Anonymization tool for GDPR

Code quality metrics and management
About Today

• What and why?
• How?
• Do you have a solution?
• Can we see it?
• Future
but first...

... allow me to reintroduce myself
my name is ...
Valentin Duricu

• Senior Software Developer  
  @ Wayfare
• CEO & Photographer  
  @ Spark Development
• Head of Development  
  @ Atelierul de Torturi și Prăjituri
• 8,5 Years of Development Experience  
  5,5 using Progress OpenEdge – since 11.1
• Hobbies: Photography, Music and Gaming
• Learn more about me here:  
  https://bit.ly/2Clc1Xo  
...AND NOW BACK TO OUR REGULARLY SCHEDULED PROGRAMMING!...
Hmm..
Problems...

• Large applications
• Duplicate code
• Using libraries => always download right version
• Hard to refactor
• Not easy to reuse code
More Problems...

• Update each client manually

• Not so easy to deploy

• Configuration problems

• ...


But why?
Why?

• Smaller testable pieces
• Easily replaceable parts
• Code reusing
• Use only what is needed
• Smaller application
HOW?
Solutions?

• Split code in components
• Component - smaller project
• Reuse components
• Deliver components
... other solutions?

- Our tool...
- Package manager
- Project dependencies
- Online Repository
What is a Package?

<table>
<thead>
<tr>
<th>Branch: master</th>
<th>puzzle-framework-sample / puzzles / sample.module /</th>
<th>Create new file</th>
<th>Upload files</th>
<th>Find file</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>vduricu Refactoring</td>
<td>Latest commit a11e3f8 on 26 May</td>
<td>----------------</td>
<td>--------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>api</td>
<td>sample project</td>
<td>5 months ago</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>route</td>
<td>Refactoring</td>
<td>4 months ago</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SampleCommand.js</td>
<td>sample project</td>
<td>5 months ago</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>index.js</td>
<td>sample project</td>
<td>5 months ago</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>package.json</td>
<td>Sample project</td>
<td>9 months ago</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is a Dependency?
What is a Dependency Manager?

```json
{
  "name": "sample",
  "version": "1.0.0",
  "description": "Sample Puzzle Framework project",
  "main": "sample_puzzle.js",
  "private": true,
  "bin": {
    "sample_puzzle-cli": "/bin/cli",
    "sample_puzzle": "/bin/sample_puzzle"
  },
  "scripts": {
    "start": "/node_modules/nodebin/bin/nodebin.js --watch . sample_puzzle.js"
  },
  "keywords": [
    "sample",
    "puzzle framework"
  ],
  "dependencies": {
    "bluebird": "^3.5.1",
    "nodebin": "^1.17.3",
    "puzzle-framework": "^1.2.0"
  }
}
```
... examples

• npm, yarn
• composer
• nuget
• ...

NPM DELIVERY

Oh boy! My package is here!

Monkeyuser.com
ARE WE THERE YET ??
• Currently a Proof of Concept

• Dedicated to OpenEdge

• Inspired by: npm, yarn, composer
3 Parts:

- Project Specification File
- Terminal (CLI) Application
- Web Repository
Communication between parts

Terminal Application

Web Repository

perform request

send response

update file

read file

application.json
Project Specification File

IT’S A SECRET
Project File

• Standard file

application.json

• Standard fields

• Easy to understand

• JSON File

  • not that verbose like XML
Required

- Name
- Version
- Dependencies (if any)
Optional

• Anything else..
• Author, Contributors
• License
• Issue Tracking
• Scripts
• OE Version
• Can have user defined fields
• Ignored by the validation algorithm
• Useful for you developers
Terminal (CLI) Application
Terminal (CLI) Application

• Your personal minion

• Does tasks for you

• Installed locally

• Works with various package repositories
Tasks

- Fetch packages
- Install packages
- Publish your package
- Run scripts defined by you for you
- Generate right ProPath
Web Repository
Web Repository

• Storage of the packages

• Package listing

• Documentation listing for CLI and File Specification
Web Repository

• Made for Cloud
public repository

• Deployable in enterprise environment
private repository
MY NAME IS
Extra thanks

• My colleague that developed:
  • UI/UX
  • Design
  • Logo

Daliana Nicolau
Le Demo
Starting point
At the end
Package definitions

```
{
    "name": "ValidityTools",
    "version": "0.1.1",
    "description": "Some validity checking tools",
    "author": "Valentin Duricu <valentin.duricu@wayfare.ro>",
    "keywords": [
        "Valid datatype",
        "valid types"
    ]
}
```
## Package structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>isValid.p</td>
<td>04.10.2018 22:27</td>
</tr>
</tbody>
</table>

---

```lisp
FUNCTION isValidCharacter RETURNS LOGICAL
    ( INPUT pcCharacter AS CHARACTER ):
    RETURN pcCharacter NE "" AND pcCharacter NE ?.
END FUNCTION.

FUNCTION isValidInteger RETURNS LOGICAL
    ( INPUT pInteger AS INTEGER ):
    RETURN pInteger NE ?.
END FUNCTION.

FUNCTION isValidDecimal RETURNS LOGICAL
    ( INPUT pdDecimal AS DECIMAL ):
    RETURN pdDecimal NE ?.
END FUNCTION.

FUNCTION isValidLogical RETURNS LOGICAL
    ( INPUT pLogical AS LOGICAL ):
    RETURN pLogical NE ?.
END FUNCTION.

FUNCTION isValidDate RETURNS LOGICAL
    ( INPUT pdtDate AS DATE ):
    RETURN pdtDate NE ?.
END FUNCTION.

FUNCTION isValidDateTime RETURNS LOGICAL
    ( INPUT pdtDateTime AS DATETIME ):
    RETURN pdtDateTime NE ?.
END FUNCTION.
```
Package publishing

Beast CLI: -
Package published

**ValidityTools**

Some validity checking tools

Valid datatype  valid types

Valentin Duricu published 0.1.1 • 2 days ago
Package update

Before

After
// Package update - code

File : ValidObject
Purpose :
Syntax :
Description :
Author(s) : ValentinDuricu
Created : Sat Oct 06 14:07:39 EST 2018
Notes :
---------------------------------------------------------------------------*/

USING Progress.Lang.*.

BLOCK-LEVEL ON ERROR UNDO, THROW.

CLASS src.ValidObject:
    METHOD PUBLIC LOGICAL isValid(INPUT pcCharacter AS CHARACTER) :
        RETURN pcCharacter NE "" AND pcCharacter NE ?.
        END METHOD.
    METHOD PUBLIC LOGICAL isValid(INPUT piInteger AS INTEGER) :
        RETURN piInteger NE ?.
        END METHOD.
    METHOD PUBLIC LOGICAL isValid(INPUT pdDecimal AS DECIMAL) :
        RETURN pdDecimal NE ?.
        END METHOD.
    METHOD PUBLIC LOGICAL isValid(INPUT plLogical AS LOGICAL) :
        RETURN plLogical NE ?.
        END METHOD.
    METHOD PUBLIC LOGICAL isValid(INPUT pdsDate AS DATE) :
        RETURN pdsDate NE ?.
        END METHOD.
    METHOD PUBLIC LOGICAL isValid(INPUT pdtDateTime AS DATETIME) :
        RETURN pdtDateTime NE ?.
        END METHOD.
END CLASS.
Package definitions

```json
{
    "name": "sports2000backend",
    "version": "1.0.0",
    "description": "Backend application for sports2000 db",
    "author": "Valentin Duricu <valentin.duricu@wayfare.ro>",
    "keywords": [
        "sports2000",
        "backend application"
    ],
    "dependencies": {
        "ValidityTools": "^0.1.0"
    }
}
```
Package structure

Sports2000Backend

/**
 File : CustomerBE.p
 Purpose :
 Syntax :
 Description :

 Author(s) : valentin.duricu
 Created : Wed Oct 03 19:07:46 BEST 2018
 Notes :

 **************************************************************************
 BLOCK-LEVEL ON ERROR UNDO, THROW.

 {ttCustomer.i}

 PROCEDURE fetchData:
 DEFINE INPUT PARAMETER piHowMany AS INTEGER NO-UNDO.
 DEFINE OUTPUT PARAMETER table FOR ttCustomer.

 DEFINE VARIABLE iCount AS INTEGER NO-UNDO.
 DEFINE BUFFER Customer FOR Customer.

 FOR EACH Customer NO-LOCK:
 CREATE ttCustomer.
 BUFFER-COPY Customer TO ttCustomer.
 iCount = iCount + 1.
 IF iCount >= piHowMany THEN
 LEAVE.
 END.
 END PROCEDURE.
Package install

Sports2000Backend

Beast CLI: _
Package structure

Sports2000Backend

- .settings
- .dbconnection
- .project
- .propath
- application.json
- application.tree.json
- CustomerBE.p
- ttCustomer.i

Date modified:
- 06.10.2018 14:30
- 06.10.2018 14:30
- 04.10.2018 22:59
- 03.10.2018 19:03
- 04.10.2018 23:03
- 03.10.2018 19:05
- 06.10.2018 13:10
- 03.10.2018 19:30
- 03.10.2018 19:26
Package publishing

Beast CLI: _
# Package update

In the directory structure:

```
entinDuricu > Workspace > PDSOE > Sports2000Backend
```

- `.settings` (06.10.2018 14:30)
- `oe_modules` (06.10.2018 14:30)
- `.dbconnection` (04.10.2018 22:59)
- `.project` (03.10.2018 19:03)
- `.propath` (04.10.2018 23:03)
- `application.json` (03.10.2018 19:05)
- `application.tree.json` (06.10.2018 13:10)
- `CustomerBE.p` (03.10.2018 19:30)
- `ttCustomer.i` (03.10.2018 19:26)

In the updated directory structure:

```
Workspace > PDSOE_Upd > Sports2000Backend
```

- `.settings` (06.10.2018 13:50)
- `oe_modules` (06.10.2018 13:50)
- `.dbconnection` (04.10.2018 22:59)
- `.project` (03.10.2018 19:03)
- `.propath` (04.10.2018 23:03)
- `application.json` (03.10.2018 19:05)
- `application.tree.json` (06.10.2018 14:12)
- `CustomerBE.cls` (highlighted)
- `CustomerBE.p` (06.10.2018 14:45)
- `ttCustomer.i` (03.10.2018 19:26)
PROCEDURE fetchData:
  DEFINE INPUT PARAMETER piHowMany AS INTEGER NO-UNDO.
  DEFINE OUTPUT PARAMETER table FOR ttCustomer.

  DEFINE VARIABLE iCount AS INTEGER NO-UNDO.
  DEFINE VARIABLE hisValid AS HANDLE NO-UNDO.

  DEFINE BUFFER Customer FOR Customer.

  RUN src/isValid.p PERSISTENT SET hisValid.
  IF NOT DYNAMIC-FUNCTION("isValidInteger" IN hisValid, piHowMany) THEN
    piHowMany = 5.

  FOR EACH Customer NO-LOCK:
    CREATE ttCustomer.
    BUFFER-COPY Customer TO ttCustomer.
    iCount = iCount + 1.
    IF iCount >= piHowMany THEN
      LEAVE.
  END.
END PROCEDURE.

CLASS CustomerBE INHERITS src.ValidityObject:
  {ttCustomer.i}

  METHOD PUBLIC VOID fetchData(OUTPUT table ttCustomer):
    fetchData(i, OUTPUT table ttCustomer).
  END METHOD.

  METHOD PUBLIC VOID fetchData(INPUT piHowMany AS INTEGER, OUTPUT table ttCustomer):
    DEFINE VARIABLE iCount AS INTEGER NO-UNDO.
    IF NOT THIS-OBJECT:isValid(piHowMany) THEN
      piHowMany = 5.
    END.

    DEFINE BUFFER Customer FOR Customer.

    FOR EACH Customer NO-LOCK:
      CREATE ttCustomer.
      BUFFER-COPY Customer TO ttCustomer.
      iCount = iCount + 1.
      IF iCount >= piHowMany THEN
        LEAVE.
    END.
END METHOD.
END CLASS.
Package definitions

```
"name": "sports2000ui",
"version": "1.0.0",
"description": "Frontend application for sports2000 db",
"author": "Valentin Duricu <valentin.duricu@wayfare.ro>",
"keywords": [
  "sports2000",
  "backend application"
],
"dependencies": {
  "sports2000backend": "^1.0.0"
},
"devDependencies": {
  "ValidityTools": "0.1.1"
}
```
Package install

Beast CLI: -
Package structure

- Sports2000UI
  - Procedure Libraries
  - oe_modules
    - boot
      - boot.p
      - propath.json
      - ttProPath.i
  - sports2000backend
    - application.json
    - application.tree.json
    - CustomerBE.p
    - ttCustomer.i
  - ValidityTools
    - src
      - isValid.p
      - application.p
  - rcode
    - src
      - CustomerWindow.w
      - application.json
      - application.tree.json
Package structure

Sports2000UI

```
block-level on error undo, throw.

define variable cProPath as character no-undo.
{oe_modules/boot/ttProPath.i}

file-info:file-name = "./oe_modules/boot/propath.json".

for each ttProPath:
cProPath = substitute("$1,$2", cProPath, ttProPath.path).
end.

propath = substitute("$1,$2", trim(cProPath, ","), propath).

if defined(beast-ttpropath) eq 0 then

global-define beast-ttpropath 1

define tmp-table ttProPath no-undo
    field path as character.
endif
```
Package definitions

```json
{
  "name": "sports2000kuib",
  "version": "1.0.0",
  "description": "Kendo UI Catalog application for sports2000 db",
  "author": "Valentin Duricu <valentin.duricu@wayfare.ro>",
  "keywords": [
    "sports2000",
    "backend application"
  ],
  "dependencies": {
    "sports2000backend": "^1.0.0"
  }
}
```
Package install

Beast CLI:
Package structure

- Sports2000KUIB
  - Procedure Libraries
  - oe_modules
  - PASOEOContent
  - rcode
  - src
    - application.json
    - application.tree.json
    - README.txt
  - Defined Services
Demo

• Live coding…
• Some coding…
• Coding?
• Copy pasting?
• …
I want a private DEMO

• Come at our booth and register for a private DEMO

• Come at our booth and signup as beta testers
Future
Future...

American rapper, singer, songwriter, and record producer
Future

• Release a Beta
• Publish the Cloud Repository
• Extend documentation
• Integration with Developer Studio
• Integration with other tools…
Valentin Duricu
valentin.duricu@wayfare.ro
THANK YOU
Links to the images & clips

- Slide 1: Spark Digital Studios, Belgrade 2018
- Slide 5: https://www.washingtontimes.com/cartoons/presidential-inauguration/and-now-back-our-regularly-scheduled-programming/
- Slide 11: https://tenor.com/view/spongebob-how-gif-4350446
- Slide 18: https://monkeyuser.com
- Slide 20: https://pixabay.com/p-1724676/?no_redirect
- Slide 29: https://tenor.com/view/minions-despicable-me-happy-cheer-excited-gif-3571108
- Slide 32: https://tenor.com/view/error-wait-computer-download-cat-gif-10480536
- Slide 35: https://tenor.com/view/my-name-is-eminem-gif-12352010
- Slide 37: https://tenor.com/view/deadpool-thats-the-coolest-name-ever-gif-11247562
- Slide 39: https://tenor.com/view/despicable-me3-despicable-me3gifs-minion-minions-mob-mentality-gif-8051851