

#Protocolo padrão entre o Cliente e o Servidor

Requisição:

Arquivo JSON contendo a seguinte estrutura:

```
{
  id: 'id da estação gerado automaticamente pelo servidor e gravado no cookie da
primeira requisição entre cliente e servidor',
  op: 'nome da operação que o servidor deverá executar',
  data: 'objeto JSON que descreve os parâmetros da operação'
}
```

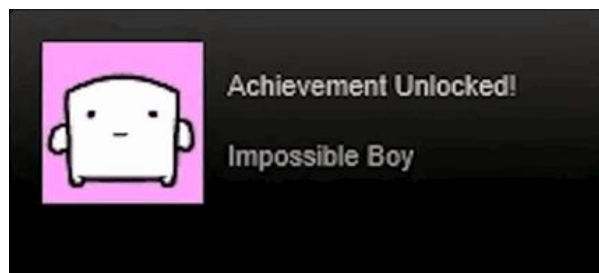
Resposta:

Arquivo JSON com a seguinte estrutura:

```
{
  response: 'código da resposta de acordo com o processamento da operação
solicitada',
  data: 'dados, em JSON, gerado pelo servidor de acordo com o requisição solicitada.
Se a resposta gerar um erro, aqui poderá ser retorna o motivo. Caso contrário, retornar o quê é
esperado pela operação'
}
```

#Operações da versão 1

Conquistas (Achievements/Trophies) (+ GamerScore/XP)



#addTrophy

Requisição:

```
id: 'id da estação/player',
op: 'add-trophy',
data: '{name: '10 coins', xp: 30, title: 'IT'S A START',
```

```
description: 'Collected first 10 coins on the game'}'
```

Resposta:

```
response: 'ok',
data: ''
```

#getTrophy

Requisição:

```
id: 'id da estação/player',
```

```
    op: 'get-trophy',
    data: '10 coins'
Resposta:
    response: 'ok',
    data: '{name: '10 coins', xp: 30, title: 'IT\'S A START',
description: 'Collected first 10 coins on the game'}'
```

#listTrophy

Requisição:

```
    id: 'id da estação/player',
    op: 'list-trophy',
    data: ''
```

Resposta:

```
    response: 'ok',
    data: '[
{name: '10 coins', xp: 30, title: 'IT\'S A START', description:
'Collected first 10 coins on the game'},
{name: 'first death', xp: 10, title: 'KEEP CALM AND PLAY',
description: 'First death on game'}, ... ]'
```

#clearTrophy

Requisição:

```
    id: 'id da estação/player',
    op: 'clear-trophy',
    data: ''
```

Resposta:

```
    response: 'ok',
    data: ''
```

Cloud-Save (Checkpoint)

#saveState

Requisição:

```
    id: 'id da estação/player',
    op: 'save-state',
    data: '{x: 130, y: 30}'
```

Resposta:

```
    response: 'ok',
    data: ''
```

#loadState

Requisição:

```
    id: 'id da estação/player',
    op: 'load-state',
    data: ''
```

Resposta:

```
    response: 'ok',
    data: '{x: 130, y: 30}'
```

Media Share (Game DVR): foto (e video)

#saveMedia

Requisição:

id: 'id da estação/player',

op: 'save-media',

data: '{mimeType: 'image/png', src: 'dados da media serializado via
método canvas.toDataURL()}'

Resposta:

response: 'ok',

data: "

#listMedia

Requisição:

id: 'id da estação/player',

op: 'list-media',

data: '{mimeType: 'image/png', start: 0, count: 10}'

Resposta:

response: 'ok',

data: '['

{mimeType: 'image/png', src: 'BDdje64_jD...'},

{mimeType: 'image/png', src: 'fDkmh62g22...'}, ...]'

#O jogador

```
// Abstrai os protocolos de comunicacao com o GAME SERVER
```

```
class ServerComm {

    static addTrophy(trophyData, callback) {
        console.log('addTrophy: ' + trophyData['name'])
        let data = {
            id: 'john-doe',
            op: 'add-trophy',
            data: trophyData // object
        }
        ServerComm.ajaxPost(data, callback)
    }

    static getTrophy(trophyName, callback) {
        console.log('getTrophy: ' + trophyName)
        let data = {
            id: 'john-doe',
            op: 'get-trophy',
            data: trophyName // string
        }
        ServerComm.ajaxPost(data, callback)
    }

    static listTrophy(callback) {
        console.log('listTrophy: ')
        let data = {
            id: 'john-doe',
            op: 'list-trophy',
            data: ''
        }
        ServerComm.ajaxPost(data, callback)
    }

    static clearTrophy(callback) {
        console.log('clearTrophy: ')
        let data = {
            id: 'john-doe',
            op: 'clear-trophy',
            data: ''
        }
        ServerComm.ajaxPost(data, callback)
    }
}
```

```

static ajaxPost(data, callback) {
  let url = "http://localhost:8000"
  $.post(url, JSON.stringify(data))
    .done(function(data, textStatus, jqXHR) {
      //console.log('ajaxPost: ' + textStatus + ", " + data)
      // data possui o json resultado
      let jsonObj = JSON.parse(data)
      callback(jsonObj)
    })
    .fail(function(jqXHR, textStatus, errorThrown) {
      console.log('ajaxPost: ' + textStatus)
      // data possui o json resultado
      let jsonObj = JSON.parse(data)
      callback(jsonObj)
    })
}

}

// teste do ServerComm
ServerComm.clearTrophy((response) => console.log('clearTrophy: ' +
JSON.stringify(response)))

let trophyData = {name: "nuked", xp: 30, title: "COME GET SOME", description:
"Nuked the baddies", done: false}
ServerComm.addTrophy(trophyData, (trophyData) => console.log('addTrophy:
'+JSON.stringify(trophyData)))

//trophyData = {name: '10 coins', xp: 30, title: 'IT'S A START', description:
'Collected first 10 coins on the game', done: false}
//ServerComm.addTrophy(trophyData, (trophyData) => console.log('addTrophy:
'+JSON.stringify(trophyData)))

trophyData = {name: "talk hand", xp: 30, title: "TALK TO THE HAND", description:
"Let NPC talking", done: "false"}
ServerComm.addTrophy(trophyData, (trophyData) => console.log('addTrophy:
'+JSON.stringify(trophyData)))

let trophyName = '10 coins'
ServerComm.getTrophy(trophyName, (trophyData) => console.log('getTrophy:
'+JSON.stringify(trophyData)))

```

```
ServerComm.listTrophy((trophyData) => console.log('listTrophy:  
' + JSON.stringify(trophyData)))
```