MetaSim API

Version 1.0 Draft 2
Status: In Progress

Overview

Table of Contents

Introduction

This document describes an API for use between MetaSim clients, servers, and engines. A MetaSim client initiates requests using the MetaSim API. A MetaSim server servers requests to the MetaSim API. MetaSim engines provide some parts of the MetaSim API. It is the responsibility of the server to initiate requests to engines on behalf of the client and aggregate engine responses into a coherent whole.

More information and discussions about the MetaSim project can be found here:
http://www.reddit.com/r/MetaSim/
https://github.com/rsimulate
http://rsimulate.com/

Goals

The goal of the API is to support the use cases defined in the Use Cases document https://docs.google.com/document/d/1OS5pBWxhLKuHQg_yUlgeodkWYvdEFZnW39rDxQxLDH0/edit?usp=sharing

Right now the API are to support these workflows:

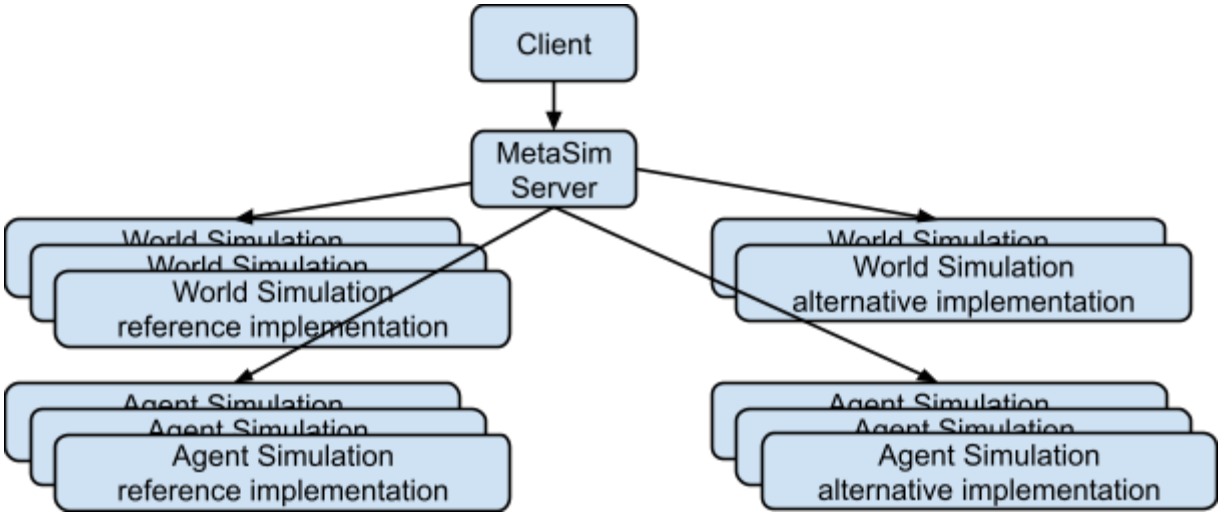
- Negotiate an api version
- Get a list of running simulations
- Add a new simulation
- Retrieve planet and agent data from a running simulation

The API needs to be able to support these workflows:

- Authentication

Architecture

A MetaSim server aggregates engines. Multiple engine implementations can be made available through the server.



Encoding

All resources MUST be encoded using JSON and utf-8.

Efficiency

WebSockets

In order to avoid polling for updated simulation information, some resources allow subscribers to receive notifications when a resource has changed through the /rel/notifications link.

Messages sent from the client to the server will be ignored. The purpose is for the server to send messages to the client when parts of the resource have changed.

Messages are always representations for the Notification resource.

Consider subscribing to Body:/rel/notifications to receive body textures and self instead of polling for body position and textures.

Document Change Process

1. Request for comments. Open up this document for commenting.
2. Wait until all comments have been added. If no comments have been added go to step 6.
3. Close the document for comments.
4. Fix the document and resolve comments until no more comments remain.
5. Go to step 1.
6. Remove draft number from document. Publish.
7. Done.

Basic Types

Names	Type	Description
HTTPMethod	String	One of “GET”, “POST”, “PUT”, “DELETE”, “HEAD”
URI	String	URI according to http://tools.ietf.org/html/rfc3986

Resources

Agent

Resource for agent
Media type: application/json

Fields

Field name	Type	Description
xyz_position	XYZPosition	Agent position
geo_position	GeoPosition	Agent position
age	Float	Age in simulated years since the agent was created.

Links

Relation name	Target resource	Required?	Description
self	Agent	Yes	GETable endpoint for this agent.
/rel/skills	Agent.Skills	Yes	GETable endpoint for this agent’s skills

Agent.Skill

Resource for agent skill

Media type: application/json

Fields

Field name	Type	Description
name	String	The name of the skill. An agent may not have two skills with the same name.
amount	Float	The amount of skill the agent has in this skill. Normalized between [0.0, 1.0]

Links
none

Agent.Skills

Resource for agent skills
Media type: application/json

Fields

Type	Description
Agent.Skill[]	A collection of all skills that apply to this agent.

Links
none

Agents

Resource for collection of agents. The agents resource is a GeoJSON “FeatureCollection” object. All elements of the feature collection MUST have a geometry type of “Point”.
Media type: application/json.

Fields

Field name	Type	Description
features	Feature[]	The current state of all agents in the simulation

Links
none

Additional Information
Agent features MAY include the following objects as GeoJSON feature properties

Properties

Field name	Type	Description
skills	Agent.Skills	The current state of the agent’s skills.

Bodies

Resource for celesial bodies
Media type: application/json

Fields

Field name	Type	Description
bodies	Body[]	Astronomical bodies participating in the simulation

Links
none

Body

Resource for a celestial body
Media type: application/json

Fields

Field name	Type	Description
xyz_position	XYZPosition	Body position.
xyz_velocity	XYZVelocity	Body velocity.
radius	Float	Radius of body in meters.

Links

Relation name	Target resource	Required?	Description
self	Body	Yes	GETable endpoint for this body.
/rel/world_texutre	Image (image/*)	No	GETable endpoint for retrieving the latest world texture.
/rel/world_texutre_night	Image (image/*)	No	GETable endpoint for retrieving the latest nighttime world texture.
/rel/cloud_texutre	Image (image/*)	No	GETable endpoint for retrieving the latest cloud texture.
/rel/heightmap_texture	Image (image/*)	No	GETable endpoint for retrieving the latest heightmap texture.
/rel/agents	Agents	No	GETable endpoint for retrieving the latest agent state.
/rel/regions	Regions	No	GETable endpoint for retrieving the latest region state.
/rel/notifications	WebSocket	No	Upgradable endpoint for receiving updated information about this body.

Engine

Resource for engine information. Data used to create new simulations.
Media type: application/json

Fields

Field name	Type	Description
name	String	Engine name, eg: “TerrainReferenceEngine”.
type	String	One of “Terrain”, “Weather”, “Agent”, “Region”.
version	String	API version, eg: “1.0”

Links
none

Engines

Resource for engine information.
Media type: application/json

Fields

Field name	Type	Description
engines	Engine[]	List of available engines.

Links
none

EngineSimulationRequest

Resource to request the creation of a new simulation.
Media type: application/json

Fields

Field name	Type	Description
simulation_id	Id	An identifier to use when identifying the simulation across engines.
simulation_href	URI	URI to the MetaSim simulation resource.

Links

none

EntryPoint

Entry point for a specific version of the MetaSimAPI
Media type: application/json

Fields
none

Links

Relation name	Target resource	Required?	Description
/rel/simulations	Simulations	No	GETable endpoint for managing simulations. This is only exposed to clients from MetaSim not engines.
/rel/engines	Engines	No	List of available engines to use in a SimulationRequest when creating a new simulation.
/rel/hosts	Hosts	No	GETable endpoint for managing hosts. This is only exposed to clients from MetaSim, not engines.

GeoPosition

3D position in lat,lon,height
Media type: application/json

Fields

Field name	Type	Description
lat	Double	Latitude position in degrees [-180.0, 180.0)
lon	Double	Longitude position in degrees [-180.0, 180.0) .
height	Double	Height position in meters relative to the center of the body.

Links
none

Host

A host is a container for running MetaSim engines. Engines can be provisioned over multiple hosts.
Media type: application/json

Fields

Field name	Type	Description
remote_location	URL	The absolute URI of the MetaSim endpoint on the host. Ex: “http://my.host.example.com/metasim”

Links

Relation name	Target resource	Required?	Description
self	Host	Yes	GETable endpoint for retrieving this Host resource.
/rel/delete	Host	Yes	DELETEableable endpoint for removing a host.

Hosts

A collection of hosts and the resources to manage hosts.
Media type: application/json

Fields

Field name	Type	Description
hosts	Host[]	Hosts available to this MetaSim instance.

Links

Relation name	Target resource	Required?	Description
self	Hosts	Yes	GETable endpoint for retrieving this Hosts resource.
/rel/add	Hosts.NewHost	Yes	POSTable endpoint for adding a host.

Hosts.NewHost

A new host is a container for running MetaSim engines. Engines can be provisioned over multiple hosts.
Media type: application/json

Fields

Field name	Type	Description
remote_location	URL	The absolute URI of the MetaSim endpoint on the host. Ex: “http://my.host.example.com/metasim”

Links
none

Link

Represents a relation from one resource to another resource.
Media type: application/json

Fields

Field name	Type	Description
rel	String	Relation name
href	URI	Target URI
method	HTTPMethod	HTTP method to use while navigating link

Links
none

MetaSim

Entry point for the MetaSim API. Accessible at /metasim on MetaSim and all engines.
Media type: application/json

Fields

Field name	Type	Description
versions	Version[]	Available versions

Links
none

Notification

Resource for sending notifications from the server to subscribers.
Media type: N/A.

Fields

Field name	Type	Description

rel	String	The rel part of a link indicating what part of the resource changed. If the value of rel is “ping” the notification can be disregarded. Ping notifications ensure that there is activity on the connection so that it does not timeout on PaaS vendors like Heroku: “The normal Heroku HTTP routing timeout rules apply to the WebSocket labs feature. Either client or server can prevent the connection from idling by sending an occasional ping packet over the connection.”
resource	*	The resource that the link points to. If the resource has a media type of application/json, the representation will be used directly for the value of this field. If the resource does not have a media type of application/json, the value of this field will be a base64 encoded value of the representation.

Links
none

Additional Information

An example is probably best here.

Consider the representation for resource A:

```
{
  “value1”: 10,
  “value2”: 20,
  “links”: [{
    “rel”:”/rel/notifications”,
```

```
    "href":"/metasim/A/1/notifications",
    "method": "UPGRADE"}, {
    "rel", "self",
    "href":"/metasim/A/1",
    "method": "GET"}, {
    "rel": /rel/a_img",
    "href": "/metasim/A/img",
    "method": "GET"}]
}
```

When subscribing to /metasim/A/1/notifications, the client can expect messages in this form:
The resource itself (application/json) has changed and this is the new representation:

```
{
  "rel": "self",
   "resource": {
     "value1": 20,
     "value2": 30,
     "links": [{
       "rel":"/rel/notifications",
       "href":"/metasim/A/1/notifications",
       "method": "UPGRADE"}, {
       "rel", "self",
       "href":"/metasim/A/1",
       "method": "GET"}, {
       "rel": /rel/a_img",
       "href": "/metasim/A/img",
       "method": "GET"}]
   }
}
```

The value at /rel/a_img has changed, and this is the new representation:

```
{
  "rel": "/rel/a_img",
  "resource": "WMCzLINTh7PUgBePMTkN/MxajP9kEIWJVpk3TcOL+T72ZO7YM9kHfim0LZrR7="
}
```

Regions

Resource for collection of regions. The regions resource is a GeoJSON “FeatureCollection” object. All elements of the feature collection MUST have a geometry type of “Polygon”.
Media type: application/json.

Fields

Field name	Type	Description
features	Feature[]	The current state of all regions in the simulation

Links
none

Additional Information
Region features MAY include the following objects as GeoJSON feature properties

Properties

Field name	Type	Description
name	String	The name of the region

Simulation

Simulation resource
Media type: application/json

Fields

Field name	Type	Description
name	String	The name of the simulation
date_created	String	The date the simulation was created.

Links

Relation name	Target resource	Required?	Description
self	Simulation	Yes	GETable endpoint for retrieving this simulation.

/rel/delete	none	Yes	DELETEable endpoint for removing a simulation
/rel/bodies	Bodies	No	GETable endpoint for retrieving information about the astronomical bodies participating in the simulation
/rel/skybox	image/*	Yes	GETable endpoint for retrieving the skybox texture

Simulations

Simulations resource for getting, adding simulations
Media type: application/json

Fields

Field name	Type	Description
active	Simulation[]	Running simulations.

Links

Relation name	Target resource	Required?	Description
/rel/add	SimulationRequest	No	POSTable endpoint for adding a simulation. This is only exposed to clients from MetaSim.
/rel/add_engine	EngineSimulationRequest	No	POSTable endpoint for adding a simulation to an engine. The 201 Created response MUST include a location response header pointing to a partially filled Simulation resource. This is only exposed from engines to MetaSim

SimulationRequest

Resource to request the creation of a new simulation.
Media type: application/json

Fields

Field name	Type	Description
name	String	The name of the simulation.
bodies_engine_name	String	Name of the bodies simulation engine to use.
terrain_engine_name	String	Name of the terrain simulation engine to use.
agent_engine_name	String	Name of the agent simulation engine to use.

Links
none

Version

An available API version starting with 1.0.
Media type: application/json

Fields

Field name	Type	Description
id	String	Version name
links	Link[]	Related links

Links

Relation name	Target resource	Required?	Description
/rel/entrypoint	Entrypoint	Yes	GETable API entrypoint for a specific version.

XYZPosition

3D position in x, y, z relative to the planet being simulated with 0,0,0 being the center of the planet.
Media type: application/json

Fields

Field name	Type	Description
x	Float	X position.
y	Float	Y position.
z	Float	Z position.

Links
none

XYZVelocity

3D velocity in x, y, z in m/s.
Media type: application/json

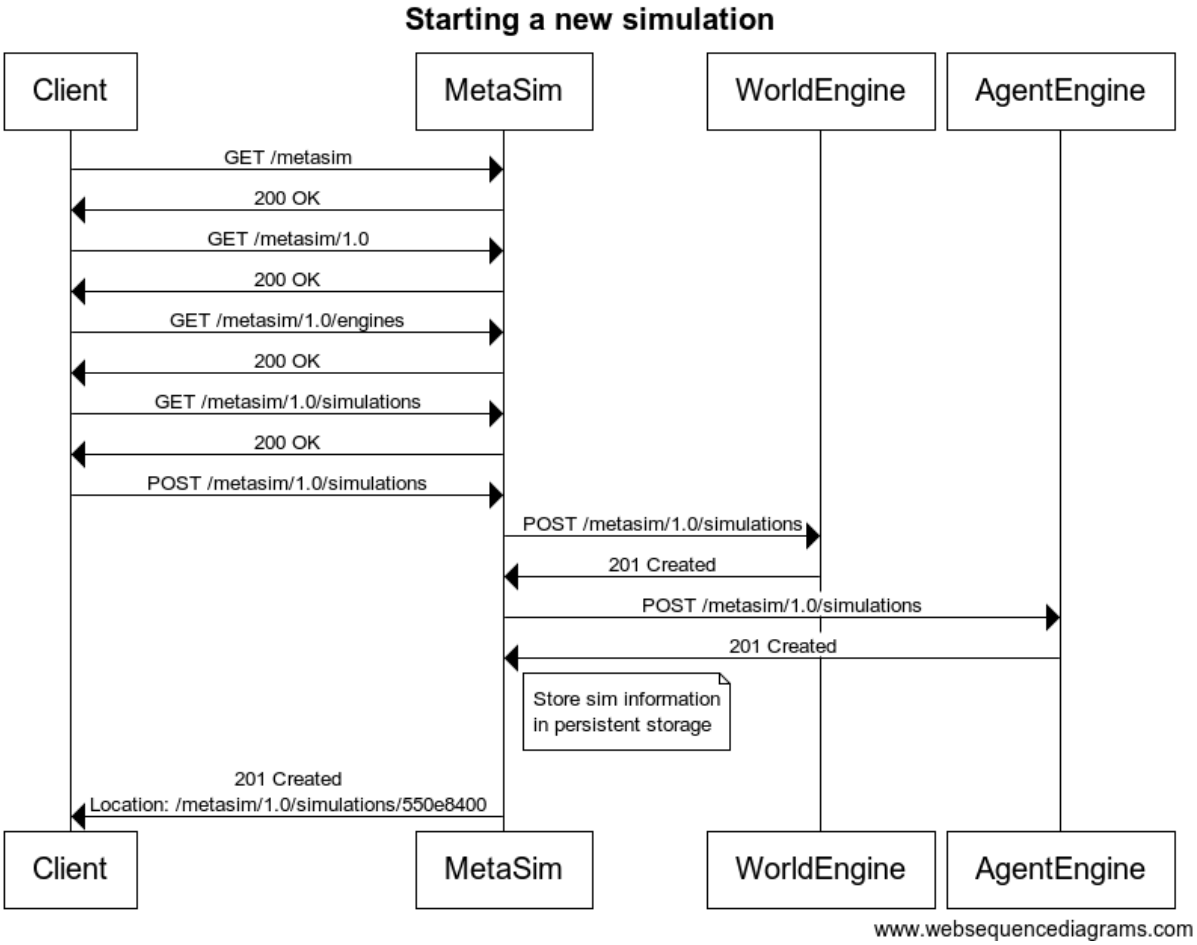
Fields

Field name	Type	Description
x	Float	X velocity.
y	Float	Y velocity.
z	Float	Z velocity.

Links
none

Examples

Starting a new simulation



http://www.websequencediagrams.com/cgi-bin/cdraw?lz=dGl0bGUgU3RhcnRpbmcgYSBuZXcgc2ltdWxhdGlvbgoKQ2xpZW50LT5NZXRhU2ltOiBHRVQgL21ldGFzaW0KAA8HLT4AIQY6IDIwMCBPSwAYHy8xLjAAATowL2VuZ2luZXMABzwAgVkKABssUE9TACQkV29ybGRFAIEgBQAUIAAgCwCCWQsyMDEgQ3JlYXRlZACCWQpBZ2VudAAwJgAgCwA_FwoKbm90ZSByaWdodCBvZiAAg1YJU3RvcmUgc2ltIGluZm9ybQCEAgVcbmluIHBlcnNpc3RlbnQgc3RvcmFnZQoAg2wUAIExCVxuTG9jAIRBBToAgmMZLzU1MGU4NDAw&s=default

Get a list of available versions

```
GET /metasim HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …

{
  “versions”: [{
    “id”: “1.0”,
```

```
    "links": [{
        "rel": "/rel/entrypoint"
        "href": "/metasim/1.0"
        "method": "GET"}]}]}
```

Pick version 1.0 and get the entrypoint

```
GET /metasim/1.0 HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …

{
  "links": [{
    "rel": "/rel/simulations"
    "href": "/metasim/1.0/simulations"
    "method": "GET"}, {
    "rel": "/rel/engines",
    "href": "/metasim/1.0/engines",
    "method": "GET"}]}
```

Get a list of the available engines

```
GET /metasim/1.0/engines HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …

{
  "engines": [{
    "name": "TerrainReferenceEngine",
    "type": "Terrain"}, {
    "name": "AgentReferenceEngine",
    "type": "Agent"}]}
```

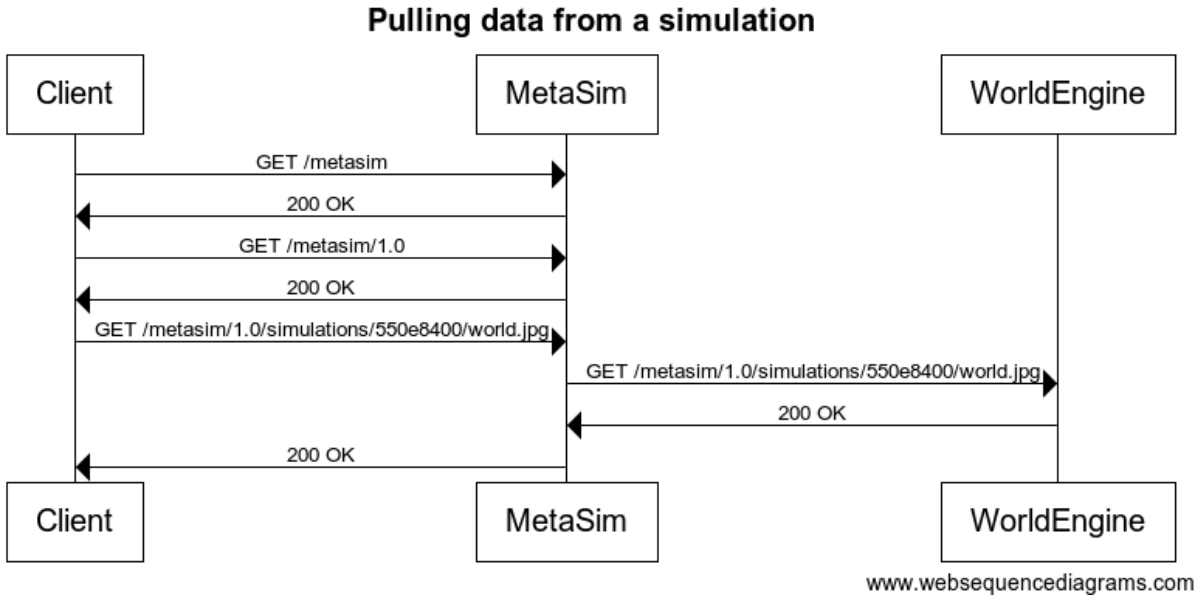
Add a new simulation

```
POST /metasim/1.0/simulations HTTP/1.1
host: localhost:9292
content-type: application/json
content-length: ...

{
  “terrain_engine_name” : “TerrainReferenceEngine”,
  “agent_engine_name” : “AgentReferenceEngine”}
```

```
HTTP/1.1 201 Created
location: /metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D
content-length: 0
```

Pulling data from a running simulation



http://www.websequencediagrams.com/?lz=dGl0bGUgUHVsbGluZyBkYXRhIGZyb20gYSBzaW11bGF0aW9uCgpDbGllbnQtPk1ldGFTaW06IEdFVCAvbWV0YXNpbQoADwctPgAhBjogMjAwIE9LABgfLzEuMAABOjAvAIEWCnMvNTUwZTg0MDAvd29ybGQuanBnAIELCldvcmxkRW5naW5lABQyADILAIFzCwCBWAgAgWEY&s=default

Get a list of available versions:

```
GET /metasim HTTP/1.1
```

```
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …

{
  “versions”: [{
    “id”: “1.0”,
    “links”: [{
        “rel”: “/rel/entrypoint”
        “href”: “/metasim/1.0”
        “method”: “GET”}]}]}
```

Pick version 1.0 and get the entrypoint

```
GET /metasim/1.0 HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …

{
  “links”: [{
    “rel”: “/rel/simulations”
    “href”: “/metasim/1.0/simulations”
    “method”: “GET”}, {
    “rel”: “/rel/engines”,
    “href”: “/metasim/1.0/engines”,
    “method”: “GET”}]}
```

Get a list of active simulations

Notice how there is an active simulation running

```
GET /metasim/1.0/simulations HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: application/json
content-length: …
```

```
{
  "simulations": [{
    "active": [{
      "links": [{
        "rel": "self"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D"
        "method": "GET"},
       {
        "rel": "/rel/delete"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D"
        "method": "DELETE"},
       {
        "rel": "/rel/world_texture"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D/world.jpg"
        "method": "GET"},
       {
        "rel": "/rel/cloud_texture"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D/cloud.png"
        "method": "GET"},
      {
        "rel": "/rel/heightmap_texture"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D/cloud.png"
        "method": "GET"},
      {
        "rel": "/rel/agents"
        "href":
"/metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D/agents"
        "method": "GET"}]}]

    "links": [{
        "rel": "/rel/add"
        "href": "/metasim/1.0/simulations"
        "method": "POST"}]}]}
```

Get the latest world texture from the simulation

```
GET /metasim/1.0/simulations/21EC2020-3AEA-1069-A2DD-08002B30309D/world.jpg
HTTP/1.1
host: localhost:9292
```

```
HTTP/1.1 200 OK
content-type: image/jpg
ETag: “686897696a7c876b7e”
content-length: …

<image content here>
```