Mkey+
Documentation

Laboratoire Informatique et Systématique - Museum National d'Histoire Naturelle

October 11, 2013

Contents

1 Global description **2**
- 1.1 Package composition . . . 2
  - 1.1.1 mkey.html . . . 2
  - 1.1.2 mkey-mobile.html . . . 2
  - 1.1.3 description_page.html . . . 2
  - 1.1.4 index.html . . . 2
  - 1.1.5 Jquery . . . 2
  - 1.1.6 Bootstrap . . . 3
  - 1.1.7 mkey.js . . . 3
  - 1.1.8 mkey-interaction-function.js . . . 3
  - 1.1.9 mkey-spipoll-interaction.js . . . 3
  - 1.1.10 mkey-mobile.js . . . 3
  - 1.1.11 mkey_lang.js . . . 3
  - 1.1.12 PHP . . . 3
- 1.2 Installation . . . 3
  - 1.2.1 Requirements . . . 3
  - 1.2.2 Install . . . 4
  - 1.2.3 Advanced installation : Mkey webservice compilation and deployment . . . 4

2 Mkey+ IHM **5**
- 2.1 Descriptor's block . . . 5
- 2.2 Item's block . . . 6

3 Mkey+ API **8**
- 3.1 Javascript Object : *mkey* . . . 8
  - 3.1.1 mkey linked Javascript Object . . . 8
  - 3.1.2 function . . . 10
- 3.2 mkeyMobile . . . 15

A Example : Mkey+ static clients installation **16**

Chapter 1

Global description

Mkey+ static clients are static web clients using static *HTML 5* and *JavaScript* libraries, used to interface with Mkey+ webservice (using ***Mkey+ Api*** / ***Xper3 Api***).

1.1 Package composition

Four different clients are available in the WebContent folder. The first HTML file is (index.html) an index and a redirection html file which opens *mkey.html* or *mkey-mobile.html* depending on the client navigator.

1.1.1 mkey.html

Basic **Mkey**+ interface, this interface contains every basic mkey interface. User can customize every view by adding interface functions after having initialized the mkey object (see 3.1). This file contains absolute and relative path to Mkey+ javascript and css library (see 1.2).

1.1.2 mkey-mobile.html

Basic **Mkey**+ interface for mobile identification. (the mobile interface used for debuging is */mobile/mkey-mobile.html*)

1.1.3 description_page.html

Empty page, which is used to display the item's description exported from the **Mkey**+ interface

1.1.4 index.html

index file used to link to the mobile or desktop **Mkey**+ interface.

1.1.5 Jquery

Basic Jquery library

- jquery-2.0.3.min.js *mandatory*
- jquery-ui.js (1.10.3) *optional (desktop only)*
- jquery.Jcrop.min.js (0.9.12) *optional (spipoll & desktop only)*
  Jquery plug-in, jcrop is used in combination with PHP to crop image.

1.1.6 Bootstrap

Basic Bootstrap library

- bootstrap.min.js (2.0.3) *optional (desktop only)*
- FontAwsome

1.1.7 mkey.js

Basic mkey javascript library. *mandatory (desktop only)*

1.1.8 mkey-interaction-function.js

Contains every interaction between the html page and mkey object. *optional (desktop only)*

1.1.9 mkey-spipoll-interaction.js

Contains every additional interaction dependant on spipoll. *optional (spipoll & desktop only)*

1.1.10 mkey-mobile.js

Basic mkey javascript library. *mandatory (mobile only)*

1.1.11 mkey_lang.js

Contains every text message in several language and functions to update and retrieve them. *mandatory*

1.1.12 PHP

Constant.php

This file contain every constant used in the PHP files. When intalling **Mkey**+, this file should be wisely initialized (see 1.2).

crop.php

This file is used to crop the image.

upload_image.php

This file is used to upload images in the image sub-folder named mkey_upload

upload_sdd.php

This file is used to upload sdd in the sdd sub-folder named mkey_upload

1.2 Installation

1.2.1 Requirements

Installing **Mkey**+ static clients require an Apache web server with an enabled php module. (if PHP is not enabled, some feature will not work such as SDD import or image manipulation).

1.2.2 Install

Copy the WebContent folder where the static client should be installed and rename it with any name. The js and css files have been minified using Yui. The subfolder mkey_upload containing the subfolder image and sdd should have a public permission (see A.1).

Constant.php should be overriden to match the server configuration (see A.2), this file can be found in the *WebContent/php* folder. Depending on the utilisation the index.html file which redirects the connexion to both mkey-mobile.html or mkey.html may be overriden too (see A.3). The path to library used in Mkey+ can be both relative or absolute (via URLs), comment or uncomment which type or url's have to be used (see A.4).

After having configured index.html note that users have to check the mkey initialization function (see 3.1).

1.2.3 Advanced installation : Mkey webservice compilation and deployment

> Warning : If you do not have acces to the source repository of **Xper3 API**, **Mkey+ API** and **Mkey+ Webservice**, go immediatly to Tomcat deployment

Mkey+ static client is in constant relation with his associated webservice, the **Mkey+** webservice. This webservice is referenced when initializing the mkey javascript object (see 3.1) and in the Constant.php file (see 1.1.12).

Maven installation

Proceed in the following order :

1. Compile Xper3 API using the `mvn install` command, and add the generated xper3API.jar file in Mkey+ API lib's, and in the Mkey+ Webservice web library.
2. Compile Mkey+ API using the `mvn install` command, and add the generated mkeyAPI.jar file in the Mkey+ Webservice web library.
3. Compile Mkey+ Webservice using the `mvn install` command.

You should now have the mkey.war file.

Tomcat deployment

The mkey.war file should be copied in the webapps folder of a tomcat server. The tomcat server does not need to be restarted. In order to use the right service change every path to this web service (if the .war file name is mkey.war then the web service is called by http://theServeurDomaineName:****/mkey) (see 1.2).

Chapter 2

Mkey+ IHM

Mkey+ Static Client is designed in functionnal blocks. Depending on the displayed interface, two to three blocks are shown Fig.(2.1).

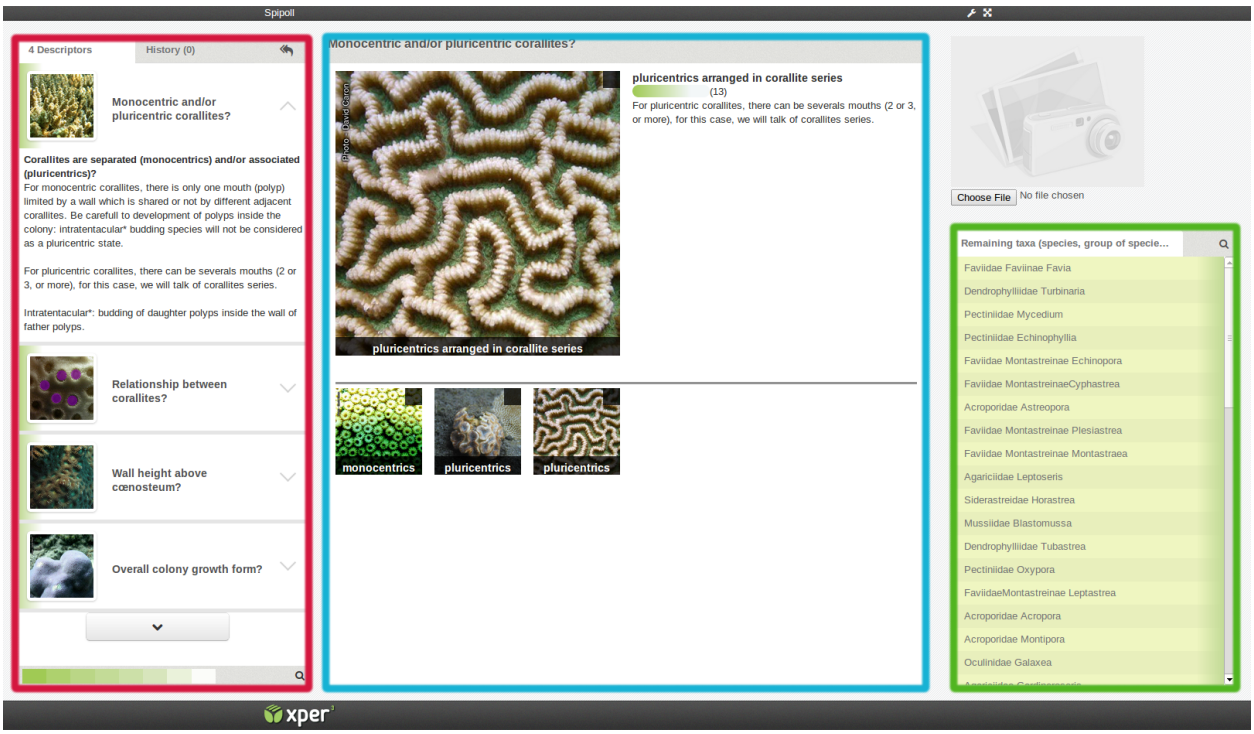

Figure 2.1: RED : the descriptor block,BLUE : the optionnal descriptor block, GREEN : the item's block.

2.1 Descriptor's block

This block contains a list of descriptors in order of discriminant weight (see Fig.2.2). Clicking on a descriptor enable makes his description editable by the user. Different options are also available on this block, such as looking at the previously described descriptor in the "history" view, reinitializing the current identification, adding descriptor or finding a specific descriptor.

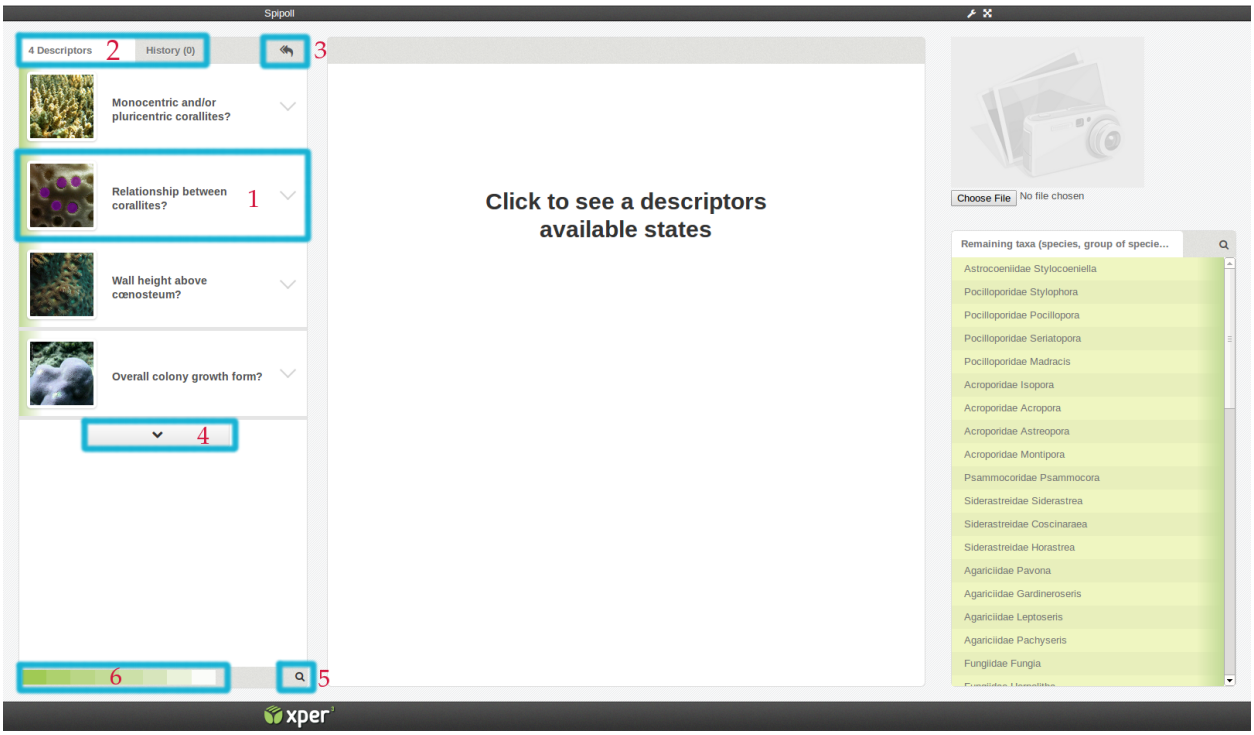

Figure 2.2: The Descriptor block is composed by a list of descriptor (1). User can expend this list by clicking on the button (2) or switch between the available and the already described descriptors (2). Neophyte user can have an idea of the discriminating power by looking on (6) while the expert user can find a specific descriptor by clicking on (5). Identification reinitialisation is fired when clicking on (3).

2.2 Item's block

The item's block (Fig.2.3) is a list of every item contained in the source SSD. Items are split into two category. The first one are the remaining items (item which match the current identification), and the seconde one the eliminated items (which diverge from the identification with one or more descriptor). The eliminated items are ordered in descrease similarity power order. The similarity is hightlighted by a more or less long green bar. Clicking on an item open his description. User may also acces a specific item by clicking on the loop icone.

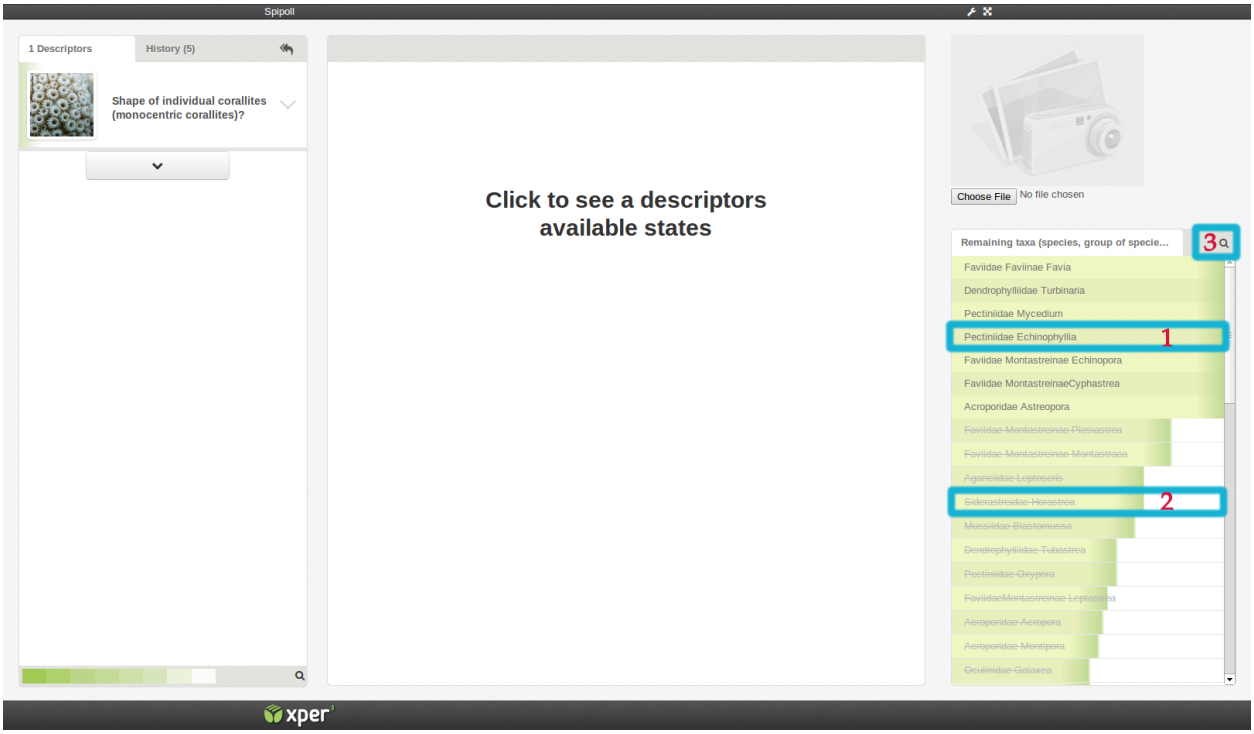

Figure 2.3: The item's block is composed of the list of every items. Each item (1), can be clicked on to display its entire description. Discarded items (2) are always displayed, depending on the similarity score the green bar is more or less expanded. Expert users can find a specific item by clicking on (3)

Chapter 3

Mkey+ API

3.1 Javascript Object : *mkey*

The mkey.js (1.1.7) defines a Javascript *mkey* object in the window scope. This object is used to interact with the Javascript mkey.js file private function.

First, to initialize this object, users need to use the *initMkey* function, here is an example of *mkey* initialization :

```
<script type="text/javascript">
  var mkeyServerURL = 'some public url to the Mkey+ Webservice';
  var sddURL = 'some public url';
  var interfaceID = 'some integer';
  initMkey(sddURL,mkeyServerURL,interfaceID);
</script>
```

Listing 3.1: mkey Javascript object initialization

The first parameter is the SDD url and can be an empty string, in this case mkey will be initialize d but won't display anything. The second and third parameter are mandatory and must be valid. InterfaceIDs values are 0 for default, 1 for row design and 2 for row square design. Additionnal configuration can be added for Javascript expert user by modifying the source code.

After having initialize this object, user have acces to the *mkey* object and its functions.

```
<script type="text/javascript">
  var mkeyServerURL = 'some public url to the Mkey+ Webservice';
  var sddURL = 'some public url';
  var interfaceID = 'some integer';
  initMkey(sddURL,mkeyServerURL,interfaceID);

  // Object can be accessed either by window.mkey or mkey
  mkey.setNumberOfDescriptor(10)

</script>
```

Listing 3.2: Example using mkey function

For more function see below.

3.1.1 mkey linked Javascript Object

mkey uses different data structure, in order to simplify futher notation we will define here some structures and reuse them later in the document.

Item

This object represents an item which can be described in a knowledge base. For taxonomists, it usually is a taxon.

item :
{
name : String, an item's name
alternativeName : String, an item's alternative name
detail : String, an item's detailed description
resourceIds : [long], the IDs of the resources associated to an item
id : long, an item's id, unique identifier generated sequentially by Mkey+
}

Descriptor

This object is a tool that serves to describe Items, essentialy a Character for taxonomists. A Descriptor can be described with States (if it is a categorical Descriptor), or a Quantitative Measure (If it is a quantitative Descriptor).
descriptor :
{
name : String, a Descriptor's name
detail : String, a Descriptor's detailed description
resourceIds : [long], the IDs of the resources associated to a Descriptor
stateIds : [long], the IDs of the states associated to a Descriptor
inapplicableState : [long], the states of a parent Descriptor for which a Descriptor is innaplicable
isCategoricalType : Boolean, true if the Descriptor is a categorical Descriptor, false otherwise
isQuantitativeType : Boolean, true if the Descriptor is a quantitative Descriptor, false otherwise
isCalculatedType : Boolean, true if the Descriptor is a calculated Descriptor, false otherwise
id : long, this Descriptor's id, an unique identifier generated sequentially by Mkey+
}

State

this object is a component of categorical Descriptors, e.g. for a Descriptor named "Color of the eye", its States could be "Blue", "Black", etc...
state :
{
name : String, a state's name
detail : String, a state's detailed description
resourceIds : [long], the IDs of the resources associated to a state
id : long, a state's id, an unique identifier generated sequentially by Mkey+
}

Resource

this object is a storage object, and is used to store media resources, which can be associated to several objects, such as Items, Descriptors, States, etc...
resource :
{
name : String, a resource's name
author : String, a resource's author
type : String, a resource's media type ("video","image","sound")
url : String, a resource's url
legend : String, a resource's legend
keywords : String, a resource's keywords
id : long, a resource's id, an unique identifier generated sequentially by Mkey+
}

DescriptionElement

This object stores the description of an Item, according to a single Descriptor, it represents the content of a single cell of the taxa / characters matrix. It may contain the list of selected States if the Descriptor is a categorical Descriptor, or a QuantitativeMeasure object, if the Descriptor is a quantitative Descriptor.
descriptionElement :

{
calculatedStates : [state], the calculated states representing an item's description
contextualWeight : int, the weigth of this description element
quantitativeMeasure : quantitativeMeasure, the quantitative measure re- -prensenting an item description
states : [state], the states representing an item's description
unknown : *boolean*, true if this description element is unkwnow
}

QuantitativeMeasure

This object is associated to a DescriptionElement object, for a given quantitative Descriptor and Item, it contains the quantitative measures used to describe a specific Item for a given quantitative Descriptor.
quantitativeMeasure :

{
min : long, this QuantitativeMeasure's minimum value
max : long, this QuantitativeMeasure's maximum
mean : long, this QuantitativeMeasure's mean
}

Description

This object represent a step in the identification process. Information stored have been selected by the user.
description :

{
descriptor : descriptor selectedStates : [*int*] ,
quantitativeMeasure : {min:*int*, max:*int*, mean:*int*}
}

3.1.2 function

getIdentificationData

Prototype function which can be used to retrieve every identification data. The data is retrieved in the callback function sent in parameter.

Parameter :

1. *function* callback(data) **(mandatory)**
data = {Items, Descriptors, States, Resources, DescriptorRootId, InvertedDependancyTable, descriptorsScoreMap}
Items = [item]
Descriptors = [descriptor]
States = [state]
Resources = [resource]
DescriptorRootId = [*int*]
InvertedDependancyTable = [{*int*:[*int*]}]
descriptorsScoreMap = {*int*:*int*}

submitDescription

Prototype function which can be used to submit a description. Information about remaining items and descriptors is retrieved in the callback function given in parameter.

Parameter :

1. [{ *int* : description}]descriptionList
2. [*int*] remainingItemsIDs (**mandatory**)
3. [*int*] discardedDescroptorsIds (**mandatory**)
4. *boolean* withScoreMap (**mandatory**)
5. *boolean* withGlobalWeigth (**mandatory**)
6. *function* callback(data) (**mandatory**)
   data = {remainingItems, discardedDescriptorsInIteration,descriptorsScoreMap}
   remainingItems = [item]
   discardedDescriptorsInIteration = [descriptor]
   descriptorsScoreMap = {*int*:*int*}

descriptionList,remainingItemsIDs,discardedDescroptorsIds, withScoreMap,withGlobalWeigth,callback

computeSimilarity

Prototype function which computes a similarity score based on the differences between a description and a list of items. The similarity computed for each item is returned in the data parameter of the callback function.

Parameter :

1. [description] descriptionsArray (**mandatory**)
2. [*int*] itemToEvaluate (**mandatory**)
3. *function* callback(data) (**mandatory**)
   data = {similarityMap}
   similarityMap = {*int*:*int*}

changeHistory

Prototype function which sets the history with the description array given in parameter. The data is then retrieved in the callback function given in parameter.

Parameter :

1. [description] descriptionArray (**mandatory**)
2. *function* callback(data) (**mandatory**)
   data = {remainingItems, discardedDescriptorsInIteration,descriptorsScoreMap, descriptions}
   remainingItems = [item]
   discardedDescriptorsInIteration = [descriptor]
   descriptorsScoreMap = {*int*:*int*}
   descriptions = [description]

setShowScore

Modify the identification interface to show or not the discriminant power next to the descriptor.

Parameter :

1. *boolean* showScore **(mandatory)**
2. *boolean* reload **(optionnal)**

setGraphicDisplayScore

Modify the identification interface to graphically (with a green gradient) display the discriminant power.

Parameter :

1. *boolean* graphicallyDisplayScore **(mandatory)**
2. *boolean* reload **(optionnal)**

setShowNumberOfRemainingItem

Modify the identification interface to show or not the number of remaining items if a specific state is selected (does not support multiple state selection).

Parameter :

1. *boolean* showNumberOfRemainingItem **(mandatory)**
2. *boolean* reload **(optionnal)**

setGraphicDisplayRemainingItem

Modify the identification interface to graphically (with a green gradiant) display the number of remaining items.

Parameter :

1. *boolean* graphicallyDisplayNumberOfRemainingItem **(mandatory)**
2. *boolean* reload **(optional)**

setNumberOfDescriptor

Change the number of descriptor displayed by default in the descriptor's block.

Parameter :

1. *int* numb **(mandatory)**

setSddFileUrl

Change the SDD file to the one passed in parameter.

Parameter :

1. *string(url)* SDDurl **(mandatory)**
2. *boolean* reloadIdentification **(optional)**
3. *function* callback **(optional)**

setWebserviceURL

Change the webservice URL, fire the callback function after having initialized every parameter.

Parameter :

1. *string(url)* urlWebService **(mandatory)**

setScoreType

(Not Implemented Yet) Change the score type (default = xper original sort)

Parameter :

1. *int* scoreID **(optional)**

setNumberOfStatesImg

Change the number of state displayed on the same line (row and rowSquare design only).

Parameter :

1. *int* numberOfStat **(mandatory)**

setPrincipalStateDim

Change the size of the focused state's image (rowSquare design only)

Parameter :

1. *int* size (between 1 and 11) **(mandatory)**

setRowDisplay

Change the global display of **Mkey**+ to row display or row display square. Set rowDisplay to true, set the row display, setting rowDisplay to true AND secondaryRowDisplay to true, set the row square design.

Parameter :

1. *boolean* rowDisplay **(mandatory)**
2. *boolean* secondaryRowDisplay **(optional)**
3. *boolean* reloadUI **(optional)**

setDefaultDisplay

Reset **Mkey**+ to default.

Parameter :

1. *boolean* reloadUI **(optional)**

setShowEndIdentification

Display or do not display an end identification interface that resume the resultat obtained.

Parameter :

1. *boolean* showEndIdentification **(mandatory)**

setShowStateWhitoutTaxon

Display or not the state which do not have any remaining taxon matching it.

Parameter :

1. *boolean* showStateWhitoutTaxon **(mandatory)**

setOnlyPositiveScore

Set if the descriptor which are not discriminant should to be shown. If set to true, the identification will end only when there is no descriptor left.

Parameter :

1. *boolean* showOnlyPositiveScore **(mandatory)**
2. *boolean* reloadUI **(optional)**

lockIdentification

Set the SDD file url to empty, in order to stop the identification process.

Parameter :

1. *function* callback

showConfigurationOption

Show or hide the configuration button. This option is used to enable or not the configuration menu.

Parameter :

1. *boolean* showOnlyPositiveScore **(mandatory)**
2. *boolean* reloadUI **(optional)**

setFixedDimension

Change the basic display from "fluid" (100%) to fixed dimension in pixel. This option only work on the default interface.

Parameter :

1. *boolean* showOnlyPositiveScore **(mandatory)**
2. *boolean* reloadUI **(optional)**

changeInterface

Change the global interface display to one of the predefined types (0 : default, 1 : rowDisplay, 2 : rowSquare, 3 : default with fixed dimension and without sdd, 4 : publish mode, 99 : spipoll mode.)

Parameter :

1. *int* theNewInterfaceType **(mandatory)**
2. *boolean* reloadUI **(optional)**

3.2 mkeyMobile

Mkey has a responsive interface named mkeyMobile. This simplifyed interface, enable users to identify taxa on mobile devices.

No futher information are disponible for now.

Appendix A

Example : Mkey+ static clients installation

```
#!/bin/sh

#Basic Shell Script to update mkey.html library

#delete the Old version of Mkey
rm -rf /var/www/.../mkeyplus

#Move WebContent (new mkeyplus) to identificationToolsWebSite and rename it to mkeyplus
mv WebContent /var/www/.../mkeyplus

#Make the mkey_upload sub folder accessible for download and links
chmod -R 777 /var/www/.../mkeyplus/mkey_upload
```

Listing A.1: sh script to update Mkey+ static clients on any server

```
//The Tomcat serveur where mkey+ Webservice is running
$host_tomcat="http://some.tomcat.serveur:****/";
//The current Apache serveur where the web client is deployed
$host_apache="http://some.apache.serveur(:80)/";
//The absolute path to mkey_upload on the apache serveur machine
$uploads_dir="/var/www/.../mkey/mkey_upload/";

//The relative path to mkey_upload ( do not change )
$uploads_dir_short ="mkey/mkey_upload/";
//The relative path to image/ ( do not change )
$uploads_dir_image ="image/";
// The relative path to sdd/ ( do not change )
$uploads_dir_sdd = "sdd/";
```

Listing A.2: PHP code inside Constant.php to update when reinstalling Mkey+ static clients

```
...
<script type="text/javascript">
//Test if the navigator is using a mobile user agent
if (navigator.userAgent.match(/(android|iphone|blackberry|symbian|symbianos|symbos|
    netfront|model-orange|javaplatform|iemobile|windows phone|samsung|htc|opera mobile|
    opera mobi|opera mini|presto|huawei|blazer|bolt|doris|fennec|gobrowser|iris|maemo
    browser|mib|cldc|minimo|semc-browser|skyfire|teashark|teleca|uzard|uzardweb|meego|
    nokia|bb10|playbook)/gi) || navigator.userAgent.match(/ipad/gi)) {
  //If a mobile user agent is used launch the mkey-mobile.html file
  // The "mobile/mkey-mobile.html" can be changed or removed to only launch the desktop
      version
  window.location = window.location+"mobile/mkey-mobile.html";
}
else{
  //Else launch the desktop version
```

```
    // Change "mkey.html" to launch another page
    window.location = window.location+"mkey.html";
  }
  </script>
  ...
```

Listing A.3: Example of index.html customization

```
<!-- ********************************************************************************** -->
<!-- ************************** Mkey with relative Path ***************************** -->
<!-- ********************** to use in identificationkey.fr ************************** -->

<title>Mkey +</title>
<!-- The styles -->
<link rel="stylesheet" href="http://docbrown.snv.jussieu.fr/xper3/resources/css/xper3.css" />
<link rel="stylesheet" href="./lib/bootstrap/css/bootstrap.min.css" />
<link rel="stylesheet" href="//netdna.bootstrapcdn.com/font-awesome/3.2.1/css/font-awesome.css" >
<link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" />
<link rel="stylesheet" href="./css/mkey.css" />
<link rel="stylesheet" href="./css/integrationStandalone.css" />
<link rel="stylesheet" href="./css/override-xper3.css" />

<!-- The javascript -->

<script src="http://code.jquery.com/jquery-2.0.3.min.js"></script>
<script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script>
<script src="./lib/bootstrap/js/bootstrap.min.js" ></script>
<script src="./js/mkey-interaction-function.js"></script>
<script src="./js/lang/mkey_lang.js"></script>
<script src="./js/mkey.js" ></script>

<script src="./lib/Highcharts3_1/js/highcharts.js"></script>
<script src="./lib/Highcharts3_1/js/modules/exporting.js"></script>

<!-- ************************** END - Mkey with relative Path ************************ -->
<!-- ********************************************************************************** -->


<!-- ********************************************************************************** -->
<!-- ************************** Mkey with absolute Path ***************************** -->
<!-- ************************** to use in xper3 publish ***************************** -->

<!-- <title></title> -->

<!-- The styles -->

<!-- <link rel="stylesheet" href="http://docbrown.snv.jussieu.fr/xper3/resources/css/xper3.css" /> -->
<!-- <link rel="stylesheet" href="http://www.identificationkey.fr/mkeyplus/lib/bootstrap/css/bootstrap.min.css" /> -->
<!-- <link rel="stylesheet" href="//netdna.bootstrapcdn.com/font-awesome/3.2.1/css/font-awesome.css" > -->
<!-- <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.3/themes/smoothness/jquery-ui.css" /> -->
<!-- <link rel="stylesheet" href="http://www.identificationkey.fr/mkeyplus/css/mkey.css" /> -->
<!-- <link rel="stylesheet" href="http://www.identificationkey.fr/mkeyplus/css/override-xper3.css" /> -->

<!-- Javascript -->

<!-- <script src="http://code.jquery.com/jquery-2.0.3.min.js"></script> -->
<!-- <script src="http://code.jquery.com/ui/1.10.3/jquery-ui.js"></script> -->
<!-- <script src="http://www.identificationkey.fr/mkeyplus/lib/bootstrap/js/bootstrap.min.js" ></script> -->
```

```
<!-- <script src="http://www.identificationkey.fr/mkeyplus/js/mkey.js" ></script> -->
<!-- <script src="http://www.identificationkey.fr/mkeyplus/js/mkey-interaction-function.js" ></script> -->
<!-- <script src="http://www.identificationkey.fr/mkeyplus/js/lang/mkey_lang.js" ></script> -->


<!-- ************************* END - Mkey with absolute Path ********************** -->
<!-- ******************************************************************************** -->
```

Listing A.4: Example of local path to library implementation