The submission flow through the api is designed to be very similar to the UX flow. It consists of 2 main phases - insample and outsample flow.

1a) Run simulation
1b) Check result
2a) Submit simulation (if eligible)
2b) Check portfolio

domain - https://alphaverse.alpha-grep.com

Setup

Screenshot added at end of the file

1) Login to Alphaverse through a browser
2) Open the network tab and make any request (e.g., run a simulation)
3) Copy the ‘Authorization’ value from the above request’s headers. It should start with ‘Bearer’
4) Add this key value pair to the header of all subsequent api calls.

Note - this token is valid for 24 hours. You will need a new token after that. Just follow the same steps.

Steps

1a) Run simulation

endpoint - **api/v1/insample** (POST)

This endpoint is designed to initiate a simulation.

Sample Payload

```
{
        "code": "alpha = open - close",
        "settings": {
                "truncation": 0.1,
                "pasteurize": true,
                "dataset": "CHINA500",
                "region": "china",
                "decay": 6,
                "neutral": "market"
        },
        “competition”: “AG_EXAMPLE”

}
```

All fields are required

Field descriptions:

code (string) - Must be a valid alpha.
settings.truncation (int) - Truncation value
settings.pasteurize (bool) - Should pasteurize or not
settings.decay (int) - Decay value
settings.region (string) - Should be either ['us' or 'china']
settings.dataset (string) - Value depends on region. 'us' datasets ['US1000', 'US2000', 'US3000'], 'china' datasets - ['CHINA500', 'CHINA1000', 'CHINA2000']
settings.neutral (int) - Value depends on region. 'us' neutrals are ['none', 'market', 'sub-sector', 'industry', 'sub-industry', 'sector'], 'china neutrals are ['none', 'market', 'sub-sector', 'industry', 'sector']
competition: Competition name to simulate this in

Response code should be **202**

Sample Response

```
{
        taskId: "738a72cf-223e-4dbc-9db6-aa75 a11d816b"
}
```

taskId (string) - **Save this id**. It is a unique uuid to identify your simulation. You will need it to check results and submit the alpha.

1b) Check results

endpoint - **api/v1/result?task_id={taskId}** (GET)

This endpoint is designed retrieve simulation results

Provide the **taskId** in the previous step to see the results of a simulation. Results will only be available after simulation is complete, which takes around a minute.

Response code should be **200**

Look out for **is_eligible_for_submission** property in the result. Only if it's **true**, should you submit your alpha. Note - Do not submit ineligible alphas. They won't be considered for scoring, but will count towards your API quota usage.

Is_eligible_for_submission is **True** if **robustness checks** and **correlation check** pass.

2a) Submit simulation

endpoint **- /api/v1/outsample** (POST)

This endpoint is designed to initiate a submission.

Sample Payload

```
{
        "taskId": “db01f7cb-c9c1-4942-bf66-70d85b417f02”
}
```

The payload should only have **taskId**.

Response code should be **202**

Sample Response

```
{
        taskId: "db01f7cb-c9c1-4942-bf66-70d85b417f02",
        osTaskId: “90d5dd66-03bd-4e4e-a2e4-c55b0cb3c6d8”
}
```

2b) Check OS results

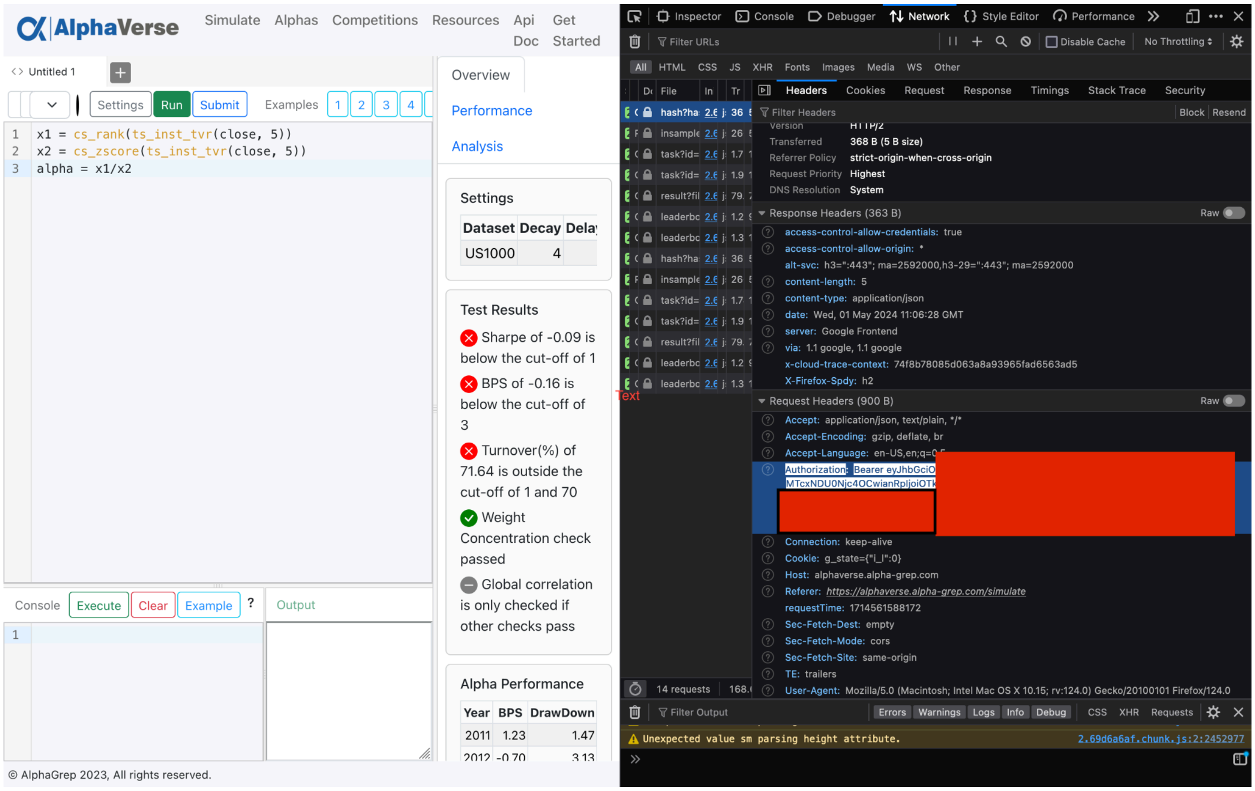
Individual OS results won’t be available anymore. You will be able to view your overall portfolio, which updates daily. We recommend viewing this in the UI.

Endpoint - **tasks/portfolio?competition={competition}&region={region}** (GET)

Example - tasks/portfolio?competition=AG_EXAMPLE&region=us

Appendix

Where to find the authorization token



```
x1 = cs_rank(ts_inst_tvr(close, 5))
x2 = cs_zscore(ts_inst_tvr(close, 5))
alpha = x1/x2
```

Settings

Dataset	Decay	Delay
US1000	4	

Test Results

- ☒ Sharpe of -0.09 is below the cut-off of 1
- ☒ BPS of -0.16 is below the cut-off of 3
- ☒ Turnover(%) of 71.64 is outside the cut-off of 1 and 70
- ✓ Weight Concentration check passed
- ⊖ Global correlation is only checked if other checks pass

Alpha Performance

Year	BPS	DrawDown
2011	1.23	1.47
2012	-0.70	3.13

© AlphaGrep 2023, All rights reserved.

```
hash?ha  2.6 j 36
insample 2.6 j 26
task?id= 2.6 j 1.7
task?id= 2.6 j 1.9
result?fil 2.6 j 79.
leaderbo 2.6 j 1.2
leaderbo 2.6 j 1.3
hash?ha  2.6 j 36
insample 2.6 j 26
task?id= 2.6 j 1.7
task?id= 2.6 j 1.9
result?fil 2.6 j 79.
leaderbo 2.6 j 1.2
leaderbo 2.6 j 1.3
```

```
Version            HTTP/2
Transferred        368 B (5 B size)
Referrer Policy    strict-origin-when-cross-origin
Request Priority   Highest
DNS Resolution     System

Response Headers (363 B)
access-control-allow-credentials: true
access-control-allow-origin: *
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
content-length: 5
content-type: application/json
date: Wed, 01 May 2024 11:06:28 GMT
server: Google Frontend
via: 1.1 google, 1.1 google
x-cloud-trace-context: 74f8b78085d063a8a93965fad6563ad5
X-Firefox-Spdy: h2

Request Headers (900 B)
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0
Authorization: Bearer [redacted]
Connection: keep-alive
Cookie: g_state={"i_l":0}
Host: alphaverse.alpha-grep.com
Referer: https://alphaverse.alpha-grep.com/simulate
requestTime: 1714561588172
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
TE: trailers
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:124.0) Gecko/20100101 Firefox/124.0
```

```
⚠ Unexpected value sm parsing height attribute.    2.69d6a6af.chunk.js:2:2452977
```

Sample python code for Insample simulation

import requests

# Define the API endpoint
url = 'https://alphaverse.alpha-grep.com/api/v1/insample'
token = # add me. Should start with “Bearer ….“

payload = {
    "code": "alpha = open - close",
    "settings": {
        "truncation": 0.1,

```
        "pasteurize": True,
        "dataset": "US1000",
        "region": "us",
        "decay": 6,
        "neutral": "market",
    },
    "competition": "APL_2024",
}

headers = {'Authorization': token}
# Send a GET request to the API endpoint
response = requests.post(url, headers=headers, json=payload, verify=False)


# Check if the request was successful (status code 200)
if response.status_code == 202:
    # Extract and print the response data
    data = response.json()
    print(data)
else:
    # Print an error message if the request failed
    print('Error:', response.status_code)
```

Sample python code for Outsample simulation

```
import requests

# Define the API endpoint
url = 'https://alphaverse.alpha-grep.com/api/v1/outsample'
token = #Add me "Bearer …"

payload = {"taskId": "123"}

headers = {'Authorization': token}
# Send a GET request to the API endpoint
response = requests.post(url, headers=headers, json=payload, verify=False)
data = response.json()

# Check if the request was successful (status code 200)
if response.status_code == 202:
    # Extract and print the response data
    data = response.json()
    print(data)
else:
```

```
# Print an error message if the request failed
print(data)
print('Error:', response.status_code)
```