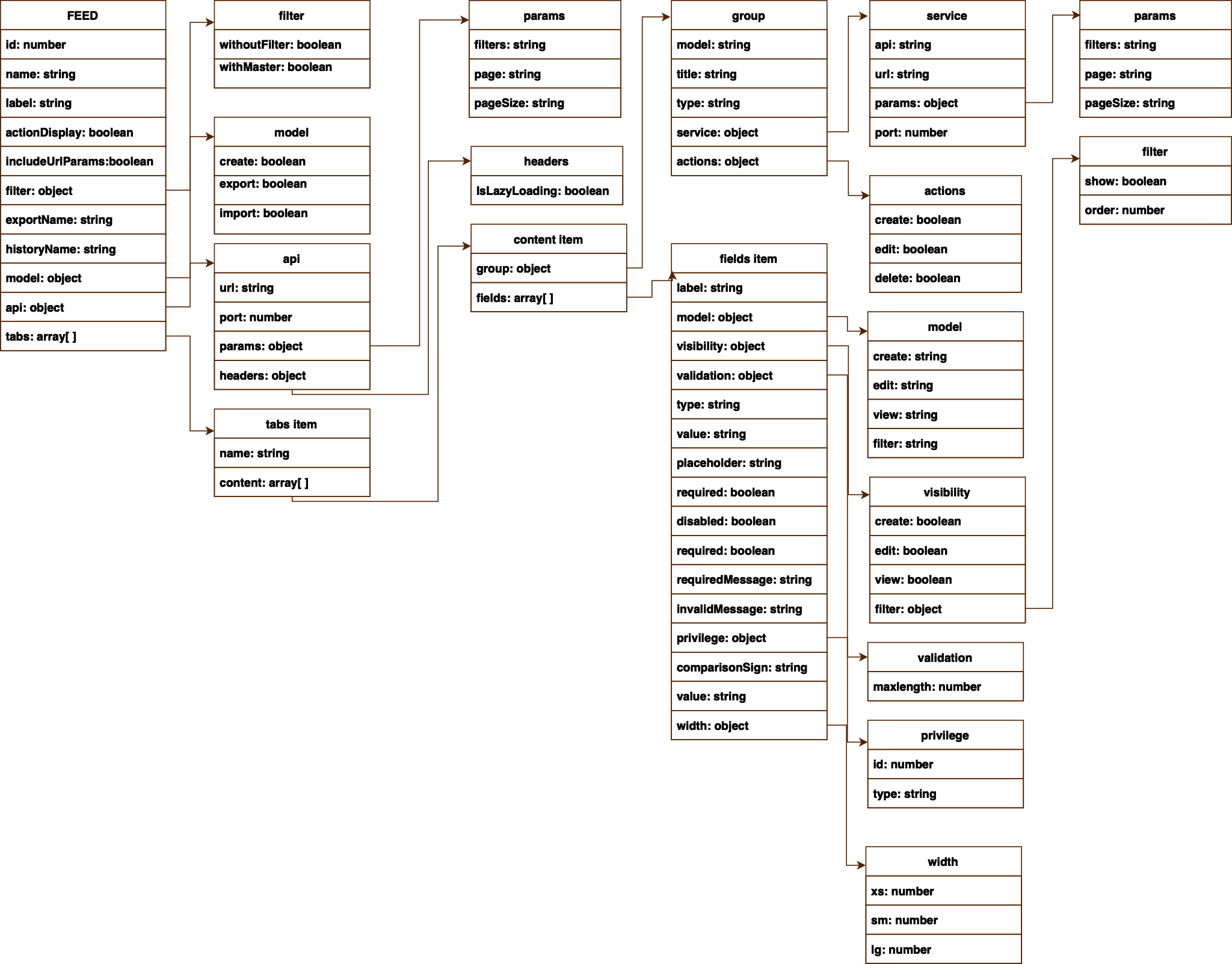
**Klasör Adı**

|  |  |
| --- | --- |
| **Klasör Adı** | **Görev** |
| Main Folder | Projenin tüm dosyalarını barındıran ana dizin. |
| Src | Kaynak kodlarının bulunduğu source dizini. |
| Public | Favicon, index html benzeri public dosyaların bulunduğu dizin. |
| Assets | Projelerde kullanılan assetlerin tutulduğu dizin. |

|  |  |
| --- | --- |
| **Klasör Adı** | **Görev** |
| Contexts | Context api ile oluşturulan providerlerın tutulduğu dizin. |
| Css | index.css ve component/sayfa bazlı cıs dosyalarının bulunduğu dizin. |
| Hooks | Custom olarak yazılan hookların bulunduğu dizin. |
| Libs | Projede kullanılan yardımcı fonksiyonların bulunduğu dizin. |
| Pages | Bir ekranın tamamına denk gelen componentlerin bulunduğu dizin. Örneğin login ekranı. |
| Routes | Projelerde bulunan router tanımlamalarının bulunduğu dizin. |
| Validations | Form validasyonlarının bulunduğu dizin. |
| App | Projeye dair tanımlamaların bulunduğu dizin. Örnek olarak Constant, Enum, Api gibi. |
| Components | Componentlerin tutulduğu dizin. |
| Slices | Global storage tanımlamalarının bulunduğu dizin. |
| Feed | Json formatında bulunana birden fazla feed dosyasının tutulduğu dizin. |

## Temel Feed yapısı

Bu bölümde temel feed yapısı ve sık kullanılan key’ler hakkında bilgi verilmiştir.



Feed bazı keyler ve işlevleri

**Key**

|  |  |
| --- | --- |
| **Key** | **İşlev** |
| name | Sayfanın Route tanımıdır ayrıca uıniqe olmak zorundadır |
| label | Sayfanın ekranda görünecek olan adı |
| actionDisplay | Data table’da backend’den gelen EntityPermissionsa göre işlemler kolunun gösterilmesi |
| includeUrlParams | Ekran açıldığında ilk atılan sorguda url’deki parametlerin dahil edilmesi |
| Filter | Ekran açıldığında default atılacak olan sorgu için tanımlamalar |

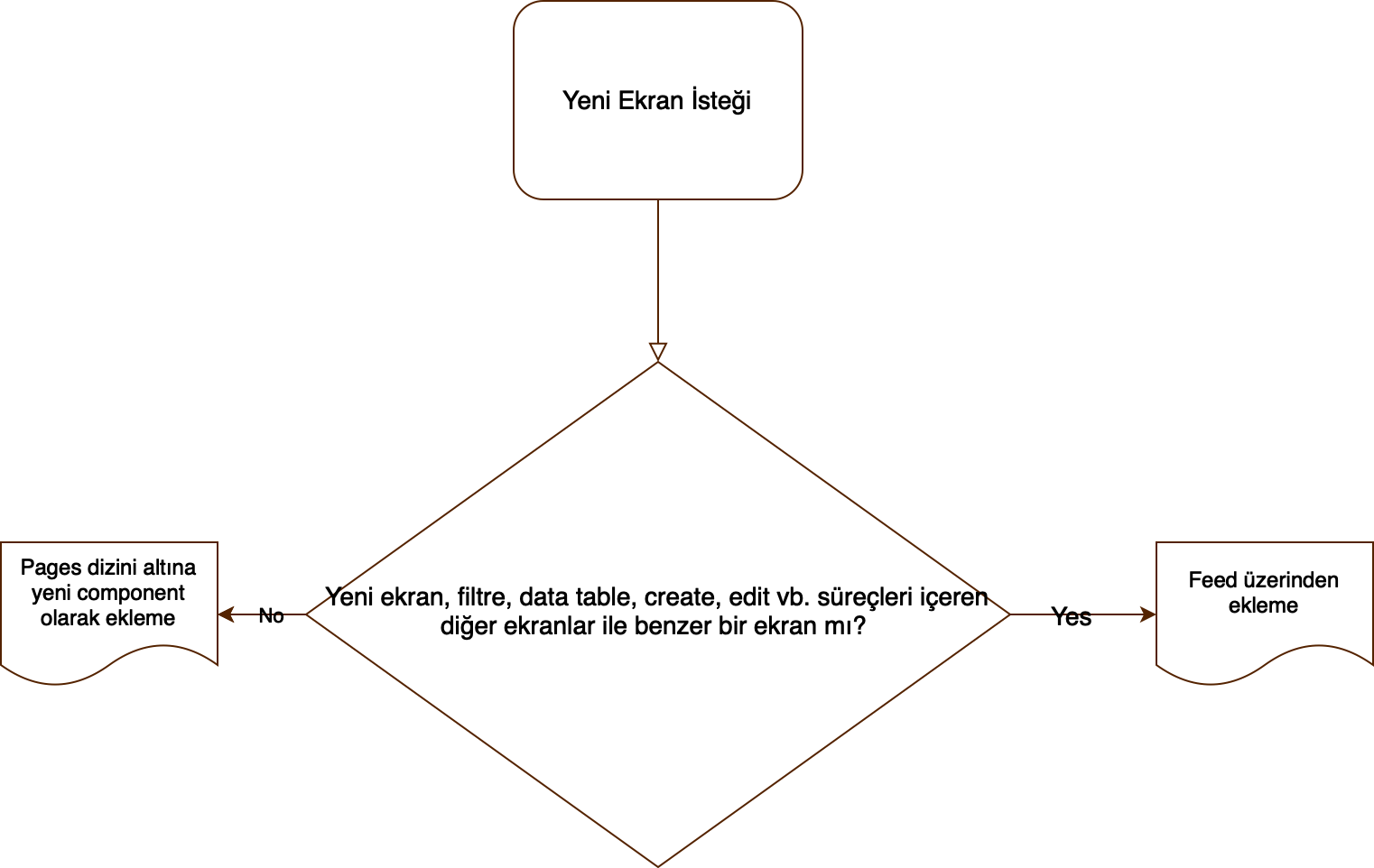
|  |  |
| --- | --- |
| **Key** | **İşlev** |
| exportName | AssetIndex’te iken ilgili ekrandadndaki verilerin export edilmesi durumu için |
| historyName | Tarihçe için gidilmesi gereken adres tanımı |
| Model | Ekranın AssetEdit,Create ve View durumu için tanımlamalar |
| api | Api bilgilerinin tanımlanması için |
| tabs | Her ekran en az bir tab’dan oluşmaktadır, daha fazla varsa array içerisine eklenir. |
| withoutFilter | Index modunda iken filtreleme işlemlerinde filtre zorunluluğu durumu, true ise ekran ilk açıldığında filtre sorgusu atar |
| withMaster | Index modunda Sigorta şirketi filtre zorunluluğu durumu, false ise master Identifier ile istek atılabilir |
| Model->Create | Create Ekranına geçiş izni, false ise buton görünmez |
| Model->Export | Index modunda ekrandan rapor alma |
| Model->Import | Index modunda excel aktarma durumu, true ise excel import işlemi yapılabilir |
| Api->URL | Endpoint tanımı |
| Api->Port | Varsa endpointin çalıştığı port |
| Api->Params | Filtre sorgularına default eklenecek parametreler |
| Api-Headers | Sorgularda readers içerisine eklenmek istenen parametreler |
| Tabs->Name | Tab’lı yapıda Tab ismi, Tab adı “Genel” ise ekranda görünmez, diğer isimler görünür. |
| Tabs->Content | Tab içeriğinin tutulduğu obje |
| Params->Filters | Filtre sorgularında Parametreye eklenecek filtreler |
| Params->page | Filtre sorgularında page değeri |
| Params->pageSize | Filtre sorgularında default pageSize |
| Headers->isLazyLoadig | Filtre sorgularında isLazyLoading değişken durumu |
| Content->Group | Bir Tab içerisinde birden fazla content olabilir. Her content için varsa eğer özel Group tanımlamaları yapılır, örneğin aynı ekranda bir grup için farklı bir endpointe istek atmak yada grup ismi belirlemek gibi. Spesifik bir durum yoksa null olarak eklenir |
| Content->Fields | Eklenecek fieldların tutulduğu yer, Dizi olarak tutar. |

|  |  |
| --- | --- |
| **Key** | **İşlev** |
| Group->model | Gösterilcek yada düzenlenecek veriler nested bir object içinde ise tanımlanır. Örneğin body:{  ornekNestedObject:{ Deger1:null, Deger2:null  }  }  gibi, bu durumda model ismi ornekNestedObject olur, filedlar da bu obje altındaki değerler ile eşleşir |
| Group->title | Grubun gösterilecek ismi |
| Group->Type | Grup tipi, tablo componenti yada form tipinde olabilir. |
| Group->Service | Varsa eğer gruba özel servis tanımının yapıldığı yer, yapılmazsa parenttan beslenir. |
| Group->Actions | Grubun create, edit, delete gibi işlemleri için izin tanımlaması |
| Filed->Label | Ekranda gösterilecek Field ismi |
| Filed->Model | Filed’ın create, edit, view gibi durumlarda denk geldiği değer adının tanımlandığı yer |
| Filed->visibility | Field’ın create, edit, view gibi durumlarda görünüp görünmemesi durumu. Örnek olarak bir Field sadece create ekranında görünmesi isteniyorsa create değeri true olur, diğer değerler false olarak tanımlanır. |
| Filed->validation | Validasyon tanımlamalar |
| Filed->type | Field Tipi, örnek olarak, selecet, text, checkbox gibi. |
| Field->value | Varsa eğer field’ın default değer tanımı |
| Field->placeHolder | Field’a ati place holder değeri |
| Field->required | Field'ın zorunlu olma durumu |
| Field->disabled | Field'ın default olarak disable olma durumu |
| Field->requiredMessage | Zorunlu olduğu durumlarda doldurulmadığında gösterilecek mesaj |
| Field->invalidMessage | Geçersiz değer durumlarında gösterilecek olan mesaj |
| Field->privilege | Filed’a ait rol tanımı. |
| Field->comparisionSign | Field filtrede tanımlı bir field ise eğer filtreleme durumunda sorguya eklenecek değer ile karşılaştırma için eklenen işaret. Örneğin değer  == 0 yada değer >= 0 gibi |
| Field->width | Field'ın ekrandaki width değer tanımlaması |
| Field->Model->create | Create ekranında field’ın body içerisinde denk geleceği key tanımı |
| Field->Model->edit | Edit ekranında field’ın body içerisinde denk geleceği key tanımı |
| Field->Model->view | View datatable gösterilecek olan değerin key tanımı |
| Field->Model->filter | Index’te iken filtreleme durumunda field’ın denk geleceği key |

|  |  |
| --- | --- |
| **Key** | **İşlev** |
| Field->Visibility->Create | Create ekranında gösterilmesi izni |
| Field->Visibility->Edit | Edit ekranında gösterilmesi izni |
| Field->Visibility->View | Index ekranında gösterilmesi izni |
| Field->Visibility->Filter | Index ekranında filtrede gösterilmesi tanımı |
| Field->Visibility->Filter->Show | Index ekranında filtrede gösterilmesi izni |
| Field->Visibility->Filter->Order | Index ekranında filtrede gösterileceği konum |
| Field->Validation->maxLength | Maksimum uzunluk validasyonu |
| Field->Privilege->Id | Tanımlanacak olan Role ait Id değeri |
| Field->Privilege->Type | Role dair izin, örneğin field’ın gösterilmesi yada düzenlenebilmesi gibi |
| Field->Width->xs|sm|lg | Responsive durumlarda field’a özel width tanımı |
| Group->Service->Url | Gruba özel servis varsa endpointi tanımlaması |
| Group->Service->params | Ekran açıldığda atılacak sorgunun alacağı değer, Örneğin InsuredId değeri yada ContactId değeri gibi denk geleceği değer tanımı |
| Group->Service->port | Servis port tanımlaması |
| Group->Actions->create | Gruba ait create izni |
| Group->Actions->edit | Gruba ait edit izni |
| Group->Actions->delete | Gruba ait delete izni |
| Group->Service->Params->filter | Group->Service->Params’da açıklanan filter tanımı |
| Group->Service->Params->page | Default sorguya eklecek page değeri |
| Group->Service->Params-  >pageSize | Default sorguya eklecek pageSize değeri |

# Sıfırdan bir ekran nasıl eklenir?

Projelerde yaygın kullanılan mimariye göre yeni bir ekran eklerken aşağıdaki gibi bir karar mekanizması olur.



Ekran klasik feed yapısına uygunsa feed dizini altına yeni bir json uzantılı dosya olarak eklenir ve yukarıda anlatılan feed özelliklerine göre düzenlenir. Son olarak ekran sidebar’da görüntülenecek bir ekran ise sidebar.js üzerine id’si yazılarak eklenir. Ekran diğer feed’ler içersinde kullanılacak bir ekran ise yapıya uygun olacak şekilde entitiyName key’i ile referans verilerek tanımlanır.

Ekran feed yapısına uygun olmayan spesifik özellikler barındıran bir ekran olacaksa eğer, pages dizini altında yeni bir component oluşturularak eklenir. Eklenen ekran routes altında handle edilerek kullanılabilir.