

---

# **Simulator Documentation**

***Release 0.1***

**Hua Wei**

**Jul 10, 2020**



---

## Contents:

---

<b>1</b>	<b>Mobility Intervention of Epidemics Challenge</b>	<b>3</b>
1.1	Goal . . . . .	3
1.2	Support . . . . .	3
1.3	License . . . . .	3
<b>2</b>	<b>Simulator</b>	<b>5</b>
2.1	Simulator Modeling . . . . .	5
2.1.1	Human Mobility Model . . . . .	5
2.1.2	Disease Transmission Model . . . . .	6
2.1.3	Health Status of an Individual . . . . .	6
2.2	Mobility Intervention Actions . . . . .	8
2.3	Evaluation Metrics . . . . .	8
<b>3</b>	<b>Competition Scenarios and Leaderboard Ranking</b>	<b>9</b>
3.1	Competition Scenarios . . . . .	9
3.1.1	1. Default . . . . .	9
3.1.2	2. Higher infection rates . . . . .	9
3.1.3	3. More areas in the city . . . . .	9
3.1.4	4. Larger initial infected population . . . . .	9
3.1.5	5. Larger range of start-working time . . . . .	9
3.2	Leaderboard Ranking . . . . .	10
<b>4</b>	<b>Try it yourself</b>	<b>11</b>
4.1	Installation Guide . . . . .	11
4.1.1	Install in your environment . . . . .	11
4.1.2	Use a docker . . . . .	12
4.2	Run Simulation . . . . .	13
4.2.1	Initiate engine . . . . .	13
4.2.2	Simulate one step . . . . .	13
4.2.3	Sample codes . . . . .	13
4.3	Results . . . . .	14
4.3.1	Submission . . . . .	14
4.3.2	Logs . . . . .	15
4.4	Sample codes for an example policy . . . . .	17
<b>5</b>	<b>APIs</b>	<b>19</b>
5.1	Simulation Initialization API . . . . .	19

5.2	Simulation Config API . . . . .	19
5.3	Data Access API . . . . .	20
5.4	Intervention API . . . . .	22
5.5	Other API . . . . .	23

Contents:



---

## Mobility Intervention of Epidemics Challenge

---

[Home Page.](#)

The challenge would like to gather the global research community to apply recent advances in data science and AI to generate new insights in support of the fight against pandemics like COVID-19.

The challenge will be evaluated on a human mobility and epidemic transmission simulator. Participants will design different strategies to minimize the spread of the virus. The winners will be invited to present at the workshop.

### 1.1 Goal

In this challenge, we aim to look for effective human mobility intervention policies during an epidemic outbreak. Generally, our goal is to minimize the total number of infected people and, at the same time, minimize the intervention on human mobility.

### 1.2 Support

If you are having issues, please let us know. We have a mailing list located at: [epidemic-challenge@google-groups.com](mailto:epidemic-challenge@google-groups.com)

### 1.3 License

The project is licensed under the BSD license.





### 2.1 Simulator Modeling

The simulator simulates individual mobility in a city of  $R$  areas with  $M$  people. Each area belongs to one of the three categories: working, residential, and commercial. An individual is associated with two fixed POIs: one for residential, and one for working.

Default parameters in the simulator			
$\mathcal{A}$	11	M	10000

#### Note:

- In the competition, we have five simulated scenarios, whose parameters will be specified in the [next section](#).

#### 2.1.1 Human Mobility Model

Our simulator simulates the human mobility from 8 A.M. to 10 P.M. with one simulation step corresponding with one hour in the real world. An individual has different modes of mobility during weekdays and weekends.

On weekdays, an individual will start from residential area to working area at a certain time  $T_{start}^d \sim U(t_{s1}^d, t_{s2}^d)$ , and stay there for  $T_{work} \sim U(t_{w1}, t_{w2})$  hours. After work, they may visit a nearby commercial area (randomly sampled from  $K_{com}$  nearest areas of the working area) with a probability of  $P_{com}^d$  and stay there for  $T_{com}^d \sim U(t_{c1}^d, t_{c2}^d)$  hours. Then, they will return to residential area.

On weekends, people may visit a random commercial area at a certain time  $T_{start}^e \sim U(t_{s1}^e, t_{s2}^e)$  with a probability  $P_{com}^e$  and stay there for  $T_{com}^e \sim U(t_{c1}^e, t_{c2}^e)$  hours. After that, they will return to residential area.

$t_{s1}^d$	1	$t_{s2}^d$	3	$t_{w1}$	7	$t_{w2}$	10
$t_{c1}^d$	1	$t_{c2}^d$	2	$t_{s1}^e$	1	$t_{s2}^e$	5
$t_{c1}^e$	1	$t_{c2}^e$	2	$P_{com}^d$	0.1	$P_{com}^e$	0.3
$K$	1						

## 2.1.2 Disease Transmission Model

The disease can transmit from an infected individual through two kinds of contacts:

- Acquaintance contacts: An individual has a fixed small group of acquaintance contacts with size  $K_l \sim U(l_{c1}, l_{c2})$  in his/her residential area, and a fixed group of acquaintance contacts with size  $K_w \sim U(w_{c1}, w_{c2})$  in his/her working area. Note that not all the individuals in the same residential/working area are the acquaintance contacts of the individual. At each timestamp, there is a probability  $P_c$  for an individual to get infected from an infected acquaintance contact.
- Stranger contacts: An individual could be in contact with strangers visiting the same area at the same time. At each timestamp, there is probability  $P_s$  for an individual to get infected from an infected stranger contact.

$l_{c1}$	1	$l_{c2}$	6	$w_{c1}$	5	$w_{c2}$	15
$P_c$	2.5e-2			$P_s$	5e-3		

---

**Note:** The parameters are calibrated to align with the  $R_0$  of COVID-19 (2 ~ 2.5) provided by World Health Organization:

- World Health Organization. (2020, May 8). Report of the WHO-China Joint Mission on Coronavirus Disease 2019 (COVID-19). Retrieved May 8, 2020, from <https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf>
- 

## 2.1.3 Health Status of an Individual

An individual's health status follows the stages below:

Health status	Description	Infected	Contagious	Symptoms	Immune
1. Susceptible	liable to be infected				
2. Pre-symptomatic	before the onset of symptoms	✓	✓		
3. Symptomatic	showing symptoms compatible with infection	✓	✓	✓	
4. Critical	symptomatic with severe acute respiratory illness	✓	✓	✓	
5. Recovered	recovered and resistant				✓

- Stage 1. **Susceptible**: Liable to be infected
- **Stage 2. Pre-symptomatic infected: Infected and undiscovered**
  - From Stage 1 to Stage 2, people can get infected via contact with infected people, with different probabilities from their contacts.
- **Stage 3. Symptomatic infected: Infected and showing signs and symptoms**
  - From Stage 2 to Stage 3, there is an incubation period of  $INC \sim U(inc_1, inc_2)$  days.
- **Stage 4. Symptomatic infected with critical health condition**
  - From Stage 3 to Stage 4, there is a development time period  $d \sim \mathcal{N}(\mu, \phi)$ .
- **Stage 5. Recovered: recovered and resistant**
  - Each infected individual can recover through self-recovery or hospitalization. He/she will recover after being hospitalized consecutively for  $TREAT$  days, and become immune to the disease. He/she can also self-recover after  $RECOVER \sim U(rec_1, rec_2)$  days.

$inc_1$	1	$inc_2$	5	$\mu$	2	$\phi$	3	$TREAT$	15
$rec_1$	15	$rec_2$	30						

**Note:** Terms are in align with recent variations of the Susceptible-Infected-Resistant (SIR) compartment models in the context of Epidemic modeling and WHO guidelines:

1. Ferretti, L., Wymant, C., Kendall, M., Zhao, L., Nurtay, A., Abeler-Dörner, L., ... & Fraser, C. (2020). Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*.
2. World Health Organization. (2020, April 24). Situation reports. Retrieved April 24, 2020, from <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports/>

## 2.2 Mobility Intervention Actions

We can provide 5 levels of mobility intervention to each individual:

- Level 0 - No intervene: The individual can move normally.
- Level 1 - Confine: An individual is confined in the neighborhood that he/she lives in, in contact with his/her acquaintance contacts and stranger contacts in the residential area.
- Level 2 - Quarantine: The individual is quarantined at home, in contact with acquaintance contacts sharing the same residential area.
- Level 3 - Isolate: The individual is isolated, even from the acquaintance contacts living in the same residential area.
- Level 4 - Hospitalize: The individual is under treatment in the hospital.

	In Contact with?	
Intervention	acquaintance contacts	stranger contacts
#0: No Intervene	✓ (Residential and working)	✓ (Residential, working and commercial)
#1: Confine	✓ (Residential only)	✓ (Residential only)
#2: Quarantine	✓ (Residential only)	
#3: Isolate		
#4: Hospitalize		

---

### Note:

- When an individual is intended with multiple interventions , only the highest level of intervention will be applied.
  - Intervention actions are only effective when being set at the start of one day.
- 

## 2.3 Evaluation Metrics

We first define two basic metrics:

- $I$ : the accumulated number of infected people on day  $T$ .
- $Q$ : the weighted sum of  $N_v$ , where if an individual is under intervention  $v$  for one day, it will put towards adding 1 towards  $N_v$  ( $v \in \{hospitalized, isolated, quarantined, confined\}$ ):
  - $Q = \lambda_h * N_{hospitalized} + \lambda_i * N_{isolated} + \lambda_q * N_{quarantined} + \lambda_c * N_{confined}$

Based on these two basic metrics, we calculate the following score for this competition.

$$Score = exp\{\frac{I}{\theta_I}\} + Q_w * exp\{\frac{Q}{\theta_Q}\}.$$

Our goal is to minimize the score, evaluated on the 60th day of simulation.

$\theta_I$	500	$\theta_Q$	10000	$Q_w$	1.0	$T$	60
$\lambda_h$	1.0	$\lambda_i$	0.5	$\lambda_q$	0.3	$\lambda_c$	0.2

---

## Competition Scenarios and Leaderboard Ranking

---

### 3.1 Competition Scenarios

In the official competition, we provide five scenarios, each has the following specified parameters:

#### 3.1.1 1. Default

In this scenario, the default parameters are specified in [the previous section](#). Unless specified, the parameters in the following scenarios are the same as the default scenario.

#### 3.1.2 2. Higher infection rates

In this scenario, we aim to simulate a disease with higher infection rates. Specifically, infection rate from acquaintance contact  $P_c = 0.05$ , infection rate from stranger contact  $P_s = 0.01$ . Their detailed description can be found [here](#).

#### 3.1.3 3. More areas in the city

In this scenario, there will be 98 areas in the city,  $\mathcal{A} = 98$ .

#### 3.1.4 4. Larger initial infected population

In this scenario, there is a large infected population at the beginning of the simulation. Specifically, there are 300 infected individuals at the first day of simulation.

#### 3.1.5 5. Larger range of start-working time

In this scenario, each individual will have a larger range of time to go to work. Specifically, the start-working time  $T_{start}^d \sim U(t_{s1}^d, t_{s2}^d)$  with  $t_{s1}^d = 1$  and  $t_{s2}^d = 8$ . Their detailed description can be found [here](#).

## 3.2 Leaderboard Ranking

The final ranking is based on the sum of rankings over all 5 scenarios.

We provide a [Starter-Kit](#) as guidelines for setting up the simulator and submit results. Here, we go through them in detail, and introduce the basic background about the simulator, template strategies and intervening APIs supported the simulator.

## 4.1 Installation Guide

Our simulator can run in multiple platforms. Here, we provide two kinds of installation.

### 4.1.1 Install in your environment

1. Check that you have python 3 installed. Other version of python might work, however, we only tested on python with version  $\geq 3.6$ .

2. Preparation

2.1 Install boost library and make sure you have libboost path added to your system path. We suggest to use [boost 1.58.0](#).

- Mac OS: *brew install boost*
- Linux:
  - Installation with *sudo apt-get install libboost-dev*
  - Locate libboost\_system.so with *locate libboost\_system.so*
  - Add the last step paths to your system path with *export PATH={your paths}:\$PATH*
- Windows: Please refer to [install-boost-build](#)

2.2 Specify python environment

Make sure you have compatible python environment that marches with the simulator. Currently, we support the following versions of python:

- Mac OS: python 3.6, 3.7
- Linux: python 3.6

3. Clone Starter-Kit from GitHub and checkout at *master* branch.

```
git clone https://github.com/prescriptive-analytics/starter-kit.git
```

4. Go to Simualtor project's root directory and run the following to test the installation

```
import simulator
eng = simulator.Engine
```

---

**Note:** You might need to rename the `.so` file that corresponds with your system and python to `simulator.so`.

---

### 4.1.2 Use a docker

1. Install docker with the [docker instruction](#)
2. Pull the docker image. Please pull from docker hub, with the following command in your terminal.

```
docker pull episim2020/simulator:latest
```

4. Clone Starter-Kit from GitHub and checkout at *master* branch.

```
git clone https://github.com/prescriptive-analytics/starter-kit.git
```

3. Create a docker container and map starter-kit into the container by typing the following command in your terminal.

```
docker run -it -v path/to/starter-kit/in/your/local/computer:path/to/starter-kit/in/
↪docker/container episim2020/simulator
```

4. You should have entered the container. Please navigate to the starter-kit folder in docker container, then you can run the following command to start an experiment.

```
python example.py
```

---

#### Note:

1. For other uses of docker, please refer to [docker run](#).
2. Please pay attention to the security of your files, since docker container will be granted the access to change your files in the folders that you have mapped into the container. Please use carefully at your own risk.
3. The dockerfile to build this image is also attached [here](#). You can build your own image for personalized use. For this approach, please download the specified [anaconda](#). version. You need to put it in the same folder as the docker file. (Remember to change the file name in the dockerfile if you are using a different version.) Then, you can run the following command to build an image.

```
docker build -t simulator -f simulator.Dockerfile
```

4. Docker container will be destroyed after you exit. If you wish to install your own package, we recommend you to build your own image based on our image. Please refer to [this link](#)

---



## 4.2 Run Simulation

### 4.2.1 Initiate engine

```
import simulator
eng = simulator.Engine(thread_num=1, write_mode="append", specified_run_name="test",
↳ scenario="scenario1")
eng.reset() # reset() should be called right after the create of engine
```

- `thread_num`: number of threads.
- `specified_run_name`: results saving folder name.
- `write_mode`: mode of saving simulation results, `write` will overwrite results from different rounds of simulation in the same `specified_run_name` folder, `append` will append the results from current simulation round to existing result files.
- `scenario`: the scenario to choose to run the experiment. Possible choices are `scenario1`, `scenario2`, `scenario3`, `scenario4`, `scenario5`, and `submit`. All other arguments will be invalid.

### 4.2.2 Simulate one step

To simulate one step, simply call `eng.next_step()`.

```
eng.next_step()
```

### 4.2.3 Sample codes

Here we provide a sample code for running our simulator, which can be found in the starter kit - [example.py](#).

```
import simulator
import os
import json

period = 840

engine = simulator.Engine(thread_num=1, write_mode="write", specified_run_name="test",
↳ scenario="scenario1")

engine.reset()
for i in range(period):
    engine.next_step()
    engine.get_current_time()

    # Here, we give the example to get the information about individual with id 1
    individual_id = 1
    engine.get_individual_visited_history(individual_id)
    engine.get_individual_infection_state(individual_id)
    engine.get_individual_visited_history(individual_id)

    # Here, we give the example to get the information about region with id 1
    region_id = 1
    engine.get_area_infected_cnt(region_id)
```

(continues on next page)

(continued from previous page)

```
# Here, we give the example to set actions for individual with id 1, 2, 3, and 4, ↵
↵respectively
engine.set_individual_confine_days({1: 5}) # {individualID: day}
engine.set_individual_quarantine_days({2: 5}) # {individualID: day}
engine.set_individual_isolate_days({3: 5}) # {individualID: day}
engine.set_individual_to_treat({4: True}) # {individualID: day}

del engine
```

## 4.3 Results

During simulation, the simulator will generate the submission file sub\_xxx.txt and log files. xxx corresponds with your specified\_run\_name when initiating the engine simulator. Engine(specified\_run\_name="xxx").

### 4.3.1 Submission

In order to generate the submission, you need to select the scenario as “submit”. This will run the simulation for 5 scenarios, with each scenario for 3 rounds. Every round will have a length of 840 steps (60 simulation days). Every 840 steps, the simulator will automatically start a new round. Every 840\*3 steps, the simulator will automatically switch to a new scenario.

```
import simulator
import os
import json

period = 840

engine = simulator.Engine(thread_num=1, write_mode="write", specified_run_name="test",
↵ scenario="submit")

engine.reset()
for ind_round in range(15):
    # Here, we have 15 rounds of testing. Each round contains 840 steps.
    # Each of the 5 scenarios will be run for 3 times. But their order is unknown, ↵
    ↵here.
    for i in range(period):

        engine.next_step()
        engine.get_current_time()

        # Here, we give the example to get the information about individual with id 1
        individual_id = 1
        engine.get_individual_visited_history(individual_id)
        engine.get_individual_infection_state(individual_id)
        engine.get_individual_visited_history(individual_id)

        # Here, we give the example to get the information about region with id 1
        region_id = 1
        engine.get_area_infected_cnt(region_id)

        # Here, we give the example to set actions for individual with id 1, 2, 3, ↵
    ↵and 4 respectively
```

(continues on next page)

(continued from previous page)

```
engine.set_individual_confine_days({1: 5}) # {individualID: day}
engine.set_individual_quarantine_days({2: 5}) # {individualID: day}
engine.set_individual_isolate_days({3: 5}) # {individualID: day}
engine.set_individual_to_treat({4: True}) # {individualID: day}

del engine
```

Before submission, make sure:

- You are running the simulation for 840 time steps (60 simulation days in simulator).
- You are required to set the engine scenario to “submit” with `simulator.Engine(scenario="submit")`, and run 840 steps in each scenario for 3 times of your subsequent codes.
- You are supposed to use one model to run over five scenarios.
- Please upload the `sub_XXX.txt` to the website.

Here we provide a sample code of simulation that matches with submission requirements, which can be found [here](#).

### 4.3.2 Logs

We also provide simulation logs to competitors.

1. The city-wide daily log file `cnt_XXX.txt`.
2. The area level daily log file `hex_cnt_XXX.txt`.
3. The city-wide daily `r` file `r0_XXX.txt`.

Their Formats are as follows:

1. ‘cnt\_XXX.txt’:

#	Name	Data Tpye	Example Data	Description
0	day	int	0	Current day in simulation
1	hospitalizeNum	int	0	# of hospitalized people
2	isolateNum	int	0	# of isolated people
3	quarantineNum	int	0	# of quarantined people
4	confineNum-free_num	int	0	# of confined people
5	free	int	201	# of people without intervention
6	CurrentHealthy	int	199	# of people that are not infected
7	CurrentInfected	int	2	# of infected cases
7	CurrentEffective	int	2	# of infected cases without any intervention
9	CurrentSusceptible	int	199	# of susceptible people
10	CurrentIncubation	int	2	# of pre-symptomatic cases
11	CurrentDiscovered	int	0	# of symptomatic cases
12	CurrentCritical	int	0	# of critical cases
13	CurrentRecovered	int	0	# of recovered cases
14	AccDiscovered	int	0	Accumulated # of symptomatic cases
15	AccCritical	int	0	Accumulated # of critical cases
16	AccAcquaintance	int	0	Accumulated # of infected through stranger contacts
17	AccStranger	int	0	Accumulated # of infected through acquaintance contacts
18	measurement	int	2	an example measurement

2. *hex\_cnt\_XXX.txt*: Area-level replay data file.

#	header	Data Tpye	Example Data	Description
0	day	int	0	Current day in simulation
1	area_id	int	0	area id
2	lat	double	114.05019	latitude
3	lng	double	30.445043	langitude
4	CurrentSusceptible	int	26	# of susceptible cases
5	CurrentIncubation	int	0	# of pre-symptomatic cases
6	CurrentDiscovered	int	0	# of discovered cases
7	CurrentCritical	int	0	# of critical cases
8	CurrentRecovered	int	0	# of recovered cases
9	CurrentInfected	int	0	# of infected cases
10	free	int	26	# of people without intervention

3. “r0\_XXX.txt”: daily R-value (effective reproduction number).

#	header	Data Tpye	Example Data	Description
0	day	int	0	Current day in simulation
1	r	double	0.889	R value

**Note:** The calculation of R is based on:

- Fred Brauer. (2010, July). Epidemic Models I: Reproduction Numbers and Final Size Relations. Summer 2010 Thematic Program on the Mathematics of Drug Resistance in Infectious Diseases, Toronto, Canada.

## 4.4 Sample codes for an example policy

Here we provide a sample code that implements an example policy, which would hospitalize the discovered cases and furthermore, isolate individuals with high (estimated) probability of getting infected. The probability of an individual getting infected is estimated through his/her contacting history with discovered cases. The codes can also be found in the starter kit - `example_policy.py`.

```
import simulator
import numpy as np
import argparse

parser = argparse.ArgumentParser()
parser.add_argument("--scenario", type=str, help='scenario',default='scenario1')
parser.add_argument("--save_name", type=str, help='save_name',default='test')

parser.add_argument("--control_period", type=int, help='total_period',default=14)
parser.add_argument("--prob_s", type=float, help='prob_s',default=0.005)
parser.add_argument("--i_threshold", type=float, help='isolate_threshold',default=0.
↪001)
parser.add_argument("--i_days", type=int, help='isolate_threshold',default=5)

args = parser.parse_args()

# Parameters
pop = 10000
total_period = 840 if args.scenario != 'submit' else 12600
control_period = args.control_period
prob_s = args.prob_s
isolate_threshold = args.i_threshold
isolate_days= args.i_days

# Initialize engine
file_name = args.save_name + '_' + str(args.i_threshold) + '_' + str(args.i_days)
engine = simulator.Engine(thread_num=8, write_mode="write", specified_run_name=file_
↪name, scenario=args.scenario)
engine.reset()

# Start
for control_period_ in range(total_period // control_period):
    # Get current observations
    current_intervene = np.array([engine.get_individual_intervention_state(i) for i_
↪in range(pop)])
    current_infection = np.array([engine.get_individual_infection_state(i) for i in_
↪range(pop)])
    current_symptomatic = np.bitwise_or(current_infection == 3, current_infection ==_
↪4)
    current_symptomatic_set = set(np.where(current_symptomatic)[0])
    current_notsymptomatic_set = set(np.where(~current_symptomatic)[0])
    # Set treat
    set_treat = np.bitwise_and(current_symptomatic, current_intervene != 5)
    set_treat = np.where(set_treat)[0]
    set_individual_hospitalize = {i:True for i in set_treat}
    engine.set_individual_to_treat(set_individual_hospitalize)
    # Estimate the prob of being (not) infected
    current_discovered = set_treat
    num_areas = len(engine.get_all_area_category())
    prob_not_infected_by_discovered = np.ones(10000) # Initialize prob_not_infected_
↪by_discovered
```

(continues on next page)

(continued from previous page)

```

prob_not_infected_by_discovered[current_symptomatic] = 0
for loc_id in range(num_areas):
    his = engine.get_area_visited_history(loc_id) # find past individuals that_
↪visited the area
    for his_ in his: # for one hour
        his_set = set(his_)
        # symptomatic/non-symptomatic individuals visited loc_id during this hour
        his_discovered = his_set & current_symptomatic_set
        his_healthy = his_set & current_notsymptomatic_set
        # estimate the probability of being infected by symptomatic individuals
        sum_prob_infected = (1-prob_not_infected_by_discovered[list(his_
↪healthy)]).sum()
        his_prob_infect = prob_s * len(his_discovered) / (len(his_)+1e-7)
        prob_not_infected_by_discovered[list(his_healthy)] *= 1-his_prob_infect
    # Set isolate
    set_isolate = np.bitwise_and(prob_not_infected_by_discovered < 1 - isolate_
↪threshold, current_intervene == 1)
    set_isolate = np.where(set_isolate)[0]
    set_individual_isolate = {i:isolate_days for i in set_isolate}
    engine.set_individual_isolate_days(set_individual_isolate)
    # Simulate
    for i in range(control_period):
        engine.next_step()

```

## 5.1 Simulation Initialization API

```
Engine(thread_num=1, write_mode="append", specified_run_name="test",
scenario="scenario1"):
```

- **Args:**
  - `thread_num`: number of threads.
  - `specified_run_name`: results saving folder name.
  - `write_mode`: mode of saving simulation results, `write` will overwrite results from different rounds of simulation in the same `specified_run_name` folder, `append` will append the results from current simulation round to existing result files.
  - `scenario`: the scenario to choose to run the experiment. Possible choices are `scenario1`, `scenario2`, `scenario3`, `scenario4`, `scenario5`, and `submit`. All other arguments will be invalid.
- **Return:** an initialized engine without `reset()`, should call `reset()` after this function.

## 5.2 Simulation Config API

```
reset(seed=False):
```

- Reset the simulation, this should be called after we creat an engine with `Engine()`
- Reset random seed if `seed` is set to `True`

```
set_random_seed(seed):
```

- Set seed of random generator to `seed`
- Input format: `int`

```
next_step():
```

- Simulate one step, a simulation step indicates one hour in the real world.

---

**Note:**

1. `reset()` should be called every time we create an engine with `Engine()`
  2. All the IDs for men and areas start from 0.
- 

## 5.3 Data Access API

`get_individual_residential_area(individualID):`

- Args: individualID - id for individual
- Return: the ID of this individual's residential area.

`get_individual_working_area(individualID):`

- Args: individualID - id for individual
- Return: the ID of this individual's working area.

`get_individual_residential_acq(individualID):`

- Args: individualID - id for individual
- Return: an list of individual IDs of this individual's acquaintance contacts in residential area.

`get_individual_working_acq(individualID):`

- Args: individualID - id for individual
- Return: an list of individual IDs of this individual's acquaintance contacts in working area.

`get_individual_infection_state(individualID):`

- Args: individualID - id for individual
- Return: infection status of this individual, 1: susceptible or pre-symptomatic, 3: symptomatic, 4: critical, 5: recovered.

---

**Note:** The returned infection state is 1 when an individual is susceptible or pre-symptomatic.

---

`get_individual_intervention_state(individualID):`

- Args: individualID - id for individual
- Return: instervention status of this individual, 0: no such individual id, 1: without intervention, 2: confined, 3: quarantined, 4: isolated, 5: hospitalized.

`get_area_visited_history(areaID):`

- Args: areaID - id for the area
- Return: a 2D list of the visited history of one area in past 5 days. Each of the inner 1D list represents the history for one hour. The order of the list is chronological, with the earliest time appearing the first in the list. Specifically, areaID=-1 stands for hospital and isolation.

`[[individualID1, individualID2, individualID3, ...], [individualID7, individualID8,]]`

`get_individual_visited_history(individualID):`



- Args: individualID
- Return: a 1D list of the id of the areas that he/she has visited in past 5 days. The order of the list is chronological, with the earliest time appearing the first in the list.

[areaID1, areaID2, ...].

get\_individual\_visited\_history\_with\_p\_infection(individualID): (This API is deprecated.)

- Args: individualID
- Return: a 2D list of the probabilities of getting infected (from acquaintance contacts and stranger contacts) in the areas that he/she has visited in past 5 days. The order of the list is chronological, with the earliest time appearing the first in the list. This should be corresponding with get\_individual\_visited\_history.

[[p\_acq1, p\_stranger1], [p\_acq2, p\_stranger2], ...]

---

**Note:** The calculation of the probability is based on the SIR model from this paper:

- William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character, 115(772):700–721, 1927.
- 

get\_all\_area\_category():

- Return: a dictionary with all area id as the keys, and the category of the area as the value, 0: residential, 1: working, 2: commercial

get\_area\_contained\_individual():

- Return: a dictionary with all area id as the keys, and the list of individualID who live in this area as the value

get\_area\_infected\_cnt(areaID):

- Args: areaID
- Return: an int representing the number of infected (symptomatic and critical) people in this area

get\_life\_count():

- Return the total number of people not in hospital of the whole environment.

get\_infect\_count(): (This API is deprecated.)

- Return the number of infected people in the whole environment.

get\_hospitalize\_count():

- Return the number of hospitalized people in the whole environment.

get\_isolate\_count():

- Return the number of isolated people in the whole environment.

get\_quarantine\_count():

- Return the number of quarantined people in the whole environment.

get\_confine\_count():

- Return the number of confined people in the whole environment.

get\_stranger\_count()

- Return the number of stranger contacts.

get\_acquaintance\_count()

- Return the number of acquaintance contacts.

`get_current_time()`:

- Get simulation time (in hour)
- Return a `int`, starting from 0

`get_current_hour()`:

- Get simulation time (in hour of day)
- Return a `int`, ranging from 0 to 13

`get_current_day()`:

- Get simulation time (in day)
- Return a `int`, starting from 0

## 5.4 Intervention API

Intervention APIs are only effective when being called at the start of one day.

`set_individual_isolate_days(days_to_isolate)`:

- **Args: days\_to\_isolate**
  - a dictionary with `individualID` as key and days for each person to be isolated as value. The days should be positive integers ( $\geq 1$ ). If the value of day is smaller than 1, the corresponding individual to isolate 1 day.

`set_individual_quarantine_days(days_to_quarantine)`:

- **Args: days\_to\_quarantine**
  - a dictionary with `individualID` as key and days for each person to be quarantined as value. The days should be positive integers ( $\geq 1$ ). If the value of day is smaller than 1, the corresponding individual to quarantine 1 day.

`set_individual_confine_days(days_to_confine)`:

- **Args: days\_to\_confine**
  - a dictionary with `individualID` as key and days for each person to be confined as value. The days should be positive integers ( $\geq 1$ ). If the value of day is smaller than 1, the corresponding individual to confine 1 day.

`set_individual_to_treat(if_treat)`

- **Args: if\_treat**
  - a dictionary with `individualID` as key and whether he/she is sent to be treated as value. Once set true, he/she will be staying in hospital for `days` (`= 15`).

---

### Note:

- When an individual is intended with multiple interventions, only the highest level of intervention will be applied.
- When an individual is intended with multiple interventions at different days, the later intervention will update the older ones. For example, an individual is intended to be isolated  $N$  days at the  $i$ -th day. If later on day  $i + j$ -th ( $j < N$ ), he/she is set up to be confined, his/her intervention status will be updated to be confined, starting from  $i + j$ -th day.

- Intervention actions are only effective when being set at the start of one day.
- 

## 5.5 Other API

TBD