



APIs Short User Guide (v.1)

Date: February 6th, 2019

Edited by: Teodoro Laino



Summary

| | |
|----------------------------------------------------------|----|
| 1. INTRODUCTION | 3 |
| 1.1. Fair use policy | 3 |
| 2. API calls | 3 |
| 2.1. Creation and Recovery of an API key | 3 |
| 2.1.1. Common data structure of API calls | 4 |
| 2.2. First, create a project | 5 |
| 2.2.1. API response code | 5 |
| 2.2.2. Example with curl | 6 |
| 2.3. How to create a new prediction | 6 |
| 2.3.1. API response code | 7 |
| 2.3.2. API Response code (in case of errors) | 8 |
| 2.3.3. Example with curl | 8 |
| 2.4. Recover prediction attempts using the prediction ID | 10 |
| 2.4.1. API response code | 10 |
| 2.4.2. Example with curl | 12 |
| 2.5. List all stored projects | 13 |
| 2.5.1. API response code | 13 |
| 2.5.2. Example with curl | 14 |
| 2.6. List all attempts in a project | 16 |
| 2.6.1. API response code | 16 |
| 2.6.2. Example with curl | 18 |



1. INTRODUCTION

This is a short user guide to allow the usage of the IBM® RXN for Chemistry services in third party applications through APIs calls.

The entire system is based on an authentication mechanism via APIs keys. Moreover, any actions performed through APIs will be appearing as well in the user dashboard when accessing the service through the web-services.

Please, note that the access to the portal through APIs is done under best effort and subject to changes without prior notice. APIs are provided on an "as is" basis, without warranty of any kind.

1.1. Fair use policy

In order to guarantee a fair usage of the free service to anybody on Earth, the calls made via the API interface will be limited to max. 6 requests per minute with a minimum time of 2 seconds between two consecutive calls. Please, note that the system orchestrating the API calls ignores if a call is successfully executed or not. For example, let's consider the following scenario:

| | | | | | | |
|-------|-------|-------|-------|-------|--------|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | | | | | | |
| 100ms | 300ms | 500ms | 700ms | 900ms | 2200ms | |

The call (1) will be successfully executed. From call (2) to call (6) the system will return an error message (**503 Service Unavailable**) which means that the server is currently unable to handle the request due to a usage policy (too short time between consecutive calls). The implication is that this is a temporary condition which will be alleviated at the beginning of the next minute.

Call (7) is respecting the 2 seconds minimum time but is actually the 7th request done within the same minute and for this reason will return the same error (503). All the counters will be reset at the start of the subsequent clock minute.

If interested in high-throughput usage of these APIs please get in touch with the IBM® RXN for Chemistry team (<https://rxn.res.ibm.com>) for dedicated cloud instances or on premise installation.

2. API calls

2.1. Creation and Recovery of an API key

To use the IBM® RXN for Chemistry services you have first to generate an API key.

To generate the token, log-in into your personal account page of RXN and go on your profile page (top-right corner of the main dashboard). Below your personal information, you will find an API key box. Click on the "GENERATE" button and it will create your own token (starting with apk-...) to use in the subsequent API calls:



Unlike the authentication mechanism for accessing the services through a browser, this API key does not expire (there is no time session limit) and must not be shared with other people to preserve your own privacy.

2.1.1. Common data structure of API calls

All API calls need to have the base URL of the services: *BaseUrl: rxn.res.ibm.com*

Moreover, the header of all calls must contain:

Content-Type: application/json
Authorization: apk-...

The structure of the answer to the request is:

```
{
  Payload: {}
  "metadata": {
    "uiMessages": {
      "errors": [],
      "infos": [],
      "warnings": []
    }
  }
}
```

Payload contains the real answer to the service request and the metadata section contains errors, info or warnings (if any).



2.2. First, create a project

To run prediction, similarly to the web-interface, you have first to create a project folder where your prediction will be stored.

```
URL: BaseUrl+/rxn/api/api/v1/projects
Method: POST
Body
{
  "name": "Test API",
  "invitations": [{
    "email": "email@email.nop"
  }]
}
```

Mandatory fields:

```
name
invitations (can be empty, but must be present)
```

2.2.1. API response code

```
{"payload": {
  "id": "5c2c95fed6cb7600019e6f17",
  "createdOn": 1546425854026,
  "createdBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
  "modifiedOn": 1546425854026,
  "modifiedBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
  "name": "Test API",
  "description": null,
  "attempts": [],
  "visibility": null,
  "computedFields": {},
  "embed": {},
  "metadata": {},
  "metadata": {...}
}}
```

The API response code contains several information among which, the “id” is the most relevant to proceed with the prediction capabilities.



2.2.2. Example with curl

```
~> curl --data '{"name": "Test API", "invitations":[]}' --header "Content-Type: application/json" --header "Authorization:apk-have-here-your-own-key" -X POST https://rxn.res.ibm.com/rxn/api/api/v1/projects
```

Output:

```
{"payload":{"id":"5c532f56d6cb7600019ea34NV","createdOn":1548955478522,"createdBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63cNV","modifiedOn":1548955478522,"modifiedBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63cNV","name":"Test API","description":null,"attempts":[],"visibility":null,"computedFields":{"embed":{"metadata":{"uiMessages":{"errors":[],"infos":[],"warnings":[]}}}}}
```

2.3. How to create a new prediction

To create a new prediction it is necessary to know the project folder ID (see 2.2. First, create a project) where the result of the prediction will be stored. As seen in the previous section the ID is a parameter returned by the API response code.

In our example, for the project “Test API” we obtained a project ID of “id”:“5c2c95fed6cb7600019e6f17”. Given a smile, representing a single or multiple number of input molecules (such as:

C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].[H]), the API call for the prediction is constructed as:

```
URL: BaseUrl + /rxn/api/api/v1/predictions/pr?projectId=...
Method: POST
Body
{
  "reactants": "C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].[H]"
}
```



Mandatory fields:

projectId = the project ID returned when the project folder is created
reactants = the smile representing the set of starting molecules

2.3.1. API response code

As for the project ID, the response code of a prediction contains the prediction ID, which is needed to recover later the result of the prediction.

```
{"payload": {
  "id": "5c2cab97d6cb7600019e6f21",
  "createdOn": 1546431383554,
  "createdBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
  "modifiedOn": 1546431383554,
  "modifiedBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
  "request": {
    "reactants":
      "C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].
      [H]",
    "reagents": [],
    "mol": ""
  },
  "projectId": "5c2c95fed6cb7600019e6f17",
  "taskId": "e8fc2e28-a11d-429a-9821-3e409ef739a1",
  "mol": "",
  "type": "pr",
  "status": "NEW",
  "attempts": null,
  "computedFields": {},
  "embed": {},
  "metadata": {}
},
"metadata": {...}
}
```



2.3.2. API Response code (in case of errors)

In case of malformed or erroneous SMILES, or in case of prediction errors, the API response will try to do its best to provide a feedback on why the prediction request failed.

An example of a failure:

```
{
  "payload": null,
  "metadata": {
    "uiMessages": {
      "errors": [
        {
          "code": null,
          "message": "ExplicitValenceError, RDKit ERROR: [12:39:34] Explicit
valence for atom # 0 H, 3, is greater than permitted - reactants: [H](C)(C)C.C.CC,
400, RDKitError",
          "type": "ERROR",
          "fieldId": null,
          "target": "TOAST"
        }
      ],
      "infos": [],
      "warnings": []
    }
  }
}
```

2.3.3. Example with curl

```
curl --data '{"reactants" :
"C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].
[H]", "mol":""}' --header "Content-Type: application/json" --header "
Authorization:apk-have-here-your-own-key" -X POST
https://rxn.res.ibm.com/rxn/api/api/v1/predictions/pr?projectId=5c532f56d6cb7600019ea342
```



Output:

```
{"payload":{"id":"5c532fa7d6cb7600019ea345","createdOn":1548955559319,"createdBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","modifiedOn":1548955559319,"modifiedBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","request":{"reactants":"C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].[H]","reagents":[],"mol":"","projectId":"5c532f56d6cb7600019ea342","taskId":"7604bfc3-de1c-4569-b991-73efb1c7d738","mol":"","type":"pr","status":"NEW","attempts":null,"computedFields":{"embed":{"metadata":{"uiMessages":{"errors":[],"infos":[],"warnings":[]}}}}1
```



2.4. Recover prediction attempts using the prediction ID

Using the prediction ID returned in the API prediction response code, one can recover the attempt outcome generated in the corresponding prediction:

URL: *BaseUrl + /rxn/api/api/v1/predictions/<IDPREDICTION>*
Method: *GET*

Mandatory fields:

IDPREDICTION (es: *5c2cab97d6cb7600019e6f21*)

2.4.1. API response code

La risposta a questa richiesta contiene oltre che lo smiles i dati riguardanti l'accuratezza della risposta AttentionWeights (SVG in Base64) e le immagini delle molecole (SVG in Base64)

```
{
  "payload": {
    "id": "5c2cab97d6cb7600019e6f21",
    "createdOn": 1546431383554,
    "createdBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
    "modifiedOn": 1546431395414,
    "modifiedBy": "system",
    "request": {
      "reactants":
        "C(O)1=C(O)C=C(C2=C(O)C=C([H])C([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].
        [H]",
      "reagents": [],
      "mol": ""
    },
    "projectId": "5c2c95fed6cb7600019e6f17",
    "taskId": "e8fc2e28-a11d-429a-9821-3e409ef739a1",
    "mol": "",
    "type": "pr",
    "status": "SUCCESS",
    "attempts": [
      {
        "id": "5c2caba3d6cb7600019e6f22",
        "createdOn": 1546431395402,
```



```
"createdBy": "system",
"modifiedOn": 1546431395451,
"modifiedBy": "system",
"name": "Test API test_20190102_12:16:35.401",
"projectId": "5c2c95fed6cb7600019e6f17",
"smiles": "Oc1cc(-c2c(O)cccc2-c2ccccc2)cc(O)c1O.[H].[H].[H]>>Oc1cc(-
c2c(O)cccc2-c2ccccc2)cc(O)c1O",
"attentionWeights": "<?xml version='1.0' encoding='utf-8'
standalone='no'?>\n<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\" \n
... .. y='42.960226' />\n </clipPath>\n </defs>\n</svg>\n",
"confidence": 0.13361498713493347,
"message": "ok",
"reactionImage": "<?xml version='1.0' encoding='iso-8859-1'?>\n<svg
version='1.1' baseProfile='full'\n      xmlns='http://www.w3.org/2000/svg'\n
xmlns:rdkit='http://www.rdkit.org/xml'\ ... .. <path d='M 695.891,96.8815
700.569,100' style='fill:none;fill-rule:evenodd;stroke:#000000;stroke-
width:2px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1'
/>\n</svg>\n",
"predictionId": "5c2cab97d6cb7600019e6f21",
"feedbacks": null,
"firstAttempt": true,
"predictTime": null,
"totalTime": null,
"computedFields": {},
"embed": {},
"metadata": {}
}
],
"computedFields": {},
"embed": {},
"metadata": {}
},
"metadata": {...}
}
}
```



2.4.2. Example with curl

```
curl --header "Content-Type: application/json" --header " Authorization: apk-  
have-here-your-own-key" -X GET  
https://rxn.res.ibm.com/rxn/api/api/v1/predictions/5c532fa7d6cb7600019ea345
```

Output:

```
{"payload":{"id":"5c5aaf22d6cb7600019ea809","createdOn":1549446946367,"cr  
eatedBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8",  
"modifiedOn":1549446950420,  
"modifiedBy":"system","request":{"reactants":"C(O)1=C(O)C=C(C2=C(O)C=C([H])C  
([H])=C2C2=C([H])C=C([H])C=C2)C=C1O.[H].[H].[H]","reagents":[],"mol":"","proje  
ctId":"5c532f56d6cb7600019ea342","taskId":"af0e8aa5-c2a9-4819-9e41-  
ad7e40421090","mol":"","type":"pr","status":"SUCCESS","attempts":[{"id":"5c5a  
af26d6cb7600019ea80a","createdOn":1549446950408,"createdBy":"system","m  
odifiedOn":1549446950583,"modifiedBy":"system","name":"Test  
API_20190206_09:55:50.407","projectId":"5c532f56d6cb7600019ea342","smiles  
":"Oc1cc(-c2c(O)cccc2-c2cccc2)cc(O)c1O.[H].[H].[H]>>Oc1cc(-c2c(O)cccc2-  
c2cccc2)cc(O)c1O","attentionWeights":"<?xml version='1.0' encoding='utf-  
8' standalone='no' ?>\n<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG  
1.1//EN\" \n \"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd\">\n [... ...  
...]  
","confidence":0.13361498713493347,"message":"ok","reactionImage":"<?xml  
version='1.0' encoding='iso-8859-1' ?>\n[.... ... ]"},  
"predictionId":"5c5aaf22d6cb7600019ea809","feedbacks":null,"firstAttempt":tru  
e,"predictTime":null,"totalTime":null,"computedFields":{"embed":{"metadata"  
:{"computedFields":{"embed":{"metadata":{"uiMessages":{"  
errors":[],"infos":[],"warnings":[]}}}}}}}
```



2.5. List all stored projects

URL: *BaseUrl +/rxn/api/api/v1/projects/*
Method: *GET*

2.5.1. API response code

Returns the list of projects belonging to API key owner.

```
{
  "payload": {
    "content": [{
      ""id": "5c2c95fed6cb7600019e6f17",
      "createdOn": 1546425854026,
      "createdBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
      "modifiedOn": 1546425854026,
      "modifiedBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
      "name": "Test API",
      "description": null,
      "attempts": [],
      "visibility": null,
      "computedFields": {},
      "embed": {},
      "metadata": {}, {
        ""id": "5c2c95fed6cb7600019e6f18",
        "createdOn": 1546425854026,
        "createdBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
        "modifiedOn": 1546425854026,
        "modifiedBy": "99f7928e-5992-484e-a0dc-9638a788ab47",
        "name": "Test API 2",
        "description": null,
        "attempts": [],
        "visibility": null,
        "computedFields": {},
        "embed": {},
        "metadata": {}, {
          ...
        }
      }
    ]
  },
  "totalPages": 1,
  "last": true,
  "totalElements": 7,
  "numberOfElements": 7,
  "sort": null,
  "first": true,
  "size": 20,
}
```




```
ull,"computedFields":{},"embed":{},"metadata":{}},{ "id":"5c49e8ded6cb7600019e9732","createdOn":1548347614461,"createdBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","modifiedOn":1548347614461,"modifiedBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","name":"Webinar","description":null,"attempts":[],"visibility":null,"computedFields":{},"embed":{},"metadata":{}},{ "id":"5c532f56d6cb7600019ea342","createdOn":1548955478522,"createdBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","modifiedOn":1548955478522,"modifiedBy":"ba7a9ee4-6de6-44bc-b0b0-6eb17dc63ca8","name":"Test API","description":null,"attempts":[],"visibility":null,"computedFields":{},"embed":{},"metadata":{}}, {"totalPages":1,"last":true,"totalElements":7,"numberOfElements":7,"sort":null,"first":true,"size":20,"number":0},"metadata":{"uiMessages":{"errors":[],"infos":[],"warnings":[]}}}
```

Which is equivalent to the view one can get using the browser interface:

The screenshot displays the 'My Projects' interface in IBM RXN. At the top, there are navigation tabs: 'IBM RXN', 'Projects', 'Collaborators', 'Molecules list', 'Challenge', and 'Learn'. The user's name 'Teodoro Laino' is visible in the top right. The main heading is 'My Projects' with a 'Total: 7' badge. A search bar 'Search by name' and a 'Sort by' dropdown are present. A 'Start New Project' button is on the left. The projects are listed in a grid:

- test**: Total Reactions Collected: 2. Owner: Teodoro Laino. Created 27-08-2018, Updated 27-08-2018.
- test on Monday**: Total Reactions Collected: 2. Owner: Teodoro Laino. Created 27-08-2018, Updated 27-08-2018.
- test 2**: Total Reactions Collected: 0. Owner: Teodoro Laino. Created 17-10-2018, Updated 17-10-2018.
- press day**: Total Reactions Collected: 1. Owner: Teodoro Laino. Created 08-11-2018, Updated 08-11-2018.
- webinar18**: Total Reactions Collected: 0. Owner: Teodoro Laino. Created 15-11-2018, Updated 15-11-2018.
- Webinar**: Total Reactions Collected: 3. Owner: Teodoro Laino. Created 24-01-2019, Updated 24-01-2019.
- Test API**: Total Reactions Collected: 0. Owner: Teodoro Laino. Created 31-01-2019, Updated 31-01-2019.



2.6. List all attempts in a project

URL: *BaseUrl*

*+ /rxn/api/api/v1/projects/<PROJECTID>/attempts?raw={}&page=0&size=8
&sort=createdOn,DESC*

Method: *GET*

Mandatory fields:

- *PROJECTID (es: 5c2c95fed6cb7600019e6f17)*
-

Optional fields:

- *raw (must be always set equal to: {})*
 - *page (ie.. 0, 1, 2)*
 - *size (ie. 8 elements per page)*
 - *sort: (createdOn, DESC (ASC|DESC))*
-

In case these fields (defining the paging of the results) are not defined, the entire list of attempts will be retrieved (it may take a long time!).

2.6.1. API response code

Returns the list of attempts belonging to the given project ID, per page and sorted (according requested criteria).

```
{
  "payload": {
    "content": [
      {
        "id": "5b894010b72b87000121f186",
        "createdOn": 1535721488737,
        "createdBy": "system",
        "modifiedOn": 1535721488817,
        "modifiedBy": "system",
        "name": "testfb_20180831_13:18:08.737",
        "projectId": "5b759c5e300be10001fc9dc6",
```




```
"smiles":
"O.OC1CC2(CCC3(CCCC3)C2)CC1O>>O=C1CC2(CCC3(CCCC3)C2)CC1O",
  "attentionWeights": "<?xml version='1.0' encoding='utf-8'
standalone='no'?\n<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\"
... .. y='42.960226'/?>\n </clipPath>\n </defs>\n</svg>\n",
  "confidence": 0.5353702306747437,
  "message": "ok",
  "reactionImage": "<?xml version='1.0' encoding='iso-8859-
1'?>\n<svg version='1.1' baseProfile='full'\n
xmlns='http://www.w3.org/2000/svg'\n
xmlns:rdkit='http://www.rdkit.org/xml' \ ... .. <path d='M 695.891,96.8815
700.569,100' style='fill:none;fill-rule:evenodd;stroke:#000000;stroke-
width:2px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1'
/>\n</svg>\n",

"predictionId": "5b89400cb72b87000121f185",
  "feedbacks": null,
  "firstAttempt": true,
  "predictTime": null,
  "totalTime": null,
  "computedFields": {},
  "embed": {},
  "metadata": {}
},
{
  "id": "5b7708d4ec909700010e48a9",
  "createdOn": 1534527700063,
  "createdBy": "system",
  "modifiedOn": 1534527700101,
  "modifiedBy": "system",
  "name": "testfb_20180817_17:41:40.062",
  "projectId": "5b759c5e300be10001fc9dc6",
  "smiles":
"COC(=O)C1=CC2C=CC1CC2.COC(=O)CC1CCCC(O)C1>>COC(=O)C1=CC2CCC1CC2",
  "attentionWeights": "<?xml version='1.0' encoding='utf-8'
standalone='no'?\n<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\"
... .. y='42.960226'/?>\n </clipPath>\n </defs>\n</svg>\n",
  "confidence": 0.08226326107978821,
  "message": "ok",
  "reactionImage": "<?xml version='1.0' encoding='iso-8859-1'?>\n<svg
version='1.1' baseProfile='full'\n      xmlns='http://www.w3.org/2000/svg'\n
xmlns:rdkit='http://www.rdkit.org/xml' \ ... .. <path d='M 695.891,96.8815
700.569,100' style='fill:none;fill-rule:evenodd;stroke:#000000;stroke-
width:2px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1'
/>\n</svg>\n",
  "predictionId": "5b7708cdec909700010e48a8",
  "feedbacks": null,
  "firstAttempt": true,
  "predictTime": null,
```



```
"totalTime": null,
"computedFields": {},
"embed": {},
"metadata": {}
}},
"totalPages": 1,
"last": true,
"totalElements": 2,
"numberOfElements": 2,
"sort": [{
  "direction": "DESC",
  "property": "createdOn",
  "ignoreCase": false,
  "nullHandling": "NATIVE",
  "descending": true,
  "ascending": false
}],
"first": true,
"size": 8,
"number": 0
},
"metadata": {...}
}
```

2.6.2. Example with curl

```
curl --header "Content-Type: application/json" --header " Authorization:apk-
have-here-your-own-key" -X GET -X GET
https://rxn.res.ibm.com/rxn/api/api/v1/projects/5c532f56d6cb7600019ea342/attempts?raw=&page=0&size=8&sort=createdOn,DESC
```

Output:

```
{"payload":{"content":[{"id":"5c5aaf26d6cb7600019ea80a","createdOn":154944
6950408,"createdBy":"system","modifiedOn":1549446950583,"modifiedBy":"syst
em","name":"Test
API_20190206_09:55:50.407","projectId":"5c532f56d6cb7600019ea342","smiles
":"Oc1cc(-c2c(O)cccc2-c2cccc2)cc(O)c1O.[H].[H].[H]>>Oc1cc(-c2c(O)cccc2-
c2cccc2)cc(O)c1O","attentionWeights":"<?xml version='1.0' encoding='utf-
8' [... ...
...],"confidence":0.13361498713493347,"message":"ok","reactionImage":"<?xml
version='1.0' encoding='iso-8859-1'?>\n<svg version='1.1' [... ...
```



```
...],"predictionId":"5c5aaf22d6cb7600019ea809","feedbacks":null,"firstAttempt":true,"predictTime":null,"totalTime":null,"computedFields":{},"embed":{},"metadata":{}},{\"id\":\"5c532fadd6cb7600019ea346\",\"createdOn\":1548955565399,\"createdBy\":\"system\",\"modifiedOn\":1548955565542,\"modifiedBy\":\"system\",\"name\":\"Test API_20190131_17:26:05.398\",\"projectId\":\"5c532f56d6cb7600019ea342\",\"smiles\":\"Oc1cc(-c2c(O)cccc2-c2cccc2)cc(O)c1O.[H].[H].[H]>>Oc1cc(-c2c(O)cccc2-c2cccc2)cc(O)c1O\",\"attentionWeights\":\"<?xml version='1.0' encoding='utf-8' [... ...]\",\"confidence\":0.13361498713493347,\"message\":\"ok\",\"reactionImage\":\"<?xml version='1.0' encoding='iso-8859-1'?>\n<svg version='1.1' [... ...]\"}],\"predictionId\":\"5c532fa7d6cb7600019ea345\",\"feedbacks\":null,\"firstAttempt\":true,\"predictTime\":null,\"totalTime\":null,\"computedFields\":{},\"embed\":{},\"metadata\":{}},{\"totalPages\":1,\"last\":true,\"totalElements\":2,\"numberOfElements\":2,\"sort\":{\"direction\":\"DESC\",\"property\":\"createdOn\",\"ignoreCase\":false,\"nullHandling\":\"NATIVE\",\"descending\":true,\"ascending\":false}},\"first\":true,\"size\":8,\"number\":0,\"metadata\":{\"uiMessages\":{\"errors\":[],\"infos\":[],\"warnings\":[]}}
```

Which is equivalent to the view you can have using the browser and navigating in the “API Test” project folder:

The screenshot shows the IBM RXN web interface for the "API Test" project. The main heading is "Attempts" with a sub-heading "Total: 2". There is a search bar for "Search saved reactions by name" and a "Date desc" filter. Two reaction cards are displayed:

- Test API_20190206_09:55:50.407**: Confidence: 0.13, CreatedBy: system, Created: 06-02-2019. The reaction shows a complex polycyclic aromatic hydrocarbon derivative reacting with H₂ to form a similar derivative.
- Test API_20190131_17:26:05.398**: Confidence: 0.13, CreatedBy: system, Created: 31-01-2019. The reaction shows a similar complex polycyclic aromatic hydrocarbon derivative reacting with H₂ to form another similar derivative.

At the bottom, there is a pagination control showing "Items per page: 8" and "1 of 1 pages".