



ClassCAD

Safe (Interpreter) API

What we want and how it could be solved



AWV

Today's used Execution-Command

```
{  
  "command": "Execute",  
  "ccCommand": "__C.Calculator.Add(5,5);",  
  "streamData": {  
    "streamKey1": "tX1tcx27gWcGI0//T8=",  
  }  
}
```

- ClassCAD interpreter command is a string within a JSON structure
- All commands could be executed
 - Service functions like CADH_DoThis();
 - Methods of an object
 - Any interpreter code



AWV

New business and its risks

- As ClassCAD is moving to the cloud
 - ... any command could **influence the infrastructure**
 - CADH_DeleteFile(...)
 - ... any command could **influence other users using the same instance**
 - CADH_SetVar(...)
 - CADH_SetGPURTessellationEnabled(...)
 - ... any JavaScript developer must know interpreter syntax (`_C.A({5,6,9});`)
- Already existing risks
 - As the API is everything what can be executed, how to create a **documentation** for the API?
 - How to **maintain** the code, as every change could break the client-application? There is no API contract.



AWV Safe API proposal

```
{  
  "command": "Execute",  
  "ccCommand": "_C.Calculator.Add(5,5);",  
  "streamData": {  
    "streamKey1": "tX1tcx27gWcGI0//T8=",  
  }  
}
```

String

"CommandString;"

"CommandString; CommandString;",

```
{  
  "command": "Execute",  
  "ccCommand": [{"Math.Add": [5,5]}],  
  "streamData": {  
    "streamKey1": "tX1tcx27gWcGI0//T8=",  
  }  
}
```

JSON-Array

[{"Namespace.Command": [command parameters]}]

**[
 {"Namespace.Command ": [command parameters]},
 {"Namespace.Command ": [command parameters]}
]**



AWV Advantages

- Instances can be configured to accept only “Safe API” calls
 - For development instances will be able to run any code as today
- Only registered Interfaces will accept calls who match to a method with the same name.
- As the API is well defined we should be able to create a documentation about it 😊
- Of course, there will be some additional work to write the interfaces, but you will never get security for free!