

# Code standard

## Server (BrainCloud: JavaScript)

Scripts Naming (Voir BrainCloud:Scripts)

---

- On organise le nom des scripts de la manière suivantes :  
{folderName}\_{subFolderName}\_{ScriptName}
- Dans le cas des scripts appellable par le client on ajoute "Cmd" comme prefix du "ScriptName"
- Exemples:
  - acc1\_CmdCreateTeam (pour les commandes de create team)
  - ent1\_Utils (pour les scripts utilaire spécifique à un jeu)
  - acc1\_Bills
  - CmdSaveSegment (Pour une commande générique)

## Server (GS: JavaScript)

### Collection

- **shortenedCollectionName = collectionName (collectionKey)**
  - eval = evaluation (studentUid)
  - evalTeam = evaluationTeam (teamUid)
  - result = results (simUid)
  - sim = simulation (simUid)
  - team = teams (teamU
  - user = userdetails (userUid)

## How to name **Events** (generic code)

- **{Action}{Collection}{WhatItDoes}** PascalCase
  - GetSimGroup
  - CreateTeam
  - CreateSimCustomEval
  - GetEvalScenario
  - JoinSim
  - JoinTeam
  - ReopenSimYear
  - EditUserStudent

## How to get a **Collection**

- ```
var {shortenedCollectionName}Entry = API.getItem("{collectionName}", {collectionKey}).document();
```

```
var {shortenedCollectionName}Data = {shortenedCollectionName}Entry.getData();
```

## Set a **Response**

- USE `sendMessageTo{Receiver}({specificToTheMethod}, {messageType}, {messageData})` when you **MODIFY** data
  - **messageType**
    - Actions
      - deleted, changed, created
    - {dataObject}\_{field}\_{action}
      - ex.: team\_cash\_changed
    - {dataObject}\_{fieldOne}\_{fieldTwo}\_{etc.}\_{action}
      - ex.: sim\_year\_yearState\_changed
    - {dataObject}\_{fieldParent}\_{fieldChild}\_{etc.}\_{action}
      - ex.: team\_yearData\_factory\_changed
  - **messageData**
    - Information you want to send to the **Client**.
    - send `teamUid` and/or `studentUid` when you send message to **teacher** (so that the teacher can filter the message)
- use `Spark.setScriptData('{ObjectName}', {Object})` when you **GET** data

## Client (VS: C#)

# Standard

- **Cases**

- **Class** PascalCase
- **Methods** PascalCase
- **Private members** \_camelCase
- **Public members** camelCase

- **Document**

- **Formatting order** (each of those section should be surrounded by region with sub-region if necessary)
  1. **Public members**
  2. **Public property**
  3. **Constructors**
  4. **Methods**
  5. **Private property**
  6. **Private Members**

- Whenever possible surround **code** with `#region MyRegion {code} #endregion` to make the **.cs** clearer
- use **Stubs** when the interface only need partial information instead of loading all the data.
  - exemples : studentStub, EvalStudentStub.
- if you want to do **Lazy loading** or **Lazy Initialization** make a Property (the reference is always private)
  - **public** : under public members
  - **private** : over private members
  - ex.:

```
private Dictionary<string, TextMeshProUGUI> TeamEvalTexts
{
    get
    {
        if (_teamEvalTexts == null)
        {
            _teamEvalTexts = new Dictionary<string, TextMeshProUGUI>();
            _teamEvalTexts[Simulation.SIM_PERF_NAME] = simPerfNoteText;
            _teamEvalTexts[Simulation.LOGBOOK_NAME] = logbookNoteText;
        }
        return _teamEvalTexts;
    }
}
private Dictionary<string, TextMeshProUGUI> _teamEvalTexts;
```

- Plugin edition :

```
//UNIKSIM BEGIN [name] [reason]
//old code in comment
...//UNIKSIM END
```

## Data

- **SetListeners** in the data **constructors**
- the data **Update** itself and fire a relevant `Action`
- Filter the **Response** with `teamUid` and/or `studentUid` when the **teacher** receive the message

# Client (Unity)

## Buttons

- **ConfirmButton** are always to the right
- **CancelButton** are always to the left

## Prefabs

- Use the basic **Prefab** when ever you need them. (NormalButton, EditButton, InputText, Dropdown, etc.)

## Popup

- **X on the top-right corner VS CancelButton**
  - When there is no action, only information, put X on the top-right corner and close on click outside.
  - When there is an action, inputs, put a CancelButton with a confirmation to close.

# Google sheets

- **shared\_text** is used by all the projects.
- Each project has a Google Sheet containing their project specific texts.

---

Revision #9

Created Fri, Jul 5, 2019 2:26 PM by [Frédéric Messier](#)

Updated Mon, Jun 13, 2022 2:35 PM by [Philippe](#)