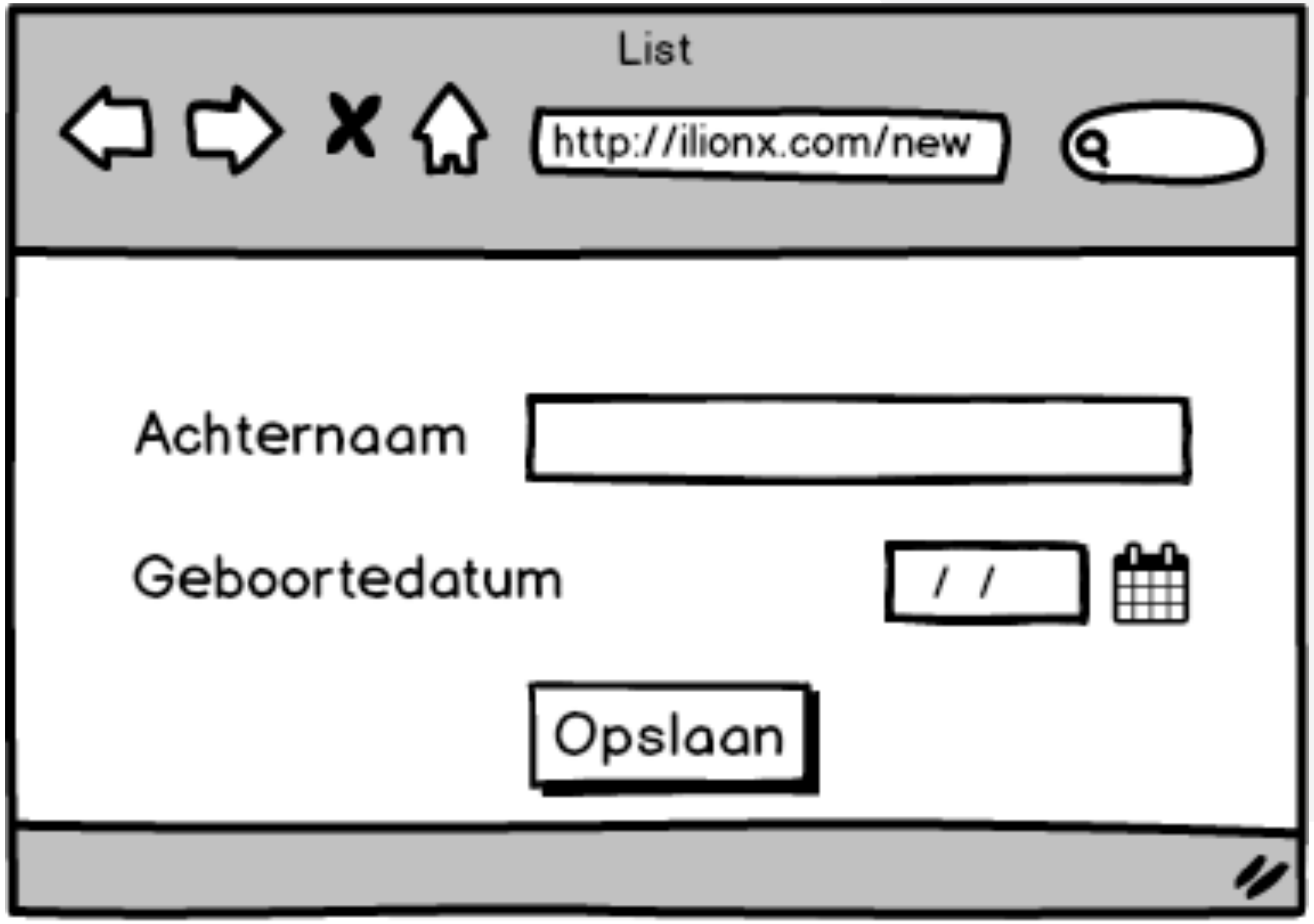
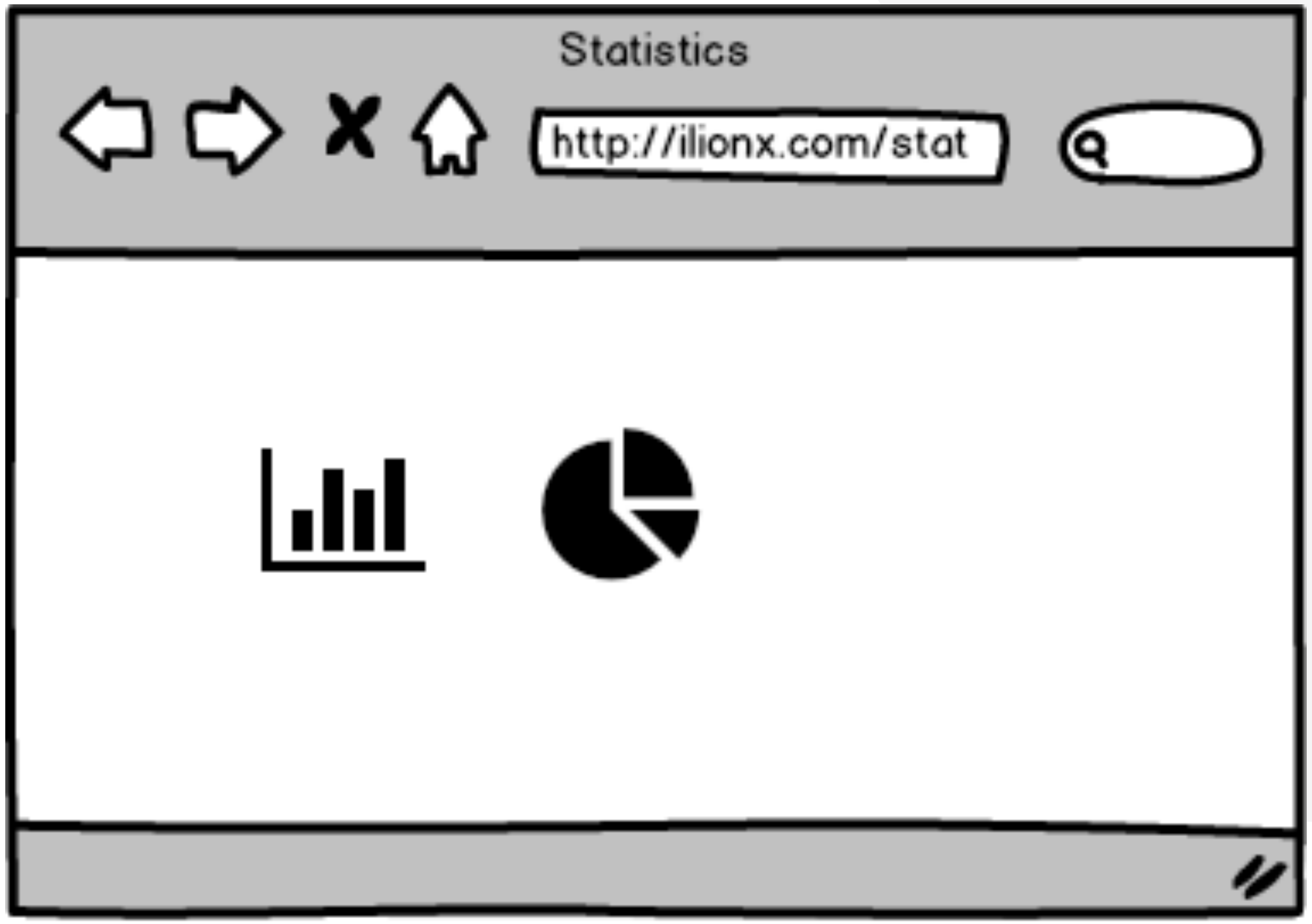
Routing

Routing

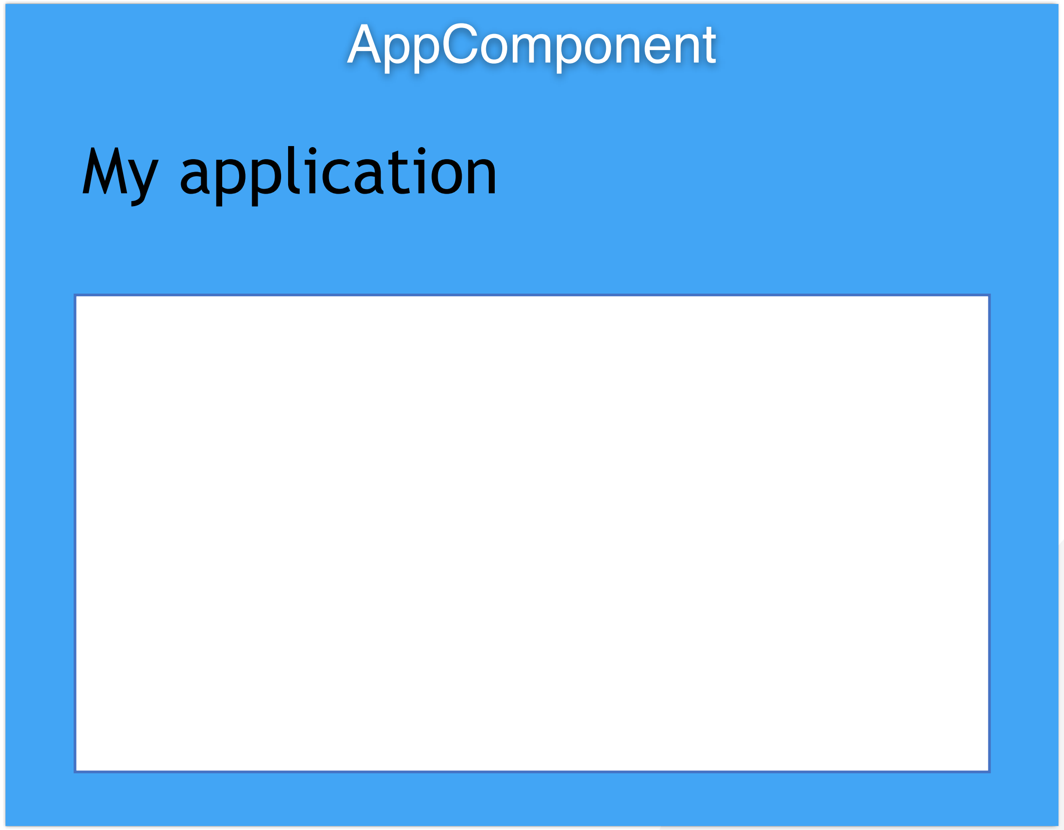

List
http://ilionx.com/new
Achternaam
Geboortedatum
/ /
Opslaan

Routing

www.ilionx.com/statistics



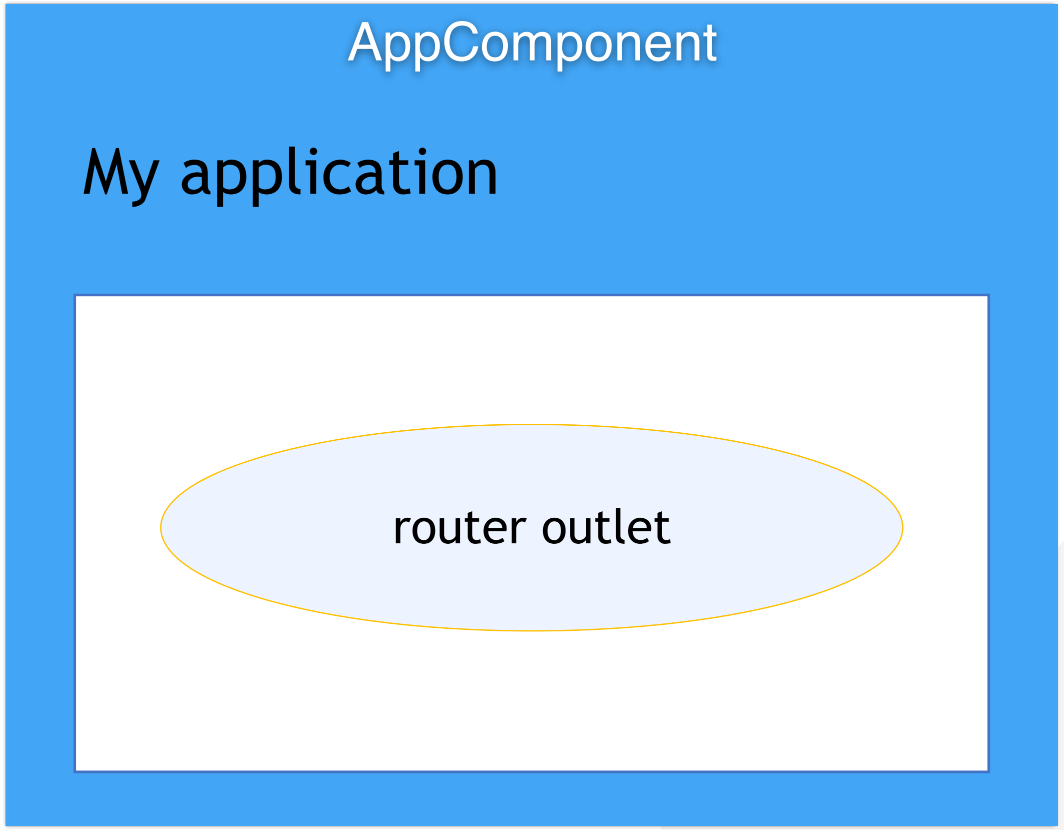
Router outlet



```
<div id="app-component">

  <h1>My application</h1>



</div>
```

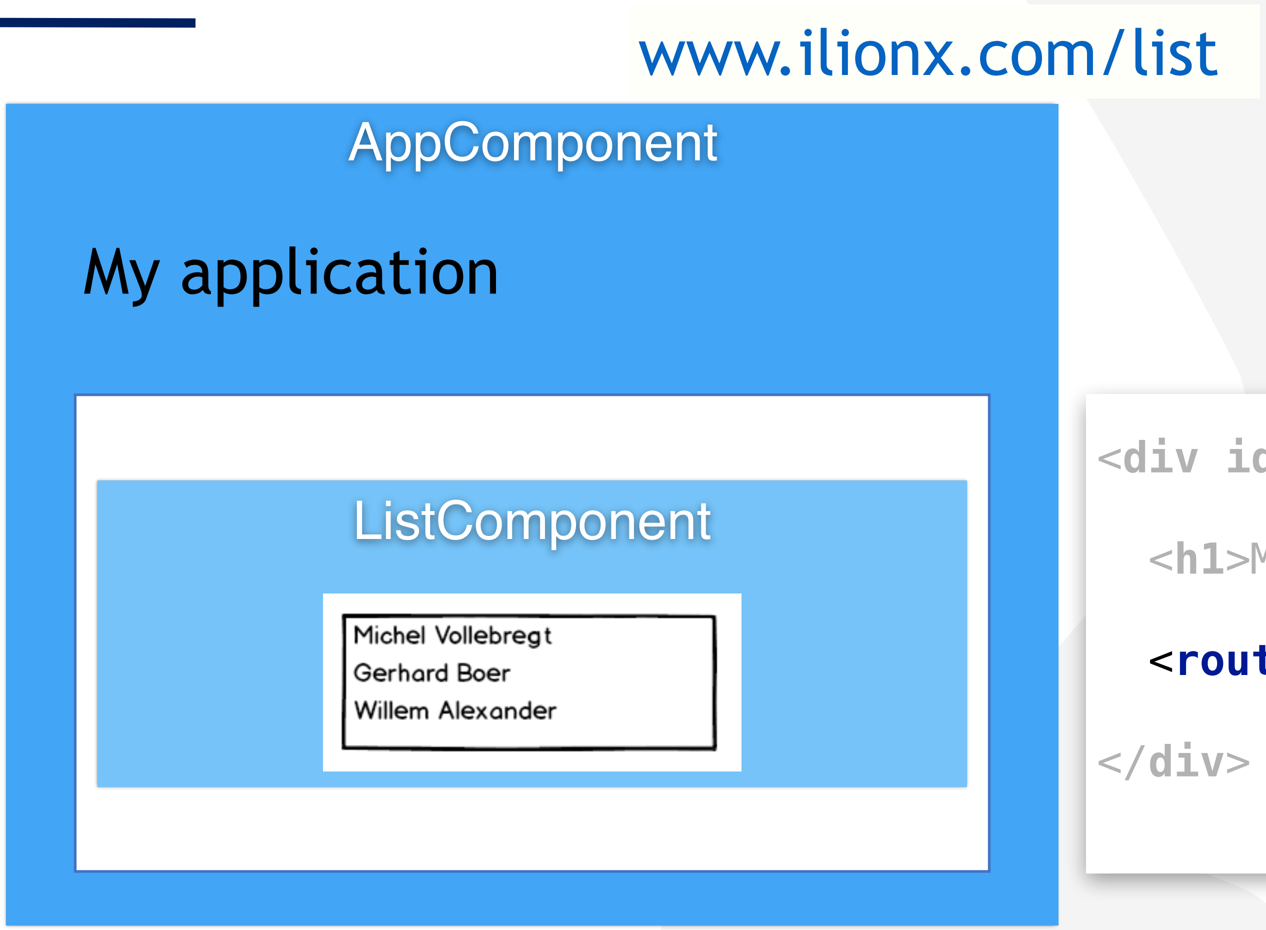
ilionx
Your partner in digital business

Router outlet



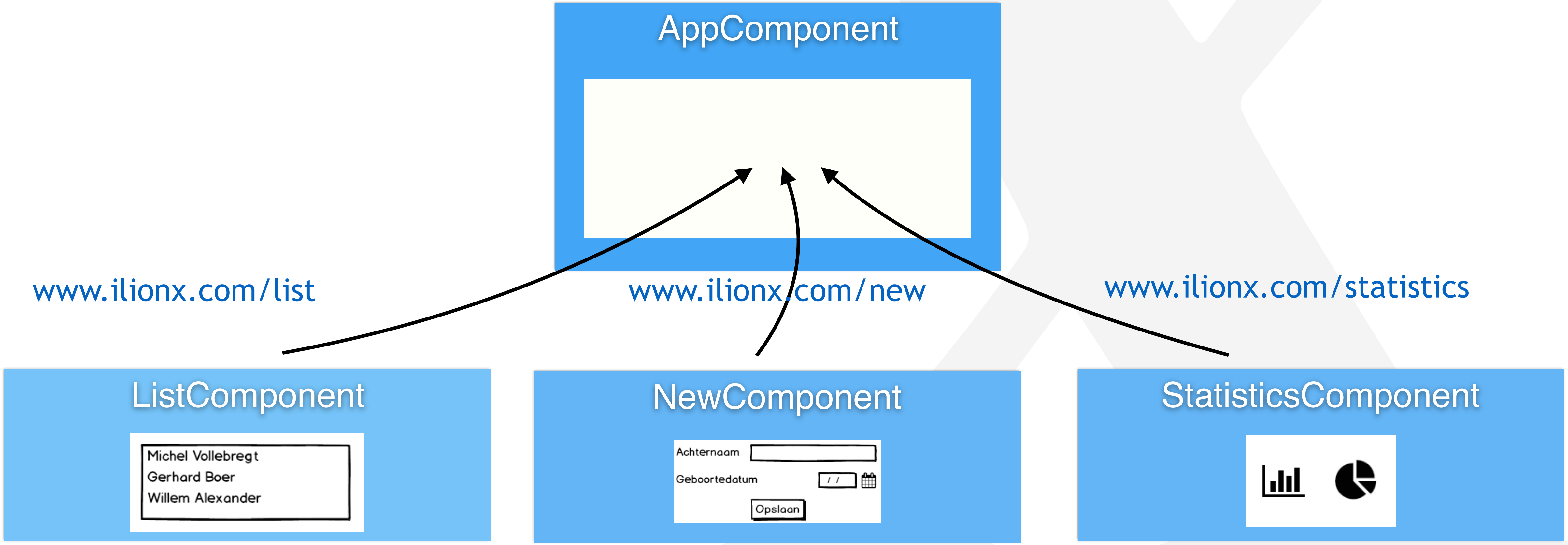
```
<div id="app-component">
  <h1>My application</h1>
  <router-outlet></router-outlet>
</div>
```

Router outlet



```
<div id="app-component">

  <h1>My application</h1>

  <router-outlet></router-outlet>

</div>
```

Router outlet



Router outlet

- Placeholder for component
- Content dependant on URL
- (activate) / (deactivate)

Configuring routes

Define array of Routes

app-routing.module.ts

```
const APP_ROUTES : Route[] = [




];
```

ilionx
Your partner in digital business

Configuring routes

Define array of Routes

app-routing.module.ts

```
const APP_ROUTES : Route[] = [



    { path: 'list',       component: ListComponent },
    { path: 'new',        component: NewComponent },
    { path: 'statistics', component: StatisticsComponent }

];
```

Configuring routes - redirect

Redirect from empty path, with pathMatch

app-routing.module.ts

```
const APP_ROUTES : Route[] = [

    { path: '',           redirectTo: 'list', pathMatch: 'full' },

    { path: 'list',       component: ListComponent },
    { path: 'new',        component: NewComponent },
    { path: 'statistics', component: StatisticsComponent }

];
```

ilionx
Your partner in digital business

Configuring routes - 404

404, '**' needs to be last route

app-routing.module.ts

```
const APP_ROUTES : Route[] = [

    { path: '',            redirectTo: 'list', pathMatch: 'full' },

    { path: 'list',        component: ListComponent },
    { path: 'new',         component: NewComponent },
    { path: 'statistics', component: StatisticsComponent },

    { path: '**',          component: NotFoundComponent }
];
```

ilionx
Your partner in digital business

Configuring routes

Export AppRoutingModule

app-routing.module.ts

```
const APP_ROUTES : Route[] = [

    { path: '',            redirectTo: 'list', pathMatch: 'full' },

    { path: 'list',        component: ListComponent },
    { path: 'new',         component: NewComponent },
    { path: 'statistics', component: StatisticsComponent }

];

export const AppRoutingModule = RouterModule.forRoot(APP_ROUTES);
```

ilionx
Your partner in digital business

Configuring routes

Import AppRoutingModule in AppModule

app.module.ts

```
@NgModule({
   declarations: [ ContactsComponent ],
   imports: [ ],
   exports: []
})
export class AppModule {


}
```

ilionx
Your partner in digital business

Configuring routes

Import AppRoutingModule in AppModule

app.module.ts

```
@NgModule({
   declarations: [ ContactsComponent ],
   imports: [ AppRoutingModule ],
   exports: []
})
export class AppModule {


}
```

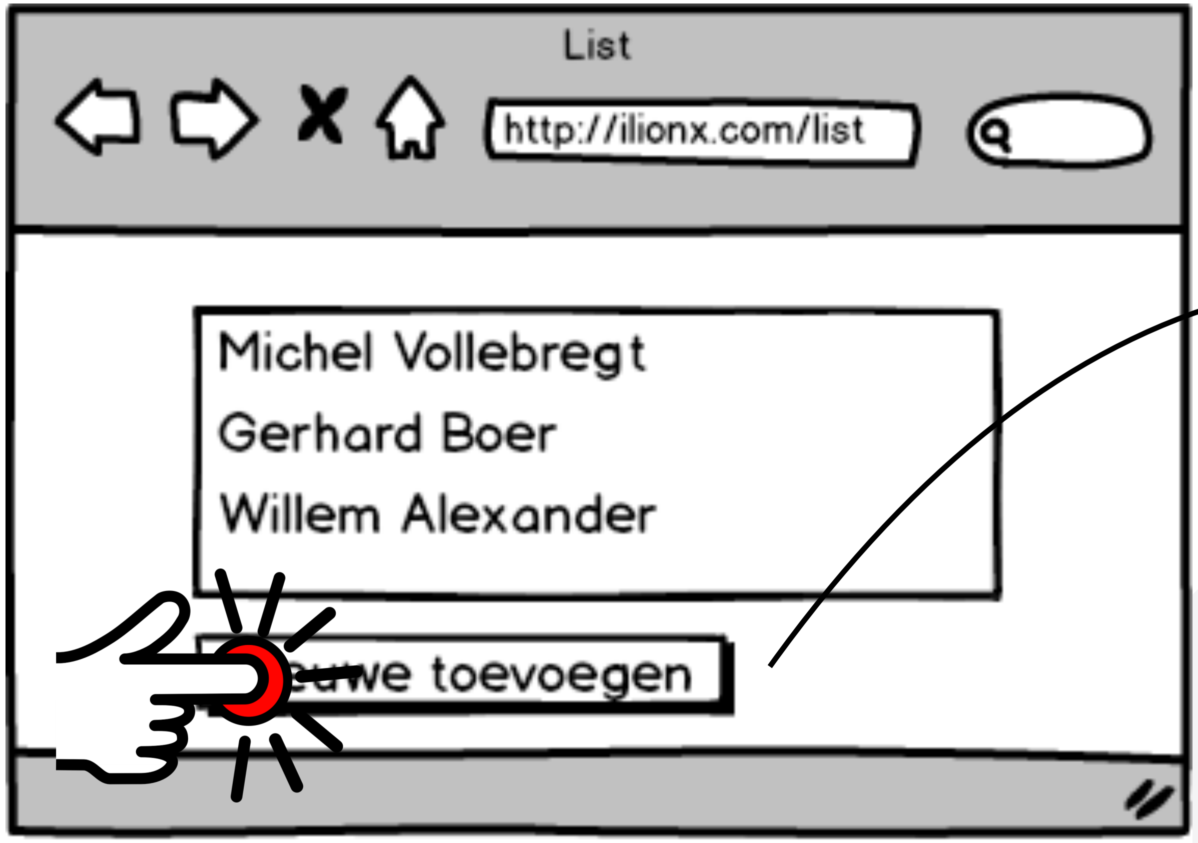
Recap

- `<router-outlet>`
- `app-routes.module.ts`
- `RouterModule.forRoot()`

Navigating through the application

- routerLink directive
- Router service

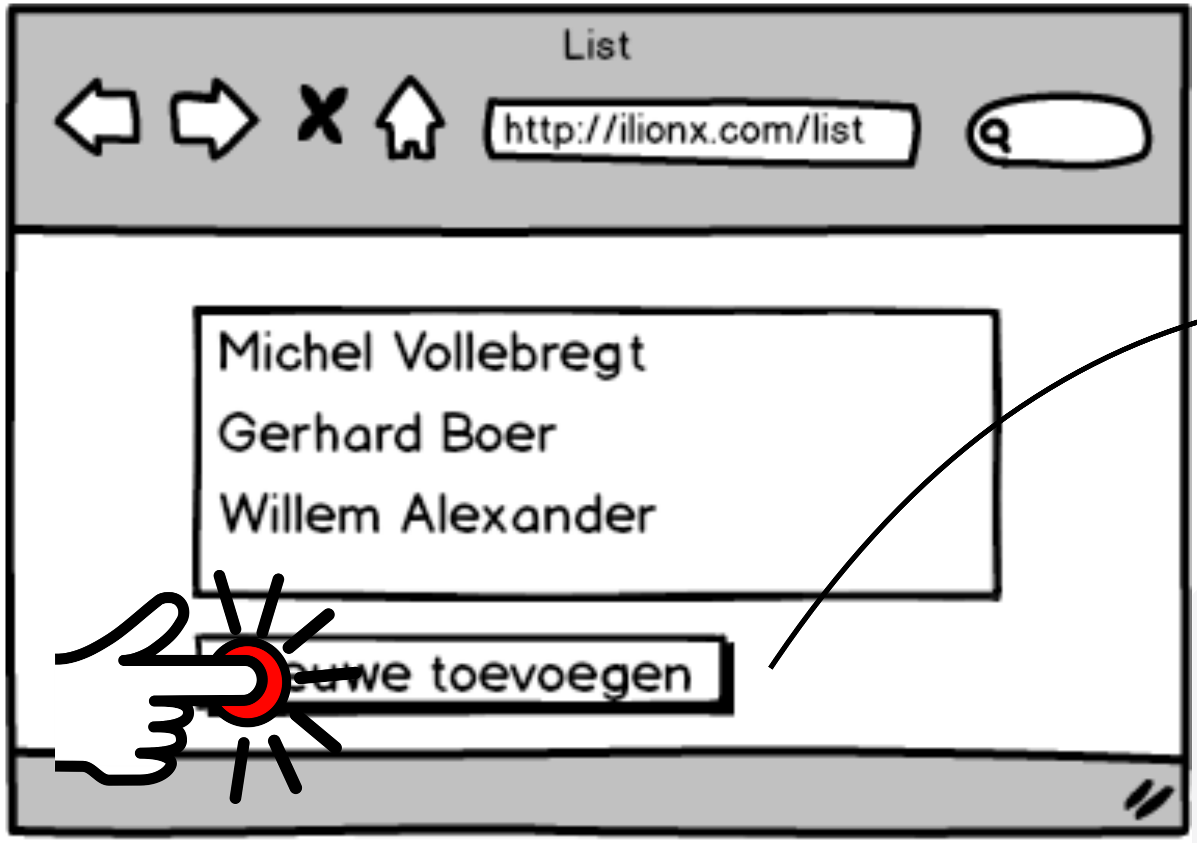
routerLink directive



Created with Balsamiq - www.balsamiq.com

a href=?

routerLink directive



Created with Balsamiq - www.balsamiq.com

```
<div ...>

  <a [routerLink]="['/new']">
    Add
  </a>

</div>
```

ilionx
Your partner in digital business

Router link

RouterLink points to the path in Routes[]

```
{ path: 'list',       component: ListComponent },
{ path: 'new',        component: NewComponent },
{ path: 'statistics', component: StatisticsComponent }
```

```
<div ...>

  <a [routerLink]=“[‘/new’]”>
    Add
  </a>

</div>
```

Router service

- Go to new URL from TypeScript
- Navigating based on backend response
- Redirect from Guards

Router service

Navigate imperatively

```
@Component({ selector: 'contacts' })
export class ContactsComponent implements OnInit {

  constructor(private router: Router) {}

  ngOnInit() {
      this.contactsService.fetchContact()
      .subscribe(
        (contact) => {
          this.router.navigate(['/contact', contact.name])
        }
      )
  }
```

ilionx
Your partner in digital business

Router service

Navigate imperatively

```
@Component({ selector: 'contacts' })
export class ContactsComponent implements OnInit {

  constructor(private router: Router) {}

  ngOnInit() {
      this.contactsService.fetchContact()
      .subscribe(
        (contact) => {
          this.router.navigate(['/contact', contact.name])
        }
      )
  }
```

Route parameters

```
const APP_ROUTES : Route[] = [
    { path: 'contact/:id', component: ContactComponent }
];
```

Your partner in digital business

Getting route parameters

3 examples

route.component.ts

```
constructor(private activatedRoute: ActivatedRoute) { }

ngOnInit() {
  let id:string = this.activatedRoute.snapshot.paramMap.get('id');
  this.handleRoute(id);

  this.activatedRoute.paramMap
    .subscribe((params: ParamMap) => this.handleRoute(params.get('id'));

  this.activatedRoute.paramMap.pipe(
      map((params: ParamMap) => params.get('id'))
    ).subscribe((id: string) => this.handleRoute(id);
}
```

ilionx
Your partner in digital business

Getting route parameters

Inject ActivatedRoute

route.component.ts

```
constructor(private activatedRoute: ActivatedRoute) { }

ngOnInit() {
  let id:string = this.activatedRoute.snapshot.paramMap.get('id');
  this.handleRoute(id);

  this.activatedRoute.paramMap
    .subscribe((params: ParamMap) => this.handleRoute(params.get('id'));

  this.activatedRoute.paramMap.pipe(
      map((params: ParamMap) => params.get('id'))
    ).subscribe((id: string) => this.handleRoute(id);
}
```

ilionx
Your partner in digital business

Getting route parameters

Snapshot gives you the value at that time

route.component.ts

```
constructor(private activatedRoute: ActivatedRoute) { }

ngOnInit() {
  let id:string = this.activatedRoute.snapshot.paramMap.get('id');
  this.handleRoute(id);

  this.activatedRoute.paramMap
    .subscribe((params: ParamMap) => this.handleRoute(params.get('id'));

  this.activatedRoute.paramMap.pipe(
      map((params: ParamMap) => params.get('id'))
    ).subscribe((id: string) => this.handleRoute(id);
}
```

Getting route parameters

Listen for changes with Observables

route.component.ts

```
constructor(private activatedRoute: ActivatedRoute) { }

ngOnInit() {
  let id:string = this.activatedRoute.snapshot.paramMap.get('id');
  this.handleRoute(id);

  this.activatedRoute.paramMap
    .subscribe((params: ParamMap) => this.handleRoute(params.get('id'));

  this.activatedRoute.paramMap.pipe(
      map((params: ParamMap) => params.get('id'))
    ).subscribe((id: string) => this.handleRoute(id);
}
```

ilionx
Your partner in digital business

Getting route parameters

Using Observable operators

route.component.ts

```
constructor(private activatedRoute: ActivatedRoute) { }

ngOnInit() {
  let id:string = this.activatedRoute.snapshot.paramMap.get('id');
  this.handleRoute(id);

  this.activatedRoute.paramMap
    .subscribe((params: ParamMap) => this.handleRoute(params.get('id'));

  this.activatedRoute.paramMap.pipe(
      map((params: ParamMap) => params.get('id'))
    ).subscribe((id: string) => this.handleRoute(id);
}
```

ilionx
Your partner in digital business

Recap

- :param
- ActivatedRoute
- paramMap / queryMap

Child Routes

- Combining routes
- Feature modules with own Routes
- Lazy loading

Child Routes

Combining routes

```
const APP_ROUTES : Route[] = [
    { path: '',               redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',           component: ListComponent },
    { path: 'new',            component: NewComponent },

    { path: 'statistics', component: StatisticsComponent }




];
```

Your partner in digital business

Child Routes

Combining routes

```
const APP_ROUTES : Route[] = [
    { path: '',           redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',       component: ListComponent },
    { path: 'new',        component: NewComponent },

    { path: 'statistics', component: StatisticsComponent }

    { path: ‘statistics/bar', component: BarComponent }
    { path: ‘statistics/pie‘, component: PieComponent }
    { path: ‘statistics/line', component: LineComponent }
    { path: ‘statistics/column', component: ColumnComponent }
    { path: ‘statistics/scatter', component: ScatterComponent }
];
```

Child Routes

Combining routes

```
const APP_ROUTES : Route[] = [
    { path: '',             redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',         component: ListComponent },
    { path: 'new',          component: NewComponent },

    { path: 'statistics', children: [
        { path: ‘’, component: StatisticsComponent }
        { path: ‘bar', component: BarComponent }
        { path: ‘pie‘, component: PieComponent }
        { path: ‘line', component: LineComponent }
        { path: ‘column', component: ColumnComponent }
        { path: ‘scatter', component: ScatterComponent }
]];
```

Feature module routes

Split the route definition

```
const APP_ROUTES : Route[] = [
    { path: '',              redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',          component: ListComponent },
    { path: 'new',           component: NewComponent },
-------------------------------------------------------------------
    { path: 'statistics', children: [
        { path: ‘', component: StatisticsComponent }
        { path: ‘bar', component: BarComponent }
        { path: ‘pie‘, component: PieComponent }
        { path: ‘line', component: LineComponent }
        { path: ‘column', component: ColumnComponent }
        { path: ‘scatter', component: ScatterComponent }
]];
```

ilionx
Your partner in digital business

Feature module routes

Split the route definition

```
const APP_ROUTES : Route[] = [
    { path: '',               redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',           component: ListComponent },
    { path: 'new',            component: NewComponent }];
```

```
const STATISTICS_ROUTES : Route[] = [
    { path: 'statistics', children: [
        { path: '', component: StatisticsComponent }
        { path: 'bar', component: BarComponent }
        { path: 'pie', component: PieComponent }
        { path: 'line', component: LineComponent }
        { path: 'column', component: ColumnComponent }
        { path: 'scatter', component: ScatterComponent }
]];
```

ilionx
Your partner in digital business

Feature module routes

Now in its own file close to the module

statistics-routing.module.ts

```
const STATISTICS_ROUTES : Route[] = [
    { path: 'statistics', children: [
        { path: ‘’, component: StatisticsComponent }
        { path: ‘bar', component: BarComponent }
        { path: ‘pie‘, component: PieComponent }
        { path: ‘line', component: LineComponent }
        { path: ‘column', component: ColumnComponent }
        { path: ‘scatter', component: ScatterComponent }
]];
```

Feature module routes

Use RouterModule.forChild

statistics-routing.module.ts

```
const STATISTICS_ROUTES : Route[] = [
    { path: 'statistics', children: [
        { path: '', component: StatisticsComponent }
        { path: 'bar', component: BarComponent }
        { path: 'pie', component: PieComponent }
        { path: 'line', component: LineComponent }
        { path: 'column', component: ColumnComponent }
        { path: 'scatter', component: ScatterComponent }
]];

export const StatisticsRoutingModule =
                      RouterModule.forChild(STATISTICS_ROUTES);
```

ilionx
Your partner in digital business

Feature module routes

Import StatisticsRoutingModule in StatisticsModule

statistics.module.ts

```
@NgModule({
   declarations: [ StatisticsComponent ],
   imports: [ StatisticsRoutingModule ],
   exports: []
})
export class StatisticsModule {


}
```

ilionx
Your partner in digital business

Feature module routes

Import StatisticsModule in AppModule

app.module.ts

```
@NgModule({
   declarations: [ ContactsComponent ],
   imports: [ StatisticsModule, AppRoutingModule ],
   exports: []
})
export class AppModule {


}
```

ilionx
Your partner in digital business

Lazy loading

Using loadChildren

```
const APP_ROUTES : Route[] = [
    { path: '',             redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',         component: ListComponent }





];
```

```
const STATISTICS_ROUTES : Route[] = [
    { path: 'statistics', children: [
        { path: '', component: StatisticsComponent }
        { path: 'bar', component: BarComponent }
        { path: 'pie', component: PieComponent }
]];
```

Your partner in digital business

Lazy loading

Using loadChildren

```
const APP_ROUTES : Route[] = [
    { path: '',             redirectTo: 'list', pathMatch: 'full' },
    { path: 'list',         component: ListComponent }

    { path: 'statistics',
      loadChildren: ‘app/statistics/statistics.module#StatisticsModule'
    }
];
```

```
const STATISTICS_ROUTES : Route[] = [

        { path: ‘’, component: StatisticsComponent }
        { path: ‘bar', component: BarComponent }
        { path: ‘pie‘, component: PieComponent }
]
```

ilionx
Your partner in digital business

Lazy loading

Remove StatisticsModule in AppModule

app.module.ts

```
@NgModule({
   declarations: [ ContactsComponent ],
   imports: [ StatisticsModule, AppRoutingModule ],
   exports: []
})
export class AppModule {


}
```

ilionx
Your partner in digital business

Lazy loading

Remove StatisticsModule in AppModule

app.module.ts

```
@NgModule({
   declarations: [ ContactsComponent ],
   imports: [ AppRoutingModule ],
   exports: []
})
export class AppModule {


}
```

ilionx
Your partner in digital business

Recap

- Mapping URL - component
- router-outlet, routerLink, routerLinkActive
- Routing configurable per module
- Lazy loading

Code

Your partner in digital business