bw2-rest-api Documentation

Release 1.2

Chris Mutel and ETH Zürich

March 06, 2015

Contents

This is documentation for version **1** of the Brightway2 REST API.

CHAPTER 1

URLS

- `/api/database/`: Get list of available databases. Method: GET.
- `/api/database/<database>/`: CRD for databases and database metadata. Database metadata sent as JSON. Methods: GET, POST, DELETE.

Note: To create a database with the JSON backend, the POST request body defining database metadata (which is a hash table) should include the following: `{"backend": "json"}`.

- `/api/database/<database>/activity/`: Get list of activities for database `database`. Method: GET.
- `/api/database/<database>/activity/<activity>/`: CRUD for activities. Methods: GET, POST, PUT, DELETE.

CHAPTER 2

Headers for POST/PUT Data

Data in POST and PUT bodies should be JSON. The `Content-Type` header should be set to `application/json` for the request bodies to be decoded correctly.

CHAPTER 3

Examples

3.1 /api/database/

```
curl 127.0.0.1:8000/api/database/
```

Returns:

```
["biosphere", "ecoinvent 2.2", "playground"]
```

3.2 /api/database/<database>/

```
curl -X POST -H "Content-Type:  application/json" -d '{"foo":"bar", "backend":
"json"}' 127.0.0.1:8765/api/database/nonsense/
```

Returns: 201.

```
curl 127.0.0.1:8765/api/database/nonsense/
```

Returns:

```
{"depends": [], "foo": "bar", "backend": "json"}
```

```
curl -X DELETE 127.0.0.1:8765/api/database/nonsense/
```

Returns: 204.

3.3 /api/database/<database>/activity/

```
curl 127.0.0.1:8765/api/database/playground/activity/
```

Returns:

```
[["playground", "hi there"]]
```

3.4 /api/database/<database>/activity/<activity>/

```
curl 127.0.0.1:8765/api/database/playground/activity/hi%20there/
```

Returns:

```
{"hey everybody": ":)", "key": ["playground", "hi there"]}
```

```
curl 127.0.0.1:8765/api/database/ecoinvent%202.2/activity/489ef3bdb7c07e91aaa53ac2ab229d2f/
```

Returns:


```
{
   "name":"electricity, production mix photovoltaic, at plant",
   "categories":[
      "photovoltaic",
      "power plants"
   ],
   "exchanges":[
      {
         "comment":"(2,2,1,1,1,3); Calculation with average module efficiency",
         "loc":1.3481510673418475,
         "code":3721,
         "group":4,
         "name":"Energy, solar, converted",
         "uncertainty type":2,
         "categories":[
            "resource",
            "in air"
         ],
         "negative":false,
         "url":"/api/database/biosphere/activity/b529ebb9304c5c31c4850b0f2ac4363e/",
         "amount":3.8503,
         "scale":0.043409845976280094,
         "location":null,
         "input":[
            "biosphere",
            "b529ebb9304c5c31c4850b0f2ac4363e"
         ],
         "type":"biosphere",
         "unit":"megajoule"
      }, ...
   ],
   "unit":"kilowatt hour",
   "location":"JP"
}
```


The details of what fields are required in the brightway2 documentation. Note that the exchange fields "name", "unit", "location", "categories", and "url" are added by the API for convenience, and don't need to be added when creating or updating datasets (they are stripped when saving the dataset).

```
curl -X POST -H "Content-Type: application/json" -d '{"name":"a name",
"exchanges": [{"input": ["playground", "hi there"], "amount": 1}]}'
127.0.0.1:8765/api/database/playground/activity/new-process/
```

Returns: 201.

```
curl -X PUT -H "Content-Type: application/json" -d '{"name":"a different
name!", "exchanges": [{"input": ["playground", "hi there"], "amount": 1}]}'
127.0.0.1:8765/api/database/playground/activity/new-process/
```

Returns: 200.

```
curl -X DELETE 127.0.0.1:8765/api/database/playground/activity/new-process/
```

Returns: 204.

CHAPTER 4

Running the API

Installation will create a script `bw2-restapi` that can be invoked to run the API. See `bw2-restapi -h` for script options.

CHAPTER 5

Database API

class bw2restapi.database.**DatabaseAPI**

The base URL for activities is:

/api/database/<database name>/

The following REST methods are available: GET, POST, DELETE.

delete(*database*)

Delete an existing database.

Response codes:

- 200: Resource delete succeeded
- 404: Resource not found

get(*database*)

Get database metadata.

Response codes:

- 200: Resource retrieved
- 404: Resource not found

post(*database*)

Create a new database.

Database metadata data is optional. If provided, it should be a JSON hash table, and its values will be passed to the Database.register() method. To create a database with the JSON backend, send `{"backend": "json"}`.

Response codes:

- 201: Database created
- 400: Can't decipher post data to valid JSON hash table
- 409: Conflict - resource already exists

CHAPTER 6

Activity API

class bw2restapi.activity.**ActivityAPI**

The base URL for activities is:

/api/database/<database name>/activity/<activity key>/

The following REST methods are available: GET, POST, PUT, DELETE.

delete(*database*, *activity*)

Delete an existing activity dataset.

Response codes:

- 200: Resource delete succeeded
- 404: Resource not found

get(*database*, *activity*)

Get an activity dataset.

Response codes:

- 200: Resource retrieved
- 404: Resource not found

post(*database*, *activity*)

Create a new activity dataset.

Response codes:

- 201: Activity dataset created
- 400: No JSON body could be decoded
- 404: Database not found
- 409: Conflict - resource already exists

put(*database*, *activity*)

Update an existing activity dataset.

Response codes:

- 200: Resource update succeeded
- 400: No JSON body could be decoded
- 404: Resource not found

A

D

G

P