## Services the wrong way

Logging.service.ts

export class LoggingService{

    logStatus(status: string)

    {

        console.log('A server status changed, new status: ' + status);

    }

}

Account.component.ts

import { Component, EventEmitter, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css']

})

export class AccountComponent {

  @Input() account: {name: string, status: string};

  @Input() id: number;

  @Output() statusChanged = new EventEmitter<{id: number, newStatus: string}>();

  onSetTo(status: string) {

    this.statusChanged.emit({id: this.id, newStatus: status});

    let service = new LoggingService()

    service.logStatus(status);

  }

}

## Injecting the Logging Service into Angular

Account.component.ts

import { Component, EventEmitter, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  providers: [LoggingService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  @Output() statusChanged = new EventEmitter<{ id: number, newStatus: string }>();

  constructor(private loggingService: LoggingService) {

  }

  onSetTo(status: string) {

    this.statusChanged.emit({ id: this.id, newStatus: status });

    //let service = new LoggingService()

    //service.logStatus(status);

    this.loggingService.logStatus(status);

  }

}

Newaccount.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  providers: [LoggingService]

})

export class NewAccountComponent {

  @Output() accountAdded = new EventEmitter<{name: string, status: string}>();

  constructor(private loggingService: LoggingService){}

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountAdded.emit({

      name: accountName,

      status: accountStatus

    });

    this.loggingService.logStatus(accountStatus);

  }

}

## Creating a Data Service

Accounts.service.ts

export class AccountService {

    accounts = [

        {

          name: 'Master Account',

          status: 'active'

        },

        {

          name: 'Testaccount',

          status: 'inactive'

        },

        {

          name: 'Hidden Account',

          status: 'unknown'

        }

      ];

      addAccount(name: string, status: string){

        this.accounts.push({name: name, status: status});

      }

      updateStatus(id: number, status: string){

        this.accounts[id].status = status;

      }

}

App.component.ts

import { OnInit } from '@angular/core';

import { Component } from '@angular/core';

import { AccountService } from './accounts.service';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css'],

  providers: [AccountService]

})

export class AppComponent implements OnInit {

  accounts: { name: string, status: string }[] = [];

  constructor(private accountService: AccountService) {

  }

  ngOnInit() {

    this.accounts = this.accountService.accounts

  }

}

App.component.html

<div class="container">

  <div class="row">

    <div class="col-xs-12 col-md-8 col-md-offset-2">

      <app-new-account ></app-new-account>

      <hr>

      <app-account \*ngFor="let acc of accounts; let i = index" [account]="acc" [id]="i"

        ></app-account>

    </div>

  </div>

</div>

New-account.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  providers: [LoggingService, AccountService]

})

export class NewAccountComponent {

  constructor(private loggingService: LoggingService, private accountService: AccountService) { }

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountService.addAccount(accountName, accountStatus);

    this.loggingService.logStatus(accountStatus);

  }

}

Account.component.ts

import { Component,  Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  providers: [LoggingService, AccountService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

  }

  onSetTo(status: string) {

    this.accountService.updateStatus(this.id, status);

    this.loggingService.logStatus(status);

  }

}

## Hierarchical Injector

Appmodule

AppComponent

Individual component

## How many instances of Service should there be?

Account.component.ts

import { Component, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  providers: [LoggingService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

  }

  onSetTo(status: string) {

    this.accountService.updateStatus(this.id, status);

    this.loggingService.logStatus(status);

  }

}

Newaccount.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  providers: [LoggingService]

})

export class NewAccountComponent {

  constructor(private loggingService: LoggingService, private accountService: AccountService) { }

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountService.addAccount(accountName, accountStatus);

    this.loggingService.logStatus(accountStatus);

  }

}

## Service in AppModule

App.module.ts

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule } from '@angular/forms';

import { AppComponent } from './app.component';

import { AccountComponent } from './account/account.component';

import { NewAccountComponent } from './new-account/new-account.component';

import { LoggingService } from './logging.service';

import { AccountService } from './accounts.service';

@NgModule({

  declarations: [

    AppComponent,

    AccountComponent,

    NewAccountComponent

  ],

  imports: [

    BrowserModule,

    FormsModule

  ],

  providers: [LoggingService, AccountService],

  bootstrap: [AppComponent]

})

export class AppModule { }

app.component.ts

import { OnInit } from '@angular/core';

import { Component } from '@angular/core';

import { AccountService } from './accounts.service';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css'],

  //providers: [AccountService]

})

export class AppComponent implements OnInit {

  accounts: { name: string, status: string }[] = [];

  constructor(private accountService: AccountService) {

  }

  ngOnInit() {

    this.accounts = this.accountService.accounts

  }

}

New-account.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  //providers: [LoggingService]

})

export class NewAccountComponent {

  constructor(private loggingService: LoggingService, private accountService: AccountService) { }

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountService.addAccount(accountName, accountStatus);

    this.loggingService.logStatus(accountStatus);

  }

}

Account.component.ts

import { Component, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  //providers: [LoggingService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

  }

  onSetTo(status: string) {

    this.accountService.updateStatus(this.id, status);

    this.loggingService.logStatus(status);

  }

}

## Injecting Services into other services

Accounts.service.ts

import { Injectable } from "@angular/core";

import { LoggingService } from "./logging.service";

@Injectable()

export class AccountService {

  accounts = [

    {

      name: 'Master Account',

      status: 'active'

    },

    {

      name: 'Testaccount',

      status: 'inactive'

    },

    {

      name: 'Hidden Account',

      status: 'unknown'

    }

  ];

  constructor(private loggingService: LoggingService) {

  }

  addAccount(name: string, status: string) {

    this.accounts.push({ name: name, status: status });

    this.loggingService.logStatus(status);

  }

  updateStatus(id: number, status: string) {

    this.accounts[id].status = status;

    this.loggingService.logStatus(status);

  }

}

New-account.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  //providers: [LoggingService]

})

export class NewAccountComponent {

  constructor(private loggingService: LoggingService, private accountService: AccountService) { }

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountService.addAccount(accountName, accountStatus);

   // this.loggingService.logStatus(accountStatus);

  }

}

Account.component.ts

import { Component, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  //providers: [LoggingService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

  }

  onSetTo(status: string) {

    this.accountService.updateStatus(this.id, status);

    //this.loggingService.logStatus(status);

  }

}

## Using Services for cross component communication

Accounts.service.ts

import { Injectable,EventEmitter } from "@angular/core";

import { LoggingService } from "./logging.service";

@Injectable()

export class AccountService {

  accounts = [

    {

      name: 'Master Account',

      status: 'active'

    },

    {

      name: 'Testaccount',

      status: 'inactive'

    },

    {

      name: 'Hidden Account',

      status: 'unknown'

    }

  ];

  statusUpdated = new EventEmitter<string>();

  constructor(private loggingService: LoggingService) {

  }

  addAccount(name: string, status: string) {

    this.accounts.push({ name: name, status: status });

    this.loggingService.logStatus(status);

  }

  updateStatus(id: number, status: string) {

    this.accounts[id].status = status;

    this.loggingService.logStatus(status);

  }

}

Account.component.ts

import { Component, Input, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-account',

  templateUrl: './account.component.html',

  styleUrls: ['./account.component.css'],

  //providers: [LoggingService]

})

export class AccountComponent {

  @Input() account: { name: string, status: string };

  @Input() id: number;

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

  }

  onSetTo(status: string) {

    this.accountService.updateStatus(this.id, status);

    //this.loggingService.logStatus(status);

    this.accountService.statusUpdated.emit(status);

  }

}

New-account.component.ts

import { Component, EventEmitter, Output } from '@angular/core';

import { LoggingService } from '../logging.service';

import { AccountService } from '../accounts.service';

@Component({

  selector: 'app-new-account',

  templateUrl: './new-account.component.html',

  styleUrls: ['./new-account.component.css'],

  //providers: [LoggingService]

})

export class NewAccountComponent {

  constructor(private loggingService: LoggingService, private accountService: AccountService) {

    this.accountService.statusUpdated.subscribe(

      (status: string) => alert("New Status"+ status)

    );

   }

  onCreateAccount(accountName: string, accountStatus: string) {

    this.accountService.addAccount(accountName, accountStatus);

   // this.loggingService.logStatus(accountStatus);

  }

}