# **BASE** API Specification

## SJTU-Software 2015

There are seven main modules in the design of our **BASE** project, shown as below.

1. **judge**

   - Description

     A function to find out whether the biobrick of a certain type is in the database or not, and returns result by printing "$Yes$" or "$No$".

   - Input

     *object*: the biobrick ID user wants to know

     *type*: the type of the biobrick

     *host*: the host of the database

     *dbname*: the name of the database

     *username*: the user name

     *password*: the password of the user

   - Output

     Returns result by printing "$Yes$" or "$No$".

2. **baseadvice**

   - Description

     A function to find several biobricks with higher scores based on the existing biobricks, the type of the biobrick, and its description, and returns result by printing the biobricks and their score.

   - Input

     *object*: the biobrick ID of the existing biobricks, the biobrick IDs are split by "$*$"

     *type*: the type of the biobrick

     *func*: the description of the biobrick

     *host*: the host of the database

     *dbname*: the name of the database

     *username*: the user name

     *password*: the password of the user

   - Output

     Returns result by printing the biobricks and their score.

3. **gscore**

   - Description

     A function to score the biobrick based on the features of biobrick and the weight of features, and returns the score of biobrick and scores of features.

   - Input

     *Part_status*: the part status

     *Sample_status*: the sample status

     *Part_results*: the part results

*Star_rating*: the star rating

*Uses*: the number of usage

*DNA_status*: the DNA status

*Qualitative_experience*: the biobrick's qualitative experience status

*Group_favorite*: the biobrick's favorite group

*Del*: whether the biobrick is deleted or not

*Groups*: group the biobrick belongs to

*Confirmed_times*: the number of being confirmed

*Number_comments*: the number of comments

*Ave_rating*: the average rating

*Power*: the weight of features set by user, split by "*"

- Output

  Returns the score of biobrick and scores of features.

4. **basesearch**

   - Description

     A function to search the biobrick according to the name, type or description of the biobricks, and returns the biobrick and its feature. The biobricks are ranked by the score in the descending order.

   - Input

     *object*: the biobrick ID or type or description about the biobricks, multiple keywords are split by "*"

     *type*: the type of the biobrick, *part* or *device*

     *hPower*: the weight of features set by user, split by "*"

     *limit*: the score limit, the biobrick with the lower score will be ignored

     *host*: the host of the database

     *dbname*: the name of the database

     *username*: the user name

     *password*: the password of the user

   - Output

     Returns the biobrick and its feature.

5. **evaluate_score**

   - Description

     A function to calculate the score of a device, and returns the score of the device.

   - Input

     *Device*: the biobrick ID of these parts within a device, the biobrick IDs are split by ","

   - Output

     Returns the device score.

6. **evaluate**

   - Description

     A function to judge whether parts of the device are complete and ordered, to calculate the score of the device and to search the better part, and returns the score and the best ten parts we evaluate.

   - Input

     *device*: the biobrick ID of these parts within a device, the biobrick IDs are split by ","

     *description*: the keyword of these parts within a device, the biobrick IDs are split by ",", and split by "+" if there are more than two keywords for a part.

     *cutoff*: precision of the evaluation, the default precision is 20 for low level, 40 for middle level, 80 for high level.

- Output

  In the first line, it judges the completion, 0 for uncomplete and 1 for complete(the completion means if there are Regulatory, Coding and Terminator part)

  In the second line, it judges the order, 0 for unordered and 1 for ordered(the order means if the first part is Regulatory, the last part is Terminator and the coding part is between Regulatory and Terminator)

  In the third line, it outputs the score of the device.

  The following lines show the biobrick ID the user inputs and evalute the best ten parts for each part according to their types and keywords.

7. **upload**

   - Description

     A function to upload the basic information of the new biobrick to the database.

   - Input

     *object*: the basic information of the biobrick, split by "∗"

     *type*: the type of the biobrick, *part* or *device*

     *host*: the host of the database

     *dbname*: the name of the database

     *username*: the user name

     *password*: the password of the user

   - Output

     Uploads to the database.