



BONITA PROPERTIES

February 2017

SOMMAIRE

1.	INTRODUCTION.....	3
2.	GET / PUT.....	3
3.	ADMINISTRATION	3
4.	REST API.....	3
4.1.	Usage	3
4.2.	Admin REST	4
4.3.	Security.....	5

1. Introduction

2. Get / put

3. Administration

4. REST API

4.1. Usage

Two rest API is available to get and write a key.

In the parameter are (GET):

Name : the name is mandatory

Domain : the domain is optional

Key : the key of the value

When only the name is given, all keys attached to the name is returned (but not the domain).

When a domain is given, then the all keys attached to the name + domain is returned

When a key is give, then the value of the key is returned

```
GET: /bonita/API/extension/bonitaproperties?name=REST
{
  "airport": "south",
  "color": "rootblue",
  "base": "Z57",
  "REST": "low",
  "jam": "strawberry",
  "size": "low3"
}
```

```
GET: /bonita/API/extension/bonitaproperties?name=REST&domain=sfo
{
  "": "yellow",
  "color": "blue",
  "size": "medium"
}
```

```
GET: /bonita/API/extension/bonitaproperties?name=REST&domain=sfo&key=color
{
  "color": "blue"
}
```

A variable can be write using the PUT attributes

```
PUT : /bonita/API/extension/bonitaproperties?name=REST&domain=663&key=color&value=green663
```

```
PUT : /bonita/API/extension/bonitaproperties?name=REST&key=color&value=orange
```

It's possible to put a JSON (then multiple variable at the same time)

```
PUT
/bonita/API/extension/bonitaproperties?name=REST&domain=sfo&set={%22state%22:%20%22Californi
a%22,%22country%22%20:%22Sacramento%22,%22maincity%22:%22Los%20Angeles,%20San%20Francisco%22
}
```

4.2. Admin REST

The administration REST is used to get information

Action=initialize : the database is checked. This method should be call at initialization

```
PUT /bonita/API/extension/bonitapropsadmin?action=initialize
```

Action=readall

All variables are read.

Action=readall [&name=:name] [&domain=:domain] [&hierarchy=true]

```
PUT /bonita/API/extension/bonitapropsadmin?action=readall
{
  "values": {
    "custompage_AegrotatList": {
      "walter.bates": {
        "accessTask": "WINDOW",
        "showFilterContent": "false",
        "datasource": "NotManagedBizDataDS",
        "accesstask": "FRAME",
        "examinerName": "bmal002",
        "maxResult": "2000",
        "profiledisplay": "MODERATOR",
        "myCourseSynthesis": "true"
      },
      "datasource": "NotManagedBizDataDS",
      "bmal002": {
        "accessTask": "WINDOW",
        "showFilterContent": "true",
        "datasource": "NotManagedBizDataDS",
        "accesstask": "FRAME",
        "examinerName": "bmal002",
        "maxResult": "2000",
        "profiledisplay": "ALL",
        "myCourseSynthesis": "false"
      },
      "hran821": {
        "accessTask": "WINDOW",
        "showFilterContent": "false",
        "datasource": "NotManagedBizDataDS",
```

```

        "accesstask": "FRAME",
        "examinerName": "bmal002",
        "maxResult": "2000",
        "profiledisplay": "ALL",
        "myCourseSynthesis": "false"
    },
    },
    "custompage AegrotatAdmin": {
        "datasource": "NotManagedBizDataDS",
        "showFilterFee": "false"
    },
    },
    "listevents": "<table></table>",
    "status": "OK"
}

```

If a name is given, only the data from this name is returned. Same for the domain.

When hierarchy is given, and a domain the result gives the name of not.

```

GET:
/bonita/API/extension/bonitapropsadmin?action=readall&name=custompage AegrotatList&hierarchy
=false
{ "values": { "accessTask": "WINDOW", "showFilterContent": "false", "datasource":
"NotManagedBizDataDS", "accesstask": "FRAME", "examinerName": "bmal002", "maxResult":
"2000", "profiledisplay": "MODERATOR", "myCourseSynthesis": "true" }, "listevents": "
", "status": "OK" }

```

```

GET:
/bonita/API/extension/bonitapropsadmin?action=readall&name=custompage_AegrotatList&hierarchy
=true
{ "values": { "custompage_AegrotatList": { "walter.bates": { "accessTask": "WINDOW",
"showFilterContent": "false", "datasource": "NotManagedBizDataDS", "accesstask": "FRAME",
"examinerName": "bmal002", "maxResult": "2000", "profiledisplay": "MODERATOR",
"myCourseSynthesis": "true" } } }, "listevents": "
", "status": "OK" }

```

4.3. Security

You can protect the REST API by creating rule.

- If no rule is defined for the name, then all REST API GET/PUT are allowed
- The rule is used only by the REST API, not via the JAVA.

The rule can be created by some REST API or via the Custom page.

The rule has different parameters:

- RuleName : the name of the rule.
- Name : this parameter is mandatory. The rule apply on this name.
- Domain : if a domain is given, the rule apply only for the domain
- DomainIsUser : if true, then the rule apply only if the domain is equal to the userId.
- ReadOnly : the rule apply only for the GET api
- Admin : apply only if the user is register in the Administrator profile
- ApplyAllDomaines : apply whatever the domain (the parameter is not used)
- profileName : apply only when the user is register in this profile
- roleName : apply only when the user is register in the role

- GroupPath : apply if the user is register in the group
- username : apply only for the user.

Access Right

Filter on name Refresh

rulename	name	domain	Domain Is User	ReadOnly	Admin	All Domains	Profile	Role	Group	User Name
AllReadHelen	REST		false	true	false	true				helen.kelly
DomainWrite	REST		true	false	false					
RootAccess	REST		false	false	true	true				
RootRead	REST		false	true	false					
writelax	REST	lax	false	false	false	false				

Clean this name Clean

Rule Name*

name*

Domain

☐ Domain Is User

☐ ReadOnly

☐ Admin

☐ apply all Domains

Profile Name

Role Name

GroupPath

User Name

Add new rule

For example, for the rule :

Name

Rule Name	Name	Domain	Domain Is User	ReadOnly	Admin	All Domains	Profile	Role	Group	User Name
DomainWrite	REST		True	False	False	false				
RootRead	REST		False	True		false				
RootAccess	REST		False	False	True	true				

When the user Helen.Kelly (which is not an administrator) access by the REST API :

“name=REST&key=datasource”

Type	RESTAPI	Access ?
GET	name=REST&key=datasource	Yes (RootRead)
PUT	name=REST&key=datasource&value=AA	No (RootRead is only valable in ReadOnly)
GET	Name=REST&domain=sfo&key=color	No
GET	Name=REST&domain=657&key=color	True (DomainWrite) when userid=657
PUT	Name=REST&domain=657&key=color&value=blue	True

d

HEADQUARTERS

PARIS, FRANCE

76 boulevard de la République
92100 Boulogne-Billancourt

EMEA, ASIA & LATIN AMERICA

GRENOBLE, FRANCE

32, rue Gustave Eiffel
38000 Grenoble

NORTH AMERICA

SAN FRANCISCO, USA

44 Tehama Street
San Francisco, CA 94105

NEW YORK, USA

33 Nassau Avenue
Brooklyn NY 11222



www.bonitasoft.com