Simulacrum Labs — API Reference

Version 2.9.21 | April 2026

Submit and manage jobs programmatically using the Simulacrum REST API. All endpoints are at **https://app.simulacrumlabs.com**.

Authentication

All API requests require authentication via one of:

- **API Key** (recommended for programmatic access) — include in every request:

```
Authorization: Bearer sim_live_your_key_here
```

- **Clerk JWT** (used by the web dashboard automatically)

API keys are created in the web dashboard under Settings > API Keys. The key is shown once — copy it immediately.

Quick Start: Submit a Job in 4 Steps

Step 1: Upload your project

```
curl -X POST https://app.simulacrumlabs.com/api/projects/request-upload \
  -H "Authorization: Bearer sim_live_xxx" \
  -H "Content-Type: application/json" \
  -d '{"name": "MyProject", "filename": "project.zip",
       "size_bytes": 52428800}'
```

Response includes **project_id** and **sas_url**. Upload your zip to the SAS URL:

```
curl -X PUT "RETURNED_SAS_URL" \
  -H "x-ms-blob-type: BlockBlob" \
  -H "Content-Type: application/zip" \
  --data-binary @project.zip
```

Step 2: Submit a job

```
curl -X POST https://app.simulacrumlabs.com/api/jobs/render \
  -H "Authorization: Bearer sim_live_xxx" \
  -H "Content-Type: application/json" \
  -d '{"project_id": "proj_xxx",
       "render_mode": "run_as_is",
       "python_script": "train.py",
       "num_frames": 0}'
```

Step 3: Check status

```
curl https://app.simulacrumlabs.com/api/jobs/JOB_ID \
  -H "Authorization: Bearer sim_live_xxx"
```

Step 4: Download results

```
curl https://app.simulacrumlabs.com/api/jobs/JOB_ID/download-manifest \
  -H "Authorization: Bearer sim_live_xxx"
```

Response includes download URLs for each output file.

Full Endpoint Reference

Project Upload

`POST /api/projects/request-upload`

Get a pre-signed Azure SAS URL to upload your project file.

Auth: API key or Clerk session. Account must be approved with credits > 0.

```
Request:
{
  "name": "MyProject",
  "filename": "project.zip",   // .zip or .7z
  "size_bytes": 52428800
}

Response (200):
{
  "project_id": "proj_xxx",
  "sas_url": "https://...blob.core.windows.net/...",
  "expires_in_minutes": 60
}
```

Job Submission

`POST /api/jobs/render`

Submit an annotation, training, or Docker Only job.

Auth: API key or Clerk. Must have credits > 0.

```
Request:
{
  "project_id": "proj_xxx",
  "render_mode": "annotations|run_as_is|docker_only",
  "num_frames": 60,            // 0 for training/docker
  "resolution": "1920x1080",
  "python_script": "train.py", // for run_as_is/docker
  "python_args": "--lr 0.001", // optional
  "map_path": "/Game/Maps/X",  // optional
  "sim_mode": "ComputerVision|Multirotor|Car",
  "require_gvisor": false      // optional
}

Response (201):
{
  "job_id": "render-20260405-xxx",
  "status": "queued",
  "assigned_node": "node-xxx",
  "project_id": "proj_xxx",
  "created_at": "2026-04-05T12:00:00Z"
}
```

`POST /api/jobs/sweep`

Submit a hyperparameter sweep. Each parameter combination runs on a separate GPU.

Auth: API key or Clerk. Must have credits > 0.

```
Request:
{
  "project_id": "proj_xxx",
  "parameter_grid": {
    "lr": [0.001, 0.01],
    "batch_size": [32, 64]
  },
  "python_script": "train.py",
  "map_path": "/Game/Maps/MyMap",
  "require_gvisor": false
}

Response (201):
{
  "sweep_id": "sweep-xxx",
  "total_variants": 4,
  "dispatched": 2,
  "queued": 2,
  "variant_job_ids": ["sweep-v0", ...]
}
```

Job Management

`GET /api/jobs`

List all your jobs. Add **?full=1** for detailed output.

Auth: API key or Clerk. Returns only your tenant's jobs.

```
Response (200):
{
  "jobs": [{
    "job_id": "render-xxx",
    "status": "completed|running|queued|failed",
    "frames_completed": 60,
    "total_frames": 60,
    "created_at": "ISO8601",
    "completed_at": "ISO8601"
  }]
}
```

`GET /api/jobs/:id`

Get detailed status of a specific job.

Auth: API key or Clerk.

`POST /api/jobs/:id/cancel`

Cancel a running or queued job. You are not charged for cancelled compute.

Auth: API key or Clerk.

```
Response: { "job_id": "xxx", "status": "cancelled" }
```

`POST /api/jobs/:id/complete`

Gracefully stop a training job and collect results so far.

Auth: API key or Clerk.

```
Response: { "job_id": "xxx", "status": "completing" }
```

Downloading Results

`GET /api/jobs/:id/download-manifest`

Get download URLs for all output files from a completed job.

Auth: API key or Clerk.

```
Response (200):
{
  "job_id": "render-xxx",
  "frames": [{
    "filename": "frame_0001.png",
    "size_bytes": 1048576,
    "download_url": "https://..."
  }]
}
```

Billing

`GET /api/billing/balance`

Check your current credit balance and remaining GPU hours.

Auth: API key or Clerk.

```
Response (200):
{
  "credits": 4.25,
  "gpu_hours_remaining": 5.67
}
```

API Key Management

`POST /api/keys`

Create a new API key. **Requires Clerk session** (web dashboard only).

```
Request: { "name": "My CI Key" }

Response (201):
{
  "id": "key_xxx",
  "key": "sim_live_abcdef...",
  "message": "Store securely - shown once only"
}
```

`GET /api/keys`

List all your API keys (prefix only, not full key).

Auth: Clerk session only.

`DELETE /api/keys/:id`

Revoke an API key. Takes effect immediately.

Auth: Clerk session only.

Python Example: Full Workflow

```python
import requests
import time

API = "https://app.simulacrumlabs.com"
KEY = "sim_live_your_key_here"
HEADERS = {"Authorization": f"Bearer {KEY}",
           "Content-Type": "application/json"}

# 1. Upload project
r = requests.post(f"{API}/api/projects/request-upload",
    headers=HEADERS,
    json={"name": "DroneTraining",
          "filename": "drone.zip",
          "size_bytes": 50_000_000})
upload = r.json()
project_id = upload["project_id"]

# Upload zip to Azure SAS URL
with open("drone.zip", "rb") as f:
    requests.put(upload["sas_url"],
        headers={"x-ms-blob-type": "BlockBlob",
                 "Content-Type": "application/zip"},
        data=f)

# 2. Submit training job
r = requests.post(f"{API}/api/jobs/render",
    headers=HEADERS,
    json={"project_id": project_id,
          "render_mode": "run_as_is",
          "python_script": "train.py",
          "num_frames": 0})
job_id = r.json()["job_id"]
print(f"Job submitted: {job_id}")

# 3. Poll for completion
while True:
    r = requests.get(f"{API}/api/jobs/{job_id}",
        headers=HEADERS)
    status = r.json()["status"]
    print(f"Status: {status}")
    if status in ("completed", "failed"):
        break
    time.sleep(30)

# 4. Download results
r = requests.get(
    f"{API}/api/jobs/{job_id}/download-manifest",
    headers=HEADERS)
for f in r.json().get("frames", []):
    data = requests.get(f["download_url"])
    with open(f["filename"], "wb") as out:
        out.write(data.content)
    print(f"Downloaded: {f['filename']}")
```

Error Codes

Code	Meaning
401	Missing or invalid API key / session
402	Insufficient credits
403	Account not approved or suspended
404	Job or project not found
429	Rate limited — too many requests
500	Server error — contact support

Rate Limits

Job submission: **5 jobs per hour** per account (free tier). Enterprise accounts have higher limits.

API requests: **60 requests per minute** per API key.

If rate limited, you receive a 429 response. Wait and retry.

Need Help? Email: help@simulacrumlabs.com