

Wrap Up: Use Cases and Final Thoughts



David Mann

@MannD | www.HeirloomSoftware.com



Use Case Scenarios

Error Handling

Personalization

**General UI
Goodness**



My Goal



Questions!

Think About It!



Error Handling

Logging

Friendly User Message



Error Handling

Output Decorator

Async Loading UI

**Templates &
TemplateRefs**

**Dynamic
Component**

**Component
Inheritance**

Directives



```
export function DebugOnly(  
    target: Object,  
    propertyKey: string,  
    descriptor: TypedPropertyDescriptor<any>  
) {  
    if (environment.production  
        || document.location.href.indexOf('debug=true') === -1) {  
        descriptor.value = function () { }  
        // PERF: 1k empty function calls < .5 ms  
    }  
}
```

DebugOnly Decorator

Eliminate functions when not in debug mode

```
@DebugOnly  
LogDebug(message: string) {  
    this.Log(message, SeverityLevel.Debug, false, false);  
}
```



Going Further...

```
@LogLevel(SeverityLevel.Info)
LogInfo(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}

@LogLevel(SeverityLevel.Warning)
LogWarning(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}

@LogLevel(SeverityLevel.Error)
LogError(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}

@LogLevel(SeverityLevel.CriticalError)
LogCriticalError(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}
```



Going Further...

```
@LogLevel(SeverityLevel.Info)
LogInfo(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}

export function LogLevel(level: SeverityLevel) {
    return function (
        target: Object,
        propertyKey: string,
        descriptor: TypedPropertyDescriptor<any>
    ) {
        let currentLevel = LogConfig.CurrentLevel;
        if (level < currentLevel) {
            descriptor.value = function () { };
        }
    }
}

@LogLevel(SeverityLevel.Debug)
LogCriticalError(message: string) {
    this.Log(message, SeverityLevel.Debug, false, false);
}
```





Personalization



Personalization Via...

**ngTemplate,
ngTemplateOutlet &
ngTemplateOutletContext**

Dynamic Components

Directives

Dynamic Forms





UI Goodness



UI Goodness

Pipes



Performance

HostBinding &
HostListening

ElementRef,
nativeElement &
Renderer2

Async Loading UI



That's All,
Folks

