

DEVELOPMENT TOOLS

```
ganesh — basejmp — 145x40
> basejmp

BASEJMP

BASEJMP >> help
basejmp 0.1.0

USAGE:
  basejmp [OPTIONS] [SUBCOMMAND]

FLAGS:
  --help      Prints help information
  -V, --version Prints version information

OPTIONS:
  -h, --host <host>      [default: http://admin.basejmp.io]
  -o, --output <output>  [default: rich-text] [possible values: rich-text, plain-text, json]
  -p, --port <port>      [default: 80]

SUBCOMMANDS:
  access-rule  Sub-command to handle access rules
  auth-pool    Sub-command to handle authorization pool management activities
  cache        Sub-command to handle Cache managment activities
  column       Sub-command to handle Table Column managment activities
  deploy       Sub-command to handle deployment from yaml
  domain       Sub-command to handle management of Domains
  help         Prints this message or the help of the given subcommand(s)
  hosting       Sub-command to handle hosting management activities
  lambda       Sub-command to handle management of JS Functions
  login        Login to the Basejmp back-end
  path         Sub-command to handle management of Path's
  path-group   Sub-command to handle management of Path groups,
  queue        Sub-command to handle Queue managment activities
  search       Sub-command to handle Search managment activities
  space        Sub-command to handle admin activities of a Basejmp space
```

INTERACTIVE CLI



The web based console is complemented by an interactive CLI tool that allows to achieve via command-line everything that can be achieved via the web console.

DEVELOPMENT TOOLS

DECLARATIVE YAML

Declarative YAML based configuration allows automation of app creation as well as version control of created apps.

```
hcm.yml - workarea

1  ---
2  Space:
3    SpaceName: hcm
4  ---
5  Store:
6    StoreName: employeeedb
7    SpaceName: hcm
8    StoreType: rowstore
9
10 Table:
11   TableName: employee_details
12   SpaceName: hcm
13
14 Columns:
15   - { ColumnName: employeeid, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
16   - { ColumnName: first_name, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
17   - { ColumnName: last_name, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
18   - { ColumnName: email_address, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
19   - { ColumnName: phone_number, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
20   - { ColumnName: address, ColumnType: varchar, ColumnConstraints: none, SpaceName: hcm }
21
22 TableStoreAssociation: { AssociationName: employeeedb_details, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" ,StoreId: "{{ Store.employeeedb.id }}" }
23
24 TableColumnAssociations:
25   - { AssociationName: details_id, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.employeeid.id }}" }
26   - { AssociationName: details_first_name, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.first_name.id }}" }
27   - { AssociationName: details_last_name, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.last_name.id }}" }
28   - { AssociationName: details_email_address, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.email_address.id }}" }
29   - { AssociationName: details_phone, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.phone_number.id }}" }
30   - { AssociationName: details_address, SpaceName: hcm, TableId: "{{ Table.employee_details.id }}" , ColumnId: "{{ Column.address.id }}" }
31
32 SQLAPI:
33   APIName: insert_employee
34   SpaceName: hcm
35   SQLTemplate: >-
36     INSERT INTO employee_details (employeeid, first_name, last_name, email_address, address, phone_number)
37     VALUES('{{employeeid}}', '{{first_name}}', '{{last_name}}', '{{email_address}}', '{{address}}', '{{phone_number}}')
38   StoreId: "{{ Store.employeeedb.id }}"
39   ResponseColumnNames: ""
40   ResponseContentType: "text/json"
41
42 SQLAPI:
43   APIName: get_employee
44   SpaceName: hcm
45   SQLTemplate: >-
46     SELECT employeeid, first_name, last_name, email_address, address, phone_number
47     FROM employee_details
48     WHERE employeeid = '{{employeeid}}'
49   StoreId: "{{ Store.employeeedb.id }}"
50   ResponseColumnNames: "employeeid,first_name,last_name,email_address,address,phone_number"
51   ResponseContentType: "text/json"
```