

DYALOG

Belfast 2018



TamStat 2018

Michael Baas, Stephen Mansour

Agenda

- 2 Screens old/new (optionally 3!)
- Challenges / Lessons learned



Disclaimer



Disclaimer

- I am showing MiServer, but there's more!



Disclaimer

- I am showing MiServer, but there's more!
- TamStat will run in DUI!



Disclaimer

- I am showing MiServer, but there's more!
- But it's not all!



Disclaimer

- I am showing MiServer, but there's more!
- But it's not all!
- 6/10 Wizards



Disclaimer

- I am showing MiServer, but there's more!
- But it's not all!
- Personal message to the russian delegation and Mr. Rivers: joking is the sincerest form of flattery ;-)



Up the mountain



Up the mountain

- Summary Wizard



Up the mountain

- Summary Wizard
 - „sum z“ for mouse-pushers!



Up the mountain

- Summary Wizard
 - „sum z“ for mouse-pushers!

Quantity	Center	Spread	Position	Shape
<input checked="" type="checkbox"/> count	<input checked="" type="checkbox"/> mean	<input type="checkbox"/> var	<input checked="" type="checkbox"/> min	<input type="checkbox"/> skewness
<input type="checkbox"/> sum	<input checked="" type="checkbox"/> median	<input checked="" type="checkbox"/> sdev	<input checked="" type="checkbox"/> 1 quartile	<input type="checkbox"/> kurtosis
<input type="checkbox"/> sumSquares	<input type="checkbox"/> mode	<input type="checkbox"/> range	<input checked="" type="checkbox"/> 3 quartile	
<input type="checkbox"/> product	<input checked="" type="checkbox"/> stdErr	<input type="checkbox"/> iqr	<input checked="" type="checkbox"/> max	



Up the mountain

- Summary Wizard
 - „sum z“ for mouse-pushers!

Quantity	Center	Spread	Position	Shape
<input checked="" type="checkbox"/> count	<input checked="" type="checkbox"/> mean	<input type="checkbox"/> var	<input checked="" type="checkbox"/> min	<input type="checkbox"/> skewness
<input type="checkbox"/> sum	<input checked="" type="checkbox"/> median	<input checked="" type="checkbox"/> sdev	<input checked="" type="checkbox"/> 1 quartile	<input type="checkbox"/> kurtosis
<input type="checkbox"/> sumSquares	<input type="checkbox"/> mode	<input type="checkbox"/> range	<input checked="" type="checkbox"/> 3 quartile	
<input type="checkbox"/> product	<input checked="" type="checkbox"/> stdErr	<input type="checkbox"/> iqr	<input checked="" type="checkbox"/> max	

Contents of #.TamStat.Summary (Global Scope)

count	kurtosis	mode	product	skewness	sumProduct
cv	max	percent	proportion	standardDeviation	sumSquares
geometricMean	mean	percentRank	quartile	stats	var
harmonicMean	median	percentile	range	stdErr	variance
iqr	min	percentileRank	sdev	sum	zScore



Up to

- Sum
-

Qu

☒ count

☐ sum

☐ sumSqu

☐ produc

Contents of

☒ count

☒ cv

☐ geometric

☐ harmonic

☐ iqr

Long- Winded Joke!



Up to

- Sum

○

○

Qu

☒ count

☐ sum

☐ sumSqu

☐ produc

Contents of

☒ count

☒ cv

☐ geometric

☐ harmonic

☐ iqr

Long- Winded Joke!



Up the mountain

- Summary Wizard
 - „sum z“ for mouse-pushers!
 - Demo

Quantity	Center	Spread	Position	Shape
<input checked="" type="checkbox"/> count	<input checked="" type="checkbox"/> mean	<input type="checkbox"/> var	<input checked="" type="checkbox"/> min	<input type="checkbox"/> skewness
<input type="checkbox"/> sum	<input checked="" type="checkbox"/> median	<input checked="" type="checkbox"/> sdev	<input checked="" type="checkbox"/> 1 quartile	<input type="checkbox"/> kurtosis
<input type="checkbox"/> sumSquares	<input type="checkbox"/> mode	<input type="checkbox"/> range	<input checked="" type="checkbox"/> 3 quartile	
<input type="checkbox"/> product	<input checked="" type="checkbox"/> stdErr	<input type="checkbox"/> iqr	<input checked="" type="checkbox"/> max	

Contents of #.TamStat.Summary (Global Scope)

count	kurtosis	mode	product	skewness	sumProduct
cv	max	percent	proportion	standardDeviation	sumSquares
geometricMean	mean	percentRank	quartile	stats	var
harmonicMean	median	percentile	range	stdErr	variance
iqr	min	percentileRank	sdev	sum	zScore



Uncertainty & desire for flexibility



Uncertainty & desire for flexibility

```
{ "desc": "Measures of Position",  
  "title": "position",  
  "fns": [  
    { "fn": "min " },  
    { "fn": "1 quartile" },  
    { "fn": "3 quartile" },  
    { "fn": "max" }  
  ]  
}
```



Uncertainty & desire for flexibility

```
{ "desc": "Measures of Position",  
  "title": "position",  
  "fns":  
    [ { "fn": "min " },  
      { "fn": "1 quartile" },  
      { "fn": "3 quartile" },  
      { "fn": "max" }  
    ]  
}
```

```
{ "desc": "Measures of spread",  
  "title": "spread",  
  "fns":  
    [ { "fn": "range", "defaultSelectState": false },  
      { "fn": "sdev" },  
      { "fn": "var", "defaultSelectState": false },  
      { "desc": "Interquartile range", "fn": "iqr", "defaultSelectState": false }  
    ]  
},
```



Uncertainty & desire for flexibility

```
{ "desc": "Measures of Position",  
  "title": "position",  
  "fns":  
    [ { "fn": "min " },  
      { "fn": "1 quartile" },  
      { "fn": "3 quartile" },  
      { "fn": "max" }  
    ]  
}
```

sw←JSON1>nget #.Boot.AppRoot,'assets/sumwiz.json'

```
{ "desc": "Measures of spread",  
  "title": "spread",  
  "fns":  
    [ { "fn": "range", "defaultSelectState": false },  
      { "fn": "sdev" },  
      { "fn": "var", "defaultSelectState": false },  
      { "desc": "Interquartile range", "fn": "iqr", "defaultSelectState": false }  
    ]  
},
```



Uncertainty & desire for flexibility

```
{ "desc": "Measures of Position",  
  "title": "position",  
  "fns":  
    [ { "fn": "min " },  
      { "fn": "1 quartile" },  
      { "fn": "3 quartile" },  
      { "fn": "max" }  
    ]  
}
```

```
sw←JSON1>nget #.Boot.AppRoot,'assets/sumwiz.json'
```

```
sw.fns.defaultSelectState←{{6::_true  
ω.defaultSelectState}ω.fns}sw
```

```
{ "desc": "Measures of spread",  
  "title": "spread",  
  "fns":  
    [ { "fn": "range", "defaultSelectState": false },  
      { "fn": "sdev" },  
      { "fn": "var", "defaultSelectState": false },  
      { "desc": "Interquartile range", "fn": "iqr", "defaultSelectState": false }  
    ]  
},
```



Uncertainty & desire for flexibility

```
{
  "desc": "Measures of Position",
  "title": "position",
  "fns": [
    { "fn": "min "},
    { "fn": "1 quartile"},
    { "fn": "3 quartile"},
    { "fn": "max"}
  ]
}
```

```
sw←JSON1>nget #.Boot.AppRoot,'assets/sumwiz.json'
```

```
sw.fns.defaultSelectState←{{6::_true ◊
ω.defaultSelectState}ω.fns}sw
```

```
sw.fns.desc←{{6:: ω.fn ◊ ω.desc}ω.fns}sw
```

```
{
  "desc": "Measures of spread",
  "title": "spread",
  "fns": [
    { "fn": "range", "defaultSelectState": false},
    { "fn": "sdev"},
    { "fn": "var", "defaultSelectState": false},
    { "desc": "Interquartile range", "fn": "iqr", "defaultSelectState": false}
  ]
},
```



Uncertainty & desire for flexibility

- I ❤️



Uncertainty & desire for flexibility



Uncertainty & desire for flexibility

- I ❤️ .json & JSON



Uncertainty & desire for flexibility

- I ♥ .json & JSON
- 1st learning: use JSON.



Uncertainty & desire for flexibility

- I ❤️ .json & JSON
- 1st learning: use JSON.
- 2nd learning: separate code & UI.



Uncertainty & desire for flexibility

- I ❤️ .json & JSON
- 1st learning: use JSON.
- 2nd learning: separate code & UI.
- 3d learning: avoid hardcoded lists.
(There is a chance a russian may come along and extend them! 🤔)



2nd screen



2nd screen

- Confidence Interval Wizard



2nd screen

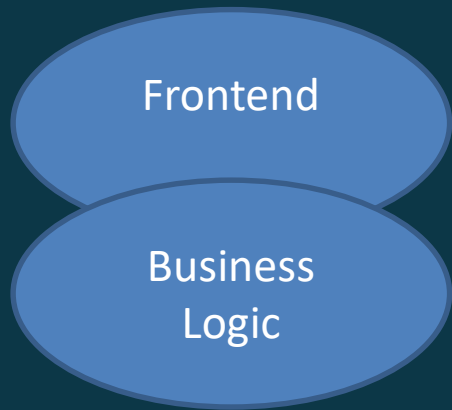
- Confidence Interval Wizard
- „deeply learned“ to separate code & application



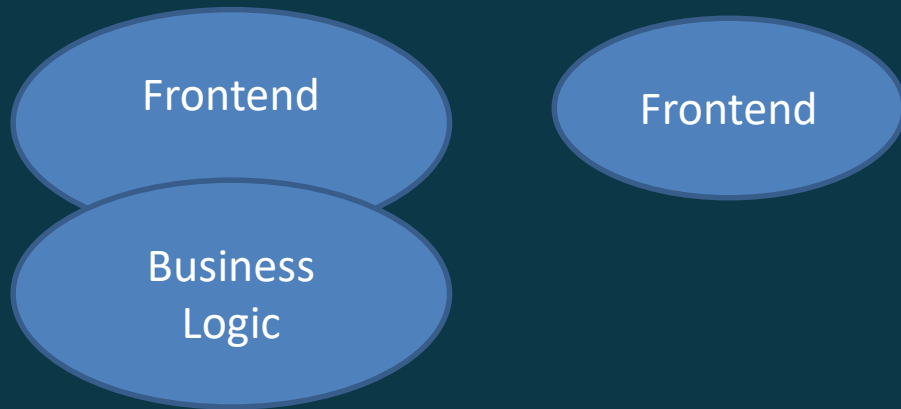
Separate Code from your UI



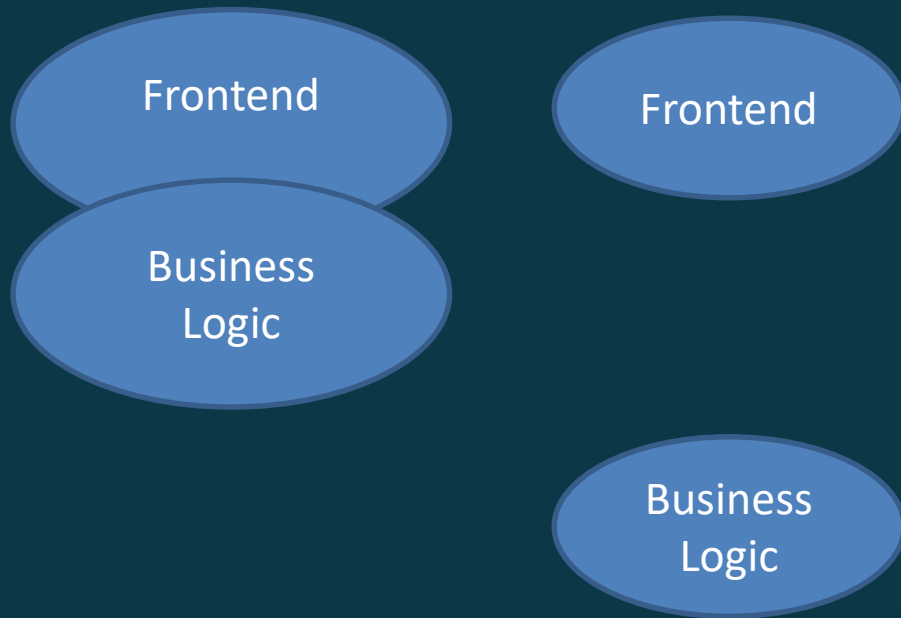
Separate Code from your UI



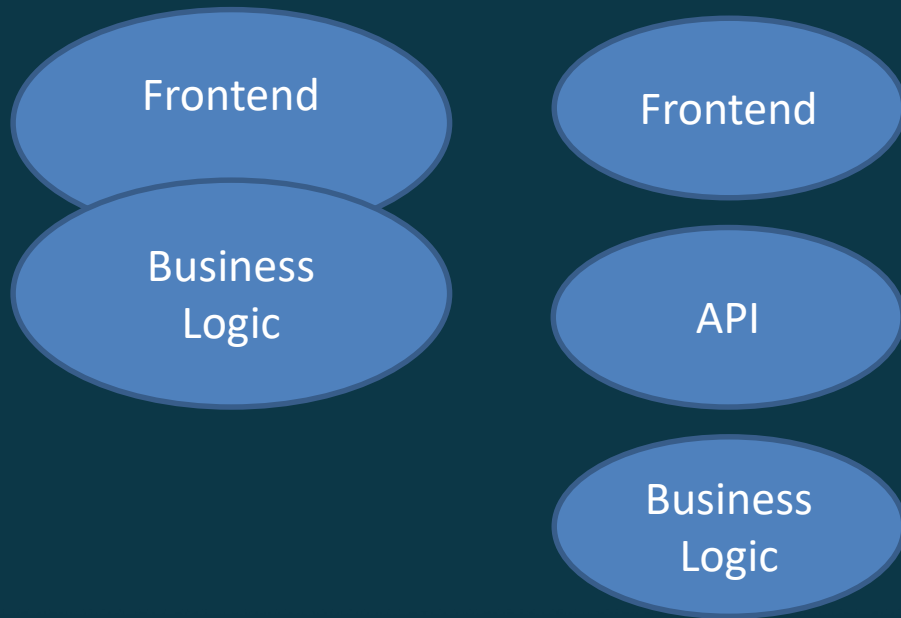
Separate Code from your UI



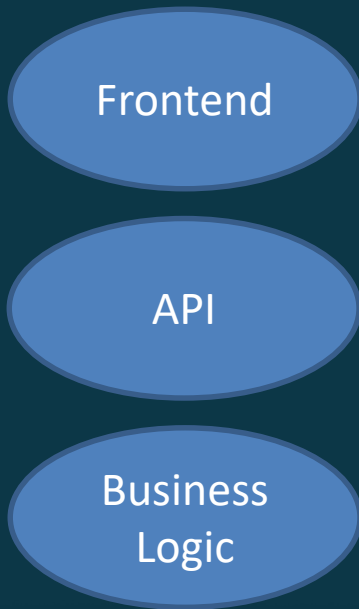
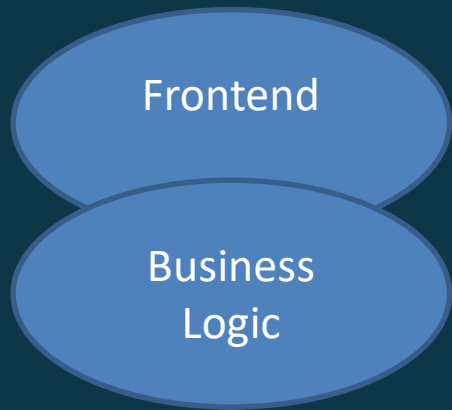
Separate Code from your UI



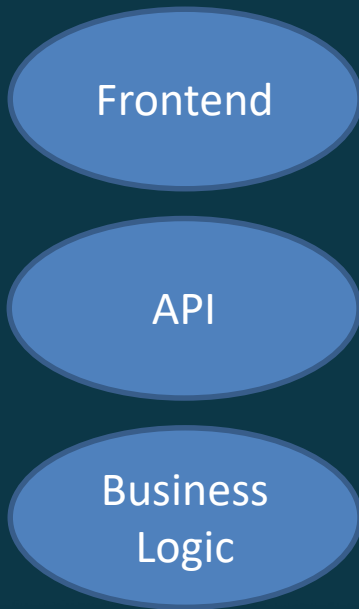
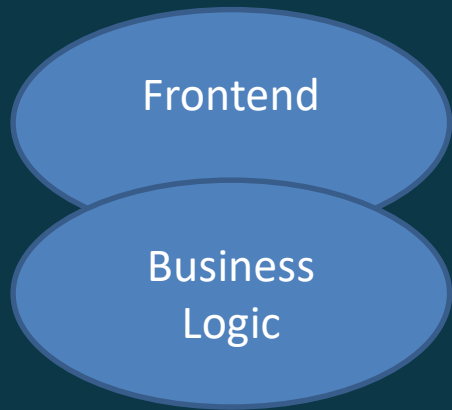
Separate Code from your UI



Separate Code from your UI



Separate Code from your UI



Separate Code from your UI

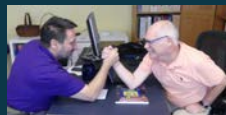
Frontend

Business
Logic

Frontend

API

Business
Logic



Separate Code from your UI

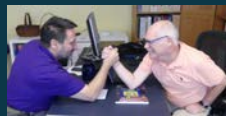
Frontend

Frontend



Business
Logic

API



Business
Logic



Separate Code from your UI

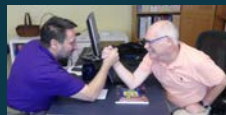
Frontend

Business
Logic

Frontend

API

Business
Logic



Separate Code from your UI

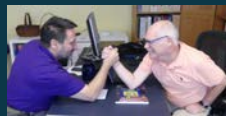
Frontend

Business
Logic

Frontend

API

Business
Logic



Separate Code from your UI

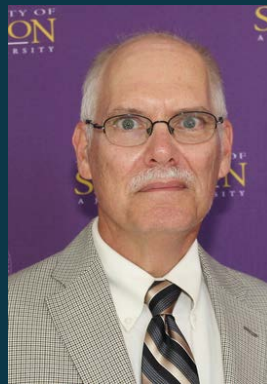
Frontend

Business
Logic

Frontend

API

Business
Logic



A flexible API

- Simple syntax:
`Res ← ConfIntAPI Arg`
- Res and Arg are namespaces!
- Easily extend fns without worrying about interfaces.
- Helps to decouple development-process
- Might be translatable into JSON
- Arg.Event selects a specific function to execute.
- Different fns may require or set different variables
- Run backend in JSONServer!
- Careful with arrays!



Summary

- API first!
 - For new Wizards, I sketch the API first
 - Functions, Inputs, Outputs
- Be dynamic!
 - No strict rules
- "relaxed" backend (not as packed/dense as before)
- easier debugging/testing



The fundamental difference

- small, cosy world
- □ WC
- Microsoft Windows on local PC
- 1024 x 768
- count pixels
- carefully position individual controls
- the cold, brutal universe
- DUI
- infinite permutations of resolutions, screensizes, operating-systems, browsers
- flow groups of things



The reward



MiServer

HtmlRenderer
WC2

TamStat
„Classic“

API

Business
Logic



The End



?



Components

- DUI with
 - Bootstrap 4.1 (using themed builds from <http://bootstrap.build>)
 - DataTables 1.10.18
 - FontAwesome 5.3 Pro
 - SharpPlot
 - ...and more



